Reference Manual of the nadc-tools API and Libraries

Richard M. van Hees
Netherlands Institute for Space Research

May 7, 2013

Abstract

This is the Reference Manual for the **nadc-tools** libraries. These libraries are developed for the Netherlands SCIAMACHY Data Center (NL-SCIA-DC). The C API code is distributed with nadc-tools and allows C programs to access the official GOME and Sciamachy level 0, 1b and 2 data products as distributed by DLR and ESRIN. The C API code also include calibration modules for GOME and Sciamachy level 1b data packets.

1 About NADC tools

1.1 Purpose and Limitations

The probably most important part of the NL-SCIA-DC software (nadc_tools) are the C-libraries, which contain functions to read Sciamachy data products into memory, dump the data as ASCII output or write the data in PDS or HDF5 format. The software is written in ANSI-C (and even assembler), and coded in such a way that you (an experienced (C-) programmer) can easily understand the code. The data extractors, included in this software package, are developed for the NL-SCIA-DC and give a good example of how to use the libraries.

The software distributed in this package contains libraries, written in ANSI-C (mostly POSIX compliant except for some ISO C99 extensions) and an IDL wrapper library (using `CALL_EXTERNAL`) with IDL functions to read GOME and Sciamachy data. The IDL interface requires sharable object libraries. Building shared libraries is easy on the supported platforms, however, do not forget to set the environment variable `LD_LIBRARY_PATH`.

This software package is written and maintained by Richard van Hees (SRON), and distributed under the GNU General Public License.

1.2 Acknowledgement

In case (part of) this dataset is used for a publication and essential to the work and the results, an offer of co-authorship to Richard van Hees (SRON) and /or Ralph Snel (SRON) is highly appreciated. Please do not hesitate to contact us in an early stage for necessary support.

In all other cases SRON should be properly acknowledged. Such an acknowledgement could state: "We acknowledge SRON for providing the nadc_tools and/or patched Sciamachy level 1b data products".

1.3 DISCLAIMER

We at SRON have developed nadc_tools as part of our commitment towards the verification and further improvement of the calibration of Sciamachy data. This software is developed for in-house usage, and generously shared with you WITHOUT ANY WARRANTY. We will try to help you with any problems, but only on a "best effort" basis. In order to continuously improve this facility, feedback from users is highly appreciated.

Note that this software package is also not supported in any way by ESA or DLR, although the nadc_tools extractors and libraries mimic some of the functionality of the gdp and EnviView toolbox.

The software package is in no way intended to replace the official data processor and should not be treated as such. Especially it is not meant to (and cannot) produce official data products. These have to be derived with ESA approved tools such as EnviView that can be downloaded for free from ESA.

It is your own responsibility to verify the results you produce with this package against official data products. If you find discrepancies, please inform SRON (not ESA or DLR) at once. Any information you give will help us to improve our software. We will relay to ESA all information to improve the official data processor.

1.4 COPYRIGHT

The software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Contents

2 General NADC Modules

2.1 ADAGUC_INIT_PARAM

Identifier	ADAGUC_INIT_PARAM
Author	R.M. van Hees
Language	ANSI C
Purpose	initializes struct with command-line parameters
Usage	`ADAGUC_INIT_PARAM( argc, argv, instrument, param );`
Input	int argc : number of parameters
	char *argv[] : parameter values
	int instrument : code for instrument en data product level
Output	struct param_record (A.1)
	*param : struct holding user-defined settings
Returns	Nothing
Comment	None

2.2 ADAUC_VERSION

Identifier	ADAUC_VERSION
Author	R.M. van Hees
Language	ANSI C
Purpose	Returns the library version numbers through arguments.
Comment	contains ADAGUCget_Version and ADAGUCshow_Version

2.2.1 ADAGUCshow_Version

Identifier	ADAGUCshow_Version
Purpose	display version of SCIA library and release date
Usage	`ADAGUCshow_Version( stream, prognm );`
Output	FILE *stream : file pointer
	char *prognm : name of the calling program
Returns	nothing
Comment	none

2.3 ENVI_PDS_DSD

Identifier	ENVI_PDS_DSD
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Data Set Description of the Envisat PDS product
Comment	contains ENVI_RD_DSD and ENVI_WR_DSD

2.3.1 ENVI_RD_DSD

Identifier	ENVI_RD_DSD
Purpose	read Data Set Description records of the Envisat PDS product
Usage	`nr_dsd = ENVI_RD_DSD( fd, mph, &dsd );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi mph : main product header (A.5)
Output	struct dsd_envi *dsd : structure for the DSD (A.6)
Returns	number of DSD read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

2.3.2 ENVI_WR_DSD

Identifier	ENVI_WR_DSD
Purpose	write Data Set Description records of the Envisat PDS product
Usage	`ENVI_WR_DSD( fd, num_dsd, dsd );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_dsd : number of DSD records
	struct dsd_envi *dsd : structure for the DSD (A.6)
Returns	nothing
Comment	none

2.4 ENVI_RD_PDS_INFO

Identifier	ENVI_RD_PDS_INFO
Author	R.M. van Hees
Language	ANSI C
Purpose	read one SCIAMACHY header-line,
	split line in keyword and keyvalue
Usage	`nbyte = ENVI_RD_PDS_INFO( fd, keyword, keyvalue );`
Input	FILE *fd : open stream pointer
Output	char *keyword : string with keyword name
	char *keyvalue : string with keyword value
Returns	number of bytes read (unsigned int)
Comment	None

2.5 ENVI_PDS_MPH

Identifier	ENVI_PDS_MPH
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Main Product Header of the Envisat PDS product
Comment	contains ENVI_RD_MPH and ENVI_WR_MPH

2.5.1 ENVI_RD_MPH

Identifier	ENVI_RD_MPH
Purpose	read Main Product Header of the Envisat PDS product
Usage	`ENVI_RD_MPH( fd, &mph );`
Input	FILE *fd : (open) stream pointer
Output	struct mph_envi *mph : structure for the MPH (A.5)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

2.5.2 ENVI_WR_MPH

Identifier	ENVI_WR_MPH
Purpose	write Main Product Header of the Envisat PDS product
Usage	`ENVI_WR_MPH( fd, mph );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi mph : structure for the MPH (A.5)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

2.6 ENVI_WR_ASCII_DSD

Identifier	ENVI_WR_ASCII_DSD
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Data Set Descriptor Records
Usage	`ENVI_WR_ASCII_DSD( param, num_dsd, dsd );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsd : number of DSD records
	struct dsd_envi *dsd : structure for the DSD records (A.6)
Returns	Nothing
Comment	None

2.7 ENVI_WR_ASCII_MPH

Identifier	ENVI_WR_ASCII_MPH
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Main Product Header
Usage	`ENVI_WR_ASCII_MPH( param, mph );`
Input	struct param_record param : struct holding user-defined settings
	struct mph_envi *mph : structure for the MPH (A.5)
Returns	Nothing
Comment	None

2.8 GET_ENVI_DSD_INDX

Identifier	GET_ENVI_DSD_INDX
Author	R.M. van Hees
Language	ANSI C
Purpose	get index to a DSD given its name
Usage	`indx_dsd = GET_ENVI_DSD_INDX( num_dsd, dsd, dsd_name );`
Input	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd[] : characteristics of DSD (A.6)
	char dsd_name[] : name of the DSD
Returns	index to DSD (unsigned int)
Comment	None

2.9 GOME_VERSION

Identifier	GOME_VERSION
Author	R.M. van Hees
Language	ANSI C
Purpose	Returns the library version numbers through arguments.
Comment	contains GOMEget_Version and GOMEshow_Version

2.9.1 GOMEget_Version

Identifier	GOMEget_Version
Purpose	get version of GOME library and release date
Usage	`GOMEget_Version( &majnum, &minnum, &relnum, reldate );`
Output	unsigned int *majnum : major revision number
	unsigned int *minnum : minor revision number
	unsigned int *relnum : release revision number
	char *reldate : release date
Returns	nothing
Comment	none

2.9.2 GOMEshow_Version

Identifier	GOMEshow_Version
Purpose	display version of GOME library and release date
Usage	`GOMEshow_Version( stream, prognm );`
Output	FILE *stream : file pointer
	char *prognm : name of the calling program
Returns	nothing
Comment	none

2.10 MERIS_VERSION

Identifier	MERIS_VERSION
Author	R.M. van Hees
Language	ANSI C
Purpose	Returns the library version numbers through arguments.
Comment	contains MERISget_Version and MERISshow_Version

2.10.1 MERISget_Version

Identifier	MERISget_Version
Purpose	get version of MERIS library and release date
Usage	`MERISget_Version( &majnum, &minnum, &relnum, reldate );`
Output	unsigned int *majnum : major revision number
	unsigned int *minnum : minor revision number
	unsigned int *relnum : release revision number
	char *reldate : release date
Returns	nothing
Comment	none

2.10.2 MERISshow_Version

Identifier	MERISshow_Version
Purpose	display version of MERIS library and release date
Usage	`MERISshow_Version( stream, prognm );`
Output	FILE *stream : file pointer
	char *prognm : name of the calling program
Returns	nothing
Comment	none

2.11 FIT_GRID_AKIMA

Identifier	FIT_GRID_AKIMA
Author	R.M. van Hees
Language	ANSI C
Purpose	interpolates values of a given spectrum to another wavelength grid
Usage	`FIT_GRID_AKIMA( FLT32_T, FLT64_T, dim_in, x_in, y_in,` `FLT64_T, FLT64_T, dim_out, x_out, y_out );`
Input	int x_type_in : data type of input values (float or double)
	int y_type_in : data type of input values (float or double)
	size_t dim_in : dimension of x_in, y_in
	void *x_in : input (wavelength) grid
	void *y_in : y values of the given spectrum
	int x_type_out : data type of output values (float or double)
	int y_type_out : data type of output values (float or double)
	size_t dim_out : dimension of x_out, y_out
	void *x_out : output (wavelength) grid
Output	void *y_out : y values of the spectrum fitted to grid
Returns	nothing
Comment	nothing

2.11.1 NADC_AKIMA_EX

Identifier	NADC_AKIMA_EX
Purpose	extrapolates the function two points left and right
Usage	`NADC_AKIMA_EX( FLT32_T, dim_in, x_in, y_in, xx, yy, st );`
Input	int x_type_id : data type of x values (float or double)
	int y_type_id : data type of y values (float or double)
	size_t dim_in : dimension of x_in and y_in
	void *x_in : x values
	void *y_in : y values
Output	double *xx : x values extrapolated by 2 points left and right
	double *yy : y values extrapolated by 2 points left and right
	double *st : slope of curve
Returns	Nothing
Comment	called by NADC_AKIMA_SU extrapolates the function two points left and right call this subroutine once before evaluating the interpolation value of the new gridpoint (using NADC_AKIMA_PO)

2.11.2 NADC_AKIMA_SU

Identifier	NADC_AKIMA_SU
Purpose	fits a polynome through a given input function for inter- and extrapolation
Usage	`NADC_AKIMA_SU( FLT32_T, FLT64_T, dim_in, x_in, y_in, a_coef, b_coef, c_coef, d_coef );`
Input	int x_type_id : data type of x values (float or double)
	int y_type_id : data type of y values (float or double)
	size_t dim_in : dimension of x_in, y_in
	void *x_in : x values
	void *y_in : y values
Output	double *a_coef : polynomial coefficients
	double *b_coef : polynomial coefficients
	double *c_coef : polynomial coefficients
	double *d_coef : polynomial coefficients
Returns	Nothing
Comment	call this subroutine once before evaluating the interpolation value of the new gridpoint (using NADC_AKIMA_PO)

2.11.3 NADC_AKIMA_PO

Identifier	NADC_AKIMA_PO
Purpose	evaluates the interpolation value on a new gridpoint x_val
Usage	`NADC_AKIMA_PO( dim_in, x_in,` `a_coef, b_coef, c_coef, d_coef, x_val );`
Input	size_t dim_in : dimension of x_in, y_in float *x_in : x values double *a_coef : polynomial coefficients double *b_coef : polynomial coefficients double *c_coef : polynomial coefficients double *d_coef : polynomial coefficients float x_val : the abscissa at which you want a y-value
Returns	the function value at point x_val
Comment	Given an n-dim. array of abscissa x_in and arrays a, b, c, d of polynomial coefficients as returned by the NADC_AKIMA_SU, this subroutine returns the interpolated value at abscissa x_val The polynomial coefficients a,b,c,d must be known from a call to subroutine 'NADC_AKIMA_SU'.

2.12 NADC_ALLOC

Identifier	NADC_ALLOC
Author	R.M. van Hees
Language	ANSI-C
Purpose	pointer testing and memory allocation
Comment	contains ALLOC_C2D, ALLOC_S2D, ALLOC_I2D, ALLOC_R2D, ALLOC_D2D ALLOC_I3D, ALLOC_I4D, FREE_2D, FREE_3D, and FREE_4D

2.12.1 ALLOC_UC2D

Identifier	ALLOC_UC2D
Purpose	allocate a unsigned character matrix with dimensions [ypix][xpix]
Usage	`plane = ALLOC_UC2D( ypix, xpix );`
Input	size_t ypix : Y dimension (slowest axis) size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.2 ALLOC_C2D

Identifier	ALLOC_C2D
Purpose	allocate a character data matrix with dimensions [ypix][xpix]
Usage	`plane = ALLOC_C2D( ypix, xpix );`
Input	size_t ypix : Y dimension (slowest axis) size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.3 ALLOC_US2D

Identifier	ALLOC_US2D
Purpose	allocate a unsigned short integer data matrix with dimensions [ypix][xpix]
Usage	`plane = ALLOC_US2D( ypix, xpix );`
Input	size_t ypix : Y dimension (slowest axis) size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.4 ALLOC_S2D

Identifier	ALLOC_S2D
Purpose	allocate a short integer data matrix with dimensions [ypix][xpix]
Usage	`plane = ALLOC_S2D( ypix, xpix );`
Input	size_t ypix : Y dimension (slowest axis) size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.5 ALLOC_I2D

Identifier	ALLOC_I2D
Purpose	allocate a integer data matrix with dimensions [ypix][xpix]
Usage	`plane = ALLOC_I2D( ypix, xpix );`
Input	size_t ypix : Y dimension (slowest axis) size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.6 ALLOC_R2D

Identifier	ALLOC_R2D
Purpose	allocate a float data matrix with dimensions [ypix][xpix]
Usage	`plane = ALLOC_R2D( ypix, xpix );`
Input	size_t ypix : Y dimension (slowest axis) size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.7 ALLOC_D2D

Identifier	ALLOC_D2D
Purpose	allocate a double data matrix with dimensions [ypix][xpix]
Usage	`plane = ALLOC_D2D( ypix, xpix );`
Input	size_t ypix : Y dimension (slowest axis) size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.8 ALLOC_S3D

Identifier	ALLOC_S3D
Purpose	allocate a short integer data cube with dimensions [zpix][ypix][xpix]
Usage	`cube = ALLOC_I3D( zpix, ypix, xpix );`
Input	size_t zpix : Z dimension (slowest axis) size_t ypix : Y dimension size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.9 ALLOC_I3D

Identifier	ALLOC_I3D
Purpose	allocate a integer data cube with dimensions [zpix][ypix][xpix]
Usage	`cube = ALLOC_I3D( zpix, ypix, xpix );`
Input	size_t zpix : Z dimension (slowest axis) size_t ypix : Y dimension size_t xpix : X dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.10 ALLOC_I4D

Identifier	ALLOC_I4D
Purpose	allocate a integer data cube with dimensions [zpix][ypix][xpix][wpix]
Usage	`cube = ALLOC_I4D( zpix, ypix, xpix, wpix );`
Input	size_t zpix : Z dimension (slowest axis) size_t ypix : Y dimension size_t xpix : X dimension size_t wpix : W dimension (fastest axis)
Returns	pointer to allocated memory
Comment	none

2.12.11 FREE_2D

Identifier	FREE_2D
Purpose	free memory allocated with ALLOC_x2D
Usage	`FREE_2D( plane );`
Input	void **plane : memory allocated with ALLOC_x2D
Returns	nothing
Comment	none

2.12.12 FREE_3D

Identifier	FREE_3D
Purpose	free memory allocated with ALLOC_x3D
Usage	`FREE_2D( plane );`
Input	void ***cube : memory allocated with ALLOC_x3D
Returns	nothing
Comment	none

2.12.13 FREE_4D

Identifier	FREE_4D
Purpose	free memory allocated with ALLOC_x4D
Usage	`FREE_4D( cube );`
Input	void ****cube : memory allocated with ALLOC_x4D
Returns	nothing
Comment	none

2.13 BinarySearch

Identifier	BinarySearch
Author	R.M. van Hees
Language	ANSI C
Purpose	searches an array[index[]] for presence of ”value”
Usage	`res = BinarySearch( dim, index, array, value );`
Input	int dim : dimension of the array to be sorted
	int *index : indices to sort ”array”
	int *array : array to be searched
	int value : value to be found in array
Returns	first occurences where array[index[res]] ¿= value
Comment	None

2.14 NADC_BITS

Identifier	NADC_BITS
Author	R.M. van Hees
Language	ANSI C
Purpose	set/read bits in a unsigned long long variable
Comment	contains Set_Bit_LL and Get_Bit_LL

2.14.1 Set_Bit_LL

Identifier	Set_Bit_LL
Purpose	set bits in a unsigned long long variable
Usage	`Set_Bit_LL( x_ull, pos );`
Input	unsigned long long *x_ull : input value
	unsigned char pos : position of bit to be set or read
Returns	Nothing
Comment	None

2.14.2 Get_Bit_LL

Identifier	Get_Bit_LL
Purpose	read bits in a unsigned long long variable
Usage	`x_ull = Get_Bit_LL( x_ull, pos );`
Input	unsigned long long x_ull : input value
	unsigned char pos : position of bit to be set or read
Returns	value of the requested bit (unsigned long long)
Comment	None

2.15 NADC_CHECK_FOR_SAA

Identifier	NADC_CHECK_FOR_SAA
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain saaFlag
Usage	`saaFlag = NADC_CHECK_FOR_SAA( latitude, longitude );`
Input	double *latitude : latitude of instrument
	double *longitude : longitude of instrument
Returns	TRUE if above SAA region
Comment	ToDo - improve SAA definition

2.16 NADC_CodeCalib_GOME

Identifier	NADC_CodeCalib_GOME
Author	R.M. van Hees
Language	ANSI C
Purpose	GOME level 1b calibration parameter coding
Usage	`calib_mask = NADC_GOME_CalibMask( calib_str );`
Input	char calib_str[] : string with encoded calibration steps
Returns	calibration mask (unsigned short)
Usage	`nr_char = NADC_GOME_CalibStr( calib_val, calib_str );`
Input	unsigned short calib_val : encoded calibration steps
Output	char calib_str[] : string with encoded calibration steps
Returns	size of the calib string
Usage	`NADC_GOME_CalibShow( FILE *stream );`
Input	FILE stream : stream to show available calibration options
Comment	None

2.17 NADC_CodeCalib_SCIA

Identifier	NADC_CodeCalib_SCIA
Author	R.M. van Hees
Language	ANSI C
Purpose	SCIA level 1b calibration parameter coding
Comment	None

2.17.1 NADC_SCIA_CalibMask

Identifier	NADC_SCIA_CalibMask
Purpose	extract calibration options from command-line string
Usage	`calib_mask = NADC_SCIA_CalibMask( calib_str );`
Input	char calib_str[] : string with encoded calibration steps
Returns	calibration mask (unsigned int)
Comment	none

2.17.2 NADC_SCIA_CalibStr

Identifier	NADC_SCIA_CalibStr
Purpose	return string with used calibration options
Usage	`nr_char = NADC_SCIA_CalibStr( calib_val, calib_str );`
Input	unsigned int calib_val : encoded calibration steps
Output	char calib_str[] : string with encoded calibration steps
Returns	size of the calib string
Comment	none

2.17.3 NADC_SCIA_CalibShow

Identifier	NADC_SCIA_CalibShow
Purpose	show all implemented calibration options
Usage	`NADC_SCIA_CalibShow( FILE *stream );`
Input	FILE stream : stream to show available calibration options
Returns	nothing
Comment	none

2.18 NADC_CodePatch_SCIA

Identifier	NADC_CodePatch_SCIA
Author	R.M. van Hees
Language	ANSI C
Purpose	Sciamachy level 1b patch parameter coding
Usage	`patch_mask = NADC_SCIA_PatchMask( patch_str );`
Input	char patch_str[] : string with encoded patch steps
Returns	patchration mask (unsigned short)
Usage	`nr_char = NADC_SCIA_PatchStr( patch_val, patch_str );`
Input	unsigned short patch_val : encoded patch steps
Output	char patch_str[] : string with encoded patch steps
Returns	size of the patch string
Usage	`NADC_SCIA_PatchShow( FILE *stream );`
Input	FILE stream : stream to show available patch options
Comment	None

2.18.1 NADC_SCIA_PatchMask

Identifier	NADC_SCIA_PatchMask
Purpose	extract patch options from command-line string
Usage	`patch_mask = NADC_SCIA_PatchMask( patch_str );`
Input	char patch_str[] : string with encoded patch options
Returns	patch mask (unsigned short)
Comment	none

2.18.2 NADC_SCIA_PatchStr

Identifier	NADC_SCIA_PatchStr
Purpose	return string with used patch options
Usage	`nr_char = NADC_SCIA_PatchStr( patch_val, patch_str );`
Input	unsigned int patch_val : encoded patch options
Output	char patch_str[] : string with encoded patch options
Returns	size of the patch string
Comment	none

2.18.3 NADC_SCIA_PatchShow

Identifier	NADC_SCIA_PatchShow
Purpose	show all implemented patch options
Usage	`NADC_SCIA_PatchShow( FILE *stream );`
Input	FILE stream : stream to show available patch options
Returns	nothing
Comment	none

2.19 NADC_CopyRight

Identifier	NADC_CopyRight
Author	R.M. van Hees
Language	ANSI C
Purpose	display copyright
Usage	`NADC_CopyRight( stream );`
Input	FILE *stream : open stream to write copyright info
Returns	nothing, function exits with EXIT_SUCCESS
Comment	none

2.20 NADC_DATE

Identifier	NADC_DATE
Author	R.M. van Hees
Language	ANSI C
Comment	contains ASCII_2_UTC, UTC_2_ASCII, ASCII_2_MJD and MJD_2_ASCII UTC_2_DATETIME, MJD_2_DATETIME, MJD_2_YMD, MJD_2_Julian and Julian_2_MJD, SciaJDAY2adaguc, Adaguc2sciaJDAY, GomeJDAY2adaguc, Adaguc2gomeJDAY

2.20.1 ASCII_2_UTC

Identifier	ASCII_2_UTC
Purpose	Converts ASCII time into ESA UTC time
Usage	`ASCII_2_UTC( ASCII_Time, &utc_day, &utc_sec );`
Input	char *ASCII_Time : given as DD-MMM-YYYY HH:MM:SS.SSS
Output	unsigned int utc_day : ESA UTC days since 01.01.1950 unsigned int utc_msec : ESA UTC milli-seconds since midnight
Returns	error status
Comment	none

2.20.2 ASCII_2_MJD

Identifier	ASCII_2_MJD
Purpose	Converts ASCII time into SCIAMACHY MJD
Usage	`ASCII_2_MJD( ASCII_Time, &mjd2000, &second, &mu_sec );`
Input	char *ASCII_Time : given as DD-MMM-YYYY HH:MM:SS.SSSSSS
Output	signed int *mjd2000 : number of days elapsed since 1.1.2000
	unsigned int *second : seconds elepsed since midnight
	unsigned int *mu_sec : micro-seconds since last second
Returns	error status
Comment	none

2.20.3 UTC_2_ASCII

Identifier	UTC_2_ASCII
Purpose	Converts ESA UTC time into a ASCII time
Usage	`UTC_2_ASCII( utc_day, utc_msec, ASCII_Time );`
Input	unsigned int utc_day : ESA UTC days since 01.01.1950
	unsigned int utc_msec : ESA UTC milli-seconds since midnight
Output	char *ASCII_Time : returned as DD-MMM-YYYY HH:MM:SS.SSS
Returns	nothing
Comment	see GOME date specification

2.20.4 MJD_2_ASCII

Identifier	MJD_2_ASCII
Purpose	Converts SCIAMACHY MJD into a ASCII time
Usage	`MJD_2_ASCII( mjd2000, second, mu_sec, ASCII_Time );`
Input	signed int mjd2000 : number of days elapsed since 1.1.2000
	unsigned int second : seconds elepsed since midnight
	unsigned int mu_sec : micro-seconds since last second
Output	char *ASCII_Time : returned as DD-MMM-YYYY HH:MM:SS.SSSSSS
Returns	nothing
Comment	see SCIAMACHY date specification

2.20.5 UTC_2_DATETIME

Identifier	UTC_2_DATETIME
Purpose	Converts ESA UTC time into a MySQL dateTime string
Usage	`UTC_2_DATETIME( utc_day, utc_msec, dateTime );`
Input	unsigned int utc_day : ESA UTC days since 01.01.1950
	unsigned int utc_msec : ESA UTC milli-seconds since midnight
Output	char *dateTime : returned as YYYY-MM-DD HH:MM:SS.SSS
Returns	nothing
Comment	see GOME date specification

2.20.6 DATETIME_2_JULIAN

Identifier	DATETIME_2_JULIAN
Purpose	Converts MySQL dateTime string to Julian day
Usage	`jday = DATETIME_2_MJD( dateTime, muSeconds );`
Input	char *dateTime : given as YYYY-MM-DD HH:MM:SS
	int muSeconds : micro-seconds since last second
Returns	Julian day (double)
Comment	none

2.20.7 MJD_2_DATETIME

Identifier	MJD_2_DATETIME
Purpose	Converts SCIAMACHY MJD into a MySQL dateTime string
Usage	`MJD_2_DATETIME( mjd2000, second, mu_sec, dateTime );`
Input	signed int mjd2000 : number of days elapsed since 1.1.2000
	unsigned int second : seconds elepsed since midnight
	unsigned int mu_sec : micro-seconds since last second
Output	char *dateTime : returned as YYYY-MM-DD HH:MM:SS.SSSSSS
Returns	nothing
Comment	see SCIAMACHY date specification

2.20.8 MJD_2_YMD

Identifier	MJD_2_YMD
Purpose	Converts SCIAMACHY MJD into a dateTime string, as used in aux. files
Usage	`MJD_2_YMD( mjd2000, second, dateTime );`
Input	signed int mjd2000 : number of days elapsed since 1.1.2000
	unsigned int second : seconds elepsed since midnight
	unsigned int mu_sec : micro-seconds since last second
Output	char *dateTime : returned as YYYYMMDD_HH:MM:SS
Returns	nothing
Comment	see SCIAMACHY date specification

2.20.9 SciaJDAY2adaguc

Identifier	SciaJDAY2adaguc
Purpose	Converts julian (2000) decimal day into a dateTime string
Usage	`SciaJDAY2adaguc( jday, dateTime );`
Input	double jday : decimal days elapsed since 1.1.2000
Output	char *dateTime : returned as yyyyMMddThhmmss
Returns	nothing
Comment	none

2.20.10 Adaguc2sciaJDAY

Identifier	Adaguc2sciaJDAY
Purpose	Converts dateTime string to julian (2000) decimal day
Usage	`jday = Adaguc2sciaJDAY2( dateTime );`
Input	char *dateTime : dateTime given as yyyyMMddThhmmss
Returns	decimal days elapsed since 1.1.2000 (double)
Comment	none

2.20.11 GomeJDAY2adaguc

Identifier	GomeJDAY2adaguc
Purpose	Converts julian (1950) decimal day into a dateTime string
Usage	`GomeJDAY2adaguc( jday, dateTime );`
Input	double jday : decimal days elapsed since 1.1.1950
Output	char *dateTime : returned as yyyyMMddThhmmss
Returns	nothing
Comment	none

2.20.12 Adaguc2gomeJDAY

Identifier	Adaguc2gomeJDAY
Purpose	Converts dateTime string to julian (1950) decimal day
Usage	`jday = Adaguc2gomeJDAY2( dateTime );`
Input	char *dateTime : dateTime given as yyyyMMddThhmmss
Returns	decimal days elapsed since 1.1.1950 (double)
Comment	see GOME date specification

2.21 NADC_ERROR

Identifier	NADC_ERROR
Author	R.M. van Hees
Language	ANSI C
Purpose	error handling and display routines
Comment	contains NADC_Err_Push, NADC_Err_Clear, NADC_Err_Keep, NADC_Err_Trace used global variables: nadc_stat and nadc_err_stack

2.21.1 NADC_Err_Push

Identifier	NADC_Err_Push
Purpose	push error message on stack
Usage	`NADC_Err_Push( mesg_num, file_name, func_name, line, desc );`
Input	NADC_err_t mesg_num : id of error message char *file_name : name of the module, where the error was issued char *func_name : name of the function, where the error was issued int line : line where the error was issued char *desc : string with additional info
Returns	nothing
Comment	none

2.21.2 NADC_Err_Clear

Identifier	NADC_Err_Clear
Purpose	clear error stack
Usage	`NADC_Err_Clear();`
Returns	nothing
Comment	none

2.21.3 NADC_Err_Keep

Identifier	NADC_Err_Keep
Purpose	save error status
Usage	`NADC_Err_Keep( do_save );`
Input	bool do_save : save current error status
Returns	nothing
Comment	none

2.21.4 NADC_Err_Trace

Identifier	NADC_Err_Trace
Purpose	write error messages to stream
Usage	`NADC_Err_Trace( stream );`
Input	FILE *stream : open stream to write messages
Returns	nothing
Comment	none

2.22 NADC_EXT_H5_ARRAY

Identifier	NADC_EXT_H5_ARRAY
Author	R.M. van Hees
Language	ANSI C
Purpose	subroutines create HDF5 unlimited arrays
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.1 NADC_CRE_H5_EArray

Identifier	NADC_CRE_H5_EArray
Purpose	create extensible HDF5 dataset
Usage	`stat = NADC_CRE_H5_EArray( locID, dset_name, rank, dims, dims_chunk, extdim, compress, fill_data, typeID, buff );`
Input	hid_t locID : HDF5 identifier of file or group char *dset_name : name of dataset int rank : number of dimensions hsize_t *dims : size of each dimension, fasted last! hsize_t *dims_chunk : chunk sizes, fasted dimension last! int extdim : index of expendable dimension unsigned int compress : compression level (zero for no compression) void *fill_data : Fill value for data (or NULL) hid_t typeID : data type (HDF5 identifier) void *buffer : buffer with data to write (or NULL)
Returns	A negative value is returned on failure
Comment	none

2.22.2 NADC_CRE_H5_EArray_uint8

Identifier	NADC_CRE_H5_EArray_uint8
Purpose	create HDF5 dataset, X-dimension fixed, Y-dimension extensible
Usage	`NADC_CRE_H5_EArray_uint8( locID, arrName, xdim, compression, buffer );`
Input	hid_t locID : HDF5 identifier of file or group char *arrName : name of dataset hsize_t xdim : size if first an fasted varying dimension int compression : compression level (0 = no compression) unsigned char *buffer : data to write
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.3 NADC_CRE_H5_EArray_uint16

Identifier	NADC_CRE_H5_EArray_uint16
Purpose	create HDF5 dataset, X-dimension fixed, Y-dimension extensible
Usage	`NADC_CRE_H5_EArray_uint16( locID, arrName, xdim, compression, buffer );`
Input	hid_t locID : HDF5 identifier of file or group char *arrName : name of dataset hsize_t xdim : size if first an fasted varying dimension int compression : compression level (0 = no compression) unsigned short *buffer : data to write
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.4 NADC_CRE_H5_EArray_int32

Identifier	NADC_CRE_H5_EArray_int32
Purpose	create HDF5 dataset, X-dimension fixed, Y-dimension extensible
Usage	`NADC_CRE_H5_EArray_int32( locID, arrName, xdim, compression, buffer );`
Input	hid_t locID : HDF5 identifier of file or group char *arrName : name of dataset hsize_t xdim : size if first an fasted varying dimension int compression : compression level (0 = no compression) int *buffer : data to write
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.5 NADC_CRE_H5_EArray_uint32

Identifier	NADC_CRE_H5_EArray_uint32
Purpose	create HDF5 dataset, X-dimension fixed, Y-dimension extensible
Usage	`NADC_CRE_H5_EArray_iunt32( locID, arrName, xdim, compression, buffer );`
Input	hid_t locID : HDF5 identifier of file or group char *arrName : name of dataset hsize_t xdim : size if first an fasted varying dimension int compression : compression level (0 = no compression) unsigned int *buffer : data to write
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.6 NADC_CRE_H5_EArray_float

Identifier	NADC_CRE_H5_EArray_float
Purpose	create HDF5 dataset, X-dimension fixed, Y-dimension extensible
Usage	`NADC_CRE_H5_EArray_float( locID, arrName, xdim, compression, buffer );`
Input	hid_t locID : HDF5 identifier of file or group char *arrName : name of dataset hsize_t xdim : size if first an fasted varying dimension int compression : compression level (0 = no compression) float : data to write
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.7 NADC_CRE_H5_EArray_struct

Identifier	NADC_CRE_H5_EArray_struct
Purpose	create HDF5 dataset, X-dimension fixed, Y-dimension extensible
Usage	`NADC_CRE_H5_EArray_struct( locID, arrName, xdim, compression, typeID, buffer );`
Input	hid_t locID : HDF5 identifier of file or group char *arrName : name of dataset hsize_t xdim : size if first an fasted varying dimension int compression : compression level (0 = no compression) hid_t typeID : HDF5 type ID of compound data void : data to write
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.8 NADC_CAT_H5_EArray

Identifier	NADC_CAT_H5_EArray
Purpose	concatenate data to a HDF5 dataset
Usage	`NADC_CAT_H5_EArray( locID, dset_name, extdim, count, buffer );`
Input	hid_t locID : HDF5 identifier of file or group
	char *dset_name : name of dataset
	int extdim : dimension to extend
	int count : number of rows to write
	void *buffer : data to write
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.9 NADC_WR_H5_EArray

Identifier	NADC_WR_H5_EArray
Purpose	write data to a HDF5 dataset (no append!)
Usage	`NADC_WR_H5_EArray( locID, dset_name, Ystart, Ycount, buffer );`
Input	hid_t locID : HDF5 identifier of file or group
	char *dset_name : name of dataset
	size_t Ystart : index of first row to overwrite (¿ 1 !)
	size_t Ycount : number of rows to write
	void *buffer : data to write
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.22.10 NADC_RD_H5_EArray

Identifier	NADC_RD_H5_EArray
Purpose	read data from a HDF5 dataset
Usage	`NADC_RD_H5_EArray_type( locID, dset_name, xdim, indexNum, metaIndex, buffer );`
Input	hid_t locID : HDF5 identifier of file or group
	char *dset_name : name of dataset
	hsize_t xdim : number of elements along fastest axis
	int indexNum : number of rows to read
	int *metaIndex : indices to rows to be read
Output	void *buffer : data buffer
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

2.23 NADC_FILES_EQUAL

Identifier	NADC_FILES_EQUAL
Author	R.M. van Hees
Language	ANSI C
Purpose	check if two files pointed to by fl_name_1 and fl_name_2, respectively, are not the same on disk
Usage	`stat = NADC_FILES_EQUAL( fl_name_1, fl_name_2 );`
Input	char *fl_name_1 : name of the first file
	char *fl_name_2 : name of the second file
Returns	on success, zero
Comment	None

2.24 NADC_FILESIZE

Identifier	NADC_FILESIZE
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain files size from operating system
Usage	`flsize = NADC_FILESIZE( flname );`
Input	char *flname : name of the file
Returns	size of file in bytes
Comment	None

2.25 NADC_FIT

Identifier	NADC_FIT
Author	R.M. van Hees
Language	ANSI C
Purpose	fitting data to a straight line y = a + b * x
Usage	`NADC_FIT( dim_arr, x_arr, y_arr, err_arr,` `&fit_a, &fit_b, &sig_a, &sig_b, &chi2_fit, &q_fit );`
Input	size_t dim_arr : number of data points float *x_arr : independent variable values (X-axis) float *y_arr : dependent variable values (Y-axis) float *err_arr : measurement errors of the data points Y
Output	float *fit_a : parameter a float *fit_b : parameter b, if NULL then a horizontal line is fitted float *sig_a : uncertainty in parameter a float *sig_b : uncertainty in parameter b, not calculated in case fit_b == NULL float *chi2_fit : chi-square of fit float *q_fit : goodness-of-fit probability (NULL will skip the calculation)
Returns	nothing
Comment	adopted from Numerical Recipes in C

2.26 NADC_FLIP

Identifier	NADC_FLIP
Author	R.M. van Hees
Language	ANSI C
Purpose	flip the data of a matrix
Usage	`NADC_FLIPx( flip, npix, xmatrix );`
Input	enum nadc_flip flip : specifies flip axis
	unsigned int npix[2] : dimensions of the matrix
In/Output	matrix matrix[2] : array to be transformed
Returns	nothing (check global error status)
Comment	”flip” specifies the operation to be performed, as follows:
	0 do not change the matrix
	1 flip the data along the first axis
	2 flip the data along the second axis
	3 flip the data along both axis
	”x” specifies the data type of the input matrix, as follows:
	”c” signed char
	”u” unsigned char
	”s” short
	”r” float

2.27 NADC_HDF5_API

Identifier	NADC_HDF5_API
Author	R.M. van Hees
Language	ANSI C
Purpose	subroutines to easily create/write/read attributes and datasets
Returns	nothing: modifies global error status
Comment	none

2.27.1 WRITE_HDF5_HISTORY

Identifier	WRITE_HDF5_HISTORY
Purpose	Write attribute in group holding user-defined settings
Usage	`WRITE_HDF5_HISTORY( file_id, instrument, param )`
Input	hid_t file_id : HDF5 file ID
	hid_t instrument : flag for Instrument & Product
	struct param_record param : struct holding user-defined settings (A.1)
Returns	nothing: modifies global error status
Comment	none

2.27.2 NADC_OPEN_HDF5_Group

Identifier	NADC_OPEN_HDF5_Group
Purpose	Open/Create a group in an exsisting HDF5-file
Usage	`grp_id = NADC_OPEN_HDF5_Group( loc_id, name );`
Input	hid_t loc_id : HDF5 object id
	char name[] : name of the group
Returns	A negative value is returned on failure.
Comment	none

2.27.3 NADC_WR_HDF5_Attribute

Identifier	NADC_WR_HDF5_Attribute
Purpose	Create/write an attribute which is attached to an object
Usage	`NADC_WR_HDF5_Attribute( loc_id, name, type_id, rank, dims, data );`
Input	hid_t loc_id : HDF5 object id
	char name[] : name of the attribute
	hid_t type_id : data type of the attribute's data
	int rank : number of dimensions
	hsize_t dims[] : dimension specification
	void *data : data for the attribute
Returns	nothing: modifies global error status
Comment	none

2.27.4 NADC_RD_HDF5_Dataset

Identifier	NADC_RD_HDF5_Dataset
Purpose	read a HDF5 dataset which is attached to loc_id
Usage	`NADC_RD_HDF5_Dataset( loc_id, name, type_id, &rank, dims, &data );`
Input	hid_t loc_id : HDF5 object id
	char name[] : name of the dataset
	hid_t type_id : data type of the attribute's data
Output	int *rank : number of dimensions
	hsize_t dims[] : dimension specification
	void **data_out : data for the dataset
Returns	nothing: modifies global error status
Comment	none

2.27.5 NADC_WR_HDF5_Dataset

Identifier	NADC_WR_HDF5_Dataset
Purpose	Create/write a dataset which is attached to an object
Usage	`NADC_WR_HDF5_Dataset( compress, loc_id, name, type_id, rank, dims, data );`
Input	hbool_t compress : sets the compression method to H5Z_DEFLATE
	hid_t loc_id : HDF5 object id
	char name[] : name of the dataset
	hid_t type_id : data type of the attribute's data
	int rank : number of dimensions
	hsize_t dims[] : dimension specification
	void *data : data for the dataset
Returns	nothing: modifies global error status
Comment	none

2.27.6 NADC_WR_HDF5_Vlen_Dataset

Identifier	NADC_WR_HDF5_Vlen_Dataset
Purpose	Create/write a variable length HDF5 dataset
Usage	`NADC_WR_HDF5_Vlen_Dataset( compress, loc_id, name, type_id, rank, dims, fill_value, vdata );`
Input	hbool_t compress : sets the compression method to H5Z_DEFLATE
	hid_t loc_id : HDF5 object id
	char name[] : name of the dataset
	hid_t type_id : data type of the attribute's data
	int rank : number of dimensions
	hsize_t dims[] : dimension specification
In/Output	void *vdata : variable length data for the dataset
	at return the memory buffers are reclaimed
Returns	nothing: modifies global error status
Comment	the memory buffers of "vdata" are reclaimed

2.27.7 Create_HDF5_NADC_FILE

Identifier	Create_HDF5_NADC_FILE
Purpose	Create HDF5 file for NADC GOME and SCIAMACHY data
Usage	`stat = Create_HDF5_NADC_FILE( instument, param );`
Input	int instrument : flag for Instrument & Product
	struct param_record param : command-line parameters (A.1)
Returns	nothing: modifies global error status
Comment	none

2.28 NADC_INFO

Identifier	NADC_INFO
Author	R.M. van Hees
Language	ANSI C
Purpose	display information on the callers display
Comment	NADC_Info_Proc, NADC_Info_Update, NADC_Info_Finish

2.28.1 NADC_Info_Proc

Identifier	NADC_Info_Proc
Purpose	display information about the progress of a certain routine
Usage	`NADC_Info_Proc( stream, proc_name, counter );`
Input	FILE *stream : pointer to open straem
	char *proc_name : name of the routine
	unsigned int counter : total number of actions performed
Returns	nothing
Comment	none

2.28.2 NADC_Info_Update

Identifier	NADC_Info_Update
Purpose	update message to user, to be called after every call of routine
Usage	`NADC_Info_Update( stream, digits, counter );`
Input	FILE *stream : pointer to open straem
	unsigned short digits : specify number of digits to be displayed
	unsigned int counter : total number of actions performed
Returns	nothing
Comment	none

2.28.3 NADC_Info_Finish

Identifier	NADC_Info_Finish
Purpose	finalise message after routine has finished
Usage	`NADC_Info_Finish( stream, digits, counter );`
Input	FILE *stream : pointer to open straem
	unsigned short digits : specify number of digits to be displayed
	unsigned int counter : total number of actions performed
Returns	nothing
Comment	none

2.29 NADC_INIT_PARAM

Identifier	NADC_INIT_PARAM
Author	R.M. van Hees
Language	ANSI C
Purpose	initializes param-structure with command-line parameters
Returns	Nothing (check global error status)
Comment	None

2.29.1 NADC_INIT_PARAM

Identifier	NADC_INIT_PARAM
Purpose	initializes param-structure
Usage	`NADC_INIT_PARAM( param );`
	`struct param_record (A.1)`
	`*param : struct holding user-defined settings`
Returns	nothing
Comment	none

2.29.2 NADC_SET_PARAM

Identifier	NADC_SET_PARAM
Purpose	initializes struct with command-line parameters
Usage	`NADC_SET_PARAM( argc, argv, instrument, param );`
Input	int argc : number of parameters
	char *argv[] : parameter values
	int instrument : code for instrument en data product level
Output	struct param_record (A.1)
	*param : struct holding user-defined settings
Returns	Nothing (check global error status)
Comment	none

2.30 NADC_INTERPOL

Identifier	NADC_INTERPOL
Author	R.M. van Hees
Language	ANSI C
Purpose	linear interpolation of an whole array
Returns	nothing
Comment	None

2.30.1 NADC_INTERPOL

Identifier	NADC_INTERPOL
Purpose	linear interpolation of an whole array
Usage	`NADC_INTERPOL( X, X_left, X_right, num_Y, Y_left, Y_right, Y );`
Input	float X : X value within range [X_left, X_right]
	float X_left : lower boundary X value: X_left ¡ X_right
	float X_right : upper boundary X value
	unsigned int num_Y : number of Y values to be calculated
	float *Y_left : Y values at X_left
	float *Y_right : Y values at X_right
Output	float *Y : Y values found by linear interpolation
Returns	nothing
Comment	None

2.30.2 NADC_INTERPOL_d

Identifier	NADC_INTERPOL_d
Purpose	linear interpolation of an whole array
Usage	`NADC_INTERPOL_d( X, X_left, X_right, num_Y, Y_left, Y_right, Y );`
Input	float X : X value within range [X_left, X_right]
	float X_left : lower boundary X value: X_left ¡ X_right
	float X_right : upper boundary X value
	unsigned int num_Y : number of Y values to be calculated
	float *Y_left : Y values at X_left
	float *Y_right : Y values at X_right
Output	double *Y : Y values found by linear interpolation
Returns	nothing
Comment	None

2.31 NADC_PYTABLE_API

Identifier	NADC_PYTABLE_API
Author	R.M. van Hees
Language	ANSI C
Purpose	subroutines to easily create/write/read attributes and datasets
Returns	status: negative value is returned on failure
Comment	none

2.31.1 PYTABLE_open_file

Identifier	PYTABLE_open_file
Purpose	Open/Create a PyTable file
Usage	`grpID = PYTABLE_open_file( filename, title );`
Input	char filename[] : file name
	char title[] : title of the PyTable database
Returns	A negative value is returned on failure.
Comment	none

2.31.2 PYTABLE_open_group

Identifier	PYTABLE_open_group
Purpose	Open/Create a group in an exsisting HDF5-file
Usage	`grpID = PYTABLE_open_group( locID, name, sz_hint );`
Input	hid_t locID : HDF5 object id
	char name[] : name of the group
Returns	A negative value is returned on failure.
Comment	none

2.31.3 PYTABLE_make_array

Identifier	PYTABLE_make_array
Purpose	create extensible HDF5 dataset
Usage	`stat = PYTABLE_make_array( locID, dset_name, title, rank, dims, extdim, typeID, dims_chunk, fill_data, compress, shuffle, fletcher32, buff );`
Input	hid_t locID : HDF5 identifier of file or group
	char *dset_name : name of dataset
	char *title :
	int rank : number of dimensions
	hsize_t *dims : size of each dimension
	int extdim : index of expendable dimension
	hid_t typeID : data type (HDF5 identifier)
	hsize_t *dims_chunk : chunk sizes
	void *fill_data : Fill value for data
	unsigned int compress : compression level (zero for no compression)
	bool shuffle : shuffel data for better compression
	bool fletcher32 :
	void *buffer : buffer with data to write (or NULL)
Returns	A negative value is returned on failure.
Comment	none

2.31.4 PYTABLE_append_array

Identifier	PYTABLE_append_array
Purpose	append data to HDF5 dataset, extending the dimension "extdim"
Usage	`stat = PYTABLE_append_array( locID, dset_name, extdim, count, buffer );`
Input	hid_t locID : HDF5 identifier of file or group
	char *dset_name : name of dataset
	int extdim : dimension to extend
	int count : number of arrays to write
	void *buffer : data to write
Returns	A negative value is returned on failure.
Comment	none

2.31.5 PYTABLE_write_records

Identifier	PYTABLE_write_records
Purpose	Write records to an HDF5 array
Usage	`stat = PYTABLE_write_records( locID, dset_name, start, step, count, buffer );`
Input	hid_t locID : HDF5 identifier of file or group
	char *dset_name : name of dataset
	hsize_t *start : index of first row to overwrite
	hsize_t *step :
	hsize_t *count : number of rows to write
	void *buffer : data to write
Returns	A negative value is returned on failure.
Comment	none

2.32 NADC_RECEIVEDATE

Identifier	NADC_RECEIVEDATE
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain modification date from operating system
Usage	`NADC_RECEIVEDATE( flname, receivedate );`
Input	char *flname : name of file
	char *receivedate : receive date (UTC)
Returns	nothing
Comment	None

2.33 NADC_SELECT

Identifier	NADC_SELECT
Author	R.M. van Hees
Language	ANSI C
Purpose	return the kk-th smallest value of an array
Usage	`val = SELECTx( kk, dim, xa );`
Input	size_t kk : index number of the array to return
	size_t dim : dimension of the array to be sorted
	array *xa : pointer to array
Returns	value of the kk-th smallest value of an array
Comment	Contains SELECTs, SELECTi, SELECTr, SELECTd
	”x” specifies the data type of the input array, as follows:
	”s” short
	”i” integer
	”r” float
	”d” double

2.34 NADC_SHOW_PARAM

Identifier	NADC_SHOW_PARAM
Author	R.M. van Hees
Language	ANSI C
Purpose	show command-line settings, as stored in the param-record
Usage	`NADC_SHOW_PARAM( instrument, param );`
Input	int instrument : code for instrument en data product level
	struct param_record (A.1)
	param : struct holding user-defined settings
Returns	Nothing
Comment	None

2.35 NADC_STRING

Identifier	NADC_STRING
Author	R.M. van Hees
Language	ANSI C
Comment	Compatible with *BSD
	contains: strlcpy, strlcat

2.35.1 strlcpy

Identifier	strlcpy
Purpose	copy a NULL-terminated string into a sized buffer
Usage	`nchar = strlcpy( dest, src, size );`
Input	char *src : where to copy the string from
	size_t size : size of destination buffer
Output	char *dest : where to copy the string to
Returns	length of the destination string
Comment	Compatible with *BSD: the result is always a valid
	NULL-terminated string that fits in the buffer (unless,
	of course, the buffer size is zero). It does not pad
	out the result like strncpy() does.

2.35.2 strlcat

Identifier	strlcat
Purpose	append a length-limited, NULL-terminated string to another
Usage	`nchar = strlcat( dest, src, size );`
Input	char *src : where to copy the string from size_t size : size of destination buffer
Output	char *dest : where to copy the string to
Returns	length of the destination string
Comment	Compatible with *BSD: the result is always a valid NULL-terminated string that fits in the buffer (unless, of course, the buffer size is zero). It does not pad out the result like strncpy() does.

2.35.3 rstrip

Identifier	rstrip
Purpose	return copy of string with trailing whitespace removed
Usage	`rstrip( dest, src );`
Input	char *src : where to copy the string from
Output	char *dest : where to copy the string to
Returns	nothing
Comment	none

2.36 NADC_USRINDX

Identifier	NADC_USRINDX
Author	R.M. van Hees
Language	ANSI C
Purpose	extent NADC_USRINP: interpret negative indices relative to maximum index, and '*' as the maximum index
Usage	`nr_indx = NADC_USRINDX( str_range, max_indx, indices );`
Input	char str_range : string holding range specification int max_indx : maximum index
Output	short *indices : array holding indices
Returns	number of indices (short)
Comment	none

2.37 NADC_USRINP

Identifier	NADC_USRINP
Author	R.M. van Hees
Language	ANSI-C
Purpose	Decode a character string into an array with elements of the requested data type
Usage	`NADC_USRINP( type, string, maxval, pntr, nrval );`
Input	valid values are: UINT8_T, INT16_T, UINT16_T, INT32_T, UINT32_T, FLT32_T, FLT_64_T char *string character string int maxval maximum number of elements in the output array
Output	int *nrval actuel number of values found
Returns	nothing (check global error status)
Comment	The input-string has to contain values separated by ”,” or ”:”. The values in the output array are converted to requested type input: string = ”1,4,8” type = FLT32_T returns: pointer to 1.0 4.0 8.0 nrval = 3 input: string = ”.6:2:.4” type = FLT32_T or FLT64_T returns: pointer to 0.6 1.0 1.4 1.8 nrval = 4

2.38 NADC_VERSION

Identifier	NADC_VERSION
Author	R.M. van Hees
Language	ANSI C
Purpose	Returns the library version numbers through arguments.
Usage	`NADCget_Version( majnum, minnum, relnum, reldate );`
Input	unsigned int majnum : major revision number unsigned int minnum : minor revision number unsigned int relnum : release revision number char reldate[] : release date
Returns	nothing
Comment	none

2.39 NADC_WR_ASCII

Identifier	NADC_WR_ASCII
Author	R.M. van Hees
Language	ANSI C
Purpose	display keyword and values uniformly
Comment	contains CRE_ASCII_FILE, CAT_ASCII_FILE, nadc_write_header, nadc_write_xxx, nadc_write_arr_xxx

2.39.1 CRE_ASCII_FILE

Identifier	CRE_ASCII_FILE
Purpose	create new file for writing (or truncate to zero length)
Usage	`outfile = CRE_ASCII_File( fl_name, extension );`
Input	char fl_name[] : name of output file (no extension!!!)
	char extension[] : extension to be concatenated (without ".")
Returns	(open) stream pointer
Comment	none

2.39.2 CAT_ASCII_FILE

Identifier	CAT_ASCII_FILE
Purpose	Open file for writing in append mode (create if non-existing)
Usage	`outfile = CAT_ASCII_File( fl_name, extension );`
Input	char fl_name[] : name of output file (no extension!!!)
	char extension[] : extension to be concatenated (without ".")
Returns	(open) stream pointer
Comment	none

2.39.3 nadc_write_text

Identifier	nadc_write_text
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_text( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	char *key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.4 nadc_write_bool

Identifier	nadc_write_bool
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_bool( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	bool key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.5 nadc_write_schar

Identifier	nadc_write_schar
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_schar( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	signed char key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.6 nadc_write_uchar

Identifier	nadc_write_uchar
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_uchar( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	unsigned char key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.7 nadc_write_short

Identifier	nadc_write_short
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_short( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	short key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.8 nadc_write_ushort

Identifier	nadc_write_ushort
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_ushort( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	unsigned short key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.9 nadc_write_int

Identifier	nadc_write_int
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_int( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	int key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.10 nadc_write_uint

Identifier	nadc_write_uint
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_uint( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	unsigned int key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.11 nadc_write_long

Identifier	nadc_write_long
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_long( fp, key_num, key_wrd, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	long key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.12 nadc_write_float

Identifier	nadc_write_float
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_float( fp, key_num, key_wrd, digits, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	int digits : number of digits (floating point only)
	float key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.13 nadc_write_double

Identifier	nadc_write_double
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_double( fp, key_num, key_wrd, digits, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd : keyword description
	int digits : number of digits (floating point only)
	double key_val : keyword value
Returns	nothing (check global error status)
Comment	none

2.39.14 nadc_write_arr_uchar

Identifier	nadc_write_arr_uchar
Purpose	write keyword and its values in uniform fashion
Usage	`nadc_write_arr_uchar( fp, key_num, key_wrd, val_ndim, val_count, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd[] : keyword description
	int val_ndim : number of dimensions
	Ps. a negative dimension flips the axis
	unsigned int val_count[] : size of the array dimensions
	unsigned char key_val[] : values for this keyword
Returns	nothing (check global error status)
Comment	none

2.39.15 nadc_write_arr_schar

Identifier	nadc_write_arr_schar
Purpose	write keyword and its values in uniform fashion
Usage	`nadc_write_arr_schar( fp, key_num, key_wrd, val_ndim, val_count, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd[] : keyword description
	int val_ndim : number of dimensions
	Ps. a negative dimension flips the axis
	unsigned int val_count[] : size of the array dimensions
	signed char key_val[] : values for this keyword
Returns	nothing (check global error status)
Comment	none

2.39.16 nadc_write_arr_short

Identifier	nadc_write_arr_short
Purpose	write keyword and its values in uniform fashion
Usage	`nadc_write_arr_short( fp, key_num, key_wrd, val_ndim, val_count, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd[] : keyword description
	int val_ndim : number of dimensions
	Ps. a negative dimension flips the axis
	unsigned int val_count[] : size of the array dimensions
	short key_val[] : values for this keyword
Returns	nothing (check global error status)
Comment	none

2.39.17 nadc_write_arr_ushort

Identifier	nadc_write_arr_ushort
Purpose	write keyword and its values in uniform fashion
Usage	`nadc_write_arr_ushort( fp, key_num, key_wrd, val_ndim, val_count, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd[] : keyword description
	int val_ndim : number of dimensions
	Ps. a negative dimension flips the axis
	unsigned int val_count[] : size of the array dimensions
	unsigned short key_val[] : values for this keyword
Returns	nothing (check global error status)
Comment	none

2.39.18 nadc_write_arr_int

Identifier	nadc_write_arr_int
Purpose	write keyword and its values in uniform fashion
Usage	`nadc_write_arr_int( fp, key_num, key_wrd, val_ndim, val_count, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd[] : keyword description
	int val_ndim : number of dimensions
	Ps. a negative dimension flips the axis
	unsigned int val_count[] : size of the array dimensions
	int key_val[] : values for this keyword
Returns	nothing (check global error status)
Comment	none

2.39.19 nadc_write_arr_uint

Identifier	nadc_write_arr_uint
Purpose	write keyword and its values in uniform fashion
Usage	`nadc_write_arr_uint( fp, key_num, key_wrd, val_ndim, val_count, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd[] : keyword description
	int val_ndim : number of dimensions
	Ps. a negative dimension flips the axis
	unsigned int val_count[] : size of the array dimensions
	unsigned int key_val[] : values for this keyword
Returns	nothing (check global error status)
Comment	none

2.39.20 nadc_write_arr_float

Identifier	nadc_write_arr_float
Purpose	write keyword and its values in uniform fashion
Usage	`nadc_write_arr_float ( fp, key_num, key_wrd, val_ndim, val_count, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd[] : keyword description
	int digits : number of digits (floating point only)
	int val_ndim : number of dimensions
	Ps. a negative dimension flips the axis
	unsigned int val_count[] : size of the array dimensions
	float key_val[] : values for this keyword
Returns	nothing (check global error status)
Comment	none

2.39.21 nadc_write_arr_double

Identifier	nadc_write_arr_double
Purpose	write keyword and its values in uniform fashion
Usage	`nadc_write_arr_double( fp, key_num, key_wrd, val_ndim, val_count, key_val );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char key_wrd[] : keyword description
	int digits : number of digits (floating point only)
	int val_ndim : number of dimensions
	Ps. a negative dimension flips the axis
	unsigned int val_count[] : size of the array dimensions
	double key_val[] : values for this keyword
Returns	nothing (check global error status)
Comment	none

2.39.22 nadc_write_header

Identifier	nadc_write_header
Purpose	write keyword and value in uniform fashion
Usage	`nadc_write_header( fp, key_num, infl_name, component );`
Input	FILE *fp : file pointer
	unsigned int key_num : (unique) number for this keyword
	char infl_name[] : name of input file
	char component : name of component
Returns	nothing (check global error status)
Comment	none

2.40 NADC_GET_XML_METADB

Identifier	NADC_GET_XML_METADB
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain host, user, passwd values from NADC configuration file
Usage	`NADC_GET_XML_METADB( fp, host, user, passwd );`
Input	FILE *fp : filename of the NADC configuration file
Output	char *host : hostname with the PostgreSQL server
	char *port : port to connect
	char *user : username of account
	char *passwd : password of account
Returns	Nothing
Comment	None

2.41 SCIA_VERSION

Identifier	SCIA_VERSION
Author	R.M. van Hees
Language	ANSI C
Purpose	Returns the library version numbers through arguments.
Comment	contains SCIAget_Version and SCIAshow_Version

2.41.1 SCIAget_Version

Identifier	SCIAget_Version
Purpose	get version of SCIA library and release date
Usage	`SCIAget_Version( &majnum, &minnum, &relnum, reldate );`
Output	unsigned int *majnum : major revision number
	unsigned int *minnum : minor revision number
	unsigned int *relnum : release revision number
	char *reldate : release date
Returns	nothing
Comment	none

2.41.2 SCIAshow_Version

Identifier	SCIAshow_Version
Purpose	display version of SCIA library and release date
Usage	`SCIAshow_Version( stream, prognm );`
Output	FILE *stream : file pointer
	char *prognm : name of the calling program
Returns	nothing
Comment	none

3 GOME Modules

3.1 GOME_CHK_SIZE

Identifier	GOME_CHK_SIZE
Author	R.M. van Hees
Language	ANSI C
Purpose	check expected product size against size on disk
Usage	`GOME_LV1_CHK_SIZE( const struct fsr1_gome fsr, const char *gomefl )`
Input	struct fsr1_gome : structure for level 1 FSR record
	char *gomefl : file name of GOME level 1 product
Usage	`GOME_LV2_CHK_SIZE( const struct fsr2_gome fsr, const char *gomefl )`
Input	struct fsr2_gome : structure for level 2 FSR record
	char *gomefl : file name of GOME level 1 product
Returns	Nothing, check nadc_err_stat
Comment	None

3.2 GOME_LV0_H5_STRUCTS

Identifier	GOME_LV0_H5_STRUCTS
Author	R.M. van Hees
Language	ANSI C
Purpose	create data structures to store GOME compound data types
Usage	`CRE_GOME_LV1_H5_STRUCTS( param );`
Input	struct param_record param : struct holding user-defined settings
Returns	Nothing
Comment	None

3.3 GOME_LV1_BDR

Identifier	GOME_LV1_BDR
Author	R.M. van Hees
Language	ANSI C
Purpose	calibrate and write selected Earth/Moon/Sun Band Data Records
Usage	`PROCESS_PCD_BDR( nband, param, fsr, fcd, nr_pcd, indx_pcd, pcd, rec );`
Input	short nband : number of spectral band [1a=0,1b,2a..]
	struct param_record param : command-line parameters (A.1)
	struct fsr1_gome *fsr : structure for File Structure Record (B.3)
	struct fcd_gome *fcd : structure for Fixed Calibration Record (B.18)
	short nr_pcd : number of Pixel Calibration Records
	short *indx_pcd : indices to selected PCDs
	struct pcd_gome *pcd : structure for Pixel Calibration Records (B.24)
	struct rec_gome *rec : Spectral Band Records (B.27)
Usage	`PROCESS_SMCD_BDR( flag_origin, nband, param, fcd, nr_scd, indx_scd, scd, rec`
Input	unsigned char flag_origin : Sun or Moon measurements
	short nband : number of spectral band [1a=0,1b,2a..]
	struct param_record param : command-line parameters (A.1)
	struct fcd_gome *fcd : Fixed Calibration Record (B.18)
	short nr_smcd : number of Pixel Calibration Records
	short *indx_scd : indices to selected PCDs
	struct smcd_gome *scd : Sun/Moon Specific Calibration Records (B.25)
	struct rec_gome *rec : Spectral Band Records (B.27)
Returns	Nothing
Comment	None

3.4 GOME_LV1_CAL_BDR

Identifier	GOME_LV1_CAL_BDR
Author	R.M. van Hees
Language	ANSI C
Purpose	calibrate Earth, Moon, Sun Spectral Band Data Records
Usage	`CALIB_PCD_BDR( calib_flag, nband, fcd, pcd, nr_rec, rec );`
Input	unsigned short calib_flag : calibration flag
	short nband : number of spectral band [1a=0,1b,2a..]
	struct fcd_gome *fcd : Fixed Calibration Data Record (B.18)
	struct pcd_gome *pcd : Pixel Specific Calibration Records (B.24)
	short nr_rec : number of band data records
Output	struct rec_gome *rec : Spectral Band Data Records (B.27)
Usage	`CALIB_SMCD_BDR( calib_flag, nband, fcd, smcd, nr_rec, rec );`
Input	unsigned short calib_flag : calibration flag
	short nband : number of spectral band [1a=0,1b,2a..]
	struct fcd_gome *fcd : Fixed Calibration Data Record (B.18)
	struct smcd_gome *smcd : Sun/Moon Specific Calibration Records (B.25)
	short nr_rec : number of band data records
Output	struct rec_gome *rec : Spectral Band Data Records (B.27)
Returns	Nothing
Comment	calibration algorithms taken from the DLR GDP version 2.

3.5 GOME_LV1_CAL_PMD

Identifier	GOME_LV1_CAL_PMD
Author	R.M. van Hees
Language	ANSI C
Purpose	calibrate EARTH or Moon/SUN PMD data
Usage	`CALIB_PCD_PMD( param, fcd, pcd, nr_pmd, pmd );`
Input	struct param_record param : struct holding user-defined settings
	struct fcd_gome *fcd : Fixed Calibration Data Record (B.18)
	struct pcd_gome *pcd : Pixel Specific Calibration Records (B.24)
	size_t nr_pmd : number of selected PMD structures
In/Output	struct pmd_gome *pmd : structure with PMD data (B.23)
Usage	`CALIB_SMCD_PMD( param, fcd, smcd, nr_pmd, pmd );`
Input	struct param_record param : struct holding user-defined settings
	struct fcd_gome *fcd : Fixed Calibration Data Record (B.18)
	struct smcd_gome *smcd : Sun/Moon Specific Calibration Records (B.25)
	size_t nr_pmd : number of selected PMD structures
In/Output	struct pmd_gome *pmd : structure with PMD data (B.23)
Returns	Nothing
Comment	None

3.6 GOME_LV1_PMD_GEO

Identifier	GOME_LV1_PMD_GEO
Author	R.M. van Hees
Language	ANSI C
Purpose	copy PMD records from IHR record for calibration
Usage	`GOME_LV1_PMD_GEO( write_pmd_geo, nr_pcd, indx_pcd, pcd, &pmd );`
Input	struct fcd_gome fcd[] : Fixed Calibration Data Record
	int nr_pcd : number of selected PCD records
	short indx_pcd[] : indices to selected PCD records
In/Output	struct pcd_gome pcd[] : Pixel Specific Calibration Records (B.24)
Returns	Nothing
Comment	None

3.7 GOME_LV1_RD_BDR

Identifier	GOME_LV1_RD_BDR
Author	R.M. van Hees
Language	ANSI C
Purpose	read data of a BCR including sub-channel data (for split-channels)
Usage	`nr_rec = GOME_LV1_RD_BDR( infl, nband, fsr, fcd, &rec_out );`
Input	FILE *infl : (open) file descriptor
	short nband : number of the spectral band [1a=0,1b,2a..]
	struct fsr1_gome *fsr : structure for the FSR record (B.3)
	struct fcd_gome *fcd : Fixed Calibration Data Record (B.18)
Output	struct rec_gome *rec : structure for a spectral band data record (B.27)
Returns	nothing
Comment	None

3.8 GOME_LV1_RD_FCD

Identifier	GOME_LV1_RD_FCD
Author	R.M. van Hees
Language	ANSI C
Purpose	Read the Fixed Calibration Record
Usage	`num_fcd = GOME_LV1_RD_FCD( infl, fsr, &fcd );`
Input	FILE *infl : (open) file descriptor
	struct fsr1_gome *fsr : structure for the FSR record (B.3)
Output	struct fcd_gome *fcd : structure for the FCD record (B.18)
Returns	number of FCD records found (modifies error status)
Comment	None

3.9 GOME_LV1_RD_FSR

Identifier	GOME_LV1_RD_FSR
Author	R.M. van Hees
Language	ANSI C
Purpose	Read the File Structure Record
Usage	`GOME_LV1_RD_FSR( infl, &fsr );`
Input	FILE *infl : (open) file descriptor
Output	struct fsr1_gome *fsr : structure for the FSR record (B.3)
Returns	nothing: modifies error status
Comment	None

3.10 GOME_LV1_RD_PCD

Identifier	GOME_LV1_RD_PCD
Author	R.M. van Hees
Language	ANSI C
Purpose	Read a Pixel Specific Calibration record
Usage	`nr_pcd = GOME_LV1_RD_PCD( infl, fsr, sph, &pcd );`
Input	FILE *infl : (open) file descriptor
	struct fsr1_gome *fsr : structure for the FSR record (B.3)
	struct sph1_gome *sph : structure for the SPH record (B.5)
Output	struct pcd_gome **pcd : structure for the PCD record (B.24)
Returns	number of PCD records found (modifies error status)
Comment	None

3.11 GOME_LV1_RD_SMCD

Identifier	GOME_LV1_RD_SMCD
Author	R.M. van Hees
Language	ANSI C
Purpose	read a Sun/Moon Specific Calibration record
Usage	`nr_smcd = GOME_LV1_RD_SMCD( infl, fsr, sph, &smcd );`
Input	FILE *infl : (open) file descriptor
	struct fsr1_gome *fsr : structure for the FSR record (B.3)
	struct sph1_gome *sph : structure for the SPH record (B.5)
Output	struct smcd_gome **smcd : structure for the SMCD record(s) (B.25)
Returns	number of SMCD records found (modifies error status)
Comment	None

3.12 GOME_LV1_RD_SPH

Identifier	GOME_LV1_RD_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	Read the Specific Product Header
Usage	`GOME_LV1_RD_SPH( infl, fsr, &sph );`
Input	FILE *infl : (open) file descriptor
	struct fsr1_gome *fsr : structure for the FSR record (B.3)
Output	struct sph1_gome *sph : structure for the SPH (B.5)
Returns	nothing: modifies global error status
Comment	None

3.13 GOME_LV1_WR_ASCII

Identifier	GOME_LV1_WR_ASCII
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump the contents of a GOME level 1b file in ASCII
Returns	Nothing
Comment	contains GOME_LV1_WR_ASCII_FSR, GOME_LV1_WR_ASCII_SPH, GOME_LV1_WR_ASCII_FCD, GOME_LV1_WR_ASCII_PCD, GOME_LV1_WR_ASCII_SMCD, GOME_LV1_WR_ASCII_REC

3.13.1 GOME_LV1_WR_ASCII_FSR

Identifier	GOME_LV1_WR_ASCII_FSR
Purpose	dump content of File Structure Record
Usage	`GOME_LV1_WR_ASCII_FSR( param, fsr );`
Input	struct param_record param : struct holding user-defined settings
	struct fsr1_gome *fsr : structure for FSR record (B.3)
Returns	nothing (check global error status)
Comment	none

3.13.2 GOME_LV1_WR_ASCII_SPH

Identifier	GOME_LV1_WR_ASCII_SPH
Purpose	dump content of Specific Product Header
Usage	`GOME_LV1_WR_ASCII_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph1_gome *sph : structure for Specific Product Header (B.5)
Returns	nothing (check global error status)
Comment	none

3.13.3 GOME_LV1_WR_ASCII_FCD

Identifier	GOME_LV1_WR_ASCII_FCD
Purpose	dump content of Fixed Calibration Data
Usage	`GOME_LV1_WR_ASCII_FCD( param, fcd );`
Input	struct param_record param : struct holding user-defined settings
	struct fcd_gome *fcd : structure for Fixed Calibration Data (B.18)
Returns	nothing (check global error status)
Comment	none

3.13.4 GOME_LV1_WR_ASCII_PCD

Identifier	GOME_LV1_WR_ASCII_PCD
Purpose	dump content of (Earth) Pixel Calibration Data
Usage	`GOME_LV1_WR_ASCII_PCD( param, nr_pcd, indx_pcd, pcd );`
Input	struct param_record param : struct holding user-defined settings
	short nr_pcd : number of Pixel Calibration Records
	short *indx_pcd : indices to selected PCDs
	struct pcd_gome *pcd : structure for Pixel Calibration Records (B.24)
Returns	nothing (check global error status)
Comment	none

3.13.5 GOME_LV1_WR_ASCII_SMCD

Identifier	GOME_LV1_WR_ASCII_SMCD
Purpose	dump content of Sun/Moon Pixel Calibration Data records
Usage	`GOME_LV1_WR_ASCII_SMCD( flag_origin, param, nr_smcd, indx_smcd, smcd );`
Input	unsigned char flag_origin : Sun or Moon measurements
	struct param_record param : struct holding user-defined settings (A.1)
	short nr_smcd : number of Pixel Calibration Records
	short *indx_smcd : indices to selected SMCDs
	struct smcd_gome *smcd : Sun/Moon Specific Calibration Records (B.25)
Returns	nothing (check global error status)
Comment	none

3.13.6 GOME_LV1_WR_ASCII_REC

Identifier	GOME_LV1_WR_ASCII_REC
Purpose	

Usage	`GOME_LV1_WR_ASCII_REC( unsigned char flag_origin, short nband,`
	`struct param_record param, (A.1)`
	`short nr_rec, short bcr_start, short bcr_count,`
	`const struct rec_gome *rec );`

Input	unsigned char flag_origin : Earth, Sun or Moon measurements
	short nband : number of spectral band [1a=0,1b,2a..]
	struct param_record param : struct holding user-defined settings (A.1)
	short nr_rec : number of measurement data records
	short bcr_start : first pixel of spectral band
	short bcr_count : number of pixels per spectral band
	struct rec_gome *rec : measurement data records (B.27)
Returns	nothing (check global error status)
Comment	none

3.14 GOME_LV1_WR_H5_FCD

Identifier	GOME_LV1_WR_H5_FCD
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 1 FCD data
Usage	`GOME_LV1_WR_H5_FCD( param, &fcd );`
Input	struct param_record param : struct holding user-defined settings
	struct fcd_gome *fcd : Fixed Calibration Data Record (B.18)
Returns	Nothing
Comment	None

3.15 GOME_LV1_WR_H5_PCD

Identifier	GOME_LV1_WR_H5_PCD
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 1 PCD data
Usage	`GOME_LV1_WR_H5_PCD( param, nr_pcd, indx_pcd, pcd );`
Input	struct param_record param : struct holding user-defined settings
	int nr_pcd : number of selected PCD records
	short *indx_pcd : indices to selected PCD records
	struct pcd_gome *pcd : Pixel Specific Calibration Record(s) (B.24)
Returns	Nothing
Comment	None

3.16 GOME_LV1_WR_H5_REC

Identifier	GOME_LV1_WR_H5_REC
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 1 REC data
Usage	`GOME_LV1_WR_H5_REC( flag_origin, param, band,` `nr_rec, bcr_start, bcr_count, rec );`
Input	unsigned char flag_origin : FLAG_EARTH/FLAG_MOON/FLAG_SUN
	short nband : band index [1a,1b,2a,2b,3,4,...]
	struct param_record param : command-line parameters (A.1)
	short nr_rec : number of band data records
	short bcr_start,bcr_count: number of pixels per spectral band
	struct rec_gome *rec : Spectral Band Data Records (B.27)
Returns	Nothing
Comment	None

3.17 GOME_LV1_WR_H5_SMCD

Identifier	GOME_LV1_WR_H5_SMCD
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 1 SMCD data to file
Usage	`GOME_LV1_WR_H5_SMCD( flag_origin, param, nr_smcd, smcd );`
Input	unsigned char flag_origin : FLAG_SUN or FLAG_MOON
	struct param_record param : command-line parameters (A.1)
	int nr_smcd : size of SMCD array
	struct smcd_gome *smcd : Sun/Moon Specific Calibration Record (B.25)
Returns	Nothing
Comment	None

3.18 GOME_LV1_WR_H5_SPH

Identifier	GOME_LV1_WR_H5_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 1 SPH data
Usage	`GOME_LV1_WR_H5_SPH( param, &sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph1_gome *sph : Specific Product Header data (B.5)
Returns	Nothing
Comment	None

3.19 GOME_LV2_RD_DDR

Identifier	GOME_LV2_RD_DDR
Author	R.M. van Hees
Language	ANSI C
Comment	contains GOME_LV2_RD_DDR

3.19.1 GOME_LV2_RD_DDR

Identifier	GOME_LV2_RD_DDR
Purpose	Read a Spectral Band Record
Usage	`nr_ddr = GOME_LV2_RD_DDR( infl, fsr, sph, &ddr );`
Input	int fid : (open) file descriptor
	struct fsr2_gome *fsr : structure for the FSR record (B.4)
	struct sph2_gome *sph : Specific Product Header (B.6)
Output	struct ddr_gome **ddr : structure for a DOAS data record (B.30)
Returns	number of DDR record read
Comment	none

3.20 GOME_LV2_RD_FSR

Identifier	GOME_LV2_RD_FSR
Author	R.M. van Hees
Language	ANSI C
Purpose	Read the File Structure Record
Usage	`GOME_LV2_RD_FSR( infl, &fsr );`
Input	FILE *infl : (open) file descriptor
Output	struct fsr2_gome *fsr : structure for the FSR record (B.4)
Comment	None

3.21 GOME_LV2_RD_SPH

Identifier	GOME_LV2_RD_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	Read the Specific Product Header
Usage	`nr_byte = GOME_LV2_RD_SPH( fid, fsr, &sph );`
Input	int *infl : (open) file descriptor
	struct fsr2_gome *fsr : structure for the FSR record (B.4)
Output	struct sph2_gome *sph : structure for the SPH (B.6)
Returns	number of bytes read (unsigned int)
	modifies global error status
Comment	None

3.22 GOME_LV2_WR_ASCII

Identifier	GOME_LV2_WR_ASCII
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump the contents of a GOME level 2 file in ASCII
Returns	Nothing
Comment	contains GOME_LV2_WR_ASCII_FSR, GOME_LV2_WR_ASCII_SPH, GOME_LV2_WR_ASCII_DDR

3.22.1 GOME_LV2_WR_ASCII_FSR

Identifier	GOME_LV2_WR_ASCII_FSR
Purpose	dump content of File Structure Record
Usage	`GOME_LV2_WR_ASCII_FSR( param, fsr );`
Input	struct param_record param : struct holding user-defined settings
	struct fsr2_gome *fsr : structure for FSR record (B.4)
Returns	nothing (check global error status)
Comment	none

3.22.2 GOME_LV2_WR_ASCII_SPH

Identifier	GOME_LV2_WR_ASCII_SPH
Purpose	dump content of Specific Product Header
Usage	`GOME_LV2_WR_ASCII_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph2_gome *sph : structure for Specific Product Header (B.6)
Returns	nothing (check global error status)
Comment	none

3.22.3 GOME_LV2_WR_ASCII_DDR

Identifier	GOME_LV2_WR_ASCII_DDR
Purpose	dump content of GOME Lv2 data record
Usage	`GOME_LV2_WR_ASCII_DDR( param, sph, nr_ddr, ddr );`
Input	struct param_record param : struct holding user-defined settings
	struct sph2_gome sph : structure with SPH record (B.6)
	short nr_ddr : number of DDR records
	struct ddr_gome *ddr : pointer to structures with DDR records (B.30)
Returns	nothing (check global error status)
Comment	none

3.23 GOME_LV2_WR_H5_DDR

Identifier	GOME_LV2_WR_H5_DDR
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 2 DDR data
Usage	`GOME_LV2_WR_H5_DDR( param, nr_ddr, ddr );`
Input	struct param_record param : struct holding user-defined settings
	int nr_ddr : number of DDR records
	struct ddr_gome *ddr : DOAS Data records (B.30)
Returns	Nothing
Comment	None

3.24 GOME_LV2_WR_H5_IRR

Identifier	GOME_LV2_WR_H5_IRR
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 2 IRR data
Usage	`GOME_LV2_WR_H5_IRR( param, nr_ddr, ddr );`
Input	struct param_record param : struct holding user-defined settings
	int nr_ddr : number of ddr records
	struct ddr_gome *ddr : DOAS Data records (B.30)
Returns	Nothing
Comment	None

3.25 GOME_LV2_WR_H5_SPH

Identifier	GOME_LV2_WR_H5_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 2 SPH data
Usage	`GOME_LV2_WR_H5_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph_gome *sph : Specific Product Header data (**??**)
Returns	Nothing
Comment	None

3.26 GOME_RD_PIR

Identifier	GOME_RD_PIR
Author	R.M. van Hees
Language	ANSI C
Purpose	Read the Product Identifier Record
Usage	`GOME_RD_PIR( infl, &pir );`
Input	FILE *infl : (open) file descriptor
Output	struct pir_gome *pir : structure for the PIR (B.2)
Returns	nothing
Comment	None

3.27 GOME_SELECT

Identifier	GOME_SELECT
Author	R.M. van Hees
Language	ANSI C
Purpose	select data according to selection criteria
Returns	one if setected, else zero
Comment	holds SELECT_BAND, SELECT_PCD, SELECT_SMCD, SELECT_DDR

3.27.1 BandChannel

Identifier	BandChannel
Purpose	returns GOME spectral band for given channel, subchannel
Usage	`nband = BandChannel( channel, short subchannel );`
Input	short channel : GOME channel [1,2,3,4]
	short subchannel : GOME sub-channel [1=a,2=b]
Returns	GOME spectral band index (starts at zero)
Comment	none

3.27.2 SELECT_BAND

Identifier	SELECT_BAND
Purpose	check(s) if a spectral band is selected
Usage	`if ( SELECT_BAND( nband, param, fcd ) ) ...`
Input	short nband : number of spectral band [1a=0,1b,2a..]
	struct param_record param : command-line parameters (A.1)
	struct fcd_gome *fcd : Fixed Calibration Record (B.18)
Returns	true, is selected
Comment	none

3.27.3 SELECT_PCD

Identifier	SELECT_PCD
Purpose	checks if a PCD record is selected
Usage	`nr_select += SELECT_PCD( param, &pcd );`
Input	struct param_record param : command-line parameters
	struct pcd_gome *pcd : Pixel Specific Calibration Record (B.24)
Returns	one if pixel is selected, else zero
Comment	none

3.27.4 SELECT_SMCD

Identifier	SELECT_SMCD
Purpose	check(s) if a SMCD is selected
Usage	`nr_select += SELECT_SMCD( param, &smcd );`
Input	struct param_record param : command-line parameters
	struct smcd_gome *smcd : Sun/Moon Specific Calibration Record (B.25)
Returns	not selected: zero, selected one
Comment	none

3.27.5 SELECT_DDR

Identifier	SELECT_DDR
Purpose	check(s) if a DDR is selected
Usage	`nr_select += SELECT_DDR( param, glr );`
Input	struct param_record param : struct holding user-defined settings
	struct glr2_gome *glr : DOAS Data Record (geolocation part) (B.8)
Returns	one if setected, else zero
Comment	none

3.28 GOME_WR_H5_PIR

Identifier	GOME_WR_H5_PIR
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write GOME level 1 PIR data
Usage	`GOME_WR_H5_PIR( param, &pir );`
Input	struct param_record param : struct holding user-defined settings
	struct pir_gome *pir : Product Identifier Record data (B.2)
Returns	Nothing
Comment	None

3.29 GOME_WR_H5_VERSION

Identifier	GOME_WR_H5_VERSION
Author	R.M. van Hees
Language	ANSI C
Purpose	add version of the software to a HDF5 file
Usage	`GOME_WR_H5_VERSION( file_id );`
Input	hid_id file_id : HDF5 file identifier
Returns	Nothing
Comment	None

4 SCIAMACHY Modules

4.1 GET_SCIA_LV0_MDS

Identifier	GET_SCIA_LV0_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	read one state with Sciamachy level 0 MDS AUX/DET/PMD records
Returns	number of records read
Comment	contains GET_SCIA_LV0_STATE_AUX, GET_SCIA_LV0_STATE_DET, GET_SCIA_LV0_STATE_PMD

4.1.1 GET_SCIA_LV0_STATE_AUX

Identifier	GET_SCIA_LV0_STATE_AUX
Purpose	read all Auxiliary MDS records of one state
Usage	`nrec = GET_SCIA_LV0_STATE_AUX( fd, info, numInfo, &aux );`
Input	FILE *fd : file pointer
	struct mds0_info *info : structure holding info about MDS records (D.18)
	unsigned int numInfo : number of info records
Output	struct mds0_aux **aux : auxiliary MDS records (D.19)
Returns	number of MDS recods read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.1.2 GET_SCIA_LV0_STATE_DET

Identifier	GET_SCIA_LV0_STATE_DET
Purpose	read all Detector MDS records of one state
Usage	`nrec = GET_SCIA_LV0_STATE_DET( fd, info, numInfo, &det );`
Input	FILE *fd : file pointer
	struct mds0_info *info : structure holding info about MDS records (D.18)
	unsigned int numInfo : number of info records
Output	struct mds0_det **det : detector MDS records (D.20)
Returns	number of MDS recods read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.1.3 GET_SCIA_LV0_STATE_PMD

Identifier	GET_SCIA_LV0_STATE_PMD
Purpose	read all PMD MDS records of one state
Usage	`nrec = GET_SCIA_LV0_STATE_PMD( fd, info, numInfo, &pmd );`
Input	FILE *fd : file pointer
	struct mds0_info *info : structure holding info about MDS records (D.18)
	unsigned int numInfo : number of info records
Output	struct mds0_pmd **pmd : PMD MDS records (D.21)
Returns	number of MDS recods read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.2 GET_SCIA_LV0_MDS_HK

Identifier	GET_SCIA_LV0_MDS_HK
Author	R.M. van Hees
Language	ANSI C
Purpose	Extract house keeping data from L0 MDS records
Comment	contains: GET_SCIA_LV0_DET_PET, GET_SCIA_LV0_MDS_ANGLES, GET_SCIA_LV0_MDS_TEMP, GET_SCIA_LV0_MDS_TIME, GET_SCIA_LV0_STATE_ANGLE, GET_SCIA_LV0_STATE_DETtemp, GET_SCIA_LV0_STATE_OBMtemp, GET_SCIA_LV0_STATE_PMDtemp

4.2.1 GET_SCIA_LV0_MDS_TIME

Identifier	GET_SCIA_LV0_MDS_TIME
Purpose	calculate Julian day as fraction number of days since 01-01-2000
Usage	`jday = GET_SCIA_LV0_MDS_TIME( source, mds_lv0 );`
Input	int source : type of L0 MDS record void *mds_lv0 : L0 MDS records (AUX, DET of PMD)
Returns	Julian day (double) of the first valid packet
Comment	none

4.2.2 GET_SCIA_LV0_MDS_TIME_ARR

Identifier	GET_SCIA_LV0_MDS_TIME_ARR
Purpose	calculate Julian day as fraction number of days since 01-01-2000 Detector MDS: time when a packet was assembled $T = T_{ICU} + bcps/16.$ Auxiliary MDS: time when a BCP was assembled $T[0:79] = T_{ICU} + bcps[0:79]/16.$ PMD MDS: time when a PMD data packet was assembled $T[0:199] = T_{ICU} + bcps[0:199]/16. + (2e-3 * delta_T - 12.5)/1e3$
Usage	`GET_SCIA_LV0_MDS_TIME_ARR( source, mds_lv0, jday );`
Input	int source : type of L0 MDS record void *mds_lv0 : L0 MDS records (AUX, DET of PMD)
Output	double *jday : Julian day
Returns	nothing
Comment	none

4.2.3 GET_SCIA_LV0_MDS_ANGLES

Identifier	GET_SCIA_LV0_MDS_ANGLES
Purpose	calculates the ASM and ASM mirrors
Usage	`GET_SCIA_LV0_MDS_ANGLES( aux, asmAngle, esmAngle );`
Input	struct mds0_aux *aux : L0 Auxiliary MDS record
Output	float *asmAngle : azimuth scan angles float *esmAngle : elevation scan angles
Returns	nothing
Comment	none

4.2.4 GET_SCIA_LV0_STATE_ANGLE

Identifier	GET_SCIA_LV0_STATE_ANGLE
Purpose	average scan angles during a number of L0 auxiliary packages
Usage	`GET_SCIA_LV0_STATE_ANGLE( num_aux, aux, &asmAngle, &esmAngle );`
Input	unsigned short num_aux : number of auxiliary packages
	struct mds0_aux *aux : L0 Auxiliary MDS records (D.19)
Output	float *asmAngle : average azimuth scan angle
	float *esmAngle : average elevation scan angle
Returns	nothing
Comment	none

4.2.5 GET_SCIA_LV0_STATE_OBMtemp

Identifier	GET_SCIA_LV0_STATE_OBMtemp
Purpose	average OBM temperature during a number of L0 auxiliary packages
Usage	`GET_SCIA_LV0_STATE_OBMtemp( sost_t_obm, num_aux, aux, &obmTemp );`
Input	bool sost_obm : return OBM temperature according to SOST
	otherwise return (wrong) SDMF pre-v3.1
	unsigned short num_aux : number of auxiliary packages
	struct mds0_aux *aux : L0 Auxiliary MDS records (D.19)
Output	float *obmTemp : temperature of the optical bench [K]
Returns	nothing
Comment	none

4.2.6 GET_SCIA_LV0_STATE_DETtemp

Identifier	GET_SCIA_LV0_STATE_DETtemp
Purpose	average Detector block temperature during a number of L0 packages
Usage	`GET_SCIA_LV0_STATE_DETtemp( num_mds, det, detTemp );`
Input	unsigned short num_mds : number of MDS records
	struct mds0_det *det : L0 Detector MDS records (D.20)
Output	float *detTemp : temperature of the detector blocks [K]
Returns	nothing
Comment	none

4.2.7 GET_SCIA_LV0_STATE_PMDtemp

Identifier	GET_SCIA_LV0_STATE_PMDtemp
Purpose	average PMD temperature during a number of L0 PMD packages
Usage	`GET_SCIA_LV0_STATE_PMDtemp( num_pmd, pmd, &pmdTemp );`
Input	unsigned short num_pmd : number of PMD packages
	struct mds0_pmd *pmd : L0 PMD MDS records (D.21)
Output	float *pmdTemp : temperature of the PMD block [K]
Returns	nothing
Comment	none

4.2.8 GET_SCIA_LV0_DET_PET

Identifier	GET_SCIA_LV0_DET_PET
Purpose	get pixel exposure time
Usage	`GET_SCIA_LV0_DET_PET( chan_hdr, pet, &vir_chan_b );`
Input	struct chan_hdr hdr : L0 MDS channel header
Output	float *pet : pixel exposure time [sec] if (*vir_chan_b != 0) pet = fltarr(2) unsigned short *vir_chan_b : virtual channel indicator
Returns	nothing
Comment	none

4.3 GET_SCIA_LV0_MDS_INFO

Identifier	GET_SCIA_LV0_MDS_INFO
Author	R.M. van Hees
Language	ANSI C
Purpose	query the contents a SCIAMACHY level 0 product
Usage	`nr_info = GET_SCIA_LV0_MDS_INFO( fd, mph, dsd, &info );`
Input	FILE *fd : (open) stream pointer struct mph_envi mph : Envisat main product header (A.5) struct dsd_envi *dsd : structure for the DSD records (A.6)
Output	struct mds0_info *info : structure holding info about MDS records (D.18)
Returns	number of MDS's read (unsigned int), error status passed by global variable "nadc_stat'
Comment	None

4.4 GET_SCIA_LV0C_MDS

Identifier	GET_SCIA_LV0C_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	convert L0 MDS_DET to L1C structure
Usage	`nr_mds = GET_SCIA_LV0C_MDS( nr_det, det, mds_1c );`
Input	unsigned int nr_det : number of MDS_DET structures struct mds0_det *det : Detector MDS records (level 0) (D.20)
Output	struct mds1c_scia *mds_1c : level 1c MDS records (C.11)
Returns	number of level 1c MDS records
Comment	None

4.5 GET_SCIA_LV1C_GEO

Identifier	GET_SCIA_LV1C_GEO
Author	R.M. van Hees
Language	ANSI C
Purpose	convert/copy gelocation records level 1b to level 1c format
Usage	`GET_SCIA_LV1C_GEON( nr_geo, geoN, nr_geo1c, geoN_1c );`
Input	unsigned int nr_geo : number of level 1b geo-records
	struct geoN_scia *geoN : Nadir gelocation records (level 1b) (C.6)
	unsigned int nr_geo1c : number of level 1c geo-records
Output	struct geoN_scia *geoN_1c : Nadir gelocation records (level 1c) (C.6)
Usage	`GET_SCIA_LV1C_GEOL( nr_geo, geoL, nr_geo1c, geoL_1c )`
Input	unsigned int nr_geo : number of level 1b geo-records
	struct geoL_scia *geoL : Limb gelocation records (level 1b) (C.5)
	unsigned int nr_geo1c : number of level 1c geo-records
Output	struct geoL_scia *geoL_1c : Limb gelocation records (level 1c) (C.5)
Usage	`GET_SCIA_LV1C_GEOC( nr_geo, geoC, nr_geo1c, geoC_1c )`
Input	unsigned int nr_geo : number of level 1b geo-records
	struct geoC_scia *geoC : Monitor gelocation records (level 1b) (C.7)
	unsigned int nr_geo1c : number of level 1c geo-records
Output	struct geoC_scia *geoC_1c : Monitor gelocation records (level 1c) (C.7)
Returns	nothing
Comment	None

4.6 GET_SCIA_LV1C_MDS

Identifier	GET_SCIA_LV1C_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	convert MDS in level 1b format to level 1c format
Usage	`nr_mds = GET_SCIA_LV1C_MDS( state, mds_1b, mds_1c );`
Input	struct state1_scia *state : structure with States of the product
	struct mds1_scia *mds_1b : structure holding level 1b MDS records (E.31)
Output	struct mds1c_scia *mds_1c : level 1c MDS records (C.11)
Returns	number of level 1c MDS records
Comment	none

4.7 GET_SCIA_LV1C_PMD

Identifier	GET_SCIA_LV1C_PMD
Author	R.M. van Hees
Language	ANSI C
Purpose	extract PMD data from the level 1b MDS
Usage	`nr_mds = GET_SCIA_LV1C_PMD( state, mds_1b, mds_pmd );`
Input	struct state1_scia *state : structure with States of the product
	struct mds1_scia *mds_1b : structure holding level 1b MDS records (E.31)
Output	struct mds1c_pmd *mds_pmd : level 1c PMD MDS record (C.12)
Returns	number of PMD records (=1)
Comment	none

4.8 GET_SCIA_LV1C_POLV

Identifier	GET_SCIA_LV1C_POLV
Author	R.M. van Hees
Language	ANSI C
Purpose	extract polarisation datasets from the level 1b MDS
Usage	`nr_mds = GET_SCIA_LV1C_POLV( state, mds_1b, mds_polV );`
Input	struct state1_scia *state : structure with States of the product
	struct mds1_scia *mds_1b : structure holding level 1b MDS records (E.31)
Output	struct mds1c_polV *mds_polV : level 1c Polarisation MDS record (C.13)
Returns	number of polarisation records (=1)
Comment	none

4.9 GET_SCIA_MAGIC_ID

Identifier	GET_SCIA_MAGIC_ID
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain unique product ID for SCIA level 0 products
Usage	`id = GET_SCIA_MAGIC_ID( productName );`
Input	char *productName : name of the Sciamachy product
Returns	unique number (unsigned int)
Comment	None

4.10 GET_SCIA_MDS1_DATA

Identifier	GET_SCIA_MDS1_DATA
Author	R.M. van Hees
Language	ANSI C
Purpose	
Usage	`nobs = GET_SCIA_MDS1_DATA( PmdScaling, chanID, state, mds_1b, pmd_1c, mds_1c, &sign_out );`
Input	bool PmdScaling : boolean flag to indicate PMD scaling
	(requires mds_1b or pmd_1c)
	unsigned char chanID : channel ID
	struct state1_scia *state : structure with States of the product (E.22)
	struct mds1_scia *mds_1b : Level 1b MDS records (or NULL) (E.31)
	struct mds1c_pmd *pmd_1c : Level 1c PMD_MDS records (or NULL) (C.12)
	struct mds1c_scia *mds_1c : Level 1c MDS records (C.11)
Output	float **sign_out : Science data of one state
Returns	number of observations (unsigned short)
Comment	None

4.10.1 RESAMPLE_1B_PmdVal

Identifier	RESAMPLE_1B_PmdVal
Purpose	re-sample PMD readouts to pixel integration time
Usage	`RESAMPLE_1B_PmdVal( numDsr, mds_1b, IndxPmd, numPmd, pixelPmd );`
Input	unsigned int numDsr : number of dataset records
	struct mds1_scia *mds_1b : level 1b MDS records (E.31)
	unsigned short indxPmd : Index to PMD [0...5]
	unsigned short numPmd : requested number of PMD values
Output	float *pixelPmd : re-sampled PMD values
Returns	nothing
Comment	none

4.10.2 RESAMPLE_1C_PmdVal

Identifier	RESAMPLE_1C_PmdVal
Purpose	re-sample PMD readouts to pixel integration time
Usage	`RESAMPLE_1C_PmdVal( pmd_1c, IndxPmd, numPmd, pixelPmd );`
Input	struct mds1c_pmd *pmd_1c : Level 1c PMD_MDS records
	unsigned short indxPmd : Index to PMD [0...5]
	unsigned short numPmd : requested number of PMD values
Output	float *pixelPmd : re-sampled PMD values
Returns	nothing
Comment	none

4.10.3 get_weight_Factors

Identifier	get_weight_Factors
Purpose	calculate weight factors
Usage	`get_weight_Factors( sampling, numVal, Val, wghtFactors );`
Input	unsigned short sampling : number of values in sample
	unsigned short numVal : number of Values
	float *Val : values
Output	float *wghtFactor : weight Factors
Returns	nothing
Comment	none

4.11 GET_SCIA_MEAN_STATE

Identifier	GET_SCIA_MEAN_STATE
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain statistics on cluster data from a state
Usage	`GET_SCIA_MEAN_STATE( do_corr_coaddf, num_mds, mds, count, coaddf, pet, mean, sdev );`
Input	bool do_corr_coaddf : correct for co-adding of readouts unsigned short num_mds : number of level 1c MDS records struct mds1c_scia *mds : level 1c MDS records (C.11)
Output	unsigned short *count : number of readouts unsigned char *coaddf : co-adding factor float *pet : pixel exposure time float *mean : average signal float *sdev : standard deviation of the signal
Returns	Nothing
Comment	None

4.12 GET_SCIA_QUALITY

Identifier	GET_SCIA_QUALITY
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain data quality
Usage	`res = GET_SCIA_QUALITY( orbit, period );`
Input	int orbit : absolute orbit number
Output	int *period : first and last orbit affected by event
Returns	data quality: (unsigned short) 0 = SCIA_Q_OK 1 = SCIA_Q_DECON 2 = SCIA_Q_RECOVER 3 = SCIA_Q_UNAVAIL 255 = SCIA_Q_UNKOWN
Comment	info taken from the SOST pages http://atmos.caf.dlr.de/projects/scops/

4.13 GET_SCIA_ROE_INFO

Identifier	GET_SCIA_ROE_INFO
Author	R.M. van Hees
Language	ANSI C
Purpose	get orbit phase and SAA flag from ROE records
Usage	`GET_SCIA_ROE_INFO( eclipseMode, jday, &absOrbit, &saaFlag, &orbitPhase );`
Input	bool eclipseMode : TRUE - orbit phase used for SDMF (v2.4) FALSE - orbit phase as used by ESA double jday : julian Day (# days since 2000-01-01)
Output	int *absOrbit : absolute orbit number float *saaFlag : in-precise SAA flag float *orbitPhase : orbit phase [0,1]
Returns	nothing error status passed by global variable “nadc_stat”
Comment	None

4.14 SCIA_LV0_H5_STRUCTS

Identifier	SCIA_LV0_H5_STRUCTS
Author	R.M. van Hees
Language	ANSI C
Purpose	create data structures to store SCIAMACHY compound data types

Usage	`CRE_SCIA_LV0_H5_STRUCTS( param );`
	`CRE_SCIA_LV1_H5_STRUCTS( param );`
	`CRE_SCIA_LV2_H5_STRUCTS( param );`
	`CRE_SCIA_OL2_H5_STRUCTS( param );`

Input	struct param_record param : struct holding user-defined settings
Returns	Nothing
Comment	None

4.14.1 CRE_SCIA_LV2_H5_STRUCTS

Identifier	CRE_SCIA_LV2_H5_STRUCTS
Purpose	create data structures to store SCIAMACHY compound data types
Usage	`CRE_SCIA_LV2_H5_STRUCTS( param );`
Input	struct param_record param : struct holding user-defined settings
Returns	nothing
Comment	nothing

4.14.2 CRE_SCIA_OL2_H5_STRUCTS

Identifier	CRE_SCIA_OL2_H5_STRUCTS
Purpose	create data structures to store SCIAMACHY compound data types
Usage	`CRE_SCIA_OL2_H5_STRUCTS( param );`
Input	struct param_record param : struct holding user-defined settings
Returns	nothing
Comment	nothing

4.15 SCIA_LV0_H5_INFO

Identifier	SCIA_LV0_H5_INFO
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write info-records from HDF5 Packet Table database
Returns	non-negative on success, negative on failure
Comment	contains SCIA_LV0_WR_H5_INFO_DB, SCIA_LV0_RD_H5_INFO_DB

4.15.1 SCIA_LV0_RD_H5_INFO_DB

Identifier	SCIA_LV0_RD_H5_INFO_DB
Purpose	read info-records to database
Usage	`num_info = SCIA_LV0_RD_H5_INFO_DB( prodName, &info );`
Input	char prodName[] : HDF5 identifier of file or group
	struct mds0_info **info : level 0 info-records (D.18)
Returns	number of info-records read from database (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.16 SCIA_LV0_MDS_INFO

Identifier	SCIA_LV0_MDS_INFO
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain inventarisation of SCIA LV0 product
Comment	contains SCIA_LV0_RD_MDS_INFO
	The information stored in the info-records allows quick access to selected MDS records, without reading first the whole file.

4.16.1 SCIA_LV0_RD_MDS_INFO

Identifier	SCIA_LV0_RD_MDS_INFO
Purpose	obtain inventarisation of Sciamachy LV0 product
Usage	`nr_info = SCIA_LV0_RD_MDS_INFO( fd, num_dsd, dsd, info );`
Input	FILE *fd : (open) stream pointer to scia-file
	unsigned int num_dsd : number of DSD records
	struct dsd_envi *dsd : structure for the DSD records (A.6)
Output	struct mds0_info **info : structure for ”info” records (D.18)
Returns	number of info records found (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.17 SCIA_LV0_PDS_SPH

Identifier	SCIA_LV0_PDS_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Specific Product Header of Level 0 product
Comment	contains SCIA_LV0_RD_SPH

4.17.1 SCIA_LV0_RD_SPH

Identifier	SCIA_LV0_RD_SPH
Purpose	read Specific Product Header of Level 0 Product
Usage	`SCIA_LV0_RD_SPH( fd, mph, &sph );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi mph : main product header (A.5)
Output	struct sph0_scia *sph : structure for the SPH (D.16)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.17.2 SCIA_LV0_WR_SPH

Identifier	SCIA_LV0_WR_SPH
Purpose	write Specific Product Header of Level 0 Product
Usage	`SCIA_LV0_WR_SPH( fd, mph, sph );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi mph : main product header (A.5)
	struct sph0_scia sph : structure for the SPH (D.16)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.18 SCIA_LV0_RD_MDS

Identifier	SCIA_LV0_RD_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	read SCIAMACHY level 0 Measurement Data Sets
Comment	contains SCIA_LV0_RD_AUX, SCIA_LV0_RD_DET, SCIA_LV0_RD_PMD, SCIA_LV0_RD_LV1_AUX, SCIA_LV0_FREE_MDS_DET, SCIA_LV0_RD_LV1_PMD Documentation: - Envisat-1 Product Specifications Volume 6: Level 0 Product Specification Ref: PO-RS-MDA-GS-2009 - Measurement Data Definition and Format description for SCIAMACHY Ref: PO-ID-DOR-SY-0032

4.18.1 SCIA_LV0_RD_MDS_DATA_HDR

Identifier	SCIA_LV0_RD_MDS_DATA_HDR
Purpose	read Data Field Header of the Packet Data Field [ISP]
Usage	`SCIA_LV0_RD_MDS_DATA_HDR( fd, data_hdr );`
Input	FILE *fd : stream pointer
Output	struct data_hdr *data_hdr : Data Field Header (D.4)
Returns	nothing
Comment	Static function The Data Field Header for each of the three packets (i.e. Detector, Auxiliary and PMD) contains standard information.

4.18.2 SCIA_LV0_RD_AUX

Identifier	SCIA_LV0_RD_AUX
Purpose	read selected SCIAMACHY level 0 Auxiliary MDS
Usage	`nr_aux = SCIA_LV0_RD_AUX( fd, info, num_info, &aux );`
Input	FILE *fd : (open) stream pointer
	struct mds0_info *info : structure holding info about MDS records (D.18)
	unsigned int num_info : number of indices to struct info
Output	struct mds0_aux **aux : Auxiliary MDS records (D.19)
Returns	number of Auxiliary MDS read (unsigned int),
	error status passed by global variable “nadc_stat”
Comment	none

4.18.3 SCIA_LV0_RD_DET

Identifier	SCIA_LV0_RD_DET
Purpose	read selected SCIAMACHY level 0 Detector MDS
Usage	`nr_det = SCIA_LV0_RD_DET( fd, info, num_info, &det );`
Input	FILE *fd : (open) stream pointer
	struct mds0_info *info : structure holding info about MDS records (D.18)
	unsigned int num_info : number of indices to struct info
	unsigned char chan_mask : mask for channel selection
Output	struct mds0_det **det : Detector MDS records (D.20)
Returns	number of Detector MDS read (unsigned int),
	error status passed by global variable “nadc_stat”
Comment	none

4.18.4 SCIA_LV0_RD_PMD

Identifier	SCIA_LV0_RD_PMD
Purpose	read selected SCIAMACHY level 0 PMD MDS
Usage	`nr_pmd = SCIA_LV0_RD_PMD( fd, info, num_info, &pmd );`
Input	FILE *fd : (open) stream pointer
	struct mds0_info *info : structure holding info about MDS records (D.18)
	unsigned int num_info : number of indices to struct info
Output	struct mds0_pmd **pmd : PMD MDS records (D.21)
Returns	number of PMD MDS read (unsigned int),
	error status passed by global variable “nadc_stat”
Comment	none

4.18.5 SCIA_LV0_FREE_MDS_DET

Identifier	SCIA_LV0_FREE_MDS_DET
Purpose	free memory allocated by SCIA_LV0_RD_DET
Usage	`SCIA_LV0_FREE_MDS_DET( num_det, det );`
Input	unsigned short num_det : number of Detector MDS
Output	struct mds0_det *det : Detector MDS records (D.20)
Returns	error status passed by global variable “nadc_stat”
Comment	none

4.18.6 SCIA_LV0_RD_LV1_AUX

Identifier	SCIA_LV0_RD_LV1_AUX
Purpose	read SCIAMACHY level 0 Auxiliary MDS as stored in a level 1b product
Usage	`SCIA_LV0_RD_LV1_AUX( fd, aux );`
Input	FILE *fd : (open) stream pointer
Output	struct mds0_aux *aux : Auxiliary MDS records (D.19)
Returns	nothing, error status passed by global variable "nadc_stat"
Comment	none

4.18.7 SCIA_LV0_RD_LV1_PMD

Identifier	SCIA_LV0_RD_LV1_PMD
Purpose	read SCIAMACHY level 0 PMD MDS as stored in a level 1b product
Usage	`SCIA_LV0_RD_LV1_PMD( fd, pmd );`
Input	FILE *fd : (open) stream pointer
Output	struct mds0_pmd *pmd : PMD MDS records (D.21)
Returns	nothing, error status passed by global variable "nadc_stat"
Comment	none

4.19 SCIA_LV0_SELECT

Identifier	SCIA_LV0_SELECT
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain indices to info-records of selected MDS records
Usage	`nr_mds = SCIA_LV0_SELECT_MDS( source, param, nr_info, info, &info_out );`
Input	int source : data source (Detector, Auxiliary, PMD) struct param_record param : struct holding user-defined settings (A.1) unsigned int nr_info : number of info-records struct mds0_info *info : structure for info-records (D.18)
Output	struct mds0_info **info_out: array with selected info-records (D.18)
Returns	number of selected records (unsigned int) error status passed by global variable "nadc_stat"
Comment	None

4.20 SCIA_LV0_WR_ASCII_INFO

Identifier	SCIA_LV0_WR_ASCII_INFO
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump collected Info-records of an SCIA level 0 file
Usage	`SCIA_LV0_WR_ASCII_INFO( param, num_info, info );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_info : number of info-records struct mds0_info *info : pointer to structure with info-records (D.18)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.21 SCIA_LV0_WR_ASCII_MDS

Identifier	SCIA_LV0_WR_ASCII_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Measurement Data Sets in ASCII
Comment	contains SCIA_WR_ASCII_LV0_AUX, SCIA_WR_ASCII_LV0_PMD, SCIA_LV0_WR_ASCII_AUX, SCIA_LV0_WR_ASCII_PMD, SCIA_LV0_WR_ASCII_DET

4.21.1 SCIA_WR_ASCII_LV0_AUX

Identifier	SCIA_WR_ASCII_LV0_AUX
Purpose	dump content of Auxiliary data packets to file
Usage	`SCIA_WR_ASCII_LV0_AUX( outfl, nr, aux );`
Input	FILE *outfl : (open) stream pointer unsigned int nr : number of Auxiliary data packets struct mds0_aux *aux : pointer to Auxiliary data packets (D.19)
Returns	nothing (check global error status)
Comment	none

4.21.2 SCIA_WR_ASCII_LV0_PMD

Identifier	SCIA_WR_ASCII_LV0_PMD
Purpose	dump content of PMD data packets to file
Usage	`SCIA_WR_ASCII_LV0_PMD( outfl, nr, pmd );`
Input	FILE *outfl : (open) stream pointer unsigned int nr : number of PMD data packets struct mds0_pmd *pmd : pointer to PMD records (D.21)
Returns	nothing (check global error status)
Comment	none

4.21.3 SCIA_LV0_WR_ASCII_AUX

Identifier	SCIA_LV0_WR_ASCII_AUX
Purpose	dump content of Auxiliary data packets
Usage	`SCIA_LV0_WR_ASCII_AUX( param, stateIndx, nr_aux, aux );`
Input	struct param_record param : struct holding user-defined settings unsigned int stateIndx : Index of State in product unsigned int nr_aux : number of Auxiliary MDS structures struct mds0_aux *aux : Level 0 Auxiliary MDS records (D.19)
Returns	nothing (check global error status)
Comment	none

4.21.4 SCIA_LV0_WR_ASCII_PMD

Identifier	SCIA_LV0_WR_ASCII_PMD
Purpose	dump content of PMD data packets
Usage	`SCIA_LV0_WR_ASCII_PMD( param, stateIndx, nr_pmd, pmd );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int stateIndx : Index of State in product
	unsigned int nr_pmd : number of PMD MDS structures
	struct mds0_pmd *pmd : Level 0 PMD MDS records (D.21)
Returns	nothing (check global error status)
Comment	none

4.21.5 SCIA_LV0_WR_ASCII_DET

Identifier	SCIA_LV0_WR_ASCII_DET
Purpose	dump content of Detector data packets
Usage	`SCIA_LV0_WR_ASCII_DET( param, stateIndx, nr_det, det );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int stateIndx : Index of State in product
	unsigned int nr_det : number of Detector MDS structures
	struct mds0_det *det : Level 0 Detector MDS records (D.20)
Returns	nothing (check global error status)
Comment	none

4.22 SCIA_LV0_WR_ASCII_SPH

Identifier	SCIA_LV0_WR_ASCII_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Specific Product Header of the level 0 product
Usage	`SCIA_LV0_WR_ASCII_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph0_scia *sph : pointer to structure with SPH record (D.16)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.23 SCIA_LV0_WR_H5_MDS

Identifier	SCIA_LV0_WR_H5_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write Measurement Data Sets in HDF5 format
Usage	`SCIA_LV0_WR_H5_AUX( param, stateIndx, nr_mds, aux );`
	`SCIA_LV0_WR_H5_DET( param, stateIndx, nr_mds, det );`
	`SCIA_LV0_WR_H5_PMD( param, stateIndx, nr_mds, pmd );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int stateIndx : index of State in product
	unsigned int nr_mds : number of MDS structures
	struct mds0_aux *aux : Auxiliary MDS records (D.19)
	struct mds0_det *det : Detector MDS records (D.20)
	struct mds0_pmd *pmd : PMD MDS records (D.21)
Returns	Nothing
Comment	None

4.23.1 SCIA_LV0_WR_H5_AUX

Identifier	SCIA_LV0_WR_H5_AUX
Purpose	define and write Auxiliary MDS records in HDF5 format
Usage	`SCIA_LV0_WR_H5_AUX( param, stateIndx, nr_mds, aux );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int stateIndx : index of State in product
	unsigned int nr_mds : number of MDS structures
	struct mds0_aux *aux : Auxiliary MDS records (D.19)
Returns	nothing
Comment	nothing

4.23.2 SCIA_LV0_WR_H5_PMD

Identifier	SCIA_LV0_WR_H5_PMD
Purpose	define and write PMD MDS records in HDF5 format
Usage	`SCIA_LV0_WR_H5_PMD( param, stateIndx, nr_mds, pmd );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int stateIndx : index of State in product
	unsigned int nr_mds : number of MDS structures
	struct mds0_pmd *pmd : PMD MDS records (D.21)
Returns	nothing
Comment	nothing

4.23.3 SCIA_LV0_WR_H5_DET

Identifier	SCIA_LV0_WR_H5_DET
Purpose	define and write Detector MDS records in HDF5 format
Usage	`SCIA_LV0_WR_H5_DET( param, info, nr_mds, det );`
Input	struct param_record param : struct holding user-defined settings
	struct mds0_scia *info : (**??**)
	unsigned int nr_mds : number of MDS structures
	struct mds0_det *det : Detector MDS records (D.20)
Returns	nothing
Comment	nothing

4.24 SCIA_LV0_WR_H5_SPH

Identifier	SCIA_LV0_WR_H5_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 0 SPH data
Usage	`SCIA_LV0_WR_H5_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph0_scia *sph : Specific Product Header data (D.16)
Returns	Nothing
Comment	None

4.25 SCIA_LV0_WR_MDS

Identifier	SCIA_LV0_WR_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	write SCIAMACHY level 0 Measurement Data Sets
Comment	contains SCIA_LV0_WR_AUX, SCIA_LV0_WR_DET, SCIA_LV0_WR_PMD, SCIA_LV0_WR_LV1_AUX, SCIA_LV0_WR_LV1_PMD Documentation: - Envisat-1 Product Specifications Volume 6: Level 0 Product Specification Ref: PO-RS-MDA-GS-2009 - Measurement Data Definition and Format description for SCIAMACHY Ref: PO-ID-DOR-SY-0032

4.25.1 SCIA_LV0_WR_AUX

Identifier	SCIA_LV0_WR_AUX
Purpose	write selected SCIAMACHY level 0 Auxiliary MDS
Usage	`nr_aux = SCIA_LV0_WR_AUX( fd, info, num_info, aux );`
Input	FILE *fd : (open) stream pointer struct mds0_info *info : structure holding info about MDS records (D.18) unsigned int num_info : number of indices to struct info struct mds0_aux *aux : Auxiliary MDS records (D.19)
Returns	number of Auxiliary MDS write (unsigned int), error status passed by global variable “nadc_stat”
Comment	none

4.25.2 SCIA_LV0_WR_DET

Identifier	SCIA_LV0_WR_DET
Purpose	write selected SCIAMACHY level 0 Detector MDS
Usage	`nr_det = SCIA_LV0_WR_DET( fd, info, num_info, det );`
Input	FILE *fd : (open) stream pointer struct mds0_info *info : structure holding info about MDS records (D.18) unsigned int num_info : number of indices to struct info struct mds0_det *det : Detector MDS records (D.20)
Returns	number of Detector MDS write (unsigned int), error status passed by global variable “nadc_stat”
Comment	none

4.25.3 SCIA_LV0_WR_PMD

Identifier	SCIA_LV0_WR_PMD
Purpose	write selected SCIAMACHY level 0 PMD MDS
Usage	`nr_pmd = SCIA_LV0_WR_PMD( fd, info, num_info, &pmd );`
Input	FILE *fd : (open) stream pointer
	struct mds0_info *info : structure holding info about MDS records (D.18)
	unsigned int num_info : number of indices to struct info
	struct mds0_pmd *pmd : PMD MDS records (D.21)
Returns	number of PMD MDS write (unsigned int),
	error status passed by global variable “nadc_stat”
Comment	none

4.25.4 SCIA_LV0_WR_LV1_AUX

Identifier	SCIA_LV0_WR_LV1_AUX
Purpose	write SCIAMACHY level 0 Auxiliary MDS as stored in a level 1b product
Usage	`nr_byte = SCIA_LV0_WR_LV1_AUX( fd, aux );`
Input	FILE *fd : (open) stream pointer
	struct mds0_aux aux : Auxiliary MDS records (D.19)
Returns	number of bytes written (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.25.5 SCIA_LV0_WR_LV1_PMD

Identifier	SCIA_LV0_WR_LV1_PMD
Purpose	write SCIAMACHY level 0 PMD MDS as stored in a level 1b product
Usage	`nr_byte = SCIA_LV0_WR_LV1_PMD( fd, pmd );`
Input	FILE *fd : (open) stream pointer
	struct mds0_pmd pmd : PMD MDS records (D.21)
Returns	number of bytes written (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.26 SCIA_LV1_CHAN2CLUS

Identifier	SCIA_LV1_CHAN2CLUS
Author	R.M. van Hees
Language	ANSI C
Purpose	combine channel and cluster selection into a mask (0/1)
Usage	`clus_mask = SCIA_LV1_CHAN2CLUS( param, state );`
Input	struct param_record param : struct holding user-defined settings
	struct state1_scia *state : structure with States of the product (E.22)
Returns	mask with clusters to be selected (unsigned long long)
Comment	None

4.27 SCIA_LV1_CORR_LOS

Identifier	SCIA_LV1_CORR_LOS
Author	R.M. van Hees
Language	ANSI C
Purpose	correct the line-of-sight azimuth and zenith angles (geoN)
Usage	`SCIA_LV1_CORR_LOS( state, mds_1b );`
Input	struct state1_scia *state : structure with States of the product
Output	struct mds1_scia *mds_1b : structure holding level 1b MDS records (E.31)
Returns	nothing
Comment	The values of the level 1b line-of-sight zenith angles are always larger than zero, and the azimuth angle jumps with 180 degrees while scanning through nadir. This function will modify these values as follows: removing the jump in the azimuth angles and returns negative zenith angles, when the original azimuth angle was larger than 180 degree. This makes interpolation much easier.

4.28 SCIA_LV1_FREE_MDS

Identifier	SCIA_LV1_FREE_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	free memory allocated by SCIA_LV1_RD_MDS, SCIA_LV1C_RD_MDS, etc
Returns	nothing
Comment	contains SCIA_LV1_FREE_MDS, SCIA_LV1C_FREE_MDS, SCIA_LV1C_FREE_MDS_PMD, SCIA_LV1C_FREE_MDS_POLV

4.28.1 SCIA_LV1_FREE_MDS

Identifier	SCIA_LV1_FREE_MDS
Purpose	free memory allocated by SCIA_LV1_RD_MDS
Usage	`SCIA_LV1_FREE_MDS( source, nr_mds, mds );`
Input	int source : source of MDS (Nadir, Limb, ...)
	unsigned short nr_mds : number of level 1b MDS
Output	struct mds1_scia *mds : level 1b MDS records (E.31)
Returns	nothing
Comment	none

4.28.2 SCIA_LV1C_FREE_MDS

Identifier	SCIA_LV1C_FREE_MDS
Purpose	free memory allocated by SCIA_LV1C_RD_MDS
Usage	`SCIA_LV1C_FREE_MDS( source, nr_mds, mds );`
Input	int source : source of MDS (Nadir, Limb, ...)
	unsigned int nr_mds : number of level 1c MDS
Output	struct mds1c_scia *mds : level 1c MDS records (C.11)
Returns	nothing
Comment	none

4.28.3 SCIA_LV1C_FREE_MDS_PMD

Identifier	SCIA_LV1C_FREE_MDS_PMD
Purpose	free memory allocated by SCIA_LV1C_RD_MDS_PMD
Usage	`SCIA_LV1C_FREE_MDS_PMD( source, pmd );`
Input	int source : source of MDS (Nadir, Limb, ...)
Output	struct mds1c_pmd *pmd : level 1c PMD MDS records (C.12)
Returns	error status passed by global variable "nadc_stat"
Comment	none

4.28.4 SCIA_LV1C_FREE_MDS_POLV

Identifier	SCIA_LV1C_FREE_MDS_POLV
Purpose	free memory allocated by SCIA_LV1C_RD_MDS_POLV
Usage	`SCIA_LV1C_FREE_MDS_POLV( source, num_pol, polV );`
Input	int source : source of MDS (Nadir, Limb, ...)
Output	struct mds1c_polV *polV : level 1c frac polarisation MDS records (C.13)
Returns	error status passed by global variable "nadc_stat"
Comment	none

4.29 SCIA_LV1_WR_H5_MDS

Identifier	SCIA_LV1_WR_H5_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	write SCIAMACHY level 1 Measurements Data Sets
Comment	Contains SCIA_LV1_RD_H5_MDS, SCIA_LV1_WR_H5_MDS, SCIA_LV1C_WR_H5_MDS, SCIA_LV1C_WR_H5_MDS_PMD, SCIA_LV1C_WR_H5_MDS_POLV

4.29.1 SCIA_LV1_RD_H5_MDS

Identifier	SCIA_LV1_RD_H5_MDS
Purpose	read SCIAMACHY level 1 Measurements Data Sets
Usage	`num_mds = SCIA_LV1_RD_H5_MDS( scia_fl, state, mds_1b );`
Input	char *scia_fl : name of the product struct state1_scia *state : states in the product (E.22) struct mds1_scia *mds_1b : MDS struct (level 1b) (E.31)
Returns	Number of MDS records read from the product (unsigned int), error status passed by global variable "nadc_stat"
Comment	None

4.29.2 SCIA_LV1_WR_H5_MDS

Identifier	SCIA_LV1_WR_H5_MDS
Purpose	write SCIAMACHY level 1 Measurements Data Sets
Usage	`SCIA_LV1_WR_H5_MDS( param, num_mds, mds_1b );`
Input	struct param_record param : struct holding user-defined settings unsigned short num_mds : number of MDS struct mds1_scia *mds_1b : MDS struct (level 1b) (E.31)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.29.3 SCIA_LV1C_WR_H5_MDS

Identifier	SCIA_LV1C_WR_H5_MDS
Purpose	write SCIAMACHY level 1c Measurements Data Sets
Usage	`SCIA_LV1C_WR_H5_MDS( param, num_mds, mds_1c );`
Input	struct param_record param : struct holding user-defined settings
	unsigned short num_mds : number of MDS
	struct mds1c_scia *mds_1c : MDS struct (level 1c) (C.11)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.29.4 SCIA_LV1C_WR_H5_MDS_PMD

Identifier	SCIA_LV1C_WR_H5_MDS_PMD
Purpose	write SCIAMACHY level 1 Measurements Data Sets
Usage	`SCIA_LV1C_WR_H5_MDS_PMD( param, pmd );`
Input	struct param_record param : struct holding user-defined settings
	unsigned short num_mds : number of MDS
	struct mds1c_pmd *pmd : PMD MDS struct (level 1c) (C.12)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.29.5 SCIA_LV1C_WR_H5_MDS_POLV

Identifier	SCIA_LV1C_WR_H5_MDS_POLV
Purpose	write SCIAMACHY level 1 Measurements Data Sets
Usage	`SCIA_LV1C_WR_H5_MDS_POLV( param, polV );`
Input	struct param_record param : struct holding user-defined settings
	unsigned short num_mds : number of MDS
	struct mds1c_polV *polV : fractional polarisation MDS (level 1c) (C.13)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.30 SCIA_LV1_LIB_DSD

Identifier	SCIA_LV1_LIB_DSD
Author	R.M. van Hees
Language	ANSI C
Purpose	Sciamachy level 1 DSD book keeping routines
Comment	contains: SCIA_LV1_EXPORT_NUM_STATE, SCIA_LV1_SET_NUM_ATTACH
	SCIA_LV1_WR_DSD_INIT, SCIA_LV1_ADD_DSD, SCIA_LV1_WR_DSD_UPDATE
	SCIA_LV1_UPDATE_SQADS, SCIA_LV1_UPDATE_LADS, SCIA_LV1_UPDATE_STATE

4.30.1 SCIA_LV1_EXPORT_NUM_STATE

Identifier	SCIA_LV1_EXPORT_NUM_STATE
Purpose	allow external program to update number of NADIR, LIMB, ... states
Usage	`SCIA_LV1_EXPORT_NUM_STATE( source, number );`
Input	int source : data source (Nadir, Limb, ...)
	unsigned short number : number of selected states
Returns	nothing
Comment	none

4.30.2 SCIA_LV1_ADD_DSD

Identifier	SCIA_LV1_ADD_DSD
Purpose	add DSD records to list of DSD written in output file
Usage	`SCIA_LV1_ADD_DSD( dsd_to_add );`
Input	struct dsd_envi *dsd_to_add
Returns	nothing
Comment	none

4.30.3 SCIA_LV1_WR_DSD_INIT

Identifier	SCIA_LV1_WR_DSD_INIT
Purpose	initialise DSD records and write them to file
Usage	`SCIA_LV1_WR_DSD_INIT( param, fp_out, num_dsd_in, dsd_in );`
Input	struct param_record param : struct holding user-defined settings FILE *fp_in : (open) stream pointer (input) FILE *fp_out : (open) stream pointer (output) unsigned int num_dsd_in : number of DSD's (input) struct dsd_envi *dsd_in : DSD's records (input) (A.6)
Returns	nothing
Comment	none

4.30.4 SCIA_LV1_SET_NUM_ATTACH

Identifier	SCIA_LV1_SET_NUM_ATTACH
Purpose	update global variables: num_attach_states, indx_attach_states
Usage	`SCIA_LV1_SET_NUM_ATTACH( param, fp_in, num_dsd_in, dsd_in );`
Input	struct param_record param : struct holding user-defined settings FILE *fp_in : (open) stream pointer (input) unsigned int num_dsd_in : number of DSD's (input) struct dsd_envi *dsd_in : DSD's records (input) (A.6)
Returns	nothing
Comment	none

4.30.5 SCIA_LV1_WR_DSD_UPDATE

Identifier	SCIA_LV1_WR_DSD_UPDATE
Purpose	update DSD records in output file
Usage	`SCIA_LV1_WR_DSD_UPDATE( fp_in, fp_out );`
Input	FILE *fp_in : (open) stream pointer (input) FILE *fp_out : (open) stream pointer (output)
Returns	nothing
Comment	IMPORTANT: call this routine just before closing the PDS file Update MPH: PRODUCT, NUM_DSD, NUM_DATA_SETS Update SPH: NO_OF_NADIR_STATES, NO_OF_LIMB_STATES, NO_OF_OCCULTATION_STATES, NO_OF_MONI_STATES Update DSD records in file (recalculate all dsd records "ds_offset")

4.30.6 SCIA_LV1_UPDATE_SQADS

Identifier	SCIA_LV1_UPDATE_SQADS
Purpose	return selected SQADS records with MDS attach
Usage	`num_sqads_out = SCIA_LV1_UPDATE_SQADS( sqads );`
In/Output	struct sqads1_scia *sqads : summary of quality flags per state (E.7)
Returns	number of selected states error status passed by global variable "nadc_stat"
Comment	none

4.30.7 SCIA_LV1_UPDATE_LADS

Identifier	SCIA_LV1_UPDATE_LADS
Purpose	return selected LADS records with MDS attach
Usage	`num_lads_out = SCIA_LV1_UPDATE_LADS( lads );`
In/Output	struct lads_scia *lads : summary of quality flags per state (C.10)
Returns	number of selected states error status passed by global variable "nadc_stat"
Comment	none

4.30.8 SCIA_LV1_UPDATE_STATE

Identifier	SCIA_LV1_UPDATE_STATE
Purpose	return selected STATE records with MDS attach
Usage	`num_state_out = SCIA_LV1_WR_STATE( state );`
In/Output	struct state1_scia *state : summary of quality flags per state (E.22)
Returns	number of selected states error status passed by global variable "nadc_stat"
Comment	none

4.31 SCIA_LV1_MFACTOR_SRS

Identifier	SCIA_LV1_MFACTOR_SRS
Author	Klaus Bramstedt (ife Bremen)
Language	ANSI C
Purpose	apply mfactor M_CAL to SMR (ESM Diffuser: first spectrum)
Usage	`SCIA_LV1_MFACTOR_SRS ( sensing_start, calibFlag, num_dsr, srs );`
Input	char *sensing_start : taken from MPH unsigned int calibFlag : bit-flag which defines how to calibrate unsigned int num_dsr : number of Sun reference spactra records
In/Output	struct srs_scia *srs : Sun reference spectrum (E.14)
Returns	nothing
Comment	nothing

4.32 SCIA_LV1_PDS_ASFP

Identifier	SCIA_LV1_PDS_ASFP
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Small Aperture Slit function parameters
Comment	contains SCIA_LV1_RD_ASFP and SCIA_LV1_WR_ASFP

4.32.1 SCIA_LV1_RD_ASFP

Identifier	SCIA_LV1_RD_ASFP
Purpose	read Small Aperture Slit function parameters
Usage	`nr_dsr = SCIA_LV1_RD_ASFP( fd, num_dsd, dsd, &asfp );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct asfp_scia **asfp : structure for ASFP parameters (E.21)
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

4.32.2 SCIA_LV1_WR_ASFP

Identifier	SCIA_LV1_WR_ASFP
Purpose	write Small Aperture Slit function parameters
Usage	`SCIA_LV1_WR_ASFP( fd, num_asfp, asfp );`
Input	FILE *fd : stream pointer
	unsigned int num_asfp : number of ASFP records
	struct asfp_scia *asfp : structure for ASFP parameters (E.21)
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

4.33 SCIA_LV1_RD_AUX

Identifier	SCIA_LV1_RD_AUX
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write auxiliary Data Packets
Comment	contains SCIA_LV1_RD_AUX and SCIA_LV1_WR_AUX

4.33.1 SCIA_LV1_RD_AUX

Identifier	SCIA_LV1_RD_AUX
Purpose	read/write auxiliary Data Packets
Usage	`nr_dsr = SCIA_LV1_RD_AUX( fd, num_dsd, dsd, &aux );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct aux_scia **aux : structure for auxiliary data packets (E.24)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.33.2 SCIA_LV1_WR_AUX

Identifier	SCIA_LV1_WR_AUX
Purpose	write auxiliary Data Packets
Usage	`SCIA_LV1_WR_AUX( fd, num_aux, aux );`
Input	FILE *fd : stream pointer
	unsigned int num_aux : number of Auxiliary records
	struct aux_scia *aux : structure for auxiliary data packets (E.24)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.34 SCIA_LV1_PDS_BASE

Identifier	SCIA_LV1_PDS_BASE
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Precise Basis of Spectral Calibration Parameters
Comment	contains SCIA_LV1_RD_BASE and SCIA_LV1_WR_BASE

4.34.1 SCIA_LV1_WR_BASE

Identifier	SCIA_LV1_WR_BASE
Purpose	read the Precise Basis of the Spectral Calibration Parameters
Usage	`nr_dsr = SCIA_LV1_RD_BASE( fd, num_dsd, dsd, &base );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct base_scia *base : structure for the Spectral BASE (E.12)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.34.2 SCIA_LV1_WR_BASE

Identifier	SCIA_LV1_WR_BASE
Purpose	write the Precise Basis of the Spectral Calibration Parameters
Usage	`SCIA_LV1_WR_BASE( fd, base );`
Input	FILE *fd : (open) stream pointer
	struct base_scia *base : structure for the Spectral BASE (E.12)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.35 SCIA_LV1_PDS_CLCP

Identifier	SCIA_LV1_PDS_CLCP
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Leakage Current Parameters (constant fraction) records
Comment	contains SCIA_LV1_RD_CLCP and SCIA_LV1_WR_CLCP

4.35.1 SCIA_LV1_RD_CLCP

Identifier	SCIA_LV1_RD_CLCP
Purpose	read Leakage Current Parameters (constant fraction) records
Usage	`nr_dsr = SCIA_LV1_RD_CLCP( fd, num_dsd, dsd, &clcp );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct clcp_scia *clcp : leakage current parameters (constant) (E.9)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.35.2 SCIA_LV1_WR_CLCP

Identifier	SCIA_LV1_WR_CLCP
Purpose	write Leakage Current Parameters (constant fraction) records
Usage	`SCIA_LV1_WR_CLCP( fd, num_clcp, clcp );`
Input	FILE *fd : stream pointer
	unsigned int num_clcp : number of CLCP records
	struct clcp_scia clcp : leakage current parameters (constant) (E.9)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.36 SCIA_LV1_PDS_DARK

Identifier	SCIA_LV1_PDS_DARK
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write average of the Dark Measurements per State
Comment	contains SCIA_LV1_RD_DARK and SCIA_LV1_WR_DARK

4.36.1 SCIA_LV1_RD_DARK

Identifier	SCIA_LV1_RD_DARK
Purpose	read average of the Dark Measurements per State
Usage	`nr_dsr = SCIA_LV1_RD_DARK( fd, num_dsd, dsd, &dark );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct dark_scia **dark : average of the dark measurements (E.26)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.36.2 SCIA_LV1_WR_DARK

Identifier	SCIA_LV1_WR_DARK
Purpose	write average of the Dark Measurements per State
Usage	`SCIA_LV1_WR_DARK( fd, num_dark, dark );`
Input	FILE *fd : stream pointer
	unsigned int num_dark : number of Dark records
	struct dark_scia *dark : average of the dark measurements (E.26)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.37 SCIA_LV1_PDS_EKD

Identifier	SCIA_LV1_PDS_EKD
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Errors on Key Data
Comment	contains SCIA_LV1_RD_EKD and SCIA_LV1_WR_EKD

4.37.1 SCIA_LV1_RD_EKD

Identifier	SCIA_LV1_RD_EKD
Purpose	read Errors on Key Data
Usage	`nr_dsr = SCIA_LV1_RD_EKD( fd, num_dsd, dsd, &ekd );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct ekd_scia *ekd : errors on key data (E.19)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.37.2 SCIA_LV1_WR_EKD

Identifier	SCIA_LV1_WR_EKD
Purpose	write Errors on Key Data
Usage	`SCIA_LV1_WR_EKD( fd, num_ekd, ekd );`
Input	FILE *fd : stream pointer
	unsigned int num_ekd : number of EKD records
	struct ekd_scia ekd : errors on key data (E.19)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.38 SCIA_LV1_PDS_LCPN

Identifier	SCIA_LV1_PDS_LCPN
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Leakage Current Parameters (newly calculated)
Comment	contains SCIA_LV1_RD_LCPN and SCIA_LV1_WR_LCPN

4.38.1 SCIA_LV1_RD_LCPN

Identifier	SCIA_LV1_RD_LCPN
Purpose	read Leakage Current Parameters (newly calculated partial set)
Usage	`nr_dsr = SCIA_LV1_RD_LCPN( fd, num_dsd, dsd, &lcpn );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct lcpn_scia **lcpn : leakage current parameters (E.25)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.38.2 SCIA_LV1_WR_LCPN

Identifier	SCIA_LV1_WR_LCPN
Purpose	write Leakage Current Parameters (newly calculated partial set)
Usage	`SCIA_LV1_WR_LCPN( fd, num_dsd, dsd, lcpn );`
Input	FILE *fd : stream pointer
	unsigned int num_lcpn : number of LCPN records
	struct lcpn_scia *lcpn : leakage current parameters (E.25)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.39 SCIA_LV1_PDS_PMD

Identifier	SCIA_LV1_PDS_PMD
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write PMD (level 0) Data Packets
Comment	contains SCIA_LV1_RD_PMD and SCIA_LV1_WR_PMD

4.39.1 SCIA_LV1_RD_PMD

Identifier	SCIA_LV1_RD_PMD
Purpose	read PMD Data Packets
Usage	`nr_dsr = SCIA_LV1_RD_PMD( fd, num_dsd, dsd, &pmd );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct pmd_scia **pmd : structure for PMD data packets (E.23)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.39.2 SCIA_LV1_WR_PMD

Identifier	SCIA_LV1_WR_PMD
Purpose	write PMD Data Packets
Usage	`SCIA_LV1_WR_PMD( fd, num_pmd, pmd );`
Input	FILE *fd : stream pointer
	unsigned int num_pmd : number of PMD records
	struct pmd_scia *pmd : structure for PMD data packets (E.23)
Returns	nothing
	error status passed by global variable "nadc_stat"
Comment	none

4.40 SCIA_LV1_PDS_PPG

Identifier	SCIA_LV1_PDS_PPG
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write PPG/Etalon Parameters records
Comment	contains SCIA_LV1_RD_PPG and SCIA_LV1_WR_PPG

4.40.1 SCIA_LV1_RD_PPG

Identifier	SCIA_LV1_RD_PPG
Purpose	read PPG/Etalon Parameters records
Usage	`nbyte = SCIA_LV1_RD_PPG( fd, num_dsd, dsd, &ppg );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct ppg_scia *ppg : PPG/Etalon Parameters (E.11)
Returns	number of data set records read (unsigned int)
	error status passed by global variable "nadc_stat"
Comment	none

4.40.2 SCIA_LV1_WR_PPG

Identifier	SCIA_LV1_WR_PPG
Purpose	write PPG/Etalon Parameters records
Usage	`SCIA_LV1_WR_PPG( fd, num_ppg, ppg );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_ppg : number of PPG records
	struct ppg_scia ppg : PPG/Etalon Parameters (E.11)
Returns	number of data set records read (unsigned int)
	error status passed by global variable "nadc_stat"
Comment	none

4.41 SCIA_LV1_PDS_PPGN

Identifier	SCIA_LV1_PDS_PPGN
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write PPG/Etalon Parameters (newly calculated)
Comment	contains SCIA_LV1_RD_PPGN and SCIA_LV1_WR_PPGN

4.41.1 SCIA_LV1_RD_PPGN

Identifier	SCIA_LV1_RD_PPGN
Purpose	read PPG/Etalon Parameters (newly calculated)
Usage	`nr_dsr = SCIA_LV1_RD_PPGN( fd, num_dsd, dsd, &ppgn );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct ppgn_scia **ppgn : PPG/Etalon Parameters (new) (E.27)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.41.2 SCIA_LV1_WR_PPGN

Identifier	SCIA_LV1_WR_PPGN
Purpose	write PPG/Etalon Parameters (newly calculated)
Usage	`SCIA_LV1_WR_PPGN( fd, num_dsd, dsd, ppgn );`
Input	FILE *fd : stream pointer
	unsigned int num_ppgn : number of PPGN records
	struct ppgn_scia *ppgn : PPG/Etalon Parameters (new) (E.27)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.42 SCIA_LV1_PDS_PSP

Identifier	SCIA_LV1_PDS_PSP
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Polarisation Sensitivity Parameters
	(nadir, limb or occultation)
Comment	contains SCIA_LV1_RD_PSPN, SCIA_LV1_RD_PSPL, SCIA_LV1_RD_PSPO
	SCIA_LV1_WR_PSPN, SCIA_LV1_WR_PSPL, SCIA_LV1_WR_PSPO

4.42.1 SCIA_LV1_RD_PSPN

Identifier	SCIA_LV1_RD_PSPN
Purpose	read Polarisation Sensitivity Parameters (nadir)
Usage	`nr_dsr = SCIA_LV1_RD_PSPN( fd, num_dsd, dsd, &pspn );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct pspn_scia **pspn : Polarisation Sensitivity Parameters (E.15)
	(nadir)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.42.2 SCIA_LV1_RD_PSPL

Identifier	SCIA_LV1_RD_PSPL
Purpose	read Polarisation Sensitivity Parameters (limb)
Usage	`nr_dsr = SCIA_LV1_RD_PSPL( fd, num_dsd, dsd, &pspl );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct psplo_scia **pspl: Polarisation Sensitivity Parameters (E.16) (limb)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.42.3 SCIA_LV1_RD_PSPO

Identifier	SCIA_LV1_RD_PSPO
Purpose	read Polarisation Sensitivity Parameters (occultation)
Usage	`nr_dsr = SCIA_LV1_RD_PSPO( fd, num_dsd, dsd, &pspo );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct psplo_scia **pspo: Polarisation Sensitivity Parameters (E.16) (occultation)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.42.4 SCIA_LV1_WR_PSPN

Identifier	SCIA_LV1_WR_PSPN
Purpose	write Polarisation Sensitivity Parameters (nadir)
Usage	`SCIA_LV1_WR_PSPN( fd, num_dsd, dsd, pspn );`
Input	FILE *fd : stream pointer
	unsigned int num_pspn : number of PSPN records
	struct pspn_scia *pspn : Polarisation Sensitivity Parameters (E.15) (nadir)
Returns	nnothing
	error status passed by global variable “nadc_stat”
Comment	none

4.42.5 SCIA_LV1_WR_PSPL

Identifier	SCIA_LV1_WR_PSPL
Purpose	write Polarisation Sensitivity Parameters (limb)
Usage	`SCIA_LV1_WR_PSPL( fd, num_pspl, pspl );`
Input	FILE *fd : stream pointer
	unsigned int num_pspl : number of PSPL records
	struct psplo_scia *pspl: Polarisation Sensitivity Parameters (E.16) (limb)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.42.6 SCIA_LV1_WR_PSPO

Identifier	SCIA_LV1_WR_PSPO
Purpose	write Polarisation Sensitivity Parameters (occultation)
Usage	`SCIA_LV1_WR_PSPO( fd, num_pspo, pspo );`
Input	FILE *fd : stream pointer
	unsigned int num_pspo : number of PSPO records
	struct psplo_scia *pspo: Polarisation Sensitivity Parameters (E.16) (occultation)
Returns	nothing
	error status passed by global variable "nadc_stat"
Comment	none

4.43 SCIA_LV1_PDS_RSP

Identifier	SCIA_LV1_PDS_RSP
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Radiance Sensitivity Parameters (nadir, limb or occultation)
Comment	contains SCIA_LV1_RD_RSPN, SCIA_LV1_RD_RSPL, SCIA_LV1_RD_RSPO SCIA_LV1_WR_RSPN, SCIA_LV1_WR_RSPL, SCIA_LV1_WR_RSPO

4.43.1 SCIA_LV1_RD_RSPN

Identifier	SCIA_LV1_RD_RSPN
Purpose	read Radiance Sensitivity Parameters (nadir)
Usage	`nr_dsr = SCIA_LV1_RD_RSPN( fd, num_dsd, dsd, &rspn );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct rspn_scia **rspn : Radiation Sensitivity Parameters (E.17) (nadir)
Returns	number of data set records read (unsigned int)
	error status passed by global variable "nadc_stat"
Comment	none

4.43.2 SCIA_LV1_RD_RSPL

Identifier	SCIA_LV1_RD_RSPL
Purpose	read Radiance Sensitivity Parameters (limb)
Usage	`nr_dsr = SCIA_LV1_RD_RSPL( fd, num_dsd, dsd, &rspl );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct rsplo_scia **rspl: Radiation Sensitivity Parameters (E.18) (limb)
Returns	number of data set records read (unsigned int)
	error status passed by global variable "nadc_stat"
Comment	none

4.43.3 SCIA_LV1_RD_RSPO

Identifier	SCIA_LV1_RD_RSPO
Purpose	read Radiance Sensitivity Parameters (occultation)
Usage	`nr_dsr = SCIA_LV1_RD_RSPO( fd, num_dsd, dsd, &rspo );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct rsplo_scia **rspo: Radiation Sensitivity Parameters (E.18) (occultation)
Returns	number of data set records read (unsigned int)
	error status passed by global variable "nadc_stat"
Comment	none

4.43.4 SCIA_LV1_WR_RSPN

Identifier	SCIA_LV1_WR_RSPN
Purpose	write Radiance Sensitivity Parameters (nadir)
Usage	`SCIA_LV1_WR_RSPN( fd, num_rspn, rspn );`
Input	FILE *fd : stream pointer
	unsigned int num_rspn : number of RSPN records
	struct rspn_scia *rspn : Radiation Sensitivity Parameters (E.17) (nadir)
Returns	nothing
	error status passed by global variable "nadc_stat"
Comment	none

4.43.5 SCIA_LV1_WR_RSPL

Identifier	SCIA_LV1_WR_RSPL
Purpose	write Radiance Sensitivity Parameters (limb)
Usage	`SCIA_LV1_WR_RSPL( fd, num_rspl, rspl );`
Input	FILE *fd : stream pointer
	unsigned int num_rspl : number of RSPL records
	struct rsplo_scia *rspl: Radiation Sensitivity Parameters (E.18) (limb)
Returns	nothing
	error status passed by global variable "nadc_stat"
Comment	none

4.43.6 SCIA_LV1_WR_RSPO

Identifier	SCIA_LV1_WR_RSPO
Purpose	write Radiance Sensitivity Parameters (occultation)
Usage	`SCIA_LV1_WR_RSPO( fd, num_rspo, rspo );`
Input	FILE *fd : stream pointer
	unsigned int num_rspo : number of RSPO records
	struct rsplo_scia *rspo: Radiation Sensitivity Parameters (E.18) (occultation)
Returns	nothing
	error status passed by global variable "nadc_stat"
Comment	none

4.44 SCIA_LV1_PDS_SCP

Identifier	SCIA_LV1_PDS_SCP
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Spectral Calibration Parameters records
Comment	contains SCIA_LV1_RD_SCP and SCIA_LV1_WR_SCP

4.44.1 SCIA_LV1_RD_SCP

Identifier	SCIA_LV1_RD_SCP
Purpose	read Spectral Calibration Parameters records
Usage	`nr_dsr = SCIA_LV1_RD_SCP( fd, num_dsd, dsd, &scp );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct scp_scia **scp : Spectral Calibration Parameters (E.13)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.44.2 SCIA_LV1_WR_SCP

Identifier	SCIA_LV1_WR_SCP
Purpose	write Spectral Calibration Parameters records
Usage	`SCIA_LV1_WR_SCP( fd, num_scp, scp );`
Input	FILE *fd : stream pointer
	unsigned int num_scp : number of SCP records
	struct scp_scia *scp : Spectral Calibration Parameters (E.13)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.45 SCIA_LV1_PDS_SCPN

Identifier	SCIA_LV1_PDS_SCPN
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Spectral Calibration Parameters (newly calculated)
Comment	contains SCIA_LV1_RD_SCPN and SCIA_LV1_WR_SCPN

4.45.1 SCIA_LV1_RD_SCPN

Identifier	SCIA_LV1_RD_SCPN
Purpose	read Spectral Calibration Parameters (newly calculated)
Usage	`nbyte = SCIA_LV1_RD_SCPN( fd, num_dsd, dsd, &scpn );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct scpn_scia **scpn : spectral calibration parameters (new) (E.28)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.45.2 SCIA_LV1_WR_SCPN

Identifier	SCIA_LV1_WR_SCPN
Purpose	write Spectral Calibration Parameters (newly calculated)
Usage	`SCIA_LV1_WR_SCPN( fd, num_dsd, dsd, scpn );`
Input	FILE *fd : stream pointer
	unsigned int num_scpn : number of SCPN records
	struct scpn_scia *scpn : spectral calibration parameters (new) (E.28)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.46 SCIA_LV1_PDS_SFP

Identifier	SCIA_LV1_PDS_SFP
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Slit function parameters
Comment	contains SCIA_LV1_RD_SFP and SCIA_LV1_WR_SFP

4.46.1 SCIA_LV1_RD_SFP

Identifier	SCIA_LV1_RD_SFP
Purpose	read Slit function parameters
Usage	`nr_dsr = SCIA_LV1_RD_SFP( fd, num_dsd, dsd, &sfp );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct sfp_scia **sfp : structure for Slit Parameters (E.20)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.46.2 SCIA_LV1_WR_SFP

Identifier	SCIA_LV1_WR_SFP
Purpose	write Slit function parameters
Usage	`SCIA_LV1_WR_SFP( fd, num_sfp, sfp );`
Input	FILE *fd : stream pointer
	unsigned int num_sfp : number of SFP records
	struct sfp_scia *sfp : structure for Slit Parameters (E.20)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.47 SCIA_LV1_PDS_SIP

Identifier	SCIA_LV1_PDS_SIP
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Static Instrument Parameters
Comment	contains SCIA_LV1_RD_SIP and SCIA_LV1_WR_SIP

4.47.1 SCIA_LV1_RD_SIP

Identifier	SCIA_LV1_RD_SIP
Purpose	read Static Instrument Parameters
Usage	`nr_dsr = SCIA_LV1_RD_SIP( fd, num_dsd, dsd, &sip );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct sip_scia *sip : structure for the SIP (E.8)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.47.2 SCIA_LV1_WR_SIP

Identifier	SCIA_LV1_WR_SIP
Purpose	write Static Instrument Parameters
Usage	`SCIA_LV1_RD_SIP( fd, num_sip, sip );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_sip : number of SIP records
	struct sip_scia sip : structure for the SIP (E.8)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.48 SCIA_LV1_PDS_SPH

Identifier	SCIA_LV1_PDS_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Specific Product Header of Level 1b Product
Comment	contains SCIA_LV1_RD_SPH and SCIA_LV1_WR_SPH

4.48.1 SCIA_LV1_RD_SPH

Identifier	SCIA_LV1_RD_SPH
Purpose	read Specific Product Header of Level 1b Product
Usage	`SCIA_LV1_RD_SPH( fd, mph, &sph );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi mph : main product header (A.5)
Output	struct sph1_scia *sph : structure for the SPH (E.6)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.48.2 SCIA_LV1_WR_SPH

Identifier	SCIA_LV1_WR_SPH
Purpose	write Specific Product Header of Level 1b Product
Usage	`SCIA_LV1_WR_SPH( fd, mph, sph );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi mph : main product header (A.5)
	struct sph1_scia sph : structure for the SPH (E.6)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.49 SCIA_LV1_PDS_SQADS

Identifier	SCIA_LV1_PDS_SQADS
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Summary of Quality flags per state records
Comment	contains SCIA_LV1_RD_SQADS and SCIA_LV1_WR_SQADS

4.49.1 SCIA_LV1_RD_SQADS

Identifier	SCIA_LV1_RD_SQADS
Purpose	read Summary of Quality flags per state records
Usage	`nr_dsr = SCIA_LV1_RD_SQADS( fd, num_dsd, dsd, &sqads );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct sqads1_scia **sqads : summary of quality flags per state (E.7)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.49.2 SCIA_LV1_WR_SQADS

Identifier	SCIA_LV1_WR_SQADS
Purpose	write Summary of Quality flags per state records
Usage	`SCIA_LV1_WR_SQADS( fd, num_sqads, sqads );`
Input	FILE *fd : stream pointer
	unsigned int num_sqads : number of SQADS records
	struct sqads1_scia *sqads : summary of quality flags per state (E.7)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.50 SCIA_LV1_PDS_SRS

Identifier	SCIA_LV1_PDS_SRS
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Sun Reference Spectrum records
Comment	contains SCIA_LV1_RD_SRS and SCIA_LV1_WR_SRS

4.50.1 SCIA_LV1_RD_SRS

Identifier	SCIA_LV1_RD_SRS
Purpose	read Sun Reference Spectrum records
Usage	`nr_dsr = SCIA_LV1_RD_SRS( fd, num_dsd, dsd, &srs );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct srs_scia **srs : Sun reference spectrum (E.14)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.50.2 SCIA_LV1_WR_SRS

Identifier	SCIA_LV1_WR_SRS
Purpose	write Sun Reference Spectrum records
Usage	`SCIA_LV1_WR_SRS( fd, num_srs, srs );`
Input	FILE *fd : stream pointer
	unsigned int num_srs : number of SRS records
	struct srs_scia *srs : Sun reference spectrum (E.14)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.51 SCIA_LV1_PDS_SRSN

Identifier	SCIA_LV1_PDS_SRSN
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Sun Reference Spectrum (newly calculated)
Comment	contains SCIA_LV1_RD_SRSN and SCIA_LV1_WR_SRSN

4.51.1 SCIA_LV1_RD_SRSN

Identifier	SCIA_LV1_RD_SRSN
Purpose	read Sun Reference Spectrum (newly calculated)
Usage	`nr_dsr = SCIA_LV1_RD_SRSN( fd, num_dsd, dsd, &srsn );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct srsn_scia **srsn : Sun reference spectrum (new) (E.29)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.51.2 SCIA_LV1_WR_SRSN

Identifier	SCIA_LV1_WR_SRSN
Purpose	write Sun Reference Spectrum (newly calculated)
Usage	`nr_dsr = SCIA_LV1_WR_SRSN( fd, num_dsd, dsd, srsn );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
	struct srsn_scia *srsn : Sun reference spectrum (new) (E.29)
Returns	number of data set records written (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.52 SCIA_LV1_PDS_STATE

Identifier	SCIA_LV1_PDS_STATE
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write States of the Product
Comment	contains SCIA_LV1_RD_STATE and SCIA_LV1_WR_STATE
	For level 1c data the returned state records reflect the actual
	number of clusters per state based on the CAL_OPTIONS DSD

4.52.1 SCIA_LV1_RD_STATE

Identifier	SCIA_LV1_RD_STATE
Purpose	read States of the Product
Usage	`nr_dsr = SCIA_LV1_RD_STATE( fd, num_dsd, dsd, &state );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct state1_scia **state : States of the product (E.22)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.52.2 SCIA_LV1_WR_STATE

Identifier	SCIA_LV1_WR_STATE
Purpose	write States of the Product
Usage	`SCIA_LV1_WR_STATE( fd, num_dsd, dsd, state );`
Input	FILE *fd : stream pointer
	unsigned int num_state : number of STATE records
	struct state1_scia *state : States of the product (E.22)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.53 SCIA_LV1_PDS_VLCP

Identifier	SCIA_LV1_PDS_VLCP
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Leakage Current Parameters (variable fraction) records
Comment	contains SCIA_LV1_RD_VLCP and SCIA_LV1_WR_VLCP

4.53.1 SCIA_LV1_RD_VLCP

Identifier	SCIA_LV1_RD_VLCP
Purpose	read Leakage Current Parameters (variable fraction) records
Usage	`nbyte = SCIA_LV1_RD_VLCP( fd, num_dsd, dsd, &vlcp );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct vlcp_scia **vlcp : leakage current parameters (variable) (E.10)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.53.2 SCIA_LV1_WR_VLCP

Identifier	SCIA_LV1_WR_VLCP
Purpose	write Leakage Current Parameters (variable fraction) records
Usage	`SCIA_LV1_WR_VLCP( fd, num_vlcp, vlcp );`
Input	FILE *fd : stream pointer
	unsigned int num_vlcp : number of VLCP records
	struct vlcp_scia *vlcp : leakage current parameters (variable) (E.10)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.54 SCIA_LV1_RD_MDS

Identifier	SCIA_LV1_RD_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	read SCIAMACHY level 1b or 1c Measurement Data Sets
Returns	number of MDS records read
Comment	None

4.54.1 SCIA_LV1_RD_MDS

Identifier	SCIA_LV1_RD_MDS
Purpose	read MDS of one state from a SCIAMACHY level 1b product
Usage	`nr_mds = SCIA_LV1_RD_MDS( fd, clus_mask, state, &mds );`
Input	FILE *fd : (open) stream pointer
	ulong64 clus_mask : mask for cluster selection
In/Output	struct state1_scia *state : structure with States of the product (E.22)
Output	struct mds1_scia **mds : structure for level 1b MDS records (E.31)
Returns	number of level 1b MDS read (unsigned int),
	error status passed by global variable “nadc_stat”
Comment	This routine allocates memory for the following variables:
	sat_flags, red_grass, glint_flags, geoN/geoL/geoC, lvl0_header,
	int_pmd, polV
	and depending up on the cluster configuration:
	clus[].sig, clus[].sigc

4.54.2 SCIA_LV1C_RD_MDS

Identifier	SCIA_LV1C_RD_MDS
Purpose	read SCIAMACHY level 1c MDS
Usage	`nr_mds = SCIA_LV1C_RD_MDS( fd, clus_mask, state, &mds );`
Input	FILE *fd : (open) stream pointer
	ulong64 clus_mask : mask for cluster selection
In/Output	struct state1_scia *state : structure with States of the product (E.22)
Output	struct mds1c_scia **mds : structure for level 1c MDS (C.11)
Returns	number of level 1c MDS read (unsigned int),
	error status passed by global variable “nadc_stat”
Comment	This routine allocates memory for the following variables:
	pixel_ids, pixel_wv, pixel_wv_err, pixel_val, pixel_err,
	geoN/geoL/geoC

4.54.3 SCIA_LV1C_RD_MDS_PMD

Identifier	SCIA_LV1C_RD_MDS_PMD
Purpose	read SCIAMACHY level 1c PMD MDS
Usage	`nr_mds = SCIA_LV1C_RD_MDS_PMD( fd, state, &pmd );`
Input	FILE *fd : (open) stream pointer
	struct state1_scia *state : structure with States of the product (E.22)
Output	struct mds1c_pmd **pmd : structure for level 1c PMD MDS (C.12)
Returns	number of level 1c PMD MDS read (unsigned int),
	error status passed by global variable “nadc_stat”
Comment	This routine allocates memory for the following variables:
	int_pmd, geoN/geoL/geoC

4.54.4 SCIA_LV1C_RD_MDS_POLV

Identifier	SCIA_LV1C_RD_MDS_POLV
Purpose	read SCIAMACHY level 1c POLV MDS
Usage	`nr_mds = SCIA_LV1C_RD_MDS_POLV( fd, state, &polV );`
Input	FILE *fd : (open) stream pointer
	struct state1_scia *state : structure with States of the product (E.22)
Output	struct mds1c_polV **polV : structure for level 1c POLV MDS (C.13)
Returns	number of level 1c POLV MDS read (unsigned int),
	error status passed by global variable "nadc_stat"
Comment	This routine allocates memory for the following variables:
	polV, geoN/geoL/geoC

4.55 SCIA_LV1_SELECT

Identifier	SCIA_LV1_SELECT
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain state-records of selected MDS records
Usage	`nr_mds = SCIA_LV1_SELECT_MDS( source, param, num_dsd, dsd, num_state, state, &mds_state );`
Input	int source : data source (Nadir, Limb, ...)
	struct param_record param : struct holding user-defined settings (A.1)
	unsigned int num_dsd : number of DSD records
	struct dsd_envi *dsd : structure with DSD records (A.6)
	unsigned int num_state : number of State records
	struct state1_scia *state : structure with States of the product (E.22)
Output	struct state1_scia **mds_state: array with selected state-records (E.22)
Returns	number of selected records (unsigned int)
	error status passed by global variable "nadc_stat"
Comment	None

4.56 SCIA_LV1_WR_ASCII_ADS

Identifier	SCIA_LV1_WR_ASCII_ADS
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Annotation Data Sets in ASCII
Returns	Nothing
Comment	contains SCIA_LV1_WR_ASCII_SQADS, SCIA_LV1_WR_ASCII_STATE, SCIA_LV1_WR_ASCII_PMD, SCIA_LV1_WR_ASCII_AUX, SCIA_LV1_WR_ASCII_LCPN, SCIA_LV1_WR_ASCII_DARK, SCIA_LV1_WR_ASCII_PPGN, SCIA_LV1_WR_ASCII_SCPN, SCIA_LV1_WR_ASCII_SRSN

4.56.1 SCIA_LV1_WR_ASCII_SQADS

Identifier	SCIA_LV1_WR_ASCII_SQADS
Purpose	dump – in ASCII Format – the SQADS
Usage	`SCIA_LV1_WR_ASCII_SQADS( param, num_dsr, sqads );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct sqads1_scia *sqads : pointer to SQADS records (E.7)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.56.2 SCIA_LV1_WR_ASCII_STATE

Identifier	SCIA_LV1_WR_ASCII_STATE
Purpose	dump – in ASCII Format – the STATE records
Usage	`SCIA_LV1_WR_ASCII_STATE( param, num_dsr, state );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct state1_scia *state : pointer to STATE records (E.22)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.56.3 SCIA_LV1_WR_ASCII_PMD

Identifier	SCIA_LV1_WR_ASCII_PMD
Purpose	dump – in ASCII Format – the Lv0 PMD records
Usage	`SCIA_LV1_WR_ASCII_PMD( param, num_dsr, pmd );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct pmd_scia *pmd : pointer to PMD records (E.23)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.56.4 SCIA_LV1_WR_ASCII_AUX

Identifier	SCIA_LV1_WR_ASCII_AUX
Purpose	dump – in ASCII Format – the Lv0 Auxiliary records
Usage	`SCIA_LV1_WR_ASCII_AUX( param, num_dsr, aux );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct aux_scia *aux : pointer to Auxiliary records (E.24)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.56.5 SCIA_LV1_WR_ASCII_LCPN

Identifier	SCIA_LV1_WR_ASCII_LCPN
Purpose	dump – in ASCII Format – the LCPN records
Usage	`SCIA_LV1_WR_ASCII_LCPN( param, num_dsr, lcpn );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct lcpn_scia *lcpn : pointer to LCPN records (E.25)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.56.6 SCIA_LV1_WR_ASCII_DARK

Identifier	SCIA_LV1_WR_ASCII_DARK
Purpose	dump – in ASCII Format – the DARK records
Usage	`SCIA_LV1_WR_ASCII_DARK( param, num_dsr, dark );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct dark_scia *dark : pointer to DARK records (E.26)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.56.7 SCIA_LV1_WR_ASCII_PPGN

Identifier	SCIA_LV1_WR_ASCII_PPGN
Purpose	dump – in ASCII Format – the PPGN records
Usage	`SCIA_LV1_WR_ASCII_PPGN( param, num_dsr, ppgn );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct ppgn_scia *ppgn : pointer to PPGN records (E.27)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.56.8 SCIA_LV1_WR_ASCII_SCPN

Identifier	SCIA_LV1_WR_ASCII_SCPN
Purpose	dump – in ASCII Format – the SCPN records
Usage	`SCIA_LV1_WR_ASCII_SCPN( param, num_dsr, scpn );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct scpn_scia *scpn : pointer to SCPN records (E.28)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.56.9 SCIA_LV1_WR_ASCII_SRSN

Identifier	SCIA_LV1_WR_ASCII_SRSN
Purpose	dump – in ASCII Format – the SRSN records
Usage	`SCIA_LV1_WR_ASCII_SRSN( param, num_dsr, srsn );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct srsn_scia *srsn : pointer to SRSN records (E.29)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57 SCIA_LV1_WR_GADS

Identifier	SCIA_LV1_WR_GADS
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Global Annotation Data Sets in ASCII
Returns	Nothing
Comment	contains SCIA_LV1_WR_ASCII_SIP, SCIA_LV1_WR_ASCII_CLCP, SCIA_LV1_WR_ASCII_VLCP, SCIA_LV1_WR_ASCII_PPG, SCIA_LV1_WR_ASCII_BASE, SCIA_LV1_WR_ASCII_SCP, SCIA_LV1_WR_ASCII_SRS, SCIA_LV1_WR_ASCII_PSPN, SCIA_LV1_WR_ASCII_PSPL, SCIA_LV1_WR_ASCII_PSPO, SCIA_LV1_WR_ASCII_RSPN, SCIA_LV1_WR_ASCII_RSPL, SCIA_LV1_WR_ASCII_RSPO, SCIA_LV1_WR_ASCII_EKD, SCIA_LV1_WR_ASCII_SFP, SCIA_LV1_WR_ASCII_ASFP

4.57.1 SCIA_LV1_WR_ASCII_SIP

Identifier	SCIA_LV1_WR_ASCII_SIP
Purpose	dump – in ASCII Format – the SIP record
Usage	`SCIA_LV1_WR_ASCII_SIP( param, sip );`
Input	struct param_record param : struct holding user-defined settings struct sip_scia *sip : pointer to SIP record (E.8)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.2 SCIA_LV1_WR_ASCII_CLCP

Identifier	SCIA_LV1_WR_ASCII_CLCP
Purpose	dump – in ASCII Format – the CLCP record
Usage	`SCIA_LV1_WR_ASCII_CLCP( param, clcp );`
Input	struct param_record param : struct holding user-defined settings struct clcp_scia *clcp : pointer to CLCP record (E.9)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.3 SCIA_LV1_WR_ASCII_VLCP

Identifier	SCIA_LV1_WR_ASCII_VLCP
Purpose	dump – in ASCII Format – the VLCP records
Usage	`SCIA_LV1_WR_ASCII_VLCP( param, num_dsr, vlcp );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct vlcp_scia *vlcp : pointer to VLCP record(s) (E.10)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.4 SCIA_LV1_WR_ASCII_PPG

Identifier	SCIA_LV1_WR_ASCII_PPG
Purpose	dump – in ASCII Format – the PPG record
Usage	`SCIA_LV1_WR_ASCII_PPG( param, ppg );`
Input	struct param_record param : struct holding user-defined settings
	struct ppg_scia *ppg : pointer to PPG record (E.11)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.5 SCIA_LV1_WR_ASCII_BASE

Identifier	SCIA_LV1_WR_ASCII_BASE
Purpose	dump – in ASCII Format – the BASE record
Usage	`SCIA_LV1_WR_ASCII_BASE( param, base );`
Input	struct param_record param : struct holding user-defined settings
	struct base_scia *base : pointer to BASE record (E.12)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.6 SCIA_LV1_WR_ASCII_SCP

Identifier	SCIA_LV1_WR_ASCII_SCP
Purpose	dump – in ASCII Format – the SCP records
Usage	`SCIA_LV1_WR_ASCII_SCP( param, num_dsr, scp );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct scp_scia *scp : pointer to SCP record(s) (E.13)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.7 SCIA_LV1_WR_ASCII_SRS

Identifier	SCIA_LV1_WR_ASCII_SRS
Purpose	dump – in ASCII Format – the SRS records
Usage	`SCIA_LV1_WR_ASCII_SRS( param, num_dsr, srs );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct srs_scia *srs : pointer to SRS record(s) (E.14)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.8 SCIA_LV1_WR_ASCII_PSPN

Identifier	SCIA_LV1_WR_ASCII_PSPN
Purpose	dump – in ASCII Format – the PSPN records
Usage	`SCIA_LV1_WR_ASCII_PSPN( param, num_dsr, pspn );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct pspn_scia *pspn : pointer to PSPN record(s) (E.15)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.9 SCIA_LV1_WR_ASCII_PSPL

Identifier	SCIA_LV1_WR_ASCII_PSPL
Purpose	dump – in ASCII Format – the PSPL records
Usage	`SCIA_LV1_WR_ASCII_PSPL( param, num_dsr, pspl );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct psplo_scia *pspl : pointer to PSPL record(s) (E.16)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.10 SCIA_LV1_WR_ASCII_PSPO

Identifier	SCIA_LV1_WR_ASCII_PSPO
Purpose	dump – in ASCII Format – the PSPO records
Usage	`SCIA_LV1_WR_ASCII_PSPO( param, num_dsr, pspo );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct psplo_scia *pspo : pointer to PSPO record(s) (E.16)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.11 SCIA_LV1_WR_ASCII_RSPN

Identifier	SCIA_LV1_WR_ASCII_RSPN
Purpose	dump – in ASCII Format – the RSPN records
Usage	`SCIA_LV1_WR_ASCII_RSPN( param, num_dsr, rspn );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct rspn_scia *rspn : pointer to RSPN record(s) (E.17)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.12 SCIA_LV1_WR_ASCII_RSPL

Identifier	SCIA_LV1_WR_ASCII_RSPL
Purpose	dump – in ASCII Format – the RSPL records
Usage	`SCIA_LV1_WR_ASCII_RSPL( param, num_dsr, rspl );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct rsplo_scia *rspl : pointer to RSPL record(s) (E.18)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.13 SCIA_LV1_WR_ASCII_RSPO

Identifier	SCIA_LV1_WR_ASCII_RSPO
Purpose	dump – in ASCII Format – the RSPO records
Usage	`SCIA_LV1_WR_ASCII_RSPO( param, num_dsr, rspo );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct rsplo_scia *rspo : pointer to RSPO record(s) (E.18)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.14 SCIA_LV1_WR_ASCII_EKD

Identifier	SCIA_LV1_WR_ASCII_EKD
Purpose	dump – in ASCII Format – the EKD record
Usage	`SCIA_LV1_WR_ASCII_EKD( param, ekd );`
Input	struct param_record param : struct holding user-defined settings struct ekd_scia *ekd : pointer to EKD record (E.19)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.15 SCIA_LV1_WR_ASCII_SFP

Identifier	SCIA_LV1_WR_ASCII_SFP
Purpose	dump – in ASCII Format – the SFP records
Usage	`SCIA_LV1_WR_ASCII_SFP( param, num_dsr, sfp );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct sfp_scia *sfp : pointer to SFP record(s) (E.20)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.57.16 SCIA_LV1_WR_ASCII_ASFP

Identifier	SCIA_LV1_WR_ASCII_ASFP
Purpose	dump – in ASCII Format – the ASFP records
Usage	`SCIA_LV1_WR_ASCII_ASFP( param, num_dsr, asfp );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_dsr : number of data sets struct asfp_scia *asfp : pointer to ASFP record(s) (E.21)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.58 SCIA_LV1_WR_ASCII_MDS

Identifier	SCIA_LV1_WR_ASCII_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump SCIAMACHY level 1 Measurement Data Sets in ASCII
Comment	contains SCIA_LV1_WR_ASCII_MDS, SCIA_LV1C_WR_ASCII_MDS, SCIA_LV1C_WR_ASCII_MDS_PMD, SCIA_LV1C_WR_ASCII_MDS_POLV

4.58.1 SCIA_LV1_WR_ASCII_MDS

Identifier	SCIA_LV1_WR_ASCII_MDS
Purpose	dump – in ASCII Format – the MDS records (Lv1b)
Usage	`SCIA_LV1_WR_ASCII_MDS( param, num_mds, mds );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_mds : number of MDS records struct mds1_scia *mds : MDS struct (level 1b) (E.31)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.58.2 SCIA_LV1C_WR_ASCII_MDS

Identifier	SCIA_LV1C_WR_ASCII_MDS
Purpose	dump – in ASCII Format – the MDS (Lv1c) records
Usage	`SCIA_LV1C_WR_ASCII_MDS( param, num_mds, mds_1c );`
Input	struct param_record param : struct holding user-defined settings unsigned int num_mds : number of MDS records struct mds1c_scia *mds_1c : MDS struct (level 1c) (C.11)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.58.3 SCIA_LV1C_WR_ASCII_MDS_PMD

Identifier	SCIA_LV1C_WR_ASCII_MDS_PMD
Purpose	dump – in ASCII Format – the PMD MDS (Lv1c)
Usage	`SCIA_LV1C_WR_ASCII_MDS_PMD( param, mds_pmd );`
Input	struct param_record param : struct holding user-defined settings struct mds1c_pmd *mds_pmd : level 1c PMD MDS struct (C.12)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.58.4 SCIA_LV1C_WR_ASCII_MDS_POLV

Identifier	SCIA_LV1C_WR_ASCII_MDS_POLV
Purpose	dump – in ASCII Format – the POLV MDS (Lv1c)
Usage	`SCIA_LV1C_WR_ASCII_MDS_POLV( param, mds_polV);`
Input	struct param_record param : struct holding user-defined settings struct mds1c_polV *mds_polV : level 1c polV MDS struct (C.13)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.59 SCIA_LV1_WR_ASCII_SPH

Identifier	SCIA_LV1_WR_ASCII_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Specific Product Header of the level 1b product
Usage	`SCIA_LV1_WR_ASCII_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph1_scia *sph : pointer to SPH structure (E.6)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.60 SCIA_LV1_WR_H5_ASFP

Identifier	SCIA_LV1_WR_H5_ASFP
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 ASFP data
Usage	`SCIA_LV1_WR_H5_ASFP( param, nr_asfp, asfp );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_asfp : number of ASFP parameters
	struct asfp_scia *asfp : small aperture slit function (E.21) parameters
Returns	Nothing
Comment	None

4.61 SCIA_LV1_WR_H5_AUX

Identifier	SCIA_LV1_WR_H5_AUX
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 Auxiliary data
Usage	`SCIA_LV1_WR_H5_AUX( param, nr_aux, aux );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_aux : number of Auxiliary data packets
	struct aux_scia *aux : structure with Auxiliary data packets (E.24)
Returns	Nothing
Comment	None

4.62 SCIA_LV1_WR_H5_BASE

Identifier	SCIA_LV1_WR_H5_BASE
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SPECTRAL_BASE data
Usage	`SCIA_LV1_WR_H5_BASE( param, base );`
Input	struct param_record param : struct holding user-defined settings
	struct base_scia *base : Spectral Base Parameters (E.12)
Returns	Nothing
Comment	None

4.63 SCIA_LV1_WR_H5_CLCP

Identifier	SCIA_LV1_WR_H5_CLCP
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 CLCP data
Usage	`SCIA_LV1_WR_H5_CLCP( param, clcp );`
Input	struct param_record param : struct holding user-defined settings
	struct clcp_scia *clcp : Leakage Current Paramameters (const) (E.9)
Returns	Nothing
Comment	None

4.64 SCIA_LV1_WR_H5_DARK

Identifier	SCIA_LV1_WR_H5_DARK
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 DARK data
Usage	`SCIA_LV1_WR_H5_DARK( param, nr_dark, dark );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_dark : number of DARK/Etalon parameters
	struct dark_scia *dark : new DARK average parameters (E.26)
Returns	Nothing
Comment	None

4.65 SCIA_LV1_WR_H5_EKD

Identifier	SCIA_LV1_WR_H5_EKD
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 EKD data
Usage	`SCIA_LV1_WR_H5_EKD( param, ekd );`
Input	struct param_record param : struct holding user-defined settings
	struct ekd_scia *ekd : Errors on Key data (E.19)
Returns	Nothing
Comment	None

4.66 SCIA_LV1_WR_H5_LCPN

Identifier	SCIA_LV1_WR_H5_LCPN
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 LCPN data
Usage	`SCIA_LV1_WR_H5_LCPN( param, nr_lcpn, lcpn );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_lcpn : number of new Leakage Current Param.
	struct lcpn_scia *lcpn : new Leakage Current Paramameters (E.25)
Returns	Nothing
Comment	None

4.67 SCIA_LV1_WR_H5_PMD

Identifier	SCIA_LV1_WR_H5_PMD
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 PMD data
Usage	`SCIA_LV1_WR_H5_PMD( param, nr_pmd, pmd );`
Input	struct param_record param : struct holding user-defined settings unsigned int nr_pmd : number of PMD data packets struct pmd_scia *pmd : structure with PMD data packets (E.23)
Returns	Nothing
Comment	None

4.68 SCIA_LV1_WR_H5_PPG

Identifier	SCIA_LV1_WR_H5_PPG
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 PPG data
Usage	`SCIA_LV1_WR_H5_PPG( param, nr_ppg, ppg );`
Input	struct param_record param : struct holding user-defined settings struct ppg_scia *ppg : PPG/Etalon parameters (E.11)
Returns	Nothing
Comment	None

4.69 SCIA_LV1_WR_H5_PPGN

Identifier	SCIA_LV1_WR_H5_PPGN
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 PPGN data
Usage	`SCIA_LV1_WR_H5_PPGN( param, nr_ppgn, ppgn );`
Input	struct param_record param : struct holding user-defined settings unsigned int nr_ppgn : number of PPGN/Etalon parameters struct ppgn_scia *ppgn : new PPG/Etalon parameters (E.27)
Returns	Nothing
Comment	None

4.70 SCIA_LV1_WR_H5_PSP

Identifier	SCIA_LV1_WR_H5_PSP
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 PSP data

Usage	`SCIA_LV1_WR_H5_PSPN( param, nr_psp, pspn );`
	`SCIA_LV1_WR_H5_PSPL( param, nr_psp, psplo );`
	`SCIA_LV1_WR_H5_PSPO( param, nr_psp, psplo );`

Input	struct param_record param : struct holding user-defined settings unsigned int nr_psp: number of nadir PSP's struct pspn_scia *pspn : Polarisation Sensitivity Parameters (E.15) (nadir) struct psplo_scia *psplo: Polarisation Sensitivity Parameters (E.16) (limb/occultation)
Returns	Nothing
Comment	None

4.71 SCIA_LV1_WR_H5_RSP

Identifier	SCIA_LV1_WR_H5_RSP
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 RSP data

Usage	`SCIA_LV1_WR_H5_RSPN( param, nr_rsp, rspn );`
	`SCIA_LV1_WR_H5_RSPL( param, nr_rsp, rsplo );`
	`SCIA_LV1_WR_H5_RSPO( param, nr_rsp, rsplo );`

Input	struct param_record param : struct holding user-defined settings unsigned int nr_rsp : number of nadir RSP's struct rspn_scia *rspn : Radiance Sensitivity Parameters (E.17) (nadir) struct rsplo_scia *rsplo: Radiance Sensitivity Parameters (E.18) (limb/occultation)
Returns	Nothing
Comment	None

4.72 SCIA_LV1_WR_H5_SCP

Identifier	SCIA_LV1_WR_H5_SCP
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SCP data
Usage	`SCIA_LV1_WR_H5_SCP( param, nr_scp, scp );`
Input	struct param_record param : struct holding user-defined settings unsigned int nr_scp : number of Spectral Calibration Params. struct scp_scia *scp : Spectral Calibration Parameters (E.13)
Returns	Nothing
Comment	None

4.73 SCIA_LV1_WR_H5_SCPN

Identifier	SCIA_LV1_WR_H5_SCPN
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SCPN data
Usage	`SCIA_LV1_WR_H5_SCPN( param, nr_scpn, scpn );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_scpn : number of Spectral Calibration Params
	struct scpn_scia *scpn : new Spectral Calibration Parameters (E.28)
Returns	Nothing
Comment	None

4.74 SCIA_LV1_WR_H5_SFP

Identifier	SCIA_LV1_WR_H5_SFP
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SFP data
Usage	`SCIA_LV1_WR_H5_SFP( param, nr_sfp, sfp );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_sfp : number of SFP parameters
	struct sfp_scia *sfp : slit function parameters (E.20)
Returns	Nothing
Comment	None

4.75 SCIA_LV1_WR_H5_SIP

Identifier	SCIA_LV1_WR_H5_SIP
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SIP data
Usage	`SCIA_LV1_WR_H5_SIP( param, sip );`
Input	struct param_record param : struct holding user-defined settings
	struct sip_scia *sip : Static Instrument Parameters (E.8)
Returns	Nothing
Comment	None

4.76 SCIA_LV1_WR_H5_SPH

Identifier	SCIA_LV1_WR_H5_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SPH data
Usage	`SCIA_LV1_WR_H5_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph1_scia *sph : Specific Product Header data (E.6)
Returns	Nothing
Comment	None

4.77 SCIA_LV1_WR_H5_SQADS

Identifier	SCIA_LV1_WR_H5_SQADS
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SQADS data
Usage	`SCIA_LV1_WR_H5_SQADS( param, nr_sqads, sqads );`
Input	struct param_record param : struct holding user-defined settings unsigned int nr_sqads : number of Summary of Quality Flags struct sqads1_scia *sqads : Summary of Quality Flags per State (E.7)
Returns	Nothing
Comment	None

4.78 SCIA_LV1_WR_H5_SRS

Identifier	SCIA_LV1_WR_H5_SRS
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SRS data
Usage	`SCIA_LV1_WR_H5_SRS( param, nr_srs, srs );`
Input	struct param_record param : struct holding user-defined settings unsigned int nr_srs : number of Sun reference spectra struct srs_scia *srs : Sun Reference Spectrum (E.14)
Returns	Nothing
Comment	None

4.79 SCIA_LV1_WR_H5_SRSN

Identifier	SCIA_LV1_WR_H5_SRSN
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 SRSN data
Usage	`SCIA_LV1_WR_H5_SRSN( param, nr_srsn, srsn );`
Input	struct param_record param : struct holding user-defined settings unsigned int nr_srsn : number of Sun reference spectra struct srsn_scia *srsn : Sun Reference Spectrum (E.29)
Returns	Nothing
Comment	None

4.80 SCIA_LV1_WR_H5_STATE

Identifier	SCIA_LV1_WR_H5_STATE
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 STATE data
Usage	`SCIA_LV1_WR_H5_STATE( param, nr_state, state );`
Input	struct param_record param : struct holding user-defined settings unsigned int nr_state : number of Summary of Quality Flags struct state1_scia *state : States of the Product (E.22)
Returns	Nothing
Comment	None

4.81 SCIA_LV1_WR_H5_VLCP

Identifier	SCIA_LV1_WR_H5_VLCP
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1 VLCP data
Usage	`SCIA_LV1_WR_H5_VLCP( param, nr_vlcp, vlcp );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_vlcp : number of var. Leakage Current Param.
	struct vlcp_scia *vlcp : Leakage Current Paramameters (var.) (E.10)
Returns	Nothing
Comment	None

4.82 SCIA_LV1_WR_MDS

Identifier	SCIA_LV1_WR_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	write SCIAMACHY level 1b/1c Measurement Data Sets
Returns	number of MDS records written
Comment	None

4.82.1 SCIA_LV1_WR_MDS

Identifier	SCIA_LV1_WR_MDS
Purpose	read MDS of one state from a SCIAMACHY level 1b product
Usage	`nr_mds = SCIA_LV1_WR_MDS( fd, num_mds, mds );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_mds : number of MDS records
	struct mds1_scia *mds : structure with level 1b MDS records (E.31)
Returns	number of written level 1b MDS records (unsigned int),
	error status passed by global variable “nadc_stat”
Comment	none

4.82.2 SCIA_LV1C_WR_MDS

Identifier	SCIA_LV1C_WR_MDS
Purpose	write SCIAMACHY level 1c MDS
Usage	`SCIA_LV1C_WR_MDS( fd, num_mds, mds );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_mds : number of MDS records
	struct mds1c_scia *mds : structure for level 1c MDS (C.11)
Returns	nothing,
	error status passed by global variable “nadc_stat”
Comment	None

4.82.3 SCIA_LV1C_WR_MDS_PMD

Identifier	SCIA_LV1C_WR_MDS_PMD
Purpose	write SCIAMACHY level 1c PMD MDS
Usage	`SCIA_LV1C_WR_MDS_PMD( fd, pmd );`
Input	FILE *fd : (open) stream pointer
	struct mds1c_scia *pmd : structure for level 1c PMD MDS (C.11)
Returns	nothing,
	error status passed by global variable "nadc_stat"
Comment	None

4.82.4 SCIA_LV1C_WR_MDS_POLV

Identifier	SCIA_LV1C_WR_MDS_POLV
Purpose	write SCIAMACHY level 1c POLV MDS
Usage	`SCIA_LV1C_WR_MDS_POLV( fd, polV );`
Input	FILE *fd : (open) stream pointer
	struct mds1c_scia *polV : structure for level 1c POLV MDS (C.11)
Returns	nothing,
	error status passed by global variable "nadc_stat"
Comment	None

4.83 SCIA_LV1C_PDS_CALOPT

Identifier	SCIA_LV1C_PDS_CALOPT
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write the calibration options GADS to SciaL1C
Comment	contains SCIA_LV1C_RD_CALOPT, SCIA_LV1C_UPDATE_CALOPT
	and SCIA_LV1C_WR_CALOPT

4.83.1 SCIA_LV1C_RD_CALOPT

Identifier	SCIA_LV1C_RD_CALOPT
Purpose	read the calibration options GADS from SciaL1C file
Usage	`nr_dsr = SCIA_LV1C_RD_CALOPT( fd, num_dsd, dsd, &calopt );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct cal_options *calopt : structure for L1c calibration options (E.32)
Returns	number of data set records read (unsigned int)
	error status passed by global variable "nadc_stat"
Comment	none

4.83.2 SCIA_LV1C_UPDATE_CALOPT

Identifier	SCIA_LV1C_UPDATE_CALOPT
Purpose	update Calibration Options GADS
Usage	`nr_dsr = SCIA_LV1C_UPDATE_CALOPT( do_init, param, &calopt );`
Input	int do_init : initialise calopt-record
	struct param_record param : struct holding user-defined settings (A.1)
Output	struct cal_options *calopt : structure for L1c calibration options (E.32)
Returns	nothing
Comment	none

4.83.3 SCIA_LV1C_WR_CALOPT

Identifier	SCIA_LV1C_WR_CALOPT
Purpose	write Calibration Options GADS to SciaL1C file
Usage	`SCIA_LV1C_WR_CALOPT( fd, num_calopt, &calopt );`
Input	FILE *fd : (open) stream pointer
	unsigned int num_calopt : number of CalOpt records
	struct cal_options calopt : structure for the calibration options (E.32)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.84 SCIA_LV1C_WR_ASCII_CALOPT

Identifier	SCIA_LV1C_WR_ASCII_CALOPT
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump the calibration options GADS to SciaL1C
Returns	Nothing
Comment	None

4.85 SCIA_LV2_RD_BIAS

Identifier	SCIA_LV2_RD_BIAS
Author	R.M. van Hees
Language	ANSI C
Purpose	read BIAS Fitting Window Application Data set
Usage	`nbyte = SCIA_LV2_RD_BIAS( fd, bias_name, num_dsd, dsd, bias );`
Input	FILE *fd : stream pointer
	char bias_name[] : PDS name for BIAS data set
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct bias_scia **bias : BIAS Fitting Window Application Data set (F.10)
Returns	number of data set records read (unsigned int)
Comment	None

4.86 SCIA_LV2_RD_CLD

Identifier	SCIA_LV2_RD_CLD
Author	R.M. van Hees
Language	ANSI C
Purpose	read Cloud and Aerosol Data sets
Usage	`nbyte = SCIA_LV2_RD_CLD( fd, num_dsd, dsd, &cld );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct cld_scia **cld : Cloud and Aerosol Data sets (F.8)
Returns	number of data set records read (unsigned int)
Comment	None

4.87 SCIA_LV2_RD_DOAS

Identifier	SCIA_LV2_RD_DOAS
Author	R.M. van Hees
Language	ANSI C
Purpose	read DOAS Fitting Window Application data sets
Usage	`nr_dsr = SCIA_LV2_RD_DOAS( fd, doas_name, num_dsd, dsd, doas );`
Input	FILE *fd : stream pointer
	char doas_name[] : PDS name for DOAS data set
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct doas_scia **doas: DOAS Fitting Window Application data set (F.9)
Returns	number of data set records read (unsigned int)
Comment	None

4.88 SCIA_LV2_RD_GEO

Identifier	SCIA_LV2_RD_GEO
Author	R.M. van Hees
Language	ANSI C
Purpose	read Geolocation Data Sets
Usage	`nr_dsr = SCIA_LV2_RD_GEO( fd, num_dsd, dsd, &geo );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct geo_scia *geo : Geolocation Data sets (F.7)
Returns	number of data set records read (unsigned int)
Comment	None

4.89 SCIA_LV2_RD_SPH

Identifier	SCIA_LV2_RD_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	read Specific Product Header of Level 2 Product
Usage	`SCIA_LV2_RD_SPH( fd, mph, &sph );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi *mph : main product header (A.5)
Output	struct sph2_scia *sph : structure for the SPH (F.4)
Returns	nothing (check global error status)
Comment	None

4.90 SCIA_LV2_RD_SQADS

Identifier	SCIA_LV2_RD_SQADS
Author	R.M. van Hees
Language	ANSI C
Purpose	read Summary of Quality Flags per State
Usage	`nr_dsr = SCIA_LV2_RD_SQADS( fd, num_dsd, dsd, &sqads );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct sqads2_scia **sqads : summary of quality flags per state (F.5)
Returns	number of data set records read (unsigned int)
Comment	None

4.91 SCIA_LV2_RD_STATE

Identifier	SCIA_LV2_RD_STATE
Author	R.M. van Hees
Language	ANSI C
Purpose	read States of the Product
Usage	`nbyte = SCIA_LV2_RD_STATE( fd, num_dsd, dsd, &state );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct state2_scia **state : states of the product (F.6)
Returns	number of data set records read (unsigned int)
Comment	None

4.92 SCIA_LV2_WR_ASCII_ADS

Identifier	SCIA_LV2_WR_ASCII_ADS
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Annotation Data Sets in ASCII
Returns	Nothing
Comment	contains SCIA_LV2_WR_ASCII_SQADS, SCIA_LV2_WR_ASCII_STATE, SCIA_LV2_WR_ASCII_GEO

4.92.1 SCIA_LV2_WR_ASCII_SQADS

Identifier	SCIA_LV2_WR_ASCII_SQADS
Purpose	dump – in ASCII Format – the SQADS
Usage	`SCIA_LV2_WR_ASCII_SQADS( param, num_dsr, sqads );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct sqads2_scia *sqads : pointer to SQADS records (F.5)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.92.2 SCIA_LV2_WR_ASCII_STATE

Identifier	SCIA_LV2_WR_ASCII_STATE
Purpose	dump – in ASCII Format – the STATE records
Usage	`SCIA_LV2_WR_ASCII_STATE( param, num_dsr, state );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct state2_scia *state : pointer to STATE records (F.6)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.92.3 SCIA_LV2_WR_ASCII_GEO

Identifier	SCIA_LV2_WR_ASCII_GEO
Purpose	dump – in ASCII Format – the records
Usage	`SCIA_LV2_WR_ASCII_GEO( param, num_dsr, geo );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct geo_scia *geo : pointer to Geolocation records (F.7)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.93 SCIA_LV2_WR_MDS

Identifier	SCIA_LV2_WR_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	dump Measurement Data Sets in ASCII
Returns	Nothing
Comment	contains SCIA_LV2_WR_ASCII_CLD, SCIA_LV2_WR_ASCII_BIAS, SCIA_LV2_WR_ASCII_DOAS

4.93.1 SCIA_LV2_WR_ASCII_CLD

Identifier	SCIA_LV2_WR_ASCII_CLD
Purpose	dump – in ASCII Format – the Cloud/Aerosol datasets
Usage	`SCIA_LV2_WR_ASCII_CLD( param, num_dsr, cld );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct cld_scia *cld : pointer to Cloud/Aerosol records (F.8)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.93.2 SCIA_LV2_WR_ASCII_BIAS

Identifier	SCIA_LV2_WR_ASCII_BIAS
Purpose	dump – in ASCII Format – the BIAS datasets
Usage	`SCIA_LV2_WR_ASCII_BIAS( mds_name, param, num_dsr, bias );`
Input	char *mds_name : name of DOAS Fitting Window Application
	struct param_record param : struct holding user-defined settings (A.1)
	unsigned int num_dsr : number of data sets
	struct bias_scia *bias : pointer to BIAS records (F.10)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.93.3 SCIA_LV2_WR_ASCII_DOAS

Identifier	SCIA_LV2_WR_ASCII_DOAS
Purpose	dump – in ASCII Format – the DOAS datasets
Usage	`SCIA_LV2_WR_ASCII_DOAS( mds_name, param, num_dsr, doas );`
Input	char *mds_name : name of DOAS Fitting Window Application
	struct param_record param : struct holding user-defined settings (A.1)
	unsigned int num_dsr : number of data sets
	struct doas_scia *doas : pointer to DOAS records (F.9)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.94 SCIA_LV2_WR_ASCII_SPH

Identifier	SCIA_LV2_WR_ASCII_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Specific Product Header of the level 2 product
Usage	`SCIA_LV2_WR_ASCII_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph2_scia *sph : pointer to SPH structure (F.4)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.95 SCIA_LV2_WR_H5_BIAS

Identifier	SCIA_LV2_WR_H5_BIAS
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIA level 2 BIAS Fitting Application Data Set(s)
Usage	`SCIA_LV2_WR_H5_BIAS( bias_name, param, nr_bias, bias );`
Input	char bias_name[] : name of BIAS data set
	struct param_record param : struct holding user-defined settings (A.1)
	unsigned int nr_bias : number of Cloud/Aerosol data sets
	struct bias_scia *bias : BIAS Fitting Application Data Set(s) (F.10)
Returns	Nothing
Comment	None

4.96 SCIA_LV2_WR_H5_CLD

Identifier	SCIA_LV2_WR_H5_CLD
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 2 Cloud/Aerosol Data Set(s)
Usage	`SCIA_LV2_WR_H5_CLD( param, nr_cld, cld );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_cld : number of Cloud/Aerosol data sets
	struct cld_scia *cld : Cloud/Aerosol Data Set(s) (F.8)
Returns	Nothing
Comment	None

4.97 SCIA_LV2_WR_H5_DOAS

Identifier	SCIA_LV2_WR_H5_DOAS
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIA level 2 DOAS Fitting Application Data Set(s)
Usage	`SCIA_LV2_WR_H5_DOAS( doas_name, param, nr_doas, doas );`
Input	char doas_name[] : name of DOAS data set
	struct param_record param : struct holding user-defined settings (A.1)
	unsigned int nr_doas : number of Cloud/Aerosol data sets
	struct doas_scia *doas : DOAS Fitting Application Data Set(s) (F.9)
Returns	Nothing
Comment	None

4.98 SCIA_LV2_WR_H5_GEO

Identifier	SCIA_LV2_WR_H5_GEO
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 2 Geolocation Data Set(s)
Usage	`SCIA_LV2_WR_H5_GEO( param, nr_geo, geo );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_geo : number of Geolocations data sets
	struct geo_scia *geo : Geolocation Data Set(s) (F.7)
Returns	Nothing
Comment	None

4.99 SCIA_LV2_WR_H5_SPH

Identifier	SCIA_LV2_WR_H5_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 2 SPH data
Usage	`SCIA_LV2_WR_H5_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph2_scia *sph : Specific Product Header data (F.4)
Returns	Nothing
Comment	None

4.100 SCIA_LV2_WR_H5_SQADS

Identifier	SCIA_LV2_WR_H5_SQADS
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 2 SQADS data
Usage	`SCIA_LV2_WR_H5_SQADS( param, nr_sqads, sqads );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_sqads : number of Summary of Quality Flags
	struct sqads2_scia *sqads : Summary of Quality Flags per State (F.5)
Returns	Nothing
Comment	None

4.101 SCIA_LV2_WR_H5_STATE

Identifier	SCIA_LV2_WR_H5_STATE
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 2 STATE data
Usage	`SCIA_LV2_WR_H5_STATE( param, nr_state, state );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_state : number of Summary of Quality Flags
	struct state2_scia *state : States of the Product (F.6)
Returns	Nothing
Comment	None

4.102 SCIA_OL2_PDS_SPH

Identifier	SCIA_OL2_PDS_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Specific Product Header of Offline Level 2 Product
Comment	contains SCIA_OL2_RD_SPH and SCIA_OL2_WR_SPH

4.102.1 SCIA_OL2_RD_SPH

Identifier	SCIA_OL2_RD_SPH
Purpose	read Specific Product Header of Level 2 Offline Product
Usage	`SCIA_OL2_RD_SPH( fd, mph, &sph );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi mph : main product header (A.5)
Output	struct sph_sci_ol *sph : structure for the SPH (F.11)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.102.2 SCIA_OL2_WR_SPH

Identifier	SCIA_OL2_WR_SPH
Purpose	write Specific Product Header of Level 2 Offline Product
Usage	`SCIA_OL2_WR_SPH( fd, mph, sph );`
Input	FILE *fd : (open) stream pointer
	struct mph_envi mph : main product header (A.5)
	struct sph_sci_ol sph : structure for the SPH (F.11)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.103 SCIA_OL2_RD_CLD

Identifier	SCIA_OL2_RD_CLD
Author	R.M. van Hees
Language	ANSI C
Purpose	read Cloud and Aerosol Data sets
Usage	`nbyte = SCIA_OL2_RD_CLD( fd, num_dsd, dsd, &cld );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct cld_sci_ol **cld : Cloud and Aerosol Data sets (F.15)
Returns	number of data set records read (unsigned int)
Comment	None

4.104 SCIA_OL2_RD_GEO

Identifier	SCIA_OL2_RD_GEO
Author	R.M. van Hees
Language	ANSI C
Purpose	read Geolocation Data Sets
Usage	`nr_dsr = SCIA_OL2_RD_NGEO( fd, num_dsd, dsd, &geo );`
	`nr_dsr = SCIA_OL2_RD_LGEO( fd, num_dsd, dsd, &geo );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct ngeo_scia *geo : Nadir Geolocation Data sets (F.13)
	struct lgeo_scia *geo : Limb/Occultation Geolocation Data sets (F.14)
Returns	number of data set records read (unsigned int)
Comment	None

4.105 SCIA_OL2_RD_LCLD

Identifier	SCIA_OL2_RD_LCLD
Author	R.M. van Hees
Language	ANSI C
Purpose	read Limb Cloud Data sets
Usage	`nbyte = SCIA_OL2_RD_LCLD( fd, num_dsd, dsd, &lcld );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct lcld_scia **lcld : Limb Cloud Data sets (F.21)
Returns	number of data set records read (unsigned int)
Comment	None

4.106 SCIA_OL2_RD_LFIT

Identifier	SCIA_OL2_RD_LFIT
Author	R.M. van Hees
Language	ANSI C
Purpose	read Limb Fitting Window Application Data sets
Usage	`nbyte = SCIA_OL2_RD_LFIT( fd, num_dsd, dsd, &lfit );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct lfit_scia **lfit : Limb Fitting Window Data sets (F.20)
Returns	number of data set records read (unsigned int)
Comment	None

4.107 SCIA_OL2_RD_NFIT

Identifier	SCIA_OL2_RD_NFIT
Author	R.M. van Hees
Language	ANSI C
Purpose	read Nadir Fitting Window Application Data sets
Usage	`nbyte = SCIA_OL2_RD_NFIT( fd, num_dsd, dsd, &nfit );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct nfit_scia **nfit : Nadir Fitting Window Data sets (F.16)
Returns	number of data set records read (unsigned int)
Comment	None

4.108 SCIA_OL2_RD_SQADS

Identifier	SCIA_OL2_RD_SQADS
Author	R.M. van Hees
Language	ANSI C
Purpose	read Summary of Quality Flags per State
Usage	`nr_dsr = SCIA_OL2_RD_SQADS( fd, num_dsd, dsd, &sqads );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi *dsd : structure for the DSDs (A.6)
Output	struct sqads_sci_ol **sqads : summary of quality flags per state (F.12)
Returns	number of data set records read (unsigned int)
Comment	None

4.109 SCIA_LV1_WR_ADS

Identifier	SCIA_LV1_WR_ADS
Author	R.M. van Hees
Language	ANSI C
Purpose	dump Annotation Data Sets in ASCII
Returns	Nothing
Comment	contains SCIA_OL2_WR_ASCII_SQADS, SCIA_OL2_WR_ASCII_NGEO, SCIA_OL2_WR_ASCII_LGEO

4.109.1 SCIA_OL2_WR_ASCII_SQADS

Identifier	SCIA_OL2_WR_ASCII_SQADS
Purpose	dump – in ASCII Format – the SQADS annotation dataset
Usage	`SCIA_OL2_WR_ASCII_SQADS( param, num_dsr, sqads );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct sqads_sci_ol *sqads: pointer to SQADS records (F.12)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.109.2 SCIA_OL2_WR_ASCII_NGEO

Identifier	SCIA_OL2_WR_ASCII_NGEO
Purpose	dump – in ASCII Format – the NADIR GEO annotation dataset
Usage	`SCIA_OL2_WR_ASCII_NGEO( param, num_dsr, geo );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct ngeo_scia *geo : pointer to Geolocation (NADIR) records (F.13)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

4.109.3 SCIA_OL2_WR_ASCII_LGEO

Identifier	SCIA_OL2_WR_ASCII_LGEO
Purpose	dump – in ASCII Format – the LIMB GEO annotation dataset
Usage	`SCIA_OL2_WR_ASCII_LGEO( param, num_dsr, geo );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct lgeo_scia *geo : pointer to Geolocation (LIMB) records (F.14)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.110 SCIA_OL2_WR_MDS

Identifier	SCIA_OL2_WR_MDS
Author	R.M. van Hees
Language	ANSI C
Purpose	dump Measurement Data Sets in ASCII
Returns	Nothing
Comment	contains SCIA_OL2_WR_ASCII_CLD, SCIA_OL2_WR_ASCII_NFIT,
	SCIA_OL2_WR_ASCII_LFIT, SCIA_OL2_WR_ASCII_LCLD

4.110.1 SCIA_OL2_WR_ASCII_CLD

Identifier	SCIA_OL2_WR_ASCII_CLD
Purpose	dump – in ASCII Format – the Cloud/Aerosol datasets
Usage	`SCIA_OL2_WR_ASCII_CLD( param, num_dsr, cld );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct cld_sci_ol *cld : pointer to Cloud/Aerosol records (F.15)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.110.2 SCIA_OL2_WR_ASCII_NFIT

Identifier	SCIA_OL2_WR_ASCII_NFIT
Purpose	dump – in ASCII Format – the Nadir Fitting Window datasets
Usage	`SCIA_OL2_WR_ASCII_NFIT( mds_name, param, num_dsr, nfit );`
Input	char *mds_name : name of Nadir Fitting Window
	struct param_record param : struct holding user-defined settings (A.1)
	unsigned int num_dsr : number of data sets
	struct nfit_scia *nfit : pointer to Fitting Window records (F.16)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.110.3 SCIA_OL2_WR_ASCII_LFIT

Identifier	SCIA_OL2_WR_ASCII_LFIT
Purpose	dump – in ASCII Format – the Limb Fitting Window datasets
Usage	`SCIA_OL2_WR_ASCII_LFIT( mds_name, param, num_dsr, lfit );`
Input	char *mds_name : name of Limb Fitting Window
	struct param_record param : struct holding user-defined settings (A.1)
	unsigned int num_dsr : number of data sets
	struct lfit_scia *lfit : pointer to Fitting Window records (F.20)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.110.4 SCIA_OL2_WR_ASCII_LCLD

Identifier	SCIA_OL2_WR_ASCII_LCLD
Purpose	dump – in ASCII Format – the Limb Clouds datasets
Usage	`SCIA_OL2_WR_ASCII_LCLD( param, num_dsr, lcld );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of data sets
	struct lcld_scia *lcld : pointer to Limb Clouds records (F.21)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.111 SCIA_OL2_WR_ASCII_SPH

Identifier	SCIA_OL2_WR_ASCII_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Specific Product Header of the Offline level 2 product
Usage	`SCIA_OL2_WR_ASCII_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph_sci_ol *sph : pointer to SPH structure (F.11)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

4.112 SCIA_OL2_WR_H5_CLD

Identifier	SCIA_OL2_WR_H5_CLD
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 2 Cloud/Aerosol Data Set(s)
Usage	`SCIA_OL2_WR_H5_CLD( param, nr_cld, cld );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_cld : number of Cloud/Aerosol data sets
	struct cld_sci_ol *cld : Cloud/Aerosol Data Set(s) (F.15)
Returns	Nothing
Comment	None

4.113 SCIA_OL2_WR_H5_GEO

Identifier	SCIA_OL2_WR_H5_GEO
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 2 Geolocation Data Set(s)
Usage	`SCIA_OL2_WR_H5_NGEO( param, nr_geo, geo );` `SCIA_OL2_WR_H5_LGEO( param, nr_geo, geo );`
Input	struct param_record param : struct holding user-defined settings unsigned int nr_geo : number of Geolocations data sets struct ngeo_scia *geo : Nadir Geolocation Data Set(s) (F.13) struct lgeo_scia *geo : Limb Geolocation Data Set(s) (F.14)
Returns	Nothing
Comment	None

4.114 SCIA_OL2_WR_H5_LCLD

Identifier	SCIA_OL2_WR_H5_LCLD
Author	R.M. van Hees
Language	ANSI C
Purpose	write SCIAMACHY level 2 Limb Clouds data sets
Usage	`SCIA_OL2_WR_H5_LCLD( lcld_name, param, nr_lcld, lcld );`
Input	char lcld_name[] : name of fitted species struct param_record param : struct holding user-defined settings (A.1) unsigned int nr_lcld : number of Nadir Fitting Windows struct lcld_scia *lcld : Limb Clouds Data Set(s) (F.21)
Returns	Nothing
Comment	None

4.115 SCIA_OL2_WR_H5_LFIT

Identifier	SCIA_OL2_WR_H5_LFIT
Author	R.M. van Hees
Language	ANSI C
Purpose	write SCIAMACHY level 2 Limb/Occultation Fitting Window Application Data Set(s)
Usage	`SCIA_OL2_WR_H5_LFIT( lfit_name, param, nr_lfit, lfit );`
Input	char lfit_name[] : name of fitted species struct param_record param : struct holding user-defined settings (A.1) unsigned int nr_lfit : number of lfit MDS struct lfit_scia *lfit : Limb/Occultation Fitting Window Data (F.20)
Returns	Nothing
Comment	None

4.116 SCIA_OL2_WR_H5_NFIT

Identifier	SCIA_OL2_WR_H5_NFIT
Author	R.M. van Hees
Language	ANSI C
Purpose	write SCIAMACHY level 2 Nadir Fitting Window Application datasets
Usage	`SCIA_OL2_WR_H5_NFIT( nfit_name, param, nr_nfit, nfit );`
Input	char nfit_name[] : name of fitted species
	struct param_record param : struct holding user-defined settings (A.1)
	unsigned int nr_nfit : number of Nadir Fitting Windows
	struct nfit_scia *nfit : Nadir Fitting Window Data Set(s) (F.16)
Returns	Nothing
Comment	None

4.117 SCIA_OL2_WR_H5_SPH

Identifier	SCIA_OL2_WR_H5_SPH
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SPH from SCIAMACHY Offline level 2 product
Usage	`SCIA_OL2_WR_H5_SPH( param, sph );`
Input	struct param_record param : struct holding user-defined settings
	struct sph_sci_ol *sph : Specific Product Header data (F.11)
Returns	Nothing
Comment	None

4.118 SCIA_OL2_WR_H5_SQADS

Identifier	SCIA_OL2_WR_H5_SQADS
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 2 SQADS data
Usage	`SCIA_OL2_WR_H5_SQADS( param, nr_sqads, sqads );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_sqads : number of Summary of Quality Flags
	struct sqads_sci_ol *sqads : Summary of Quality Flags per State (F.12)
Returns	Nothing
Comment	None

4.119 SCIA_PDS_LADS

Identifier	SCIA_PDS_LADS
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write Geolocation of the state records
Comment	contains SCIA_RD_LADS and SCIA_WR_LADS

4.119.1 SCIA_RD_LADS

Identifier	SCIA_RD_LADS
Purpose	read Geolocation of the state records
Usage	`nr_dsr = SCIA_RD_LADS( fd, num_dsd, dsd, &lads );`
Input	FILE *fd : stream pointer
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Output	struct lads_scia **lads : geolocation of states (C.10)
Returns	number of data set records read (unsigned int)
	error status passed by global variable “nadc_stat”
Comment	none

4.119.2 SCIA_LV1_WR_LADS

Identifier	SCIA_LV1_WR_LADS
Purpose	write Geolocation of the state records
Usage	`SCIA_LV1_WR_LADS( fd, num_lads, &lads );`
Input	FILE *fd : stream pointer
	unsigned int num_lads : number of LADS records
	struct lads_scia *lads : geolocation of states records (C.10)
Returns	nothing
	error status passed by global variable “nadc_stat”
Comment	none

4.120 SCIA_RD_H5_MEM

Identifier	SCIA_RD_H5_MEM
Author	R.M. van Hees
Language	ANSI C
Purpose	read table for Reticon non-linearity correction
Usage	`SCIA_RD_H5_MEM( &mem );`
Output	struct scia_memcorr *mem : (C.2)
	size_t dims[2] : dimension of matrix (hdf5 definition)
	float **matrix : memory correction matrix
Returns	Nothing
Comment	You need to release allocated memory, use SCIA_FREE_H5_MEM

4.121 SCIA_RD_H5_NLIN

Identifier	SCIA_RD_H5_NLIN
Author	R.M. van Hees
Language	ANSI C
Purpose	read table for Epitaxx non-linearity correction
Usage	`SCIA_RD_H5_NLIN( nlcorr_db, &nlin );`
Input	char *nlcorr_db : full-path to nlcorr database (or NULL)
Output	struct scia_nlincorr *nlin : (C.3)
	size_t dims[2] : dimension of matrix (hdf5 definition)
	char curve[8192] : index to correction curve
	float **matrix : non-linearity correction matrix
Returns	Nothing
Comment	You need to release allocated non-linearity, use SCIA_FREE_H5_NLIN

4.122 SCIA_RD_H5_RSP

Identifier	SCIA_RD_H5_RSP
Author	R.M. van Hees
Language	ANSI C
Purpose	read Polarisation Sensitivity Parameters from external file
Comment	- contains SCIA_RD_H5_PSPN, SCIA_RD_H5_PSPL, SCIA_RD_H5_PSPO

4.122.1 SCIA_RD_H5_PSPN

Identifier	SCIA_RD_H5_PSPN
Purpose	read Polarization Sensitivity Parameters (nadir) from HDF5 file
Usage	`num_psp = SCIA_RD_H5_PSPN( &pspn );`
Output	struct pspn_scia **pspn : Polarization Sensitivity Parameters (E.15) (nadir)
Returns	number of data set records read (unsigned short) error status passed by global variable “nadc_stat”
Comment	none

4.122.2 SCIA_RD_H5_PSPL

Identifier	SCIA_RD_H5_PSPL
Purpose	read Polarization Sensitivity Parameters (limb) from external file
Usage	`num_psp = SCIA_RD_H5_PSPL( &pspl );`
Output	struct psplo_scia **pspl : Polarization Sensitivity Parameters (E.16) (limb)
Returns	number of data set records read (unsigned short) error status passed by global variable “nadc_stat”
Comment	none

4.122.3 SCIA_RD_H5_PSPO

Identifier	SCIA_RD_H5_PSPO
Purpose	read Polarization Sensitivity Parameters (limb) from external file
Usage	`num_psp = SCIA_RD_H5_PSPO( &pspo );`
Output	struct psplo_scia **pspo : Polarization Sensitivity Parameters (E.16) (limb)
Returns	number of data set records read (unsigned short) error status passed by global variable “nadc_stat”
Comment	none

4.123 SCIA_RD_H5_RSP

Identifier	SCIA_RD_H5_RSP
Author	R.M. van Hees
Language	ANSI C
Purpose	read Radiance Sensitivity Parameters from external file
Comment	- contains SCIA_RD_H5_RSPN, SCIA_RD_H5_RSPL, SCIA_RD_H5_RSPO and SCIA_RD_H5_RSPD

4.123.1 SCIA_RD_H5_RSPN

Identifier	SCIA_RD_H5_RSPN
Purpose	read Radiance Sensitivity Parameters (nadir) from external file
Usage	`num_rsp = SCIA_RD_H5_RSPN( &rspn );`
Output	struct rspn_scia **rspn : radiance sensitivity parameters (Nadir) (E.17)
Returns	number of data set records read (unsigned short) error status passed by global variable "nadc_stat"
Comment	none

4.123.2 SCIA_RD_H5_RSPL

Identifier	SCIA_RD_H5_RSPL
Purpose	read Radiance Sensitivity Parameters (limb) from external file
Usage	`num_rsp = SCIA_RD_H5_RSPL( &rspl );`
Output	struct rsplo_scia **rspl : Radiance Sensitivity Parameters (E.18) (limb)
Returns	number of data set records read (unsigned short) error status passed by global variable "nadc_stat"
Comment	none

4.123.3 SCIA_RD_H5_RSPO

Identifier	SCIA_RD_H5_RSPO
Purpose	read Radiance Sensitivity Parameters (limb) from external file
Usage	`num_rsp = SCIA_RD_H5_RSPO( &rspo );`
Output	struct rsplo_scia **rspo : Radiance Sensitivity Parameters (E.18) (limb)
Returns	number of data set records read (unsigned short) error status passed by global variable "nadc_stat"
Comment	none

4.123.4 SCIA_RD_H5_RSPD

Identifier	SCIA_RD_H5_RSPD
Purpose	read keydata to calculate Radiance Sensitivity Parameters (diffuser measurement) from external file
Usage	`Read_SCIA_H5_RSPD( &rspd_key );`
Output	struct rspd_key *key : keydata radiance sensitivity parameters (E.38)
Returns	number of data set records read (unsigned short) error status passed by global variable "nadc_stat"
Comment	none

4.124 SCIA_RD_H5_STRAY

Identifier	SCIA_RD_H5_STRAY
Author	R.M. van Hees
Language	ANSI C
Purpose	read straylight correction matrix
Usage	`SCIA_RD_H5_STRAY( &stray );`
Output	struct scia_straycorr *stray : (C.4)
	size_t dims[2] : dimension of matrix (hdf5 definition)
	float *grid_in : pixel grid of input array (dims[1])
	float *grid_out : pixel grid of output array (dims[0])
	float **matrix : straylight correction matrix
Returns	Nothing
Comment	You need to release allocated memory, use SCIA_FREE_H5_STRAY

4.125 SCIA_RD_MFACTOR

Identifier	SCIA_RD_MFACTOR
Author	K. Bramstedt
Language	ANSI C
Purpose	read in m-factors from external database
Usage	`SCIA_RD_MFACTOR( mftype, sensing_start, calibFlag, mfactor );`
Input	enum mf_type mftype : which type is requested
	char *sensing_start : taken from MPH
	unsigned int calibFlag : bit-flag which defines how to calibrate
Output	float *mfactor : array holding mfactor
Returns	nothing
Comment	nothing

4.125.1 Scia_rd_aux_mfactor

Identifier	Scia_rd_aux_mfactor
Purpose	read m-factor for science channels from database
	with auxiliary files.
Usage	`Scia_rd_H5_mfactor( mftype, sensing_start, mfactor );`
Input	enum mf_type mftype : which type is requested
	char* sensing_start : sensing start time yyyymmdd_hhmmss
Output	float *mfactor : array to write mfactor
Returns	nothing
Comment	none

4.125.2 Scia_rd_H5_mfactor

Identifier	Scia_rd_H5_mfactor
Purpose	read m-factor for science channels from H5 database.
Usage	`Scia_rd_H5_mfactor( mftype, sensing_start, mfactor );`
Input	enum mf_type mftype : which type is requested
	char* sensing_start : sensing start time yyyymmdd_hhmmss
Output	float *mfactor : array to write mfactor
Returns	nothing
Comment	none

4.126 SCIA_WR_ASCII_LADS

Identifier	SCIA_WR_ASCII_LADS
Author	R.M. van Hees
Language	ANSI C
Purpose	Dump Geolocation of State
Usage	`SCIA_WR_ASCII_LADS( param, num_dsr, lads );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int num_dsr : number of LADS records
	struct lads_scia *lads : structure for the LADS records (C.10)
Returns	Nothing
Comment	None

4.127 SCIA_WR_H5_LADS

Identifier	SCIA_WR_H5_LADS
Author	R.M. van Hees
Language	ANSI C
Purpose	define and write SCIAMACHY level 1b/2 LADS data
Usage	`SCIA_WR_H5_LADS( param, nr_lads, lads );`
Input	struct param_record param : struct holding user-defined settings
	unsigned int nr_lads : number of Geolocation of States
	struct lads_scia *lads : Geolocation of States (C.10)
Returns	Nothing
Comment	None

4.128 SCIA_WR_H5_MPH

Identifier	SCIA_WR_H5_MPH
Author	R.M. van Hees
Language	ANSI C
Purpose	write the Main Product Header of a PDS SCIAMACHY file
Usage	`SCIA_WR_H5_MPH( param, mph );`
Input	struct param_record param : struct holding user-defined settings
	struct mph_envi *mph : Main Product Header data (A.5)
Returns	Nothing
Comment	None

4.129 SCIA_WR_H5_VERSION

Identifier	SCIA_WR_H5_VERSION
Author	R.M. van Hees
Language	ANSI C
Purpose	add version of the software to a HDF5 file
Usage	`SCIA_WR_H5_VERSION( hdf5_name );`
Input	const char hdf5_name[] : name of the HDF5 output file
Returns	Nothing
Comment	None

5 NADC PostgreSQL Modules

5.1 GOME_LV1_DEL_ENTRY

Identifier	GOME_LV1_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a GOME level 1 product
Usage	`GOME_LV1_DEL_ENTRY( conn, flname, softVersion )`
Input	PGconn *conn : PostgreSQL connection handle
	char *flname : filename of the GOME Level 1 product
	char *softVersion : S/W version of the GOME Level 1 product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”flname” from tables:
	meta__1P and tileinfo_meta__1P

5.2 GOME_LV1_WR_SQL_META

Identifier	GOME_LV1_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of Gome Lv1 product to table ”meta__1P”
Usage	`meta_id = GOME_LV1_WR_SQL_META( conn, gomefl, sph, fsr );`
Input	PGconn *conn : PostgreSQL connection handle
	char *gomefl : name of GOME file
	struct sph1_gome *sph : structure for SPH record (B.5)
	struct fsr1_gome *fsr : structure for FSR record (B.3)
Returns	value of serial variable “pk_meta”
	error status passed by global variable “nadc_stat”
Comment	None

5.3 GOME_LV1_WR_SQL_TILE

Identifier	GOME_LV1_WR_SQL_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of Gome Lv1 product to database
Usage	`GOME_LV1_WR_SQL_TILE( conn, meta_id, version,` `num_pcd, indx_pcd, pcd );`
Input	PGconn *conn : PostgreSQL connection handle
	int meta_id : value of primary key meta__1p.pk_meta
	char *version : software version from SPH
	short num_pcd : number of PCD records
	short *indx_pcd : indices to relevant PCD records
	struct pcd_gome *pcd : structures for PCD data (B.24)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.4 GOME_LV2_DEL_ENTRY

Identifier	GOME_LV2_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a GOME level 2 product
Usage	`GOME_LV2_DEL_ENTRY( conn, flname, softVersion );`
Input	PGconn *conn : PostgreSQL connection handle
	char *flname : filename of the GOME Level 2 product
	char *softVersion : S/W version of the GOME Level 2 product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”flname” from tables:
	meta__2P and tileinfo_meta__2P

5.5 GOME_LV2_WR_SQL_META

Identifier	GOME_LV2_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of Gome Lv2 product to table ”meta__2P”
Usage	`meta_id = GOME_LV2_WR_SQL_META( conn, gomefl, sph,fsr );`
Input	PGconn *conn : PostgreSQL connection handle
	char *gomefl : name of GOME file
	struct sph2_gome *sph : structure for SPH record (B.6)
	struct fsr2_gome *fsr : structure for FSR record (B.4)
Returns	value of serial variable “pk_meta”
	error status passed by global variable “nadc_stat”
Comment	None

5.6 GOME_LV2_WR_SQL_TILE

Identifier	GOME_LV2_WR_SQL_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of Gome Lv2 product to database
Usage	`GOME_LV2_WR_SQL_TILE( conn, meta_id, version, num_ddr, ddr );`
Input	PGconn *conn : PostgreSQL connection handle
	int meta_id : value of primary key meta__2p.pk_meta
	char *version : software version from SPH
	short num_ddr : number of DDR records
	struct ddr_gome *ddr : structures for DDR data (B.30)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.7 NADC_CONNECT_DB

Identifier	NADC_CONNECT_DB
Author	R.M. van Hees
Language	ANSI C
Purpose	connect to meta/pixel database (reading XML configuration file)
Usage	`NADC_CONNECT_DB( conn, database );`
Input	PGconn *conn : PostgreSQL connection handle
	char *database : name of database to be used
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.8 NADC_FRESCO_DEL_ENTRY

Identifier	NADC_FRESCO_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a Fresco product
Usage	`NADC_FRESCO_DEL_ENTRY( conn, prodName );`
Input	PGconn *conn : PolstgreSQL connection handle
	char *prodName : filename of the Fresco product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”prodName” from tables:
	meta_fresco, tile_fresco, tile_meta_fresco

5.9 NADC_FRESCO_WR_SQL_META

Identifier	NADC_FRESCO_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of KNMI Fresco product to table ”meta_fresco”
Usage	`metaID = NADC_FRESCO_WR_SQL_META( conn, hdr, fresco );`
Input	PGconn *conn : PostgreSQL connection handle
	struct fresco_hdr *hdr : header of Fresco product (**??**)
	struct fresco_rec *fresco : records with Fresco data (**??**)
Returns	value of the primary key of the new entry
	error status passed by global variable “nadc_stat”
Comment	None

5.10 NADC_FRESCO_WR_SQL_TILE

Identifier	NADC_FRESCO_WR_SQL_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a KNMI Fresco product to database
Usage	`NADC_FRESCO_WR_SQL_TILE( conn, prodName, num_rec, rec );`
Input	PGconn *conn : PostgreSQL connection handle
	char *prodName : filename of the Fresco product
	unsigned int num_rec : number of tiles to write
	struct fresco_rec *rec : pointer to Fresco tile records (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.11 NADC_MCFS_DEL_ENTRY

Identifier	NADC_MCFS_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a SCIA Ife MCFS product
Usage	`NADC_MCFS_DEL_ENTRY( conn, prodName );`
Input	PGconn *conn : PolstgreSQL connection handle
	char *prodName : filename of the SCIA MCFS product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”prodName” from tables:
	meta_mcfs, tile_mcfs, mcfs_tileinfo

5.12 NADC_MCFS_WR_SQL_META

Identifier	NADC_MCFS_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of SRON MCFS product to table ”meta_mcfs”
Usage	`NADC_MCFS_WR_SQL_META( conn, hdr );`
Input	PGconn *conn : PostgreSQL connection handle
	struct mcfs_hdr *hdr : header of MCFS product (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.13 NADC_MCFS_WR_SQL_TILE

Identifier	NADC_MCFS_WR_SQL_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a Sciamachy MCFS product to database
Usage	`NADC_MCFS_WR_SQL_TILE( conn, sciafl, num_rec, rec );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : filename of Sciamachy level 2 product
	unsigned short num_rec : number of tiles to write
	struct mcfs_rec *rec : pointer to MCFS records (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.14 NADC_SCIA_L1B_NAME

Identifier	NADC_SCIA_L1B_NAME
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain Sciamachy level 1b product name from database
Usage	`NADC_SCIA_L1B_NAME( absOrbit, fileCounter, l1b_product );`
Input	unsigned int absOrbit :
	unsigned short fileCounter :
Output	char *l1b_product :
Returns	Nothing
Comment	None

5.15 NADC_TOGOMI_DEL_ENTRY

Identifier	NADC_TOGOMI_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a Togomi product
Usage	`NADC_TOGOMI_DEL_ENTRY( conn, prodName );`
Input	PGconn *conn : PolstgreSQL connection handle
	char *prodName : filename of the Togomi product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”prodName” from tables:
	meta_togomi, tile_togomi, tile_meta_togomi

5.16 NADC_TOGOMI_WR_SQL_META

Identifier	NADC_TOGOMI_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of KNMI TOGOMI product to table ”meta_togomi”
Usage	`metaID = NADC_TOGOMI_WR_SQL_META( conn, hdr );`
Input	PGconn *conn : PostgreSQL connection handle
	struct togomi_hdr *hdr : header of TOGOMI product (**??**)
Returns	value of the primary key of the new entry
	error status passed by global variable “nadc_stat”
Comment	None

5.17 NADC_TOGOMI_WR_SQL_TILE

Identifier	NADC_TOGOMI_WR_SQL_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a KNMI Togomi product to database
Usage	`NADC_TOGOMI_WR_SQL_TILE( conn, prodName, num_rec, rec );`
Input	PGconn *conn : PostgreSQL connection handle
	char *prodName : filename of the Togomi product
	unsigned int num_rec : number of tiles to write
	struct togomi_rec *rec : pointer to Togomi tile records (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.18 NADC_TOSOMI_DEL_ENTRY

Identifier	NADC_TOSOMI_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a Tosomi product
Usage	`NADC_TOSOMI_DEL_ENTRY( conn, prodName );`
Input	PGconn *conn : PolstgreSQL connection handle
	char *prodName : filename of the Tosomi product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”prodName” from tables:
	meta_tosomi, tile_tosomi, tile_meta_tosomi

5.19 NADC_TOSOMI_WR_SQL_META

Identifier	NADC_TOSOMI_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of KNMI Tosomi product to table ”meta_tosomi”
Usage	`metaID = NADC_TOSOMI_WR_SQL_META( conn, hdr, tosomi );`
Input	PGconn *conn : PostgreSQL connection handle
	struct tosomi_hdr *hdr : header of Tosomi product (**??**)
	struct tosomi_rec *tosomi : records with Tosomi data (**??**)
Returns	value of the primary key of the new entry
	error status passed by global variable “nadc_stat”
Comment	None

5.20 NADC_TOSOMI_WR_SQL_TILE

Identifier	NADC_TOSOMI_WR_SQL_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a KNMI Tosomi product to database
Usage	`NADC_TOSOMI_WR_SQL_TILE( conn, prodName, num_rec, rec );`
Input	PGconn *conn : PostgreSQL connection handle
	char *prodName : filename of the Tosomi product
	unsigned int num_rec : number of tiles to write
	struct tosomi_rec *rec : pointer to Tosomi tile records (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.21 SCIA_DEL_ENTRY_IMAP_CH4

Identifier	SCIA_DEL_ENTRY_IMAP_CH4
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a SCIA SRON IMAP product
Usage	`SCIA_DEL_ENTRY_IMAP_CH4( conn, prodName );`
Input	PGconn *conn : PolstgreSQL connection handle
	char *prodName : filename of the SCIA IMAP product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”prodName” from tables:
	meta_imap, tile_imap, imap_2_tileinfo

5.22 SCIA_DEL_ENTRY_IMAP_HDO

Identifier	SCIA_DEL_ENTRY_IMAP_HDO
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a SCIA SRON IMAP product
Usage	`SCIA_DEL_ENTRY_IMAP_HDO( conn, prodName );`
Input	PGconn *conn : PolstgreSQL connection handle
	char *prodName : filename of the SCIA IMAP product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”prodName” from tables:
	meta_imap, tile_imap, imap_2_tileinfo

5.23 SCIA_DEL_ENTRY_IMLM_CO

Identifier	SCIA_DEL_ENTRY_IMLM_CO
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a SCIA SRON IMLM product
Usage	`SCIA_DEL_ENTRY_IMLM_CO( conn, prodName );`
Input	PGconn *conn : PolstgreSQL connection handle
	char *prodName : filename of the SCIA IMLM product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”prodName” from tables:
	meta_imlm_co, tile_imlm_co

5.24 SCIA_DEL_ENTRY_IMLM_H2O

Identifier	SCIA_DEL_ENTRY_IMLM_H2O
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a SCIA SRON IMLM product
Usage	`SCIA_DEL_ENTRY_IMLM_H2O( conn, prodName );`
Input	PGconn *conn : PolstgreSQL connection handle
	char *prodName : filename of the SCIA IMLM product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”prodName” from tables:
	meta_imlm_h2o, tile_imlm_h2o

5.25 SCIA_DEL_SQL_DMOP

Identifier	SCIA_DEL_SQL_DMOP
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a Sciamachy DMOP product (SOST)
Usage	`SCIA_DEL_SQL_DMOP( conn, numRec, dmop )`
Input	PGconn *conn : PostgreSQL connection handle
	unsigned int numRec : number of DMOP records
	struct dmop_rec *dmop : records with DMOP data (C.14)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries from tables: stateinfo

5.26 SCIA_LV0_DEL_ENTRY

Identifier	SCIA_LV0_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a Sciamachy level 0 product
Usage	`SCIA_LV0_DEL_ENTRY( conn, flname );`
Input	PGconn *conn : PostgreSQL connection handle
	char *flname : filename of the Sciamachy Level 0 product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”flname” from tables:
	meta__0P, stateinfo_meta__0P and reverse stateinfo

5.27 SCIA_LV0_MATCH_STATE

Identifier	SCIA_LV0_MATCH_STATE
Author	R.M. van Hees
Language	ANSI C
Purpose	write state information of Sciamachy Lv0 product to database
Usage	`SCIA_LV0_MATCH_STATE( conn, mph, numState, state );`
Input	PGconn *conn : PostgreSQL connection handle
	struct mph_envi *mph : structure for MPH record (A.5)
	unsigned short numState : number of state records
	struct mds0_sql *state : structure for STATE records (C.16)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.28 SCIA_LV0_WR_SQL_META

Identifier	SCIA_LV0_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of Sciamachy Lv0 product to table ”meta__0P”
Usage	`SCIA_LV0_WR_SQL_META( conn, sciafl, mph );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : name of Sciamachy file
	struct mph_envi *mph : tructure for MPH record (A.5)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.29 SCIA_LV1_DEL_ENTRY

Identifier	SCIA_LV1_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a Sciamachy level 1b product
Usage	`SCIA_LV1_DEL_ENTRY( conn, flname );`
Input	PGconn *conn : PostgreSQL connection handle
	char *flname : filename of the Sciamachy Level 1b product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”flname” from tables:
	meta__1P, stateinfo_meta__1P and reverse stateinfo

5.30 SCIA_LV1_MATCH_STATE

Identifier	SCIA_LV1_MATCH_STATE
Author	R.M. van Hees
Language	ANSI C
Purpose	write state information of Sciamachy Lv1b product to database
Usage	`SCIA_LV1_MATCH_STATE( conn, mph, numState, lads, sqads, state );`
Input	PGconn *conn : PostgreSQL connection handle
	struct mph_envi *mph : structure for MPH record (A.5)
	unsigned short numState : number of state records
	struct lads_scia *lads : structure for LADS records (C.10)
	struct sqads1_scia *sqads : structure for SQADS records (E.7)
	struct state1_scia *state : structure for STATE records (E.22)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.31 SCIA_LV1_WR_SQL_AUX

Identifier	SCIA_LV1_WR_SQL_AUX
Author	R.M. van Hees
Language	ANSI C
Purpose	
Usage	`SCIA_LV1_WR_SQL_AUX( conn, mph, num_dsd, dsd );`
Input	PGconn *conn : PostgreSQL connection handle
	struct mph_envi *mph : structure for MPH record (A.5)
	unsigned int num_dsd : number of DSDs
	struct dsd_envi dsd : structure for the DSDs (A.6)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Nothing

5.32 SCIA_LV1_WR_SQL_META

Identifier	SCIA_LV1_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of Sciamachy Lv1b product to table ”meta__1P”
Usage	`SCIA_LV1_WR_SQL_META( conn, sciafl, mph, sph );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : name of Sciamachy file
	struct mph_envi *mph : structure for MPH record (A.5)
	struct sph1_scia *sph: structure for SPH record (E.6)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.33 SCIA_LV1_WR_SQL_TILE

Identifier	SCIA_LV1_WR_SQL_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a Sciamachy Lv1b product to database
Usage	`SCIA_LV1_WR_SQL_TILE( conn, sciafl, intg_time, num_mds, mds );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : filename of Sciamachy level 1 product
	unsigned short intg_time : integration time (1/16 second)
	unsigned int num_mds : number of MDS records
	struct mds1_scia *mds : pointer to structure for MDS records (E.31)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.34 SCIA_OL2_DEL_ENTRY

Identifier	SCIA_OL2_DEL_ENTRY
Author	R.M. van Hees
Language	ANSI C
Purpose	remove entries in database for a Sciamachy level 2 product
Usage	`SCIA_OL2_DEL_ENTRY( conn, flname );`
Input	MYSQL *mysql : PostgreSQL connection handle
	char *flname : filename of the Sciamachy Level 2 product
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	Removes entries related to file ”flname” from tables:
	meta__2P, stateinfo_meta__2P, tileinfo_meta__2P, cld__2P, no2__2P,
	ozone__2P, and reverse stateinfo

5.35 SCIA_OL2_MATCH_STATE

Identifier	SCIA_OL2_MATCH_STATE
Author	R.M. van Hees
Language	ANSI C
Purpose	write state information of Sciamachy Lv2 product to database
Usage	`SCIA_OL2_MATCH_STATE( conn, mph, numState, state );`
Input	PQconn *conn : PostgreSQL connection handle
	struct mph_envi *mph : structure for MPH record (A.5)
	unsigned short numState : number of state records
	struct state2_scia *state : structure for STATE records (F.6)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.36 SCIA_OL2_WR_SQL_CLD

Identifier	SCIA_OL2_WR_SQL_CLD
Author	R.M. van Hees
Language	ANSI C
Purpose	write cloud information of a Sciamachy Lv2 product to database
Usage	`SCIA_OL2_WR_SQL_CLD( conn, sciafl, num_cld, cld );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : name of SCIA lv2 product
	unsigned int num_cld : number of cloud/aerosol records
	struct cld_sci_ol *cld : pointer to structure for CLD records (F.15)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.37 SCIA_OL2_WR_SQL_META

Identifier	SCIA_OL2_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of Sciamachy Lv2 offline to table ”meta__2P”
Usage	`SCIA_OL2_WR_SQL_META( conn, sciafl, l1b_product, mph, sph );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : name of Sciamachy file
	char *l1b_product name of the SCIA L1b product
	struct mph_envi *mph : structure for MPH record (A.5)
	struct sph_sci_ol *sph: structure for SPH record (F.11)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.38 SCIA_OL2_WR_SQL_NFIT

Identifier	SCIA_OL2_WR_SQL_NFIT
Author	R.M. van Hees
Language	ANSI C
Purpose	write Nadir fits of SCIA Lv2 offline product to database
Usage	`SCIA_OL2_WR_SQL_NFIT( conn, sciafl, nfit_name, num_nfit, nfit );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : name of SCIA lv2 product
	char *nfit_name : name of the Nadir fit specie
	unsigned int num_nfit : number of fitting window records
	struct nfit_scia *nfit : pointer to structure for NFIT records (F.16)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.39 SCIA_IMAP_WR_SQL_META

Identifier	SCIA_IMAP_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of SRON IMAP product to table ”meta_imap_ch4”
Usage	`SCIA_WR_SQL_CH4_META( conn, hdr );`
Input	PGconn *conn : PostgreSQL connection handle
	struct imap_hdr *hdr : header of Imap product (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.40 SCIA_WR_SQL_CH4_TILE

Identifier	SCIA_WR_SQL_CH4_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a Sciamachy IMAP product to database
Usage	`SCIA_WR_SQL_CH4_TILE( conn, sciafl, num_rec, rec );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : filename of Sciamachy level 2 product
	unsigned int num_rec : number of tiles to write
	struct imap_rec *rec : pointer to IMAP records (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.41 SCIA_WR_SQL_CO_META

Identifier	SCIA_WR_SQL_CO_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of SRON IMLM product to table ”meta_imlm_co”
Usage	`SCIA_WR_SQL_CO_META( conn, hdr );`
Input	PGconn *conn : PostgreSQL connection handle
	struct imlm_hdr *hdr : header of Imlm product (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.42 SCIA_WR_SQL_CO_TILE

Identifier	SCIA_WR_SQL_CO_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a Sciamachy IMLM product to database
Usage	`SCIA_WR_SQL_CO_TILE( conn, sciafl, num_rec, rec );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : filename of Sciamachy level 2 product
	unsigned int num_rec : number of tiles to write
	struct imlm_rec *rec : pointer to IMLM records (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.43 SCIA_WR_SQL_DMOP

Identifier	SCIA_WR_SQL_DMOP
Author	R.M. van Hees
Language	ANSI C
Purpose	write
Usage	`SCIA_WR_SQL_DMOP( conn, numRec, dmop );`
Input	PGconn *conn : PostgreSQL connection handle
	unsigned int numRec : number of DMOP records
	struct dmop_rec *dmop : records with DMOP data (C.14)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.44 SCIA_WR_SQL_H2O_META

Identifier	SCIA_WR_SQL_H2O_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of SRON IMLM product to table ”meta_imlm_h2o”
Usage	`SCIA_WR_SQL_H2O_META( conn, hdr );`
Input	PGconn *conn : PostgreSQL connection handle
	struct imlm_hdr *hdr : header of Imlm product (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.45 SCIA_WR_SQL_H2O_TILE

Identifier	SCIA_WR_SQL_H2O_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a Sciamachy IMLM product to database
Usage	`SCIA_WR_SQL_H2O_TILE( conn, sciafl, num_rec, rec );`
Input	PGconn *conn : PostgreSQL connection handle
	char *sciafl : filename of Sciamachy level 2 product
	unsigned int num_rec : number of tiles to write
	struct imlm_rec *rec : pointer to IMLM records (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.46 SCIA_IMAP_WR_SQL_META

Identifier	SCIA_IMAP_WR_SQL_META
Author	R.M. van Hees
Language	ANSI C
Purpose	write meta data of SRON IMAP product to table ”meta_imap_hdo”
Usage	`SCIA_WR_SQL_HDO_META( conn, hdr );`
Input	PGconn *conn : PostgreSQL connection handle
	struct imap_hdr *hdr : header of Imap product (**??**)
Returns	Nothing, error status passed by global variable “nadc_stat”
Comment	None

5.47 SCIA_WR_SQL_HDO_TILE

Identifier	SCIA_WR_SQL_HDO_TILE
Author	R.M. van Hees
Language	ANSI C
Purpose	write tile information of a Sciamachy IMAP product to database
Usage	`SCIA_WR_SQL_HDO_TILE( conn, sciafl, num_rec, rec );`
Input	PGconn *conn : PostgreSQL connection handle char *sciafl : filename of Sciamachy level 2 product unsigned int num_rec : number of tiles to write struct imap_rec *rec : pointer to IMAP records (**??**)
Returns	Nothing, error status passed by global variable "nadc_stat"
Comment	None

6 NADC SDMF Modules

6.1 SDMF_ARRAY

Identifier	SDMF_ARRAY
Author	R.M. van Hees
Language	ANSI C
Purpose	read/write statistics of State executions
Comment	contains SDMF_rd_uint8_Array, SDMF_rd_uint8_Matrix, SDMF_rd_int16_Array, SDMF_rd_int16_Matrix, SDMF_rd_uint16_Array, SDMF_rd_uint16_Matrix, SDMF_rd_float_Array, SDMF_rd_float_Matrix

6.1.1 SDMF_rd_string_Array

Identifier	SDMF_rd_string_Array
Purpose	read element from dataset ”arrName”
Usage	`SDMF_rd_string_Array( locID, arrName, Index, data );`
Input	hid_t locID : HDF5 identifier of file or group char arrName[] : name of dataset int Index : number of index to read
Output	unsigned char *data : string space for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.1.2 SDMF_rd_uint8_Array

Identifier	SDMF_rd_uint8_Array
Purpose	read row(s) from dataset ”arrName”
Usage	`SDMF_rd_uint8_Array( locID, arrName, numIndex, metaIndex,` `pixelRange, data );`
Input	hid_t locID : HDF5 identifier of file or group char arrName[] : name of dataset int numIndex : number of indices to read int *metaIndex : array with indices of row to read int *pixelRange : two element array containing the pixelsIDs of the first and last pixel to be read OR NULL, in case all pixels are read
Output	unsigned char *data : array for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.1.3 SDMF_rd_uint8_Matrix

Identifier	SDMF_rd_uint8_Matrix
Purpose	read row(s) from dataset ”arrName”
Usage	`SDMF_rd_uint8_Matrix( locID, arrName, Index, slabsize, dims, data );`
Input	hid_t locID : HDF5 identifier of file or group
	char arrName[] : name of dataset
	int Index : number of index to read
	int slabsize : array of elements to read in all dimensions
	int dims : number of dimensions of dataset
Output	unsigned char *data : matrix for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.1.4 SDMF_rd_int16_Array

Identifier	SDMF_rd_int16_Array
Purpose	read row(s) from dataset ”arrName”
Usage	`SDMF_rd_int16_Array( locID, arrName, numIndex, metaIndex, pixelRange, data );`
Input	hid_t locID : HDF5 identifier of file or group
	char arrName[] : name of dataset
	int numIndex : number of indices to read
	int *metaIndex : array with indices of row to read
	int *pixelRange : two element array containing the pixelsIDs
	of the first and last pixel to be read
	OR NULL, in case all pixels are read
Output	short *data : array for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.1.5 SDMF_rd_int16_Matrix

Identifier	SDMF_rd_int16_Matrix
Purpose	read slabs from dataset ”arrName”
Usage	`SDMF_rd_int16_Matrix( locID, arrName, metaIndex, slabsize, dims, data );`
Input	hid_t locID : HDF5 identifier of file or group
	char arrName[] : name of dataset
	int *metaIndex : index of orbit
	int *slabsize : size of slab to read
	int dims : dimensions of dataset, only tested for 3D
Output	short *data : array for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.1.6 SDMF_rd_uint16_Array

Identifier	SDMF_rd_uint16_Array
Purpose	read row(s) from dataset ”arrName”
Usage	`SDMF_rd_uint16_Array( locID, arrName, numIndex, metaIndex, pixelRange, data );`
Input	hid_t locID : HDF5 identifier of file or group
	char arrName[] : name of dataset
	int numIndex : number of indices to read
	int *metaIndex : array with indices of row to read
	int *pixelRange : two element array containing the pixelsIDs
	of the first and last pixel to be read
	OR NULL, in case all pixels are read
Output	unsigned short *data : array for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.1.7 SDMF_rd_uint16_Matrix

Identifier	SDMF_rd_uint16_Matrix
Purpose	read slabs from dataset ”arrName”
Usage	`SDMF_rd_uint16_Matrix( locID, arrName, metaIndex, slabsize, dims, data );`
Input	hid_t locID : HDF5 identifier of file or group
	char arrName[] : name of dataset
	int *metaIndex : index of orbit
	int *slabsize : size of slab to read
	int dims : dimensions of dataset, only tested for 3D
Output	uint16 *data : array for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.1.8 SDMF_rd_float_Array

Identifier	SDMF_rd_float_Array
Purpose	read row(s) from dataset ”arrName”
Usage	`SDMF_rd_float_Array( locID, arrName, numIndex, metaIndex, pixelRange, data );`
Input	hid_t locID : HDF5 identifier of file or group
	char arrName[] : name of dataset
	int numIndex : number of indices to read
	int *metaIndex : array with indices of row to read
	int *pixelRange : two element array containing the pixelsIDs
	of the first and last pixel to be read
	OR NULL, in case all pixels are read
Output	float *data : array for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.1.9 SDMF_rd_float_Matrix

Identifier	SDMF_rd_float_Matrix
Purpose	read slabs from dataset ”arrName”
Usage	`SDMF_rd_float_Matrix( locID, arrName, metaIndex, slabsize, dims, data );`
Input	hid_t locID : HDF5 identifier of file or group
	char arrName[] : name of dataset
	int *metaIndex : index of orbit
	int *slabsize : size of slab to read
	int dims : dimensions of dataset, only tested for 3D
Output	float *data : array for data to read
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.2 SDMF_get_clusConf

Identifier	SDMF_get_clusConf
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain cluster configuration parameters
Comment	contains SDMF_get_clusParam, SDMF_get_statePET, SDMF_get_stateCoadd, SDMF_get_stateCount SDMF_PET2StateID

6.2.1 SDMF_get_stateParam

Identifier	SDMF_get_stateParam
Purpose	obtain number of readouts for given orbit
Usage	`SDMF_get_stateParam( stateID, orbit, channel, &ipet, orbit_range );`
Input	unsigned char stateID : state ID [1..70]
	unsigned short orbit : orbit number
	unsigned short channel : channel ID [1..8]
Output	unsigned short *int_pet : pixel exposure time * 32
	int *orbit_range : orbit range with same state definition as orbit
Returns	nothing
Comment	none

6.2.2 SDMF_get_statePET

Identifier	SDMF_get_statePET
Purpose	obtain PET for given state, orbit and channel
Usage	`SDMF_get_statePET( stateID, orbit, channel, pet );`
Input	unsigned char stateID : state ID
	unsigned short orbit : orbit number
	unsigned short channel : channel ID or zero for all channels
Output	float : pixel exposure time (s)
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.2.3 SDMF_get_stateCoadd

Identifier	SDMF_get_stateCoadd
Purpose	obtain co-add factor for given state, orbit and cluster
Usage	`coaddf = SDMF_get_stateCoadd( stateID, orbit, clusID );`
Input	unsigned char stateID : state ID
	unsigned short orbit : orbit number
	unsigned short clusID : cluster ID
Returns	co-adding factor (unsigned char),
	error status passed by global variable “nadc_stat”
Comment	none

6.2.4 SDMF_get_stateCount

Identifier	SDMF_get_stateCount
Purpose	obtain number of complete readouts for given state and orbit
Usage	`count = SDMF_get_stateCount( stateID, orbit );`
Input	unsigned char stateID : state ID
	unsigned short orbit : orbit number
Returns	number of complete readouts (unsigned short),
	error status passed by global variable “nadc_stat”
Comment	none

6.2.5 SDMF_PET2StateID

Identifier	SDMF_PET2StateID
Purpose	obtain state ID for given orbit, channel and PET
Usage	`stateID = SDMF_PET2StateID( orbit, channel, pet );`
Input	unsigned short orbit : orbit number
	unsigned short channel : channel ID [1..8]
	float pet : pixel exposure time (s)
Returns	state ID (unsigned char), or UCHAR_MAX on failure
Comment	none

6.3 SDMF_DARK

Identifier	SDMF_DARK
Author	R.M. van Hees
Language	ANSI C
Comment	contains SDMF_rd_darkTable

6.3.1 SDMF_rd_darkTable

Identifier	SDMF_rd_darkTable
Purpose	read darkTable records from SDMF dark-current database
Usage	`SDMF_rd_darkTable( locID, &numIndx, metaIndx, &mtbl );`
Input	hid_t locID : HDF5 identifier of file or group
	int *metaIndx : array with indices to requested records
In/Output	int *numIndx : input: dimension metaIndx (or zero)
Output	output:
	struct mtbl_dark_rec **mtbl : Dark meta-data records to read (G.9)
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	none

6.4 SDMF_get_BDPM

Identifier	SDMF_get_BDPM
Author	R.M. van Hees
Language	ANSI C
Purpose	read Dead/Bad pixels mask from SDMF databases
Comment	contains SDMF_get_BDPM_24 and SDMF_get_BDPM_30

6.4.1 SDMF_get_BDPM_24

Identifier	SDMF_get_BDPM_24
Purpose	obtain bad/dead pixel mask from SRON Monitoring database (v2.4)
Usage	`SDMF_get_BDPM_24( orbit, bdpm );`
Input	unsigned short absOrbit : absolute orbitnumber
Output	unsigned char *bdpm : bad/dead pixel mask
Returns	flag: FALSE (no mask found) or TRUE error status passed by global variable “nadc_stat”
Comment	none

6.4.2 SDMF_get_BDPM_30

Identifier	SDMF_get_BDPM_30
Purpose	obtain bad/dead pixel mask from SRON Monitoring database (v3.0)
Usage	`SDMF_get_BDPM_30( absOrbit, bdpm );`
Input	unsigned short absOrbit : absolute orbitnumber
Output	unsigned char *bdpm : bad/dead pixel mask
Returns	flag: FALSE (no mask found) or TRUE error status passed by global variable “nadc_stat”
Comment	none

6.5 SDMF_get_fileEntry

Identifier	SDMF_get_fileEntry
Author	R.M. van Hees
Language	ANSI C
Purpose	select SDMF 2.4 entry
Usage	`found = SDMF_get_fileEntry( sdmfDB, orbit, fileEntry );`
Input	enum sdmf24_db sdmfDB : select type of calibration int orbit : requested orbit number
Output	char *fileEntry : name of the selected file
Returns	boolean value: TRUE when entry is found
Comment	SDMF24_STATE mimics data averaging of Hans Schrijver

6.6 SDMF_get_FittedDark

Identifier	SDMF_get_FittedDark
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain (fitted) dark correction parameters
Comment	contains SDMF_get_FittedDark, SDMF_get_FittedDark_30 SDMF_get_FittedDark_24

6.6.1 SDMF_get_FittedDark_24

Identifier	SDMF_get_FittedDark_24
Purpose	obtain dark correction parameters (SDMF v2.4.1)
Usage	`found = SDMF_get_FittedDark_24( channel, orbit, analogOffs, darkCurrent, analogOffsError, darkCurrentError, chiSquareFit );`
Input	unsigned short channel : channel ID or zero for all channels
	unsigned short absOrbit : orbit number
In/Output	float *analogOffs : analog offset (BU)
	float *darkCurrent : leakage current (BU/s)
	float *analogOffsError : analog offset error (or NULL)
	float *darkCurrentError : leakage current error (or NULL)
	float *chiSquareFit : chiSquare of fit (or NULL)
Returns	solution found (True or False)
	error status passed by global variable "nadc_stat"
Comment	none

6.6.2 SDMF_get_FittedDark_30

Identifier	SDMF_get_FittedDark_30
Purpose	obtain dark correction parameters (SDMF v3.0)
Usage	`found = SDMF_get_FittedDark_30( channel, orbit, analogOffs, darkCurrent, analogOffsError, darkCurrentError, chiSquareFit );`
Input	unsigned short channel : channel ID or zero for all channels
	unsigned short absOrbit : orbit number
In/Output	float *analogOffs : analog offset (BU)
	float *darkCurrent : leakage current (BU/s)
	float *analogOffsError : analog offset error (or NULL)
	float *darkCurrentError : leakage current error (or NULL)
	float *chiSquareFit : chiSquare of fit (or NULL)
Returns	solution found (True or False)
	error status passed by global variable "nadc_stat"
Comment	none

6.6.3 SDMF_get_FittedDark

Identifier	SDMF_get_FittedDark
Purpose	obtain dark correction parameters (SDMF v3.1)
Usage	`found = SDMF_get_FittedDark( channel, orbit, analogOffs, darkCurrent, analogOffsError, darkCurrentError, meanNoise, chiSquareFit, probabilityFit );`
Input	unsigned short channel : channel ID or zero for all channels
	unsigned short absOrbit : orbit number
In/Output	float *analogOffs : analog offset (BU)
	float *darkCurrent : leakage current (BU/s)
	float *analogOffsError : analog offset error (or NULL)
	float *darkCurrentError : leakage current error (or NULL)
	float *meanNoise : mean noise (or NULL)
	float *chiSquareFit : chiSquare of fit (or NULL)
	float *probabilityFit : probability of fit (or NULL)
Returns	solution found (True or False)
	error status passed by global variable “nadc_stat”
Comment	none

6.7 SDMF_get_OrbitalDark

Identifier	SDMF_get_OrbitalDark
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain orbital dark correction parameters, channel 8
Comment	contains SDMF_get_OrbitalDark, SDMF_get_OrbitalDark_30
	SDMF_get_OrbitalDark_24

6.7.1 SDMF_get_OrbitalDark_24

Identifier	SDMF_get_OrbitalDark_24
Purpose	obtain dark correction parameters (SDMF v2.4.x)
Usage	`found = SDMF_get_OrbitalDark_24( orbit, orbitPhase analogOffs, darkCurrent, analogOffsError, darkCurrentError );`
Input	unsigned short absOrbit : orbit number
	float orbitPhase : orbit phase (ESA definition)
In/Output	float *analogOffs : analog offset (BU)
	float *darkCurrent : leakage current (BU/s)
	float *analogOffsError : analog offset error (BU)
	float *darkCurrentError : leakage current error (BU/s)
Returns	solution found (True or False)
	error status passed by global variable “nadc_stat”
Comment	none

6.7.2 SDMF_get_OrbitalDark_30

Identifier	SDMF_get_OrbitalDark_30
Purpose	obtain dark correction parameters (SDMF v3.0)
Usage	`found = SDMF_get_OrbitalDark_30( orbit, orbitPhase analogOffs, darkCurrent, analogOffsError, darkCurrentError );`
Input	unsigned short absOrbit : orbit number
	float orbitPhase : orbit phase
In/Output	float *analogOffs : analog offset (BU)
	float *darkCurrent : leakage current (BU/s)
	float *analogOffsError : analog offset error (BU)
	float *darkCurrentError : leakage current error (BU/s)
Returns	solution found (True or False)
	error status passed by global variable "nadc_stat"
Comment	none

6.7.3 SDMF_get_OrbitalDark

Identifier	SDMF_get_OrbitalDark
Purpose	obtain dark correction parameters (SDMF v3.1)
Usage	`found = SDMF_get_OrbitalDark( orbit, orbitPhase analogOffs, darkCurrent, analogOffsError, darkCurrentError );`
Input	unsigned short absOrbit : orbit number
	float orbitPhase : orbit phase
In/Output	float *analogOffs : analog offset (BU)
	float *darkCurrent : leakage current (BU/s)
	float *analogOffsError : analog offset error (BU)
	float *darkCurrentError : leakage current error (BU/s)
Returns	solution found (True or False)
	error status passed by global variable "nadc_stat"
Comment	none

6.8 SDMF_get_PPG

Identifier	SDMF_get_PPG
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain PPG parameters
Comment	contains SDMF_get_PPG, SDMF_get_PPG_30

6.8.1 SDMF_get_PPG_24

Identifier	SDMF_get_PPG_24
Purpose	Read Pixel-to-Pixel Gain factors from Monitoring database (v2.4)
Usage	`SDMF_get_PPG_24( absOrbit, channel, pixelGain );`
Input	unsigned short absOrbit : absolute orbitnumber
	unsigned short channel : channel ID or zero for all channels
Output	float *pixelGain : Pixel-to-Pixel Gain factors
Returns	flag: FALSE (no mask found) or TRUE
	error status passed by global variable "nadc_stat"
Comment	none

6.8.2 SDMF_get_PPG_30

Identifier	SDMF_get_PPG_30
Purpose	obtain Pixel-to-Pixel Gain factors from SRON Monitoring database
Usage	`SDMF_get_PPG_30( absOrbit, channel, pixelGain );`
Input	unsigned short absOrbit : absolute orbitnumber
	unsigned short channel : channel ID or zero for all channels
Output	float *pixelGain : Pixel-to-Pixel Gain factors
Returns	flag: FALSE (no PPG found) or TRUE
	error status passed by global variable “nadc_stat”
Comment	none

6.9 SDMF_get_StateDark

Identifier	SDMF_get_StateDark
Author	R.M. van Hees
Language	ANSI C
Purpose	obtain (state) dark correction parameters
Comment	contains SDMF_get_StateDark, SDMF_get_StateDark_30
	SDMF_get_StateDark_24

6.9.1 SDMF_get_StateDark_24

Identifier	SDMF_get_StateDark_24
Purpose	obtain dark correction parameters (SDMF v2.4.1)
Usage	`found = SDMF_get_StateDark_24( stateID, channel, absOrbit, pet, darkSignal, darkNoise );`
Input	unsigned char stateID : state ID
	unsigned short channel : channel ID or zero for all channels
	unsigned short absOrbit : orbit number
Output	float *pet : channel Pixel Exposure Time (sec)
	float *darkSignal : dark signal (BU, memory/non-linearity)
	float *darkNoise : dark noise (BU, standard deviation)
Returns	solution found (True or False)
	error status passed by global variable “nadc_stat”
Comment	none

6.9.2 SDMF_get_StateDark_30

Identifier	SDMF_get_StateDark_30
Purpose	obtain dark correction parameters (SDMF v3.0)
Usage	`found = SDMF_get_StateDark_30( stateID, channel, absOrbit, pet, darkSignal, darkNoise );`
Input	unsigned char stateID : state ID
	unsigned short channel : channel ID or zero for all channels
	unsigned short absOrbit : orbit number
Output	float *pet : channel Pixel Exposure Time (sec)
	float *darkSignal : dark signal (BU, memory/non-linearity)
	float *darkNoise : dark noise (BU, standard deviation)
Returns	solution found (True or False)
	error status passed by global variable “nadc_stat”
Comment	none

6.9.3 SDMF_get_StateDark

Identifier	SDMF_get_StateDark
Purpose	obtain dark correction parameters (SDMF v3.1)
Usage	`found = SDMF_get_StateDark( stateID, channel, absOrbit, pet, darkSignal, darkNoise );`
Input	unsigned char stateID : state ID
	unsigned short absOrbit : orbit number
	unsigned short channel : channel ID or zero for all channels
Output	float *pet : channel Pixel Exposure Time (sec)
	float *darkSignal : dark signal (BU, memory/non-linearity)
	float *darkNoise : dark noise (BU, absolute mean deviation)
Returns	solution found (True or False)
	error status passed by global variable “nadc_stat”
Comment	none

6.10 SDMF_TRANSMISSION

Identifier	SDMF_TRANSMISSION
Author	R.M. van Hees
Language	ANSI C
Purpose	read transmission based on Sun or WLS measurements
Comment	contains SDMF_get_Transmission_24 and SDMF_get_Transmission_30

6.10.1 SDMF_get_Transmission_24

Identifier	SDMF_get_Transmission_24
Purpose	obtain transmission from SRON Monitoring database (v2.4)
Usage	`SDMF_get_Transmission_24( wls, absOrbit, channel, transmission );`
Input	bool wls : transmission based on WLS measurements
	unsigned short absOrbit : absolute orbitnumber
	unsigned short channel : channel ID or zero for all channels
Output	float *transmission : transmission factors
Returns	flag: FALSE (no mask found) or TRUE
	error status passed by global variable “nadc_stat”
Comment	none

6.10.2 SDMF_get_Transmission_30

Identifier	SDMF_get_Transmission_30
Purpose	obtain transmission from SRON Monitoring database (v3.0)
Usage	`SDMF_get_Transmission_30( wls, absOrbit, channel, transmission );`
Input	bool wlsFlag : transmission based on WLS measurements
	unsigned short absOrbit : absolute orbitnumber
	unsigned short channel : channel ID or zero for all channels
Output	float *transmission : transmission factors
Returns	flag: FALSE (no mask found) or TRUE
	error status passed by global variable “nadc_stat”
Comment	none

6.11 SDMF_HIST

Identifier	SDMF_HIST
Author	R.M. van Hees
Language	ANSI C
Purpose	read histogram from readouts of one state execution
Comment	contains: SDMF30_rd_histTable, SDMF31_rd_histTable

6.11.1 SDMF30_rd_histTable

Identifier	SDMF30_rd_histTable
Purpose	read histogram data
Usage	`SDMF30_rd_histTable( locID, numIndex, metaIndex, pixelRange, sdmf_hist );`
Input	hid_t locID : HDF5 identifier of file or group int numIndex : number of indices to read int *metaIndex : array with indices of row to read int *pixelRange : two element array containing the pixelsIDs of the first and last pixel to be read OR NULL, in case all pixels are read
Output	struct sdmf_hist1_rec *sdmf_hist : histogram-data records (G.5)
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	allocation requirements for sdmf_hist is (pixelRange != NULL) sz_sdmf_hist * indexNum * (pixelRange[1]-pixelRange[0]+1) (pixelRange == NULL) sz_sdmf_hist * indexNum * SCIENCE_PIXELS

6.11.2 SDMF30_rd_histTable

Identifier	SDMF30_rd_histTable
Purpose	read histogram data
Usage	`SDMF31_rd_histTable( locID, numIndex, metaIndex, pixelRange, sdmf_hist );`
Input	hid_t locID : HDF5 identifier of file or group int numIndex : number of indices to read int *metaIndex : array with indices of row to read int *pixelRange : two element array containing the pixelsIDs of the first and last pixel to be read OR NULL, in case all pixels are read
Output	struct sdmf_hist2_rec *sdmf_hist : histogram-data records (G.6)
Returns	nothing, error status passed by global variable “nadc_stat”
Comment	allocation requirements for sdmf_hist is (pixelRange != NULL) sz_sdmf_hist * indexNum * (pixelRange[1]-pixelRange[0]+1) (pixelRange == NULL) sz_sdmf_hist * indexNum * SCIENCE_PIXELS

6.12 SDMF_META

Identifier	SDMF_META
Author	R.M. van Hees
Language	ANSI C
Comment	contains SDMF_get_metaIndex, SDMF_get_metaIndex_range, SDMF_rd_metaTable

6.12.1 SDMF_get_metaIndex

Identifier	SDMF_get_metaIndex
Purpose	find index to entry of stateID and absOrbit
Usage	`row = SDMF_get_metaIndex( locID, absOrbit, &numIndx, metaIndx );`
Input	hid_t fid : HDF5 identifier of file or group
	int absOrbit : orbit number
In/Output	int *numIndx : input: dimension metaIndx
Output	output:
	int *metaIndx : array with indices to requested orbit
Returns	Index to element of orbitIndx to be updated (required for write)
	error status passed by global variable “nadc_stat”
Comment	none

6.12.2 SDMF_get_metaIndex_range

Identifier	SDMF_get_metaIndex_range
Purpose	find index to entry of stateID and absOrbit
Usage	`row = SDMF_get_metaIndex_range( locID, orbit_range, numIndx, metaIndx, use_neighbours );`
Input	hid_t fid : HDF5 identifier of file or group
	int *orbit_range : orbit range [lo, hi]
	int use_neighbours : when no entries in range, use their neighbours
In/Output	int *numIndx : input: dimension metaIndx (2nd pass only!)
Output	output:
	int *metaIndx : array with indices to requested orbit
	if NULL: no indices stored!
Returns	Index to element of orbitIndx to be updated (required for write)
	error status passed by global variable “nadc_stat”
Comment	May also be used as a two-pass function:
	first pass: get nr of indices found..
	row = SDMF_get_metaIndex_range(locID, orbit_range, &numIndx, NULL, use_neighbours);
	second pass: get actual data..
	row = SDMF_get_metaIndex_range(locID, orbit_range, &numIndx, metaIndx, use_neighbours);

6.12.3 SDMF_rd_metaTable

Identifier	SDMF_rd_metaTable
Purpose	read metaTable records from SDMF calibration state database
Usage	`SDMF_rd_metaTable( locID, &numIndx, metaIndx, &mtbl );`
Input	hid_t locID : HDF5 identifier of file or group
	int *metaIndx : array with indices to requested records
In/Output	int *numIndx : input: dimension metaIndx (or zero)
Output	output:
	struct mtbl_calib_rec **mtbl : State meta-data records to read (G.7)
Returns	nothing, error status passed by global variable "nadc_stat"
Comment	Note that *numIndx has to be zero and/or *metaIndx equal to NULL
	to read all metaTable records

6.13 SDMF_DBPM

Identifier	SDMF_DBPM
Author	R.M. van Hees
Language	ANSI C
Purpose	read Dead/Bad pixels mask from SDMF databases
Comment	contains SDMF_get_BDPM_24 and SDMF_get_BDPM_30

6.14 SDMF_PT_DB

Identifier	SDMF_PT_DB
Author	R.M. van Hees
Language	ANSI C
Purpose	reoutines to access the SDMF packet table databases
Comment	contains: SDMF_get_pt_orbitIndex, SDMF_get_pt_jdayIndex,
	SDMF_rd_pt_metaTable, SDMF_rd_pt_pointing,
	SDMF_rd_pt_cluster
	The modules SDMF_fill_ll_msd1c and SDMF_fill_sun_msd1c are rather
	high level read-routines, without calibration. To obtain
	calibrated values:
	num_mds = SDMF_fill_sun_msd1c(fid, absOrbit, state_id, clus_id, &mds);
	(void) strcpy(calib_str, "0+,1+,2+,5+,7k,9+");
	calib_mask = NADC_SCIA_CalibMask(calib_str);
	SCIA_L1C_CAL(calib_mask, num_mds, mds_1c);

6.15 SDMF_SIMUDARK

Identifier	SDMF_SIMUDARK
Author	P. van der Meer
Language	ANSI C
Comment	contains SDMF_rd_simudarkTable

6.15.1 SDMF_rd_simudarkTable

Identifier	SDMF_rd_simudarkTable
Purpose	read metaTable records from SDMF simultaneous dark signal parameter database
Usage	`SDMF_rd_simudarkTable( locID, &numIndx, metaIndx, &mtbl );`
Input	hid_t locID : HDF5 identifier of file or group
	int *metaIndx : array with indices to requested records
In/Output	int *numIndx : input: dimension metaIndx (or zero)
Output	output:
	struct mtbl_simudark_rec **mtbl : State meta-data records to read (G.11)
Returns	nothing, error status passed by global variable "nadc_stat"
Comment	none

A General Compound Data Types

A.1 struct param_record

Attribute	Data Type	Vector Length
flag_infile	unsigned char	1
flag_outfile	unsigned char	1
flag_check	unsigned char	1
flag_show	unsigned char	1
flag_version	unsigned char	1
flag_silent	unsigned char	1
flag_verbose	unsigned char	1
flag_cloud	unsigned char	1
flag_geoloc	unsigned char	1
flag_geomnmx	unsigned char	1
flag_period	unsigned char	1
flag_pselect	unsigned char	1
flag_subset	unsigned char	1
flag_sunz	unsigned char	1
flag_wave	unsigned char	1
use_infodb	unsigned char	1
write_pds	unsigned char	1
write_ascii	unsigned char	1
write_hdf5	unsigned char	1
flag_deflate	unsigned char	1
write_meta	unsigned char	1
write_sql	unsigned char	1
flag_sql_remove	unsigned char	1
flag_sql_replace	unsigned char	1
write_lv1c	unsigned char	1
write_subset	unsigned char	1
write_blind	unsigned char	1
write_stray	unsigned char	1
write_aux0	unsigned char	1
write_pmd0	unsigned char	1
write_aux	unsigned char	1
write_det	unsigned char	1
write_pmd	unsigned char	1
write_pmd_geo	unsigned char	1
write_polV	unsigned char	1
write_ads	unsigned char	1
write_gads	unsigned char	1
write_limb	unsigned char	1
write_moni	unsigned char	1
write_moon	unsigned char	1
write_nadir	unsigned char	1
write_occ	unsigned char	1
write_sun	unsigned char	1
write_bias	unsigned char	1
write_cld	unsigned char	1
write_doas	unsigned char	1
catID_nr	unsigned char	1
stateID_nr	unsigned char	1

Attribute	Data Type	Vector Length
clusID_nr	unsigned char	1
chan_mask	unsigned char	1
patch_scia	unsigned short	1
calib_earth	unsigned short	1
calib_limb	unsigned short	1
calib_moon	unsigned short	1
calib_sun	unsigned short	1
calib_pmd	unsigned short	1
calib_scia	unsigned int	1
hdf_file_id	int	1
hdf_sd_id	int	1
clus_mask	unsigned long long	1
catID	unsigned char	MAX_NUM_STATE
stateID	unsigned char	MAX_NUM_STATE
bgn_date	char	GOME_DATE_LENGTH
end_date	char	GOME_DATE_LENGTH
pselect	char	MAX_STRING_LENGTH
program	char	MAX_STRING_LENGTH
infile	char	MAX_STRING_LENGTH
outfile	char	MAX_STRING_LENGTH
hdf5_name	char	MAX_STRING_LENGTH
cloud	float	2
geo_lat	float	2
geo_lon	float	2
sunz	float	2
wave	float	2

A.2 struct param_adaguc

Attribute	Data Type	Vector Length
flag_show	unsigned char	1
flag_version	unsigned char	1
flag_silent	unsigned char	1
flag_verbose	unsigned char	1
flag_indir	unsigned char	1
flag_outdir	unsigned char	1
flag_clip	unsigned char	1
prodClass	char	5
clipStart	char	16
clipStop	char	16
num_infiles	unsigned short	1
name_infiles	char	MAX_ADAGUC_INFILES
indir	char	MAX_STRING_LENGTH
outdir	char	MAX_STRING_LENGTH

A.3 struct mjd_envi

Attribute	Data Type	Vector Length
days	int	1
secnd	unsigned int	1
musec	unsigned int	1

A.4 struct coord_envi

Attribute	Data Type	Vector Length
lat	int	1
lon	int	1

A.5 struct mph_envi

Attribute	Data Type	Vector Length
product	char	ENVI_FILENAME_SIZE
proc_stage	char	2
ref_doc	char	24
acquis	char	21
proc_center	char	7
proc_time	char	MAX_UTC_STRING
soft_version	char	15
sensing_start	char	MAX_UTC_STRING
sensing_stop	char	MAX_UTC_STRING
phase	char	2
cycle	short	1
rel_orbit	int	1
abs_orbit	int	1
state_vector	char	MAX_UTC_STRING
delta_ut	double	1
x_position	double	1
y_position	double	1
z_position	double	1
x_velocity	double	1
y_velocity	double	1
z_velocity	double	1
vector_source	char	3
utc_sbt_time	char	MAX_UTC_STRING
sat_binary_time	unsigned int	1
clock_step	unsigned int	1
leap_utc	char	MAX_UTC_STRING
leap_sign	short	1
leap_err	char	2
product_err	char	2
tot_size	unsigned int	1
sph_size	unsigned int	1
num_dsd	unsigned int	1
dsd_size	unsigned int	1
num_data_sets	unsigned int	1

A.6 struct dsd_envi

Attribute	Data Type	Vector Length
name	char	29
type	char	2
flname	char	ENVI_FILENAME_SIZE
offset	unsigned int	1

Attribute	Data Type	Vector Length
size	unsigned int	1
num_dsr	unsigned int	1
dsr_size	int	1

A.7 struct state_list_rec

Attribute	Data Type	Vector Length
stateID	unsigned char	1
intg_time	unsigned char	1
numObs	unsigned short	1

B GOME Compound Data Types

B.1 struct mjd_gome

Attribute	Data Type	Vector Length
days	unsigned int	1
msec	unsigned int	1

B.2 struct pir_gome

Attribute	Data Type	Vector Length
mission	char	3
sensor	char	4
orbit	char	6
nr_orbit	char	5
acquis	char	3
product	char	6
blank	char	2
proc_id	char	3
proc_date	char	9
proc_time	char	7

B.3 struct fsr1_gome

Attribute	Data Type	Vector Length
nr_sph	short	1
sz_sph	int	1
nr_fcd	short	1
sz_fcd	int	1
nr_pcd	short	1
sz_pcd	int	1
nr_scd	short	1
sz_scd	int	1
nr_mcd	short	1
sz_mcd	int	1
nr_band	short	NUM_SPEC_BANDS

Attribute	Data Type	Vector Length
sz_band	int	NUM_SPEC_BANDS

B.4 struct fsr2_gome

Attribute	Data Type	Vector Length
nr_sph	short	1
sz_sph	int	1
nr_ddr	short	1
sz_ddr	int	1

B.5 struct sph1_gome

Attribute	Data Type	Vector Length
nr_inref	short	1
inref	char	2
soft_version	char	6
calib_version	char	6
prod_version	short	1
time_orbit	unsigned int	1
time_utc_day	unsigned int	1
time_utc_ms	unsigned int	1
time_counter	unsigned int	1
time_period	unsigned int	1
pmd_entry	short	1
subset_entry	short	1
intgstat_entry	short	1
peltier_entry	short	1
status2_entry	short	1
PMD_NUMBER]	float pmd_conv[2 *	1
state_utc_day	unsigned int	1
state_utc_ms	unsigned int	1
state_orbit	unsigned int	1
state_x	float	1
state_y	float	1
state_z	float	1
state_dx	float	1
state_dy	float	1
state_dz	float	1
att_yaw	double	1
att_pitch	double	1
att_roll	double	1
att_dyaw	double	1
att_dpitch	double	1
att_droll	double	1
att_flag	int	1
att_stat	int	1
julian	double	1
semi_major	double	1
excen	double	1
incl	double	1

Attribute	Data Type	Vector Length
right_asc	double	1
perigee	double	1
mn_anom	double	1
start_time	struct mjd_gome (B.1)	1
stop_time	struct mjd_gome (B.1)	1

B.6 struct sph2_gome

Attribute	Data Type	Vector Length
inref	char	39
soft_version	char	6
param_version	char	6
format_version	char	6
nwin	short	1
nmol	short	1
height	float	1
window	float	LVL2_MAX_NWIN
mol_win	short	LVL2_MAX_NMOL
mol_name	char	LVL2_MAX_NMOL
start_time	struct mjd_gome (B.1)	1
stop_time	struct mjd_gome (B.1)	1

B.7 struct glr1_gome

Attribute	Data Type	Vector Length
sun_glint	char	1
subsetcounter	unsigned char	1
utc_date	unsigned int	1
utc_time	unsigned int	1
sat_geo_height	float	1
earth_radius	float	1
sun_zen_sat_north	float	3
sun_azim_sat_north	float	3
los_zen_sat_north	float	3
los_azim_sat_north	float	3
sun_zen_sat_ers	float	3
sun_azim_sat_ers	float	3
los_zen_sat_ers	float	3
los_azim_sat_ers	float	3
sun_zen_surf_north	float	3
sun_azim_surf_north	float	3
los_zen_surf_north	float	3
los_azim_surf_north	float	3
lon	float	NUM_COORDS
lat	float	NUM_COORDS

B.8 struct glr2_gome

Attribute	Data Type	Vector Length
pixel_nr	int	1
subsetcounter	int	1
utc_date	unsigned int	1
utc_time	unsigned int	1
sat_geo_height	float	1
earth_radius	float	1
sat_zenith	float	3
sat_sight	float	3
sat_azim	float	3
toa_zenith	float	3
toa_sight	float	3
toa_azim	float	3
lon	float	NUM_COORDS
lat	float	NUM_COORDS

B.9 struct cr1_gome

Attribute	Data Type	Vector Length
mode	short	1
type	short	1
surfaceHeight	float	1
fraction	float	1
fractionError	float	1
albedo	float	1
albedoError	float	1
height	float	1
heightError	float	1
thickness	float	1
thicknessError	float	1
topPress	float	1
topPressError	float	1

B.10 union quality_fcd

Attribute	Data Type	Vector Length
flag_fields.array_1:2	unsigned int	1
flag_fields.array_2:2	unsigned int	1
flag_fields.array_3:2	unsigned int	1
flag_fields.array_4:2	unsigned int	1
flag_fields.pmd_1:1	unsigned int	1
flag_fields.pmd_2:1	unsigned int	1
flag_fields.pmd_3:1	unsigned int	1
flag_fields.spare:5	unsigned int	1
flags	short	1

B.11 struct lv1_bcr

Attribute	Data Type	Vector Length
array_nr	short	1
start	short	1

Attribute	Data Type	Vector Length
end	short	1

B.12 struct lv1_kde

Attribute	Data Type	Vector Length
bsdf_1	float	SCIENCE_CHANNELS
bsdf_2	float	SCIENCE_CHANNELS
resp_1	float	SCIENCE_CHANNELS
resp_2	float	SCIENCE_CHANNELS
f2_1	float	SCIENCE_CHANNELS
f2_2	float	SCIENCE_CHANNELS
smdep_1	float	SCIENCE_CHANNELS
smdep_2	float	SCIENCE_CHANNELS
chi_1	float	SCIENCE_CHANNELS
chi_2	float	SCIENCE_CHANNELS
eta_1	float	SCIENCE_CHANNELS
eta_2	float	SCIENCE_CHANNELS
ksi_1	float	SCIENCE_CHANNELS
ksi_2	float	SCIENCE_CHANNELS
rfs	float	SCIENCE_PIXELS

B.13 struct lv1_ghost

Attribute	Data Type	Vector Length
symmetry	short	SCIENCE_CHANNELS
center	short	SCIENCE_CHANNELS
defocus	float	SCIENCE_CHANNELS
energy	float	SCIENCE_CHANNELS

B.14 struct lv1_leak

Attribute	Data Type	Vector Length
noise	float	1
pmd_offs	float	PMD_NUMBER
pmd_noise	float	1
dark	float	SCIENCE_PIXELS

B.15 struct lv1_hot

Attribute	Data Type	Vector Length
record	short	1
array	short	1
pixel	short	1

B.16 struct lv1_spec

Attribute	Data Type	Vector Length
coeffs	double	SCIENCE_CHANNELS
error	double	SCIENCE_CHANNELS

B.17 struct lv1_calib

Attribute	Data Type	Vector Length
eta_omega	float	CHANNEL_SIZE
response	float	CHANNEL_SIZE

B.18 struct fcd_gome

Attribute	Data Type	Vector Length
detector_flags	union quality_fcd (B.10)	1
npeltier	short	1
nleak	short	1
nhot	short	1
nspec	short	1
nang	short	1
width_conv	short	1
indx_spec	short	1
sun_date	unsigned int	1
sun_time	unsigned int	1
bsdf_0	float	1
elevation	float	1
azimuth	float	1
sun_pmd	float	PMD_NUMBER
sun_pmd_wv	float	PMD_NUMBER
stray_level	float	4
scale_peltier	float	NUM_FPA_SCALE
coeffs	float	8
filter_peltier	float	NUM_FPA_COEFFS
pixel_gain	float	SCIENCE_PIXELS
intensity	float	SCIENCE_PIXELS
sun_ref	float	SCIENCE_PIXELS
sun_precision	float	SCIENCE_PIXELS
ghost	struct lv1_ghost (B.13)	1
kde	struct lv1_kde (B.12)	1
bcr	struct lv1_bcr (B.11)	NUM_SPEC_BANDS
leak	struct lv1_leak (B.14)	1
hot	struct lv1_hot (B.15)	1
spec	struct lv1_spec (B.16)	1
calib	struct lv1_calib (B.17)	1

B.19 struct polar_gome

Attribute	Data Type	Vector Length
wv	float	NUM_POLAR_COEFFS
coeff	float	NUM_POLAR_COEFFS
error	float	NUM_POLAR_COEFFS

Attribute	Data Type	Vector Length
chi	float	1

B.20 struct mph0_gome

Attribute	Data Type	Vector Length
ProductConfidenceData	char	3
UTC_MPH_Generation	char	25
ProcessorSoftwareVersion	char	9

B.21 struct sph0_gome

Attribute	Data Type	Vector Length
sph_5	char	2
sph_6	char	21

B.22 struct ihr_gome

Attribute	Data Type	Vector Length
subsetcounter	unsigned short	1
prism_temp	unsigned short	1
averagemode	unsigned short	1
intg.stat.ch4:2	unsigned char	1
intg.stat.ch3:2	unsigned char	1
intg.stat.ch2b:2	unsigned char	1
intg.stat.ch2a:2	unsigned char	1
intg.stat.ch1b:2	unsigned char	1
intg.stat.ch1a:2	unsigned char	1
intg.stat.fpa2:1	unsigned char	1
intg.stat.fpa3:2	unsigned char	1
intg.stat.fpa4:1	unsigned char	1
intg.two_byte	unsigned short	1
pmd	unsigned short	PMD_NUMBER
peltier	short	SCIENCE_CHANNELS

B.23 struct pmd_gome

Attribute	Data Type	Vector Length
glr	struct glr1_gome (B.7)	1
value	float	PMD_NUMBER

B.24 struct pcd_gome

Attribute	Data Type	Vector Length
selected	short	1
indx_spec	short	1
indx_leak	short	1

Attribute	Data Type	Vector Length
indx_bands	short	NUM_SPEC_BANDS
dark_current	float	1
noise_factor	float	1
glr	struct glr1_gome (B.7)	1
cld	struct cr1_gome (B.9)	1
polar	struct polar_gome (B.19)	1
mph0	struct mph0_gome (B.20)	1
sph0	struct sph0_gome (B.21)	1
ihr	struct ihr_gome (B.22)	1
pmd	struct pmd_gome (B.23)	PMD_IN_GRID

B.25 struct smcd_gome

Attribute	Data Type	Vector Length
selected	short	1
indx_spec	short	1
indx_leak	short	1
indx_bands	short	NUM_SPEC_BANDS
utc_date	unsigned int	1
utc_time	unsigned int	1
north_sun_zen	float	1
north_sun_azim	float	1
north_sm_zen	float	1
north_sm_azim	float	1
sun_or_moon	float	1
dark_current	float	1
noise_factor	float	1
mph0	struct mph0_gome (B.20)	1
sph0	struct sph0_gome (B.21)	1
ihr	struct ihr_gome (B.22)	1
pmd	struct pmd_gome (B.23)	PMD_IN_GRID

B.26 union quality_rec

Attribute	Data Type	Vector Length
flag_fields.dead:2	unsigned int	1
flag_fields.hot:2	unsigned int	1
flag_fields.saturate:2	unsigned int	1
flag_fields.spectral:2	unsigned int	1
flags	short	1

B.27 struct rec_gome

Attribute	Data Type	Vector Length
pixel_flags	union quality_rec (B.26)	1
indx_psp	short	1
indx_pcd	short	1
integration	float	2
wave	float	CHANNEL_SIZE
data	float	CHANNEL_SIZE

B.28 struct irr1_gome

Attribute	Data Type	Vector Length
indx_vcd	short	1
indx_doas	short	1
indx_amf	short	1
indx_icfa	short	1
indx_stats	short	1
cloud_frac	float	1
cloud_pres	float	1
err_cloud_frac	float	1
err_cloud_pres	float	1
surface_pres	float	1
cca_cloud_frac	float	1
cca_subpixel	signed char	16
pmd_avg	float	3
pmd_sdev	float	3
pixel_color	float	16
pixel_gradient	float	1
total_vcd	float	LVL2_MAX_NMOL
error_vcd	float	LVL2_MAX_NMOL
slant_doas	float	LVL2_MAX_NMOL
error_doas	float	LVL2_MAX_NMOL
rms_doas	float	LVL2_MAX_NWIN
chi_doas	float	LVL2_MAX_NWIN
fit_doas	float	LVL2_MAX_NWIN
iter_doas	float	LVL2_MAX_NWIN
ground_amf	float	LVL2_MAX_NMOL
cloud_amf	float	LVL2_MAX_NMOL
intensity_ground	float	LVL2_MAX_NWIN
intensity_cloud	float	LVL2_MAX_NWIN
intensity_measured	float	LVL2_MAX_NWIN

B.29 struct irr2_gome

Attribute	Data Type	Vector Length
indx_vcd	short	1
indx_doas	short	1
indx_amf	short	1
ozone_temperature	float	1
ozone_ring_corr	float	1
ghost_column	float	1
cld_frac	float	1
error_cld_frac	float	1
cld_height	float	1
error_cld_height	float	1
cld_press	float	1
error_cld_press	float	1
cld_albedo	float	1
error_cld_albedo	float	1
surface_height	float	1
surface_press	float	1

Attribute	Data Type	Vector Length
surface_albedo	float	1
total_vcd	float	LVL2_V2_NMOL
error_vcd	float	LVL2_V2_NMOL
slant_doas	float	LVL2_V2_NMOL
error_doas	float	LVL2_V2_NMOL
rms_doas	float	LVL2_V2_NWIN
chi_doas	float	LVL2_V2_NWIN
fit_doas	float	LVL2_V2_NWIN
iter_doas	float	LVL2_V2_NWIN
ground_amf	float	LVL2_V2_NMOL
error_ground_amf	float	LVL2_V2_NMOL
cloud_amf	float	LVL2_V2_NMOL
error_cloud_amf	float	LVL2_V2_NMOL

B.30 struct ddr_gome

Attribute	Data Type	Vector Length
glr	struct glr2_gome (B.8)	1
irr1	struct irr1_gome (B.28)	1
irr2	struct irr2_gome (B.29)	1
ozone	float	1
error	float	1

C SCIA Compound Data Types (general)

C.1 struct scale_rec

Attribute	Data Type	Vector Length
offs	float	1
scale	float	1

C.2 struct scia_memcorr

Attribute	Data Type	Vector Length
dims	size_t	2
matrix	float	1

C.3 struct scia_nlincorr

Attribute	Data Type	Vector Length
dims	size_t	2
curve	char	1
matrix	float	1

C.4 struct scia_straycorr

Attribute	Data Type	Vector Length
dims	size_t	2
grid_in	float	1
grid_out	float	1
matrix	float	1

C.5 struct geoL_scia

Attribute	Data Type	Vector Length
pixel_type	unsigned char	1
glint_flag	unsigned char	1
pos_esm	float	1
pos_asm	float	1
sat_h	float	1
earth_rad	float	1
dopp_shift	float	1
sun_zen_ang	float	3
sun_azi_ang	float	3
los_zen_ang	float	3
los_azi_ang	float	3
tan_h	float	3
sub_sat_point	struct coord_envi (A.4)	1
tang_ground_point	struct coord_envi (A.4)	3

C.6 struct geoN_scia

Attribute	Data Type	Vector Length
pixel_type	unsigned char	1
glint_flag	unsigned char	1
pos_esm	float	1
sat_h	float	1
earth_rad	float	1
sun_zen_ang	float	3
sun_azi_ang	float	3
los_zen_ang	float	3
los_azi_ang	float	3
sub_sat_point	struct coord_envi (A.4)	1
corner	struct coord_envi (A.4)	NUM_CORNERS
center	struct coord_envi (A.4)	1

C.7 struct geoC_scia

Attribute	Data Type	Vector Length
pos_esm	float	1
pos_asm	float	1
sun_zen_ang	float	1
sub_sat_point	struct coord_envi (A.4)	1

C.8 struct gdf_para

Attribute	Data Type	Vector Length
p_bar	float	1
beta	float	1
w0	float	1

C.9 struct polV_scia

Attribute	Data Type	Vector Length
Q	float	NUM_FRAC_POLV
error_Q	float	NUM_FRAC_POLV
U	float	NUM_FRAC_POLV
error_U	float	NUM_FRAC_POLV
rep_wv	float	NUM_FRAC_POLV+1
gdf	struct gdf_para (C.8)	1
intg_time	unsigned short	1

C.10 struct lads_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
corner	struct coord_envi (A.4)	NUM_CORNERS

C.11 struct mds1c_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
rad_units_flag	signed char	1
quality_flag	signed char	1
type_mds	unsigned char	1
coaddf	unsigned char	1
category	unsigned char	1
state_id	unsigned char	1
state_index	unsigned char	1
chan_id	unsigned char	1
clus_id	unsigned char	1
dur_scan	unsigned short	1
num_obs	unsigned short	1
num_pixels	unsigned short	1
dsr_length	unsigned int	1
orbit_phase	float	1

Attribute	Data Type	Vector Length
pet	float	1
pixel_ids	unsigned short	1
pixel_wv	float	1
pixel_wv_err	float	1
pixel_val	float	1
pixel_err	float	1
geoC	struct geoC_scia (C.7)	1
geoL	struct geoL_scia (C.5)	1
geoN	struct geoN_scia (C.6)	1

C.12 struct mds1c_pmd

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality_flag	signed char	1
type_mds	unsigned char	1
category	unsigned char	1
state_id	unsigned char	1
state_index	unsigned char	1
dur_scan	unsigned short	1
num_pmd	unsigned short	1
num_geo	unsigned short	1
dsr_length	unsigned int	1
orbit_phase	float	1
int_pmd	float	1
geoL	struct geoL_scia (C.5)	1
geoN	struct geoN_scia (C.6)	1

C.13 struct mds1c_polV

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality_flag	signed char	1
type_mds	unsigned char	1
category	unsigned char	1
state_id	unsigned char	1
state_index	unsigned char	1
dur_scan	unsigned short	1
total_polV	unsigned short	1
num_diff_intg	unsigned short	1
num_geo	unsigned short	1
dsr_length	unsigned int	1
orbit_phase	float	1
intg_times	unsigned short	MAX_CLUSTER
num_polar	unsigned short	MAX_CLUSTER
polV	struct polV_scia (C.9)	1
geoL	struct geoL_scia (C.5)	1
geoN	struct geoN_scia (C.6)	1

C.14 struct dmop_rec

Attribute	Data Type	Vector Length
timeLine	char	8
dateTimeStart	char	24
dateTimeStop	char	24
muSecStart	unsigned int	1
muSecStop	unsigned int	1
orbitPhase	float	1
stateID	unsigned char	1
absOrbit	unsigned short	1

C.15 struct stateinfo_rec

Attribute	Data Type	Vector Length
softVersion	char	4
stateID	unsigned char	1
indxState	unsigned short	1
indxDMOP	unsigned int	1
jday	double	1
dtMatch	double	1

C.16 struct mds0_sql

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
stateID	unsigned char	1
nrAux	unsigned short	1
nrDet	unsigned short	1
nrPMD	unsigned short	1
obmTemp	float	1
pmdTemp	float	1
chanTemp	float	SCIENCE_CHANNELS

D SCIA Level 0 Compound Data Types

D.1 struct clusdef_rec

Attribute	Data Type	Vector Length
chanID	unsigned char	1
clusID	unsigned char	1
start	unsigned short	1
length	unsigned short	1

D.2 struct fep_hdr

Attribute	Data Type	Vector Length
gsrt	struct mjd_envi (A.3)	1

Attribute	Data Type	Vector Length
isp_length	unsigned short	1
crc_errs	unsigned short	1
rs_errs	unsigned short	1

D.3 struct packet_hdr

Attribute	Data Type	Vector Length
api.field.op_mode:5	unsigned short	1
api.field.vcid:5	unsigned short	1
api.field.:6	unsigned short	1
api.two_byte	unsigned short	1
seq_cntrl	unsigned short	1
length	unsigned short	1

D.4 struct data_hdr

Attribute	Data Type	Vector Length
category	unsigned char	1
state_id	unsigned char	1
length	unsigned short	1
rdv.field.atc_id:6	unsigned char	1
rdv.field.hsm:2	unsigned char	1
rdv.field.config_id:8	unsigned char	1
rdv.two_byte	unsigned short	1
id.field.:4	unsigned char	1
id.field.packet:4	unsigned char	1
id.field.overflow:4	unsigned char	1
id.field.:4	unsigned char	1
id.two_byte	unsigned short	1
on_board_time	unsigned int	1

D.5 struct pmtc_hdr

Attribute	Data Type	Vector Length
pmtc_1.field.ndfm:2	unsigned char	1
pmtc_1.field.:2	unsigned char	1
pmtc_1.field.phase:4	unsigned char	1
pmtc_1.field.sls:2	unsigned char	1
pmtc_1.field.wls:2	unsigned char	1
pmtc_1.field.apsm:2	unsigned char	1
pmtc_1.field.ncwm:2	unsigned char	1
pmtc_1.two_byte	unsigned short	1
scanner_mode	unsigned short	1
az_param.field.repeat:12	unsigned short	1
az_param.field.basic:4	unsigned char	1
az_param.field.h_w:4	unsigned char	1
az_param.field.rel:4	unsigned char	1
az_param.field.corr:4	unsigned char	1
az_param.field.invert:1	unsigned char	1

Attribute	Data Type	Vector Length
az_param.field.filter:1	unsigned char	1
az_param.field.centre:1	unsigned char	1
az_param.field.type:1	unsigned char	1
az_param.four_byte	unsigned int	1
elv_param.field.repeat:12	unsigned short	1
elv_param.field.basic:4	unsigned char	1
elv_param.field.:4	unsigned char	1
elv_param.field.rel:4	unsigned char	1
elv_param.field.corr:4	unsigned char	1
elv_param.field.invert:1	unsigned char	1
elv_param.field.filter:1	unsigned char	1
elv_param.field.centre:1	unsigned char	1
elv_param.field.:1	unsigned char	1
elv_param.four_byte	unsigned int	1
factor	unsigned char	6

D.6 union bench_cntrl

Attribute	Data Type	Vector Length
field.stat:1	unsigned char	1
field.temp:15	unsigned short	1
field.temp:15	unsigned short	1
field.stat:1	unsigned char	1
two_byte	unsigned short	1

D.7 struct aux_bcp

Attribute	Data Type	Vector Length
sync	unsigned short	1
bcps	unsigned short	1
flags.field.phase:4	unsigned char	1
flags.field.m:1	unsigned char	1
flags.field.d:1	unsigned char	1
flags.field.eu:1	unsigned char	1
flags.field.au:1	unsigned char	1
flags.field.pointing:6	unsigned char	1
flags.field.:2	unsigned char	1
flags.two_byte	unsigned short	1
azi_encode_cntr	unsigned int	1
ele_encode_cntr	unsigned int	1
azi_cntr_error	unsigned short	1
ele_cntr_error	unsigned short	1
azi_scan_error	unsigned short	1
ele_scan_error	unsigned short	1

D.8 struct pmtc_frame

Attribute	Data Type	Vector Length
bcp	struct aux_bcp (D.7)	NUM_LV0_AUX_BCP
bench_rad	union bench_cntrl (D.6)	1

Attribute	Data Type	Vector Length
bench_elv	union bench_cntrl (D.6)	1
bench_az	union bench_cntrl (D.6)	1

D.9 struct aux_src

Attribute	Data Type	Vector Length
pmtc	struct pmtc_frame (D.8)	NUM_LV0_AUX_PMTC_FRAME

D.10 struct chan_hdr

Attribute	Data Type	Vector Length
sync	unsigned short	1
bcps	unsigned short	1
bias	unsigned short	1
temp	unsigned short	1
channel.field.lu:2	unsigned char	1
channel.field.is:2	unsigned char	1
channel.field.id:4	unsigned char	1
channel.field.clusters:8	unsigned char	1
channel.two_byte	unsigned short	1
ratio_hdr.field.status:3	unsigned char	1
ratio_hdr.field.ratio:5	unsigned char	1
ratio_hdr.field.frame:8	unsigned char	1
ratio_hdr.two_byte	unsigned short	1
command_vis.field.cntrl:2	unsigned char	1
command_vis.field.ratio:5	unsigned char	1
command_vis.field.sec:9	unsigned short	1
command_vis.field.mode:2	unsigned char	1
command_vis.field.etf:14	unsigned short	1
command_vis.four_byte	unsigned int	1
command_ir.field.cntrl:2	unsigned char	1
command_ir.field.pet:4	unsigned char	1
command_ir.field.:2	unsigned char	1
command_ir.field.bias:3	unsigned char	1
command_ir.field.:3	unsigned char	1
command_ir.field.comp:2	unsigned char	1
command_ir.field.mode:2	unsigned char	1
command_ir.field.etf:14	unsigned short	1
command_ir.four_byte	unsigned int	1

D.11 struct chan_src

Attribute	Data Type	Vector Length
cluster_id	unsigned char	1
co_adding	unsigned char	1
sync	unsigned short	1
block_nr	unsigned short	1
start	unsigned short	1
length	unsigned short	1
data	unsigned char	1

D.12 struct h5_chan_src

Attribute	Data Type	Vector Length
cluster_id	unsigned char	1
co_adding	unsigned char	1
sync	unsigned short	1
block_nr	unsigned short	1
start	unsigned short	1
length	unsigned short	1
indx_data	unsigned short	1

D.13 struct det_src

Attribute	Data Type	Vector Length
hdr	struct chan_hdr (D.10)	1
pixel	struct chan_src (D.11)	1

D.14 struct pmd_data

Attribute	Data Type	Vector Length
sync	unsigned short	1
data	unsigned short	PMD_NUMBER
bcps	unsigned short	1
time.field.delta:15	unsigned short	1
time.field.is:1	unsigned short	1
time.two_byte	unsigned short	1

D.15 struct pmd_src

Attribute	Data Type	Vector Length
temp	unsigned short	1
packet	struct pmd_data (D.14)	NUM_LV0_PMD_PACKET

D.16 struct sph0_scia

Attribute	Data Type	Vector Length
descriptor	char	29
start_lat	double	1
start_lon	double	1
stop_lat	double	1
stop_lon	double	1
sat_track	double	1
isp_errors	unsigned short	1
missing_isps	unsigned short	1
isp_discard	unsigned short	1
rs_sign	unsigned short	1
num_error_isps	int	1
error_isps_thres	double	1
num_miss_isps	int	1

Attribute	Data Type	Vector Length
miss_isps_thres	double	1
num_discard_isps	int	1
discard_isps_thres	double	1
num_rs_isps	int	1
rs_thres	double	1
tx_rx_polar	char	6
swath	char	4

D.17 struct info_clus

Attribute	Data Type	Vector Length
chanID	unsigned char	1
clusID	unsigned char	1
coAdding	unsigned char	1
start	unsigned short	1
length	unsigned short	1

D.18 struct mds0_info

Attribute	Data Type	Vector Length
packetID	unsigned char	1
category	unsigned char	1
stateID	unsigned char	1
numClusters	unsigned char	1
length	unsigned short	1
bcps	unsigned short	1
stateIndex	unsigned short	1
offset	unsigned int	1
mjd	struct mjd_envi (A.3)	1
cluster	struct info_clus (D.17)	MAX_CLUSTER

D.19 struct mds0_aux

Attribute	Data Type	Vector Length
isp	struct mjd_envi (A.3)	1
fep_hdr	struct fep_hdr (D.2)	1
packet_hdr	struct packet_hdr (D.3)	1
data_hdr	struct data_hdr (D.4)	1
pmtc_hdr	struct pmtc_hdr (D.5)	1
data_src	struct aux_src (D.9)	1

D.20 struct mds0_det

Attribute	Data Type	Vector Length
bcps	unsigned short	1
num_chan	unsigned short	1
orbit_vector	int	8
isp	struct mjd_envi (A.3)	1
fep_hdr	struct fep_hdr (D.2)	1

Attribute	Data Type	Vector Length
packet_hdr	struct packet_hdr (D.3)	1
data_hdr	struct data_hdr (D.4)	1
pmtc_hdr	struct pmtc_hdr (D.5)	1
data_src	struct det_src (D.13)	1

D.21 struct mds0_pmd

Attribute	Data Type	Vector Length
isp	struct mjd_envi (A.3)	1
fep_hdr	struct fep_hdr (D.2)	1
packet_hdr	struct packet_hdr (D.3)	1
data_hdr	struct data_hdr (D.4)	1
data_src	struct pmd_src (D.15)	1

D.22 struct offs_size_rec

Attribute	Data Type	Vector Length
offset	unsigned int	1
length	unsigned int	1

D.23 struct h5_mds0_info

Attribute	Data Type	Vector Length
packetID	unsigned char	1
category	unsigned char	1
stateID	unsigned char	1
numClusters	unsigned char	1
length	unsigned short	1
bcps	unsigned short	1
stateIndex	unsigned short	1
offset	unsigned int	1
mjd	struct mjd_envi (A.3)	1

E SCIA Level 1b Compound Data Types

E.1 struct Clcon_scia

Attribute	Data Type	Vector Length
id	unsigned char	1
channel	unsigned char	1
type	unsigned char	1
dummy	/* unsigned char	1
pixel_nr	unsigned short	1
length	unsigned short	1
intg_time	unsigned short	1
coaddf	unsigned short	1
n_read	unsigned short	1
pet	float	1

E.2 union det_signal

Attribute	Data Type	Vector Length
field.:24	unsigned int sign	1
field.:8	signed char corr	1
field.:8	signed char corr	1
field.:24	unsigned int sign	1
four_byte	unsigned int	1

E.3 struct Sig_scia

Attribute	Data Type	Vector Length
stray	unsigned char	1
corr	signed char	1
sign	unsigned short	1

E.4 struct Sigc_scia

Attribute	Data Type	Vector Length
stray	unsigned char	1
det	union det_signal (E.2)	1

E.5 struct Clus_scia

Attribute	Data Type	Vector Length
n_sig	unsigned short	1
n_sigc	unsigned short	1
sig	struct Sig_scia (E.3)	1
sigc	struct Sigc_scia (E.4)	1

E.6 struct sph1_scia

Attribute	Data Type	Vector Length
spec_cal	char	5
saturate	char	5
dark_check	char	5
dead_pixel	char	5
key_data	char	6
m_factor	char	6
descriptor	char	29
init_version	char	38
start_time	char	MAX_UTC_STRING
stop_time	char	MAX_UTC_STRING
stripline	short	1
slice_pos	short	1
no_slice	unsigned short	1
no_nadir	unsigned short	1
no_limb	unsigned short	1
no_occult	unsigned short	1

Attribute	Data Type	Vector Length
no_monitor	unsigned short	1
no_noproc	unsigned short	1
comp_dark	unsigned short	1
incomp_dark	unsigned short	1
start_lat	double	1
start_lon	double	1
stop_lat	double	1
stop_lon	double	1

E.7 struct sqads1_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
flag_glint	unsigned char	1
flag_rainbow	unsigned char	1
flag_saa_region	unsigned char	1
missing_readouts	unsigned short	1
hotpixel	unsigned short	ALL_CHANNELS
mean_wv_diff	float	SCIENCE_CHANNELS
sdev_wv_diff	float	SCIENCE_CHANNELS
mean_diff_leak	float	ALL_CHANNELS

E.8 struct sip_scia

Attribute	Data Type	Vector Length
do_use_limb_dark	char	2
do_pixelwise	char	SCIENCE_CHANNELS+1
do_ib_oc_etn	char	PMD_NUMBER+1
do_ib_sd_etn	char	PMD_NUMBER+1
SCIENCE_CHANNELS+1]	char do_fraunhofer[5 *	1
SCIENCE_CHANNELS+1]	char do_etalon[3 *	1
IR_CHANNELS+1]	char do_var_lc_cha[4 *	1
SCIENCE_CHANNELS+1]	char do_stray_lc_cha[4 *	1
IR_PMD_NUMBER+1]	char do_var_lc_pmd[4 *	1
PMD_NUMBER+1]	char do_stray_lc_pmd[4 *	1
do_pol_point	char	NUM_FRAC_POLV+1
n_lc_min	unsigned char	1
ds_n_phases	unsigned char	1
sp_n_phases	unsigned char	1
ds_poly_order	unsigned char	1
lc_harm_order	unsigned char	1
level_2_smr	unsigned char	SCIENCE_CHANNELS
sat_level	unsigned short	SCIENCE_CHANNELS
pmd_sat_limit	unsigned short	1
startpix_6	short	1
startpix_8	short	1
alpha0_asm	float	1
alpha0_esm	float	1
ppg_error	float	1

Attribute	Data Type	Vector Length
stray_error	float	1
h_toa	float	1
lambda_end_gdf	float	1
ds_phase_boundaries	float	MaxBoundariesSIP
sp_phase_boundaries	float	MaxBoundariesSIP
lc_stray_indx	float	2
electrons_bu	float	SCIENCE_CHANNELS

E.9 struct clcp_scia

Attribute	Data Type	Vector Length
fpn	float	SCIENCE_PIXELS
fpn_error	float	SCIENCE_PIXELS
lc	float	SCIENCE_PIXELS
lc_error	float	SCIENCE_PIXELS
PMD_NUMBER]	float pmd_dark[2 *	1
PMD_NUMBER]	float pmd_dark_error[2 *	1
mean_noise	float	SCIENCE_PIXELS

E.10 struct vlcp_scia

Attribute	Data Type	Vector Length
orbit_phase	float	1
PMD_NUMBER]	float obm_pmd[IR_CHANNELS +	1
CHANNEL_SIZE]	float var_lc[IR_CHANNELS *	1
CHANNEL_SIZE]	float var_lc_error[IR_CHANNELS *	1
solar_stray	float	SCIENCE_PIXELS
solar_stray_error	float	SCIENCE_PIXELS
pmd_stray	float	PMD_NUMBER
pmd_stray_error	float	PMD_NUMBER
pmd_dark	float	IR_PMD_NUMBER
pmd_dark_error	float	IR_PMD_NUMBER

E.11 struct ppg_scia

Attribute	Data Type	Vector Length
ppg_fact	float	SCIENCE_PIXELS
etalon_fact	float	SCIENCE_PIXELS
etalon_resid	float	SCIENCE_PIXELS
wls_deg_fact	float	SCIENCE_PIXELS
bad_pixel	unsigned char	SCIENCE_PIXELS

E.12 struct base_scia

Attribute	Data Type	Vector Length
wvlen_det_pix	float	SCIENCE_PIXELS

E.13 struct scp_scia

Attribute	Data Type	Vector Length
orbit_phase	float	1
SCIENCE_CHANNELS]	double coeffs[NUM_SPEC_COEFFS *	1
num_lines	unsigned short	SCIENCE_CHANNELS
wv_error_calib	float	SCIENCE_CHANNELS

E.14 struct srs_scia

Attribute	Data Type	Vector Length
sun_spec_id	char	3
avg_asm	float	1
avg_esm	float	1
avg_elev_sun	float	1
dopp_shift	float	1
wvlen_sun	float	SCIENCE_PIXELS
mean_sun	float	SCIENCE_PIXELS
precision_sun	float	SCIENCE_PIXELS
accuracy_sun	float	SCIENCE_PIXELS
etalon	float	SCIENCE_PIXELS
pmd_mean	float	PMD_NUMBER
pmd_out_nd_out	float	PMD_NUMBER
pmd_out_nd_in	float	PMD_NUMBER

E.15 struct pspn_scia

Attribute	Data Type	Vector Length
ang_esm	float	1
mu2	double	SCIENCE_PIXELS
mu3	double	SCIENCE_PIXELS

E.16 struct psplo_scia

Attribute	Data Type	Vector Length
ang_esm	float	1
ang_asm	float	1
mu2	double	SCIENCE_PIXELS
mu3	double	SCIENCE_PIXELS

E.17 struct rspn_scia

Attribute	Data Type	Vector Length
ang_esm	float	1
sensitivity	double	SCIENCE_PIXELS

E.18 struct rsplo_scia

Attribute	Data Type	Vector Length
ang_esm	float	1
ang_asm	float	1
sensitivity	double	SCIENCE_PIXELS

E.19 struct ekd_scia

Attribute	Data Type	Vector Length
mu2_nadir	float	SCIENCE_PIXELS
mu3_nadir	float	SCIENCE_PIXELS
mu2_limb	float	SCIENCE_PIXELS
mu3_limb	float	SCIENCE_PIXELS
radiance_vis	float	SCIENCE_PIXELS
radiance_nadir	float	SCIENCE_PIXELS
radiance_limb	float	SCIENCE_PIXELS
radiance_sun	float	SCIENCE_PIXELS
bsdf	float	SCIENCE_PIXELS

E.20 struct sfp_scia

Attribute	Data Type	Vector Length
pix_pos_slit_fun	unsigned short	1
type_slit_fun	unsigned char	1
fwhm_slit_fun	float	1
f_voi_fwhm_loren	float	1

E.21 struct asfp_scia

Attribute	Data Type	Vector Length
pix_pos_slit_fun	unsigned short	1
type_slit_fun	unsigned char	1
fwhm_slit_fun	float	1
f_voi_fwhm_gauss	float	1

E.22 struct state1_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
Clcon	struct Clcon_scia (E.1)	MAX_CLUSTER
flag_mds	unsigned char	1
flag_reason	unsigned char	1
type_mds	unsigned char	1
dummy	/* unsigned char	1
category	unsigned short	1
state_id	unsigned short	1
dur_scan	unsigned short	1
longest_intg_time	unsigned short	1
num_clus	unsigned short	1
num_aux	unsigned short	1

Attribute	Data Type	Vector Length
num_pmd	unsigned short	1
num_intg	unsigned short	1
intg_times	unsigned short	MAX_CLUSTER
num_polar	unsigned short	MAX_CLUSTER
total_polar	unsigned short	1
num_dsr	unsigned short	1
indx	unsigned int	1
length_dsr	unsigned int	1
offset	unsigned int	1
offs_pmd	unsigned int	1
offs_polV	unsigned int	1
orbit_phase	float	1

E.23 struct pmd_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
mds0	struct mds0_pmd (D.21)	1

E.24 struct aux_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
mds0	struct mds0_aux (D.19)	1

E.25 struct lcpn_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
mjd_last	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
orbit_phase	float	1
PMD_NUMBER]	float obm_pmd[IR_CHANNELS +	1
fpn	float	SCIENCE_PIXELS
fpn_error	float	SCIENCE_PIXELS
lc	float	SCIENCE_PIXELS
lc_error	float	SCIENCE_PIXELS
mean_noise	float	SCIENCE_PIXELS
PMD_NUMBER]	float pmd_off[2 *	1
PMD_NUMBER]	float pmd_off_error[2 *	1

E.26 struct dark_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1

Attribute	Data Type	Vector Length
dark_spec	float	SCIENCE_PIXELS
sdev_dark_spec	float	SCIENCE_PIXELS
PMD_NUMBER]	float pmd_off[2 *	1
PMD_NUMBER]	float pmd_off_error[2 *	1
sol_stray	float	SCIENCE_PIXELS
sol_stray_error	float	SCIENCE_PIXELS
pmd_stray	float	PMD_NUMBER
pmd_stray_error	float	PMD_NUMBER

E.27 struct ppgn_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
gain_fact	float	SCIENCE_PIXELS
etalon_fact	float	SCIENCE_PIXELS
etalon_resid	float	SCIENCE_PIXELS
avg_wls_spec	float	SCIENCE_PIXELS
sd_wls_spec	float	SCIENCE_PIXELS
bad_pixel	unsigned char	SCIENCE_PIXELS

E.28 struct scpn_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
orbit_phase	float	1
srs_param	unsigned char	SCIENCE_CHANNELS
num_lines	unsigned short	SCIENCE_CHANNELS
wv_error_calib	float	SCIENCE_CHANNELS
sol_spec	float	SCIENCE_PIXELS
SCIENCE_CHANNELS]	float line_pos[3 *	1
SCIENCE_CHANNELS]	double coeffs[NUM_SPEC_COEFFS *	1

E.29 struct srsn_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
flag_neu	unsigned char	1
sun_spec_id	char	3
avg_asm	float	1
avg_esm	float	1
avg_elev_sun	float	1
dopp_shift	float	1
wvlen_sun	float	SCIENCE_PIXELS
mean_sun	float	SCIENCE_PIXELS
precision_sun	float	SCIENCE_PIXELS
accuracy_sun	float	SCIENCE_PIXELS

Attribute	Data Type	Vector Length
etalon	float	SCIENCE_PIXELS
pmd_mean	float	PMD_NUMBER
pmd_out	float	PMD_NUMBER

E.30 struct lv0_hdr

Attribute	Data Type	Vector Length
bcps	unsigned short	1
num_chan	unsigned short	1
orbit_vector	int	8
packet_hdr	struct packet_hdr (D.3)	1
data_hdr	struct data_hdr (D.4)	1
pmtc_hdr	struct pmtc_hdr (D.5)	1

E.31 struct mds1_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality_flag	signed char	1
type_mds	unsigned char	1
state_id	unsigned char	1
state_index	unsigned char	1
n_clus	unsigned short	1
n_aux	unsigned short	1
n_pmd	unsigned short	1
n_pol	unsigned short	1
dsr_length	unsigned int	1
scale_factor	unsigned char	SCIENCE_CHANNELS
sat_flags	unsigned char	1
red_grass	unsigned char	1
lv0	struct lv0_hdr (E.30)	1
geoC	struct geoC_scia (C.7)	1
geoL	struct geoL_scia (C.5)	1
geoN	struct geoN_scia (C.6)	1
int_pmd	float	1
polV	struct polV_scia (C.9)	1
clus	struct Clus_scia (E.5)	MAX_CLUSTER

E.32 struct cal_options

Attribute	Data Type	Vector Length
geo_filter	signed char	1
time_filter	signed char	1
category_filter	signed char	1
nadir_mds	signed char	1
limb_mds	signed char	1
occ_mds	signed char	1
moni_mds	signed char	1
pmd_mds	signed char	1

Attribute	Data Type	Vector Length
frac_pol_mds	signed char	1
slit_function	signed char	1
sun_mean_ref	signed char	1
leakage_current	signed char	1
spectral_cal	signed char	1
pol_sens	signed char	1
rad_sens	signed char	1
ppg_etalon	signed char	1
mem_effect_cal	signed char	1
leakage_cal	signed char	1
straylight_cal	signed char	1
ppg_cal	signed char	1
etalon_cal	signed char	1
wave_cal	signed char	1
polarisation_cal	signed char	1
radiance_cal	signed char	1
num_nadir	unsigned short	1
num_limb	unsigned short	1
num_occ	unsigned short	1
num_moni	unsigned short	1
start_lat	int	1
start_lon	int	1
end_lat	int	1
end_lon	int	1
start_time	struct mjd_envi (A.3)	1
stop_time	struct mjd_envi (A.3)	1
category	unsigned short	5
nadir_cluster	signed char	MAX_CLUSTER
limb_cluster	signed char	MAX_CLUSTER
occ_cluster	signed char	MAX_CLUSTER
moni_cluster	signed char	MAX_CLUSTER
l1b_prod_name	char	ENVI_FILENAME_SIZE

E.33 struct keydata_rec

Attribute	Data Type	Vector Length
pspn	struct pspn_scia (E.15)	NUM_KEY_PSPN
rspn	struct rspn_scia (E.17)	NUM_KEY_RSPN
pspl	struct psplo_scia (E.16)	NUM_KEY_PSPL
rspl	struct rsplo_scia (E.18)	NUM_KEY_RSPL
srs	struct srs_scia (E.14)	NUM_KEY_SRS

E.34 struct rspd_key_fix_scia

Attribute	Data Type	Vector Length
wl	float	SCIENCE_PIXELS
OBM_s_p	float	SCIENCE_PIXELS
ABS_RAD	float	SCIENCE_PIXELS
NDF	float	SCIENCE_PIXELS
NDF_s_p	float	SCIENCE_PIXELS
PPG0	float	SCIENCE_PIXELS

E.35 struct rspd_ELEV_scia

Attribute	Data Type	Vector Length
n_wl	int	1
elevat_angle	float	1
wl	float	BRDF_WAVELENGTH
sensitivity	float	BRDF_WAVELENGTH

E.36 struct rspd_EL_AZ_scia

Attribute	Data Type	Vector Length
n_wl	int	1
elevat_angle	float	1
azimuth_angle	float	1
wl	float	BRDF_WAVELENGTH
sensitivity	float	BRDF_WAVELENGTH

E.37 struct rspd_BRDF_scia

Attribute	Data Type	Vector Length
n_wl	int	1
elevat_angle	float	1
asm_angle	float	1
wl	float	BRDF_WAVELENGTH
sensitivity	float	BRDF_WAVELENGTH

E.38 struct rspd_key

Attribute	Data Type	Vector Length
key_fix	struct rspd_key_fix_scia (E.34)	1
elev_p	struct rspd_ELEV_scia (E.35)	1
elev_s	struct rspd_ELEV_scia (E.35)	1
el_az_p	struct rspd_EL_AZ_scia (E.36)	1
el_az_s	struct rspd_EL_AZ_scia (E.36)	1
brdf_p	struct rspd_BRDF_scia (E.37)	1
brdf_s	struct rspd_BRDF_scia (E.37)	1
n_elev	size_t	1
n_el_az	size_t	1
n_brdf	size_t	1

F SCIA Level 2 Compound Data Types

F.1 struct bias_record

Attribute	Data Type	Vector Length
wv_min	unsigned short	1
wv_max	unsigned short	1
nr_micro	unsigned short	1

Attribute	Data Type	Vector Length
micro_min	unsigned short	MAX_BIAS_MICRO_WIN
micro_max	unsigned short	MAX_BIAS_MICRO_WIN

F.2 struct doas_record

Attribute	Data Type	Vector Length
wv_min	unsigned short	1
wv_max	unsigned short	1

F.3 struct win_record

Attribute	Data Type	Vector Length
mol	char	5
wv_min	unsigned short	1
wv_max	unsigned short	1

F.4 struct sph2_scia

Attribute	Data Type	Vector Length
fit_error	char	5
descriptor	char	29
start_time	char	MAX_UTC_STRING
stop_time	char	MAX_UTC_STRING
bias_mol	char	MAX_BIAS_SPECIES
doas_mol	char	MAX_DOAS_SPECIES
stripline	short	1
slice_pos	short	1
no_slice	unsigned short	1
no_bias_win	unsigned short	1
no_bias_mol	unsigned short	1
no_doas_win	unsigned short	1
no_doas_mol	unsigned short	1
start_lat	double	1
start_lon	double	1
stop_lat	double	1
stop_lon	double	1
bias_win	struct bias_record (F.1)	MAX_BIAS_FITTING_WIN
doas_win	struct doas_record (F.2)	MAX_DOAS_FITTING_WIN

F.5 struct sqads2_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
flag_pqf	unsigned char	NL2_SQADS_PQF_FLAGS

F.6 struct state2_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
state_id	unsigned short	1
duration	unsigned short	1
longest_intg_time	unsigned short	1
shortest_intg_time	unsigned short	1
num_obs_state	unsigned short	1

F.7 struct geo_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
intg_time	unsigned short	1
sun_zen_ang	float	3
los_zen_ang	float	3
rel_azi_ang	float	3
sat_h	float	1
earth_rad	float	1
corner	struct coord_envi (A.4)	NUM_CORNERS
center	struct coord_envi (A.4)	1

F.8 struct cld_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality	signed char	1
quality_cld	unsigned char	1
outputflag	unsigned short	1
intg_time	unsigned short	1
numpmd	unsigned short	1
dsrlen	unsigned int	1
cloudfrac	float	1
toppress	float	1
aai	float	1
albedo	float	1
pmdcloudfrac	float	1

F.9 struct doas_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality	signed char	1
dummy	signed char	1
vcdflag	unsigned short	1
escflag	unsigned short	1
amfflag	unsigned short	1
intg_time	unsigned short	1
numfitp	unsigned short	1

Attribute	Data Type	Vector Length
numiter	unsigned short	1
dsrlen	unsigned int	1
vcd	float	1
errvcd	float	1
esc	float	1
erresc	float	1
rms	float	1
chi2	float	1
goodness	float	1
amfgnd	float	1
amfcld	float	1
reflgnd	float	1
reflcld	float	1
refl	float	1
corrpar	float	1

F.10 struct bias_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality	signed char	1
dummy	signed char	1
hghtflag	unsigned short	1
vcdflag	unsigned short	1
intg_time	unsigned short	1
numfitp	unsigned short	1
numsegm	unsigned short	1
numiter	unsigned short	1
dsrlen	unsigned int	1
hght	float	1
errhght	float	1
vcd	float	1
errvcd	float	1
closure	float	1
errclosure	float	1
rms	float	1
chi2	float	1
goodness	float	1
cutoff	float	1
corrpar	float	1

F.11 struct sph_sci_ol

Attribute	Data Type	Vector Length
dbserver	char	6
errorsum	char	5
descriptor	char	29
decont	char	42
nadir_win_uv0	char	31
nadir_win_uv1	char	31
nadir_win_uv2	char	31

Attribute	Data Type	Vector Length
nadir_win_uv3	char	31
nadir_win_uv4	char	31
nadir_win_uv5	char	31
nadir_win_uv6	char	31
nadir_win_uv7	char	31
nadir_win_uv8	char	31
nadir_win_uv9	char	31
nadir_win_ir0	char	31
nadir_win_ir1	char	31
nadir_win_ir2	char	31
nadir_win_ir3	char	31
nadir_win_ir4	char	31
nadir_win_ir5	char	31
limb_win_pth	char	31
limb_win_uv0	char	31
limb_win_uv1	char	31
limb_win_uv2	char	31
limb_win_uv3	char	31
limb_win_uv4	char	31
limb_win_uv5	char	31
limb_win_uv6	char	31
limb_win_uv7	char	31
limb_win_ir0	char	31
limb_win_ir1	char	31
limb_win_ir2	char	31
limb_win_ir3	char	31
limb_win_ir4	char	31
occl_win_pth	char	31
occl_win_uv0	char	31
occl_win_uv1	char	31
occl_win_uv2	char	31
occl_win_uv3	char	31
occl_win_uv4	char	31
occl_win_uv5	char	31
occl_win_uv6	char	31
occl_win_uv7	char	31
occl_win_ir0	char	31
occl_win_ir1	char	31
occl_win_ir2	char	31
occl_win_ir3	char	31
occl_win_ir4	char	31
start_time	char	MAX_UTC_STRING
stop_time	char	MAX_UTC_STRING
stripline	short	1
slice_pos	short	1
no_slice	unsigned short	1
no_nadir_win	unsigned short	1
no_limb_win	unsigned short	1
no_occl_win	unsigned short	1
start_lat	double	1
start_lon	double	1
stop_lat	double	1
stop_lon	double	1

F.12 struct sqads_sci_ol

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
flag_pqf	unsigned char	OL2_SQADS_PQF_FLAGS

F.13 struct ngeo_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
pixel_type	unsigned char	1
intg_time	unsigned short	1
sun_zen_ang	float	3
los_zen_ang	float	3
rel_azi_ang	float	3
sat_h	float	1
radius	float	1
subsat	struct coord_envi (A.4)	1
corner	struct coord_envi (A.4)	NUM_CORNERS
center	struct coord_envi (A.4)	1

F.14 struct lgeo_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
flag_mds	unsigned char	1
dummy	unsigned char	1
intg_time	unsigned short	1
sun_zen_ang	float	3
los_zen_ang	float	3
rel_azi_ang	float	3
sat_h	float	1
radius	float	1
tan_h	float	3
subsat	struct coord_envi (A.4)	1
tang	struct coord_envi (A.4)	3

F.15 struct cld_sci_ol

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality	signed char	1
dummy	signed char	1
intg_time	unsigned short	1
numpmdpix	unsigned short	1
cloudtype	unsigned short	1
cloudflag	unsigned short	1
aaiflag	unsigned short	1

Attribute	Data Type	Vector Length
numaeropars	unsigned short	1
fullfree	unsigned short	2
dsrlen	unsigned int	1
surfpress	float	1
cloudfrac	float	1
errcldfrac	float	1
toppress	float	1
errtoppress	float	1
cldoptdepth	float	1
errcldoptdep	float	1
cloudbrdf	float	1
errcldbrdf	float	1
effsurfrefl	float	1
erreffsrefl	float	1
aai	float	1
aaidiag	float	1
aeropars	float	1

F.16 struct nfit_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality	signed char	1
dummy	signed char	1
intg_time	unsigned short	1
numvcd	unsigned short	1
vcdflag	unsigned short	1
num_fitp	unsigned short	1
num_nfitp	unsigned short	1
numiter	unsigned short	1
fitflag	unsigned short	1
amfflag	unsigned short	1
dsrlen	unsigned int	1
esc	float	1
erresc	float	1
rms	float	1
chi2	float	1
goodness	float	1
amfgrd	float	1
erramfgrd	float	1
amfcld	float	1
erramfcld	float	1
temperature	float	1
vcd	float	1
errvcd	float	1
linpars	float	1
errlinpars	float	1
lincorrm	float	1
nlinpars	float	1
errnlinpars	float	1
nlincorrm	float	1

F.17 struct layer_rec

Attribute	Data Type	Vector Length
tangvmr	float	1
errtangvmr	float	1
vertcol	float	1
errvertcol	float	1

F.18 struct meas_grid

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
tangh	float	1
tangp	float	1
tangt	float	1
num_win	unsigned char	1
win_limits	float	2

F.19 struct state_vec

Attribute	Data Type	Vector Length
value	float	1
error	float	1
type	signed char	4

F.20 struct lfit_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
quality	signed char	1
criteria	signed char	1
method	unsigned char	1
refpsrc	unsigned char	1
num_rlevel	unsigned char	1
num_mlevel	unsigned char	1
num_species	unsigned char	1
num_closure	unsigned char	1
num_other	unsigned char	1
num_scale	unsigned char	1
intg_time	unsigned short	1
stvec_size	unsigned short	1
cmatrixsize	unsigned short	1
numiter	unsigned short	1
ressize	unsigned short	1
num_adddiag	unsigned short	1
summary	unsigned short	2
dsrlen	unsigned int	1
refh	float	1
refp	float	1
rms	float	1

Attribute	Data Type	Vector Length
chi2	float	1
goodness	float	1
tangh	float	1
tangp	float	1
tangt	float	1
corrmatrix	float	1
residuals	float	1
adddiag	float	1
mainrec	struct layer_rec (F.17)	1
scaledrec	struct layer_rec (F.17)	1
mgrid	struct meas_grid (F.18)	1
statevec	struct state_vec (F.19)	1

F.21 struct lcld_scia

Attribute	Data Type	Vector Length
mjd	struct mjd_envi (A.3)	1
dsrlen	unsigned int	1
quality	signed char	1
diag_cloud_algo	unsigned char	1
flag_normal_water	unsigned char	1
flag_water_clouds	unsigned char	1
flag_ice_clouds	unsigned char	1
hght_index_max_value_ice	unsigned char	1
flag_polar_strato_clouds	unsigned char	1
hght_index_max_value_strato	unsigned char	1
flag_noctilucent_clouds	unsigned char	1
hght_index_max_value_noctilucent	unsigned char	1
intg_time	unsigned short	1
num_tangent_hghts	unsigned short	1
num_cir	unsigned short	1
num_limb_para	unsigned short	1
max_value_cir	float	1
hght_max_value_cir	float	1
max_value_cir_ice	float	1
hght_max_value_cir_ice	float	1
max_value_cir_strato	float	1
hght_max_value_cir_strato	float	1
hght_max_value_noctilucent	float	1
tangent_hghts	float	1
cir	float	1
limb_para	float	1

G SDMF Compound Data Types

G.1 struct monitor_rec

Attribute	Data Type	Vector Length
Orbit	int	1
MagicNumber	int	1

Attribute	Data Type	Vector Length
StateCount	int	16
QualityNumber	int	1
QualitySmoothMask	int	1
Consolidated	int	1
Transmission	int	1
WLSTransmission	int	1
PixelGain	int	1
Orbital	int	1
OrbitalData	int	1
OrbitalFit	int	1
SMR	int	1
FileName	char	70

G.2 struct mtbl_pt_rec

Attribute	Data Type	Vector Length
julianDay	double	1
duration	int	1
absOrbit	int	1
entryDate	char	STR_SZ_DATE
procStage	char	1
saaFlag	bool	1
porFlag	bool	1
dummy	char	1
orbitPhase	float	1
asmAngle	float	1
esmAngle	float	1
sunAz	float	1
sunEl	float	1
longitude	float	1
latitude	float	1
obmTemp	float	1
detTemp	float	SCIENCE_CHANNELS

G.3 struct geo_pt_rec

Attribute	Data Type	Vector Length
julianDay	double	1
asmAngle	float	1
esmAngle	float	1
sunAz	float	1
sunEl	float	1

G.4 struct ftbl_rec

Attribute	Data Type	Vector Length
absOrbit	int	1
prodName	char	ENVI_FILENAME_SIZE
entryDate	char	STR_SZ_DATE

Attribute	Data Type	Vector Length
stateList	char	MAX_NUM_STATE
dummy	char	3

G.5 struct sdmf_hist1_rec

Attribute	Data Type	Vector Length
offset	unsigned int	1
binsize	unsigned char	1
coaddf	unsigned char	1
location	unsigned short	MAX_NUM_HIST1
count	unsigned short	MAX_NUM_HIST1

G.6 struct sdmf_hist2_rec

Attribute	Data Type	Vector Length
offset	unsigned int	1
binsize	unsigned char	1
coaddf	unsigned char	1
count	unsigned short	MAX_NUM_HIST2
location	unsigned short	MAX_NUM_HIST2
remainder	unsigned short	MAX_NUM_HIST2

G.7 struct mtbl_calib_rec

Attribute	Data Type	Vector Length
julianDay	double	1
duration	int	1
absOrbit	int	1
entryDate	char	STR_SZ_DATE
procStage	char	2
softVersion	char	15
saaFlag	unsigned char	1
rtsEnhFlag	unsigned char	1
vorporFlag	unsigned char	1
orbitPhase	float	1
sunSemiDiam	float	1
moonAreaSunlit	float	1
longitude	float	1
latitude	float	1
asmAngle	float	1
esmAngle	float	1
obmTemp	float	1
detectorTemp	float	SCIENCE_CHANNELS

G.8 struct mtbl_statedark_rec

Attribute	Data Type	Vector Length
absOrbit	int	1
StateId	short	1

Attribute	Data Type	Vector Length
StateCount	short	1
Tobm	float	1
Tdet	float	SCIENCE_CHANNELS
entryDate	char	STR_SZ_DATE
saaFlag	bool	1
Quality	int	1

G.9 struct mtbl_dark_rec

Attribute	Data Type	Vector Length
julianDay	double	1
absOrbit	int	1
entryDate	char	STR_SZ_DATE
saaFlag	bool	1
quality	short	1
orbitPhase	float	1
obmTemp	float	1
detectorTemp	float	SCIENCE_CHANNELS

G.10 struct mtbl_dark2_rec

Attribute	Data Type	Vector Length
julianDay	double	1
entryDate	double	1
absOrbit	unsigned short	1
quality	unsigned short	1
stateCount	unsigned short	1
statesSelected	unsigned short	1
orbitRange	unsigned short	SCIENCE_CHANNELS
obmTemp	float	1
detTemp	float	SCIENCE_CHANNELS

G.11 struct mtbl_simudark_rec

Attribute	Data Type	Vector Length
julianDate	double	1
absOrbit	int	1
entryDate	char	STR_SZ_DATE
saaFlag	bool	1
obmTemp	float	1
detTemp	float	1
quality	float	1
orbitPhase	float	1
sig_phase	float	1
phase2	float	1
sig_phase2	float	1
amp2	float	1
sig_amp2	float	1