

SGL I プロダクト I/O ツールキット 取扱説明書

初版

平成 30 年 12 月



目 次

1 HDF ライブラリと SGLI I/O ツールキット	1-1
1.1 SGLI プロダクト情報	1-1
1.2 SGTK とは	1-2
2 HDF ライブラリと SGTK のインストール	2-1
2.1 HDF5 ライブラリのインストール	2-1
2.1.1 HDF5 ライブラリの入手	2-1
2.1.2 HDF5 ライブラリのインストール手順 (Linux 環境)	2-1
2.1.3 HDF5 ライブラリのインストール手順 (Windows)	2-2
2.2 SGTK のインストール	2-3
2.2.1 Linux 環境の SGTK インストール	2-3
2.2.2 Windows 環境の SGTK インストール	2-4
3 SGTK を利用したプログラミング	3-1
3.1 プログラミングの流れ	3-2
4 機能構成	4-1
5 SGTK の関数	5-1
5.1 C 言語	5-1
5.1.1 共通関数	5-1
5.1.2 データ取得関数	5-2
5.1.3 データ設定関数	5-12
5.1.4 その他の関数	5-22
5.2 Fortran 言語	5-25
5.2.1 共通関数	5-25
5.2.2 データ取得関数	5-26
5.2.3 データ設定関数	5-35
5.2.4 その他の関数	5-44
6 ファイル定義	6-1
6.1 設定ファイル	6-1
6.2 うるう秒ファイル	6-3
6.3 アクセスラベルパラメータファイル	6-4
7 入出力データ	7-1
7.1 データ定義	7-1
7.1.1 データセットアクセスラベル	7-1
8 エラー番号	8-2
9 付録	9-1
9.1 サンプルプログラム (sample_01) について	9-1
9.1.1 Linux 環境の場合	9-1
9.1.2 Windows 環境の場合	9-2

1 HDF ライブラリと SGLI I/O ツールキット

SGLI プロダクト I/O ツールキット（以降 **SGTK** と呼びます。）は、気候変動観測衛星（GCOM-C）のデータである SGLI プロダクトデータを、C 言語や Fortran 言語のプログラムで利用するために提供されます。SGLI プロダクトデータは、レベル 1～3 に分類され、HDF（Hierarchical Data Format）というファイル形式に格納されます。SGTK は、HDF の SGLI プロダクトデータに容易にアクセスできるようにします。

1.1 SGLI プロダクト情報

SGLI プロダクトは、HDF5 フォーマットのファイルとして提供されます。

HDF (Hierarchical Data Format) は、NCSA (The National Center for Supercomputing Applications: イリノイ大学) がユーザの計算機構成に依存せず、情報を利用できるように開発したフォーマットです。HDF ファイルは、HDF4 と HDF5 フォーマットがあり、HDF5 は、HDF4 の課題（データサイズに制限がある、複数種類のデータタイプなど）を全面的に見直して提供されています。

SGTK は、HDF5 のファイルにアクセスし、ユーザプログラムで指定された情報を入出力します。

HDF ファイルは、属性情報を持つ **Attribute** 部とプロダクト情報そのものをもつ **Dataset** 部、それらを包含する **Group** 部に分けられます。SGLI プロダクトの **Attribute** 部にはメタデータが収容され、**Dataset** 部には観測データ、緯度経度データなどプロダクトデータが収容されます。**Group** 部は、ファイルシステムのディレクトリに類似した要素であり、「幾何情報」などの分類で複数のメタデータ、データセットをまとめるグループとして使用されます。

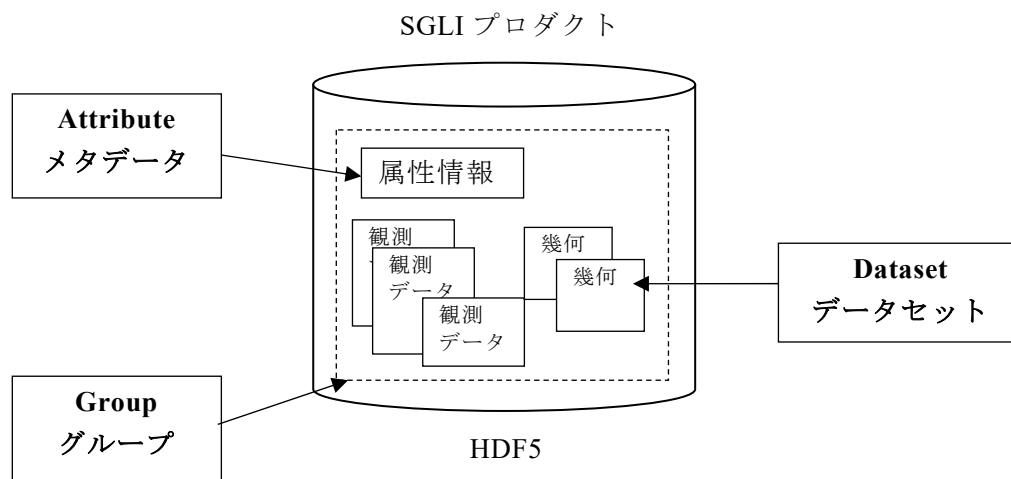


図 1-1 HDF ファイルの構成

HDF へのアクセスは、HDF ファイルをオープンし、このオープンにより返される ID を指定して Attribute、Dataset の情報の入出力を行います。SGTK では複数の Dataset を識別するため、任意の要素へのパス、またはアクセスラベル（規定のパスのエイリアス。各プログラミング言語上の識別子として提供）を使用して Dataset を特定します。

1.2 SGTK とは

SGTKは、C言語やFortran言語のプログラム上で、SGLIプロダクトデータ、DEMファイル、気象ファイルに容易にアクセスするため開発されたツールキットです。

SGTKは、SGLIプロダクト情報の入出力関数群で構成されます。

HDF へのアクセスは、HDF ファイルをオープンし、このオープンにより返される識別子 (hid_t) を指定して Attribute、各データセットの情報の入出力を行います。SGTK では、複数のデータセットを識別するため、HDF 識別番号（詳細は、[7.1.1](#) 章を参照してください。）を使用してデータセットを特定します。

(1) 動作確認環境

SGTK は、表 1-1に示す環境で動作確認を行いました。

表 1-1 動作確認環境

機種名	DELL		DELL		DELL	
OS	Red Hat Enterprise Linux 6.3 Linux 2.6(32bit,64bit)		Windows 8.1(32bit)		Windows 8.1(64bit)	
コンパイラ	C/C++言語	Fortran	C/C++言語	Fortran	C/C++言語	Fortran
	gcc, g++ Intel C++ Compiler	g77, g95,gfortran f77, f95 Intel Fortran Compiler	Visual Studio 2013,2015	G95	Visual Studio 2013,2015	G95
メモリサイズ	12GB		4GB		4GB	
HDF ライブラリ	HDF5-1.8.21					

(※1)

g95 コンパイラは、G95 の Web サイトにて公開されている以下のバイナリを使用しました。
Linux x86_64/EMT64 (32 bit D.I.) Default integer of 32 bits, compatible with older programs

(2) プラットフォーム

SGTK は、32 ビットマシン／64 ビットマシンの両方に対応します。

(3) 適用言語／使用コンパイラ

SGTK は、表 1-2に示す言語、コンパイラで使用します。

表 1-2 適用言語／使用コンパイラ

OS	C/C++言語	Fortran 言語	
		Fortran77	Fortran90/95
RedHat Enterprise Linux	gcc, g++	gfortran	gfortran, G95
	Intel C++ Compiler	Intel Fortran Compiler	
Windows8.1	Visual Studio 2015 (cl.exe)	-	G95

(4) 実数の有効精度

実数の有効精度は、データセットが示す浮動小数点型精度に依存した有効桁まで値の照合を行います。具体的には、単精度：7 桁、倍精度：15 桁になります。

(5) バイトオーダー

SGTK が入出力する HDF5 のバイトオーダーの扱いを以下に示します。

- ・入力：HDF5 に格納されているバイトオーダーに従います。
- ・出力：上書きをする場合は、HDF5 に格納されているバイトオーダーに従います。新規作成する場合は、デフォルトをリトルエンディアンとします。
(SGTK の設定にてビッグエンディアンに変更することも可能です)

2 HDF ライブラリと SGTK のインストール

SGLIプロダクトデータは、HDF5ライブラリのRelease 1.8.21を使用してアクセスします。
ここでは、HDF5ライブラリのRelease 1.8.21のインストールについて説明します。

2.1 HDF5 ライブラリのインストール

HDF5ライブラリのソースコードをコンパイルしインストールします。
プラットフォーム別に用意されているソースファイルを使用します。

2.1.1 HDF5 ライブラリの入手

HDF グループのホームページから HDF5 ライブラリを入手します。HDF5 ライブラリ Release 1.8.21 (リリース日 2018/06/05) のダウンロードのホームページを以下に示します。

<https://portal.hdfgroup.org/display/support/HDF5+1.8.21>

2018 年 8 月時点でのダウンロードの対象を表に示します。ご使用になる計算機及び OS に対応した HDF5 ライブラリを入手してください。

表 2-1 HDF5 ライブラリ

プラットフォーム	ダウンロードファイル名	備考
Linux 環境	hdf5-1.8.21.tar.gz	ソースコード
Windows 環境	CMake-hdf5-1.8.21.zip	ソースコード (CMake 版) VisualStudio2013/2015/2017 用に 32bit/64bit 用のインストールバッチファ イルが用意されています。

2.1.2 HDF5 ライブラリのインストール手順 (Linux 環境)

インストール手順を以下に示します。

(1) ライブラリの入手

ソースコードからコンパイルしインストールするため、ソースコードを入手します。

(2) HDF5 ライブラリ (ソースコード) のインストール

hdf5-1.8.21.tar.gzファイルを解凍し、ファイルを展開します。

```
$ gunzip hdf5-1.8.21.tar.gz  
$ tar xzf hdf5-1.8.21.tar
```

上記コマンドを実行するとhdf5-1.8.21というディレクトリが作成されます。

ここでは、/HDF5/sharedというディレクトリをインストール先に指定し、HDF5ライブラリをインストールする例を示します。

```
$ cd hdf5-1.8.21
```

```
$ mkdir build
```

```
$ cd build
```

```
$ ../configure --prefix=/HDF5/shared --enable-cxx
```

```
$ make
```

```
$ make check
```

```
$ make install
```

Configure の引数としてインストール先(/HDF5/shared)の指定とC++を指定します。

2.1.3 HDF5 ライブラリのインストール手順 (Windows)

インストール手順を以下に示します。

(1) HDF5 ライブラリ (ソースコード) のインストール

Windows 環境用に、ソースコードからインストールする手順です。

- ① cmake をインストールしてください。
(クロスプラットフォームのビルドシステム。<http://www.cmake.org/>)
- ② CMake-hdf5-1.8.21.zip ファイルを解凍し、ファイルを展開します。
- ③ ファイルが展開されたディレクトリ CMake-hdf5-1.8.21 にインストール用の
バッチファイル(下記一覧)が用意されていますので、Visual Studio のバージョン
と Windows の 32bit/64bit の組み合わせで選択して実行します。
 - build-VS2013-32.bat
 - build-VS2013-64.bat
 - build-VS2015-32.bat
 - build-VS2015-64.bat
 - build-VS2017-32.bat
 - build-VS2017-64.bat

C:ドライブ直下に展開されたディレクトリ CMake-hdf5-1.8.21 で
Visual Studio 2015 を使用して Window(64bit)環境用のインストール
を行う場合。

(C:¥CMake-hdf5-1.8.21¥build のディレクトリが作成され
各種ファイルが生成されます)

```
C:¥CMake-hdf5-1.8.21> build-VS2015-64.bat
```

2.2 SGTK のインストール

2.2.1 Linux 環境の SGTK インストール

(1) ファイルの展開

SGTK_SGLI_Ver1.01.tar.gzファイルを解凍し、ファイルを展開します。

```
$ tar xzf SGTK_SGLI_Ver1.01.tar.gz
```

上記コマンドを実行するとSGTK_SGLIというディレクトリが作成されます。このディレクトリには、表 2-2のファイルとディレクトリが展開されています。

表 2-2 SGTK の内容

名称	ファイル／ ディレクトリ の区別	内容	備考
include/	ディレクトリ	SGTKのインクルードファイルを格納したディレクトリです。	
lib/	ディレクトリ	SGTKのライブラリが作成されるディレクトリです。	
linux/	ディレクトリ	Linux環境用のインストーラを格納したディレクトリです。	Makefile
msvc/	ディレクトリ	Windows環境用のインストーラを格納したディレクトリです。	VisualStudioプロジェクトファイル
src/	ディレクトリ	SGTKのソースコードを格納したディレクトリです。	

(2) ライブラリのインストール

SGTKの「linux」のディレクトリ移動した後、makeコマンドを実行します。

その際、MakeFileに記載されている下記のパスを環境に応じて変更してください。

- ・「ROOTDIR」にHDF5ライブラリファイルをインストールしたパス。(HDF5/shared)
- ・「INCDIR」にHDF5ライブラリのヘッダファイルが格納されているパス

```
$ vi Makefile
ROOTDIR =/HDF5/shared
INCDIR= -I$(ROOTDIR)/include -I../include
$ make
```

ライブラリ作成の確認をします。libディレクトリの下にlibsgtk.aが作成されていれば、SGTKライブラリの作成は終了です。

2.2.2 Windows 環境の SGTK インストール

(1) ファイルの展開

SGTK_SGLI_Ver1.01.zip ファイルを解凍し、ファイルを展開します。

ファイルを解凍するとするとSGTK_SGLIというディレクトリが作成されます。このディレクトリには、表 2-3のファイルとディレクトリが展開されています。

表 2-3 SGTK の内容

名称	ファイル／ ディレクトリ の区別	内容	備考
include/	ディレクトリ	SGTKのインクルードファイルを格納したディレクトリです。	
lib/	ディレクトリ	SGTKのライブラリが作成されるディレクトリです。	
linux/	ディレクトリ	Linux環境用のSGTKを作成するためのディレクトリです。	Makefile
msvc/	ディレクトリ	Windows環境用のSGTKを作成するためのディレクトリです。	VisualStudioプロジェクトファイル
src/	ディレクトリ	SGTKのソースコードを格納したディレクトリです。	

(2) ライブラリのインストール

SGTK の「msvc」のフォルダへ移動した後、「sgtk.sln」から VisualStudio を起動します。

「プロジェクト(P)」メニューの「プロパティ(P)」から「sgtk プロパティページ」を開き、「VC++ディレクトリ」を選択して、「インクルードディレクトリ」に HDF5 をインストールした下記 4 つのディレクトリをセミコロン(;)で区切って指定します。

```
C:\Make-hdf5-1.8.21\build
C:\Make-hdf5-1.8.21\ hdf5-1.8.21\src
C:\Make-hdf5-1.8.21\ hdf5-1.8.21\hl\src
C:\Make-hdf5-1.8.21\ hdf5-1.8.21\c++\src
```

```
$(VC_IncludePath);$(WindowsSDK_IncludePath);
↓ 上記内容に追記する ($(WindowsSDK_IncludePath)はない場合がある)
C:\Make-hdf5-1.8.21\build;C:\Make-hdf5-1.8.21\ hdf5-1.8.21\src;C:\Make-hdf5-1.
8.21\ hdf5-1.8.21\c++\src;C:\Make-hdf5-1.8.21\ hdf5-1.8.21\hl\src;
$(VC_IncludePath);$(WindowsSDK_IncludePath);
```

ビルド正常終了後、ライブラリ作成の確認をします。カレントフォルダと同階層に生成される「Release¥Win64 (32bit 環境の場合は Win32)」フォルダの下に `sgtk.lib` が作成されていれば、SGTK ライブラリの作成は終了です。

3 SGTK を利用したプログラミング

Linux環境での設定内容

項目	名称	説明
インストール先	/HDF5/shared/ /SGTK_SGLI/	HDF5、IO ツールのインストール先が左記の場合の説明です。
インクルードパス	/HDF5/shared/include /SGTK_SGLI/include	Makefile に記述 例 -I/HDF5/shared/include -I/SGTK_SGLI/include
ライブラリパス	/HDF5/shared/lib /SGTK_SGLI/lib	Makefile に記述 例 -L/HDF5/shared/lib -L/SGTK_SGLI/lib
ライブラリ名	sgtk hdf5 hdf5_cpp hdf5_hl hdf5_hl_cpp	Makefile に記述 例 -lsgtk -lhdf5 -lhdf5_cpp -lhdf5_hl -lhdf5_hl_cpp
実行時に必要な 環境変数	LD_LIBRARY_PATH	ライブラリパスを追加 export LD_LIBRARY_PATH=/HDF5/shared/lib : /SGTK_SGLI/lib:\$LD_LIBRARY_PATH

※ライブラリ sgtk は同一環境で作成されたものを使用して下さい。

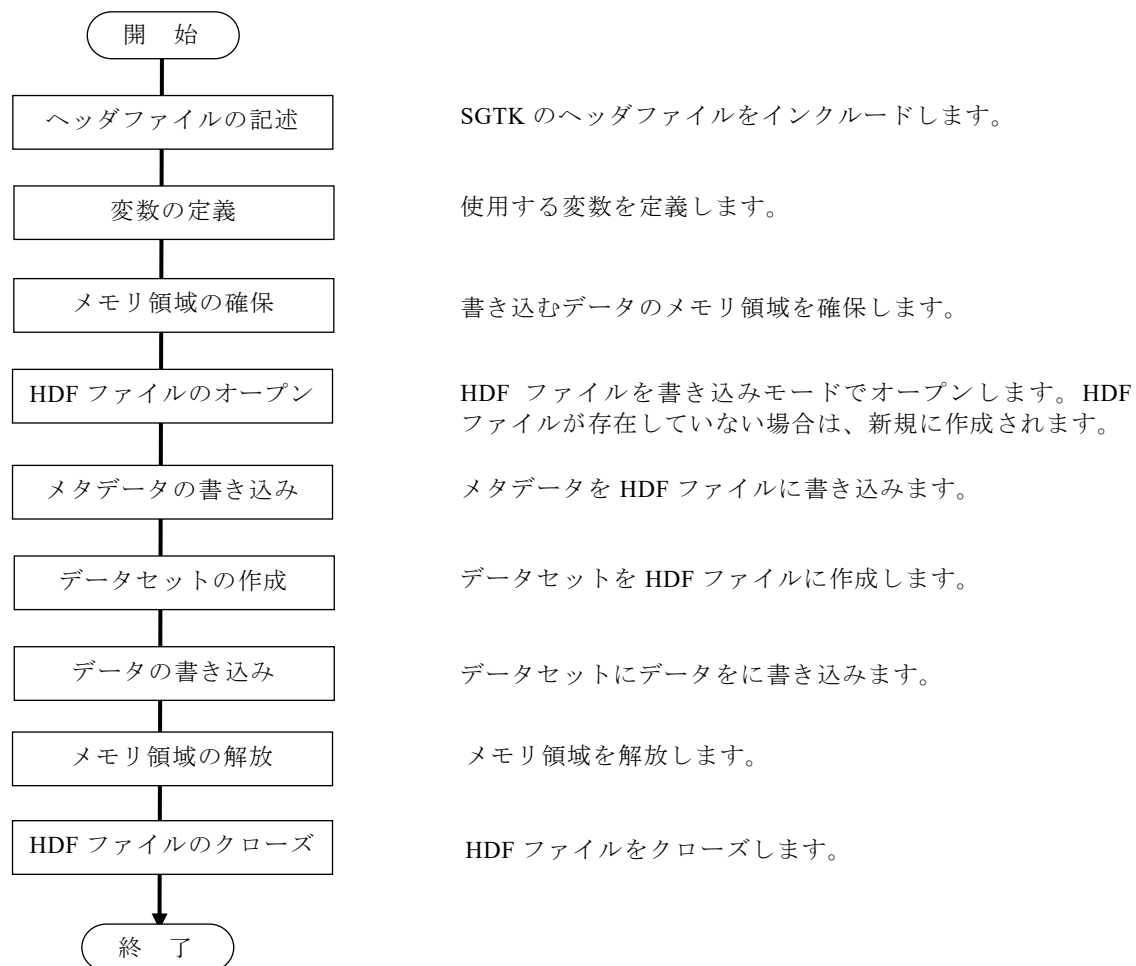
Windows環境での設定内容

項目	名称	説明
インストール先	C:\Make-hdf5-1.8.21 C:\SGTK_SGLI	
インクルードパス	C:\Make-hdf5-1.8.21\build C:\Make-hdf5-1.8.21\hdf5-1.8.21\src C:\Make-hdf5-1.8.21\hdf5-1.8.21\hl\src C:\Make-hdf5-1.8.21\hdf5-1.8.21\c++\src C:\SGTK_SGLI\include	プロパティページのインクルードディレクトリにセミコロン (;) で区切って追加。
ライブラリパス	C:\Make-hdf5-1.8.21\build\bin\Release C:\SGTK_SGLI\lib	プロパティページのライブラリディレクトリにセミコロン (;) で区切って追加。
ライブラリ名	sgtk.lib hdf5.lib hdf5_cpp.lib hdf5_hl.lib hdf5_hl_cpp.lib	プロパティページのリンカーの追加の依存ファイルに追加
実行時に必要な 環境変数	PATH Set PATH=%PATH%;C:\Make-hdf5-1.8.21\build\bin\Release;C:\SGTK_SGLI\lib	実行時に DLL を読み込むためライブラリパスを設定

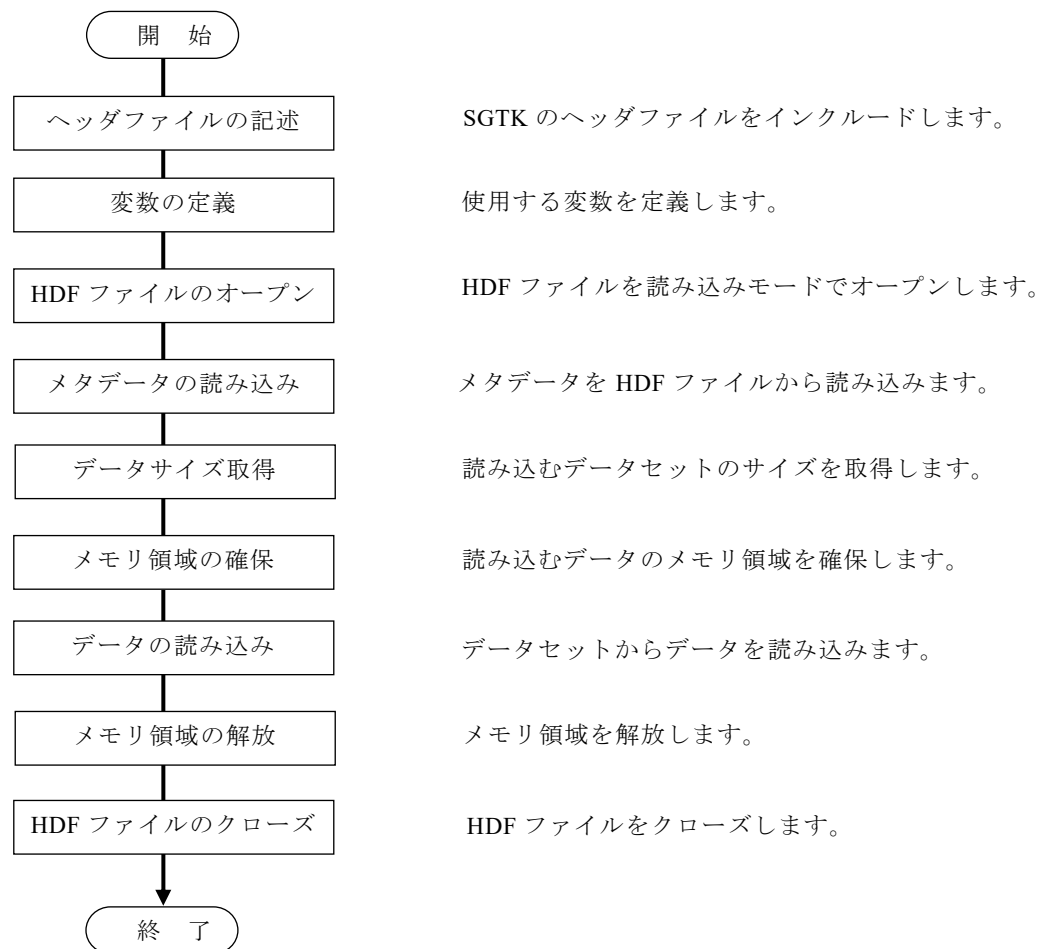
3.1 プログラミングの流れ

SGTKを使って、SGLIデータをCやFortranで読み書きする流れを以下に示します。

(1) HDF ファイルの新規作成



(2) HDF ファイルからデータの読み出し



4 機能構成

SGTK の機能構成図を

図 4-1に示します。

提供機能は、入力機能、出力機能で分けられます。

(1) 入力機能

(a) レベル 1 プロダクト (L1A、L1B) 入力機能

(b) レベル 2 プロダクト (L2) 入力機能

(c) レベル 3 プロダクト (L3) 入力機能

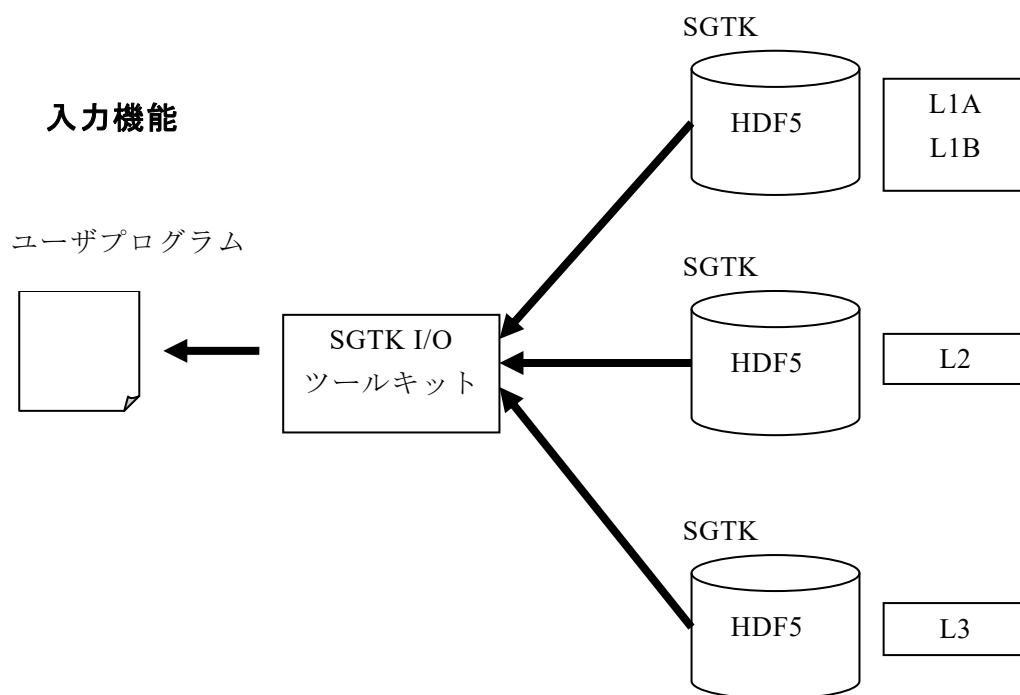


図 4-1 SGTK プロダクト入力機能

(2) 入力機能

- (a) レベル 1 プロダクト (L1A、L1B) 出力機能
- (b) レベル 2 プロダクト (L2) 出力機能
- (c) レベル 3 プロダクト (L3) 出力機能

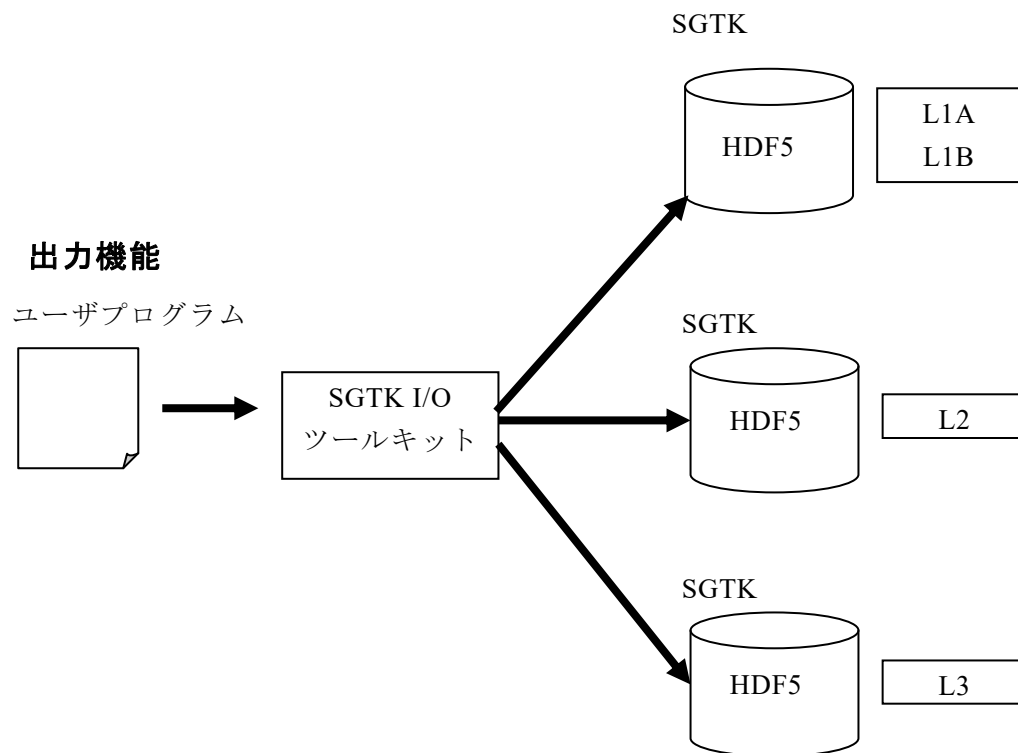


図 4-2 SGK プロダクト出力機能

5 SGTK の関数

5.1 C 言語

5.1.1 共通関数

(1) L1A、L1B、L2、L3 入力機能

アクセラブルが定義されたパラメータファイルの読み込み処理				
アクセラブル定義パラメータファイルを読み込み、パラメータテーブルを作成します。				
int SGTK_make_parameter_table (const std::string parameter_file)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:負の値が返ります。
パラメータ				
parameter_file	const std::string	input	1	パラメータファイル名

プロダクトファイルオープン(読み込み専用)				
HDFプロダクトファイルをオープンします。				
hid_t SGTK_open_file(char *file_name)				
名前	型	入出力区別	サイズ	説明
戻り値				
HDF_file_id	hid_t	output	1	正常時:HDF access file idが返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
file_name	char *	input	1	SGLI HDFファイル名

プロダクトファイルクローズ				
HDFプロダクトファイルをクローズします。				
int SGTK_close_file(hid_t HDF_file_id)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id

5.1.2 データ取得関数

(1) L1A、L1B、L2、L3 データ取得機能

データセットの次元情報取得				
プロダクトから次元情報を取得します。 - 複数のデータセットを対象とするアクセラベルが指定された場合は、エラーを返します。				
int SGTK_get_dims (hid_t HDF_file_id, char *label, int **num)				
名前	型	入出力区別	サイズ	説明
戻り値				
num	int	output	1	次元情報要素数
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	「7.1.1 データセットアクセラベル」を参照してください。
num	int **	output	N	次元情報（1～6次元の情報）

メタデータ取得(メタデータ名)(string 型)				
プロダクトのメタデータ名でメタデータ(string型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_string(hid_t HDF_file_id, char *label, char *meta, char *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	char *	output	1	メタデータ値

メタデータ取得(メタデータ名)(int 型)				
プロダクトのメタデータ名でメタデータ(int型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_int(hid_t HDF_file_id, char *label, char *meta, int *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	int *	output	1	メタデータ値

メタデータ取得(メタデータ名)(int8 型)				
プロダクトのメタデータ名でメタデータ(int8型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_int8(hid_t HDF_file_id, char *label, char *meta, int8 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	int8 *	output	1	メタデータ値

メタデータ取得(メタデータ名)(int16 型)				
プロダクトのメタデータ名でメタデータ(int16型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_int16(hid_t HDF_file_id, char *label, char *meta, int16 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	int16 *	output	1	メタデータ値

メタデータ取得(メタデータ名)(int32 型)				
プロダクトのメタデータ名でメタデータ(int32型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_int32(hid_t HDF_file_id, char *label, char *meta, int32 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	int32 *	output	1	メタデータ値

メタデータ取得(メタデータ名)(uint 型)				
プロダクトのメタデータ名でメタデータ(uint型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_uint(hid_t HDF_file_id, char *label, char *meta, uint *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	uint *	output	1	メタデータ値

メタデータ取得(メタデータ名)(uint8 型)				
プロダクトのメタデータ名でメタデータ(uint8型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_uint8(hid_t HDF_file_id, char *label, char *meta, uint8 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	uint8 *	output	1	メタデータ値

メタデータ取得(メタデータ名)(uint16 型)				
プロダクトのメタデータ名でメタデータ(uint16型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_uint16(hid_t HDF_file_id, char *label, char *meta, uint16 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	uint16 *	output	1	メタデータ値

メタデータ取得(メタデータ名)(uint32 型)				
プロダクトのメタデータ名でメタデータ(uint32型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_uint32(hid_t HDF_file_id, char *label, char *meta, uint32 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	uint32 *	output	1	メタデータ値

メタデータ取得(メタデータ名)(float 型)				
プロダクトのメタデータ名でメタデータ(float型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_float(hid_t HDF_file_id, char *label, char *meta, float *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	float *	output	1	メタデータ値

メタデータ取得(メタデータ名)(float32 型)				
プロダクトのメタデータ名でメタデータ(float32型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_float32(hid_t HDF_file_id, char *label, char *meta, float *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	float *	output	1	メタデータ値

メタデータ取得(メタデータ名)(float64)				
プロダクトのメタデータ名でメタデータ(float64型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
int SGTK_get_meta_float64(hid_t HDF_file_id, char *label, char *meta, double *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	double *	output	1	メタデータ値

整数型データ取得(int 型)				
HDFアクセラブルを指定してint型のデータセットの値を取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
Int SGK_get_dataset_int(hid_t HDF_file_id, char *label, int **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
get_value	int **	output	N	データを格納するメモリエリア

整数型データ取得(int8 型)				
HDFアクセラブルを指定してint8型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
Int SGK_get_dataset_int8(hid_t HDF_file_id, char *label, int8 **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
get_value	int8 **	output	N	データを格納するメモリエリア

整数型データ取得(int16 型)				
HDFアクセラブルを指定してint16型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
Int SGK_get_dataset_int16(hid_t HDF_file_id, char *label, int16 **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
get_value	int16 **	output	N	データを格納するメモリエリア

整数型データ取得(int32 型)				
HDFアクセラブルを指定してint32型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
Int SGK_get_dataset_int32(hid_t HDF_file_id, char *label, int32 **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
get_value	int32 **	output	N	データを格納するメモリエリア

整数型データ取得(uint 型)				
HDFアクセラブルを指定してuint型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
Int SGK_get_dataset_uint(hid_t HDF_file_id, char *label, uint **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
get_value	uint **	output	N	データを格納するメモリエリア

整数型データ取得(uint8 型)				
HDFアクセラブルを指定してuint8型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
Int SGK_get_dataset_uint8(hid_t HDF_file_id, char *label, uint8 **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
get_value	uint8 **	output	N	データを格納するメモリエリア

整数型データ取得(uint16 型)				
HDFアクセラブルを指定してuint16型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
Int SGK_get_dataset_uint16(hid_t HDF_file_id, char *label, uint16 **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
get_value	uint16 **	output	N	データを格納するメモリエリア

整数型データ取得(uint32 型)				
HDFアクセラブルを指定してuint32型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
Int SGK_get_dataset_uint32(hid_t HDF_file_id, char *label, uint32 **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
get_value	uint32 **	output	N	データを格納するメモリエリア

整数型データ取得(float 型)				
HDFアクセラブルを指定してfloat型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGK_get_dataset_float(hid_t HDF_file_id, char *label, int kind, float **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
kind	int	input	1	輝度(SGTK_KIND_RADIANCE=0)/ 反射率(SGTK_KIND_REFLECTANCE=1)/ その他(SGTK_KIND_OTHER=3) 識別
get_value	float **	output	N	データを格納するメモリエリア

整数型データ取得(float32 型)				
HDFアクセラブルを指定してfloat32型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGK_get_dataset_float32(hid_t HDF_file_id, char *label, int kind, float **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
kind	int	input	1	輝度(SGTK_KIND_RADIANCE=0)/ 反射率(SGTK_KIND_REFLECTANCE=1)/ その他(SGTK_KIND_OTHER=3) 識別
get_value	float **	output	N	データを格納するメモリエリア

整数型データ取得(float64 型)				
HDFアクセラブルを指定してfloat64型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGK_get_dataset_float64(hid_t HDF_file_id, char *label, int kind, double **get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
Kind	int	input	1	輝度(SGTK_KIND_RADIANCE=0)/ 反射率(SGTK_KIND_REFLECTANCE=1)/ その他(SGTK_KIND_OTHER=3) 識別
get_value	double **	output	N	データを格納するメモリエリア

緯度経度データ取得				
指定されたスキャン番号の範囲の緯度経度データを取得します。 - スキャン番号は、開始スキャン番号と終了スキャン番号を指定します。 終了スキャン番号が、開始スキャン番号より小さい場合は、エラーが返ります。 終了スキャン番号が指定されたHDF データセットに格納されているスキャン番号より大きい場合は、HDF データセットに格納されているデータまでを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_get_dataset_latlon(hid_t HDF_file_id, char *label_lat, char *label_lon, int kind, int start_no, int end_no, float **latlon)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label_lat	char *	input	1	データセットアクセスラベル(緯度)
label_lon	char *	input	1	データセットアクセスラベル(経度)
kind	int	input	1	輝度(SGTK_KIND_RADIANCE=0)/ 反射率(SGTK_KIND_REFLECTANCE=1)/ その他(SGTK_KIND_OTHER=3) 識別
start_no	int	input	1	取得開始スキャン番号
end_no	int	input	1	取得終了スキャン番号
latlon	float **	output	1	データを格納するメモリエリア

5.1.3 データ設定関数

(1) L1A、L1B、L2、L3 データ設定機能

HDF データセット作成				
プロダクトへデータディメンジョンサイズのHDF データセットを作成します。 ー 生成する際、HDF データセットを欠損値で初期化します。				
int SGTK_set_dims(int HDF_file_id, char *label, int *dims, int rank, int dat_type, char **dim_lbl)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:SUCCESS(0)が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char*	input	1	データセットアクセスラベル
dims	int*	input	1	次元要素数
rank	int	input	1	次元数
dat_type	int	input	1	HDF5 データタイプ
dim_lbl	char **	input	1	ディメンジョンラベル

メタデータ設定(string型)				
HDFプロダクトのメタデータ名でstring型のメタデータを設定します。				
int SGTK_set_meta_string(hid_t HDF_file_id, char *label, char *meta, char *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	char *	input	1	メタデータ値格納バッファアドレス

メタデータ設定(int型)				
HDFプロダクトのメタデータ名でint型のメタデータを設定します。				
int SGTK_set_meta_int(hid_t HDF_file_id, char *label, char *meta, int value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	int	input	1	設定値

メタデータ設定(int8型)				
HDFプロダクトのメタデータ名でint8型のメタデータを設定します。				
int SGK_set_meta_int8(hid_t HDF_file_id, char *label, char *meta, int8 value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	int8	input	1	設定値

メタデータ設定(int16型)				
HDFプロダクトのメタデータ名でint16型のメタデータを設定します。				
int SGK_set_meta_int16(hid_t HDF_file_id, char *label, char *meta, int16 value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	int16	input	1	設定値

メタデータ設定(int32型)				
HDFプロダクトのメタデータ名でint32型のメタデータを設定します。				
int SGK_set_meta_int32(hid_t HDF_file_id, char *label, char *meta, int32 value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	int32	input	1	設定値

メタデータ設定(uint型)				
HDFプロダクトのメタデータ名でuint型のメタデータを設定します。				
int SGK_set_meta_uint(hid_t HDF_file_id, char *label, char *meta, uint value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	uint	input	1	設定値

メタデータ設定(uint8型)				
HDFプロダクトのメタデータ名でint8型のメタデータを設定します。				
int SGK_set_meta_uint8(hid_t HDF_file_id, char *label, char *meta, uint8 value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	uint8	input	1	設定値

メタデータ設定(uint16型)				
HDFプロダクトのメタデータ名でuint16型のメタデータを設定します。				
int SGK_set_meta_uint16(hid_t HDF_file_id, char *label, char *meta, uint16 value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	uint16	input	1	設定値

メタデータ設定(uint32型)				
HDFプロダクトのメタデータ名でuint32型のメタデータを設定します。				
int SGK_set_meta_uint32(hid_t HDF_file_id, char *label, char *meta, uint32 value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	uint32	input	1	設定値

メタデータ設定(float型)				
HDFプロダクトのメタデータ名でfloat型のメタデータを設定します。				
int SGK_set_meta_float(hid_t HDF_file_id, char *label, char *meta, float value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	float	input	1	設定値

メタデータ設定(float32型)				
HDFプロダクトのメタデータ名でfloat32型のメタデータを設定します。				
int SGK_set_meta_float32(hid_t HDF_file_id, char *label, char *meta, float value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	float	input	1	設定値

メタデータ設定(float64型)				
HDFプロダクトのメタデータ名でfloat64型のメタデータを設定します。				
int SGTK_set_meta_float64(hid_t HDF_file_id, char *label, char *meta, double value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
meta	char *	input	1	メタデータ名
value	double	input	1	設定値

データセット設定(int型)				
HDFアクセスラベルを指定してint型のデータを設定します。				
- データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_int(hid_t HDF_file_id, char *label, int *dims, int rank, int *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	int *	input	1	データ格納バッファアドレス

データセット設定(int8型)				
HDFアクセスラベルを指定してint8型のデータを設定します。				
- データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_int8(hid_t HDF_file_id, char *label, int *dims, int rank, int8 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	int8 *	input	1	データ格納バッファアドレス

データセット設定(int16型)				
HDFアクセスラベルを指定してint16型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_int16(hid_t HDF_file_id, char *label, int *dims, int rank, int16 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	int16 *	input	1	データ格納バッファアドレス

データセット設定(int32型)				
HDFアクセスラベルを指定してint32型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_int32(hid_t HDF_file_id, char *label, int *dims, int rank, int32 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	int32 *	input	1	データ格納バッファアドレス

データセット設定(uint型)				
HDFアクセスラベルを指定してuint型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_uint(hid_t HDF_file_id, char *label, int *dims, int rank, uint *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	uint *	input	1	データ格納バッファアドレス

データセット設定(uint8型)				
HDFアクセスラベルを指定してuint8型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_uint8(hid_t HDF_file_id, char *label, int *dims, int rank, uint8 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	uint8 *	input	1	データ格納バッファアドレス

データセット設定(uint16型)				
HDFアクセスラベルを指定してuint16型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_uint16(hid_t HDF_file_id, char *label, int *dims, int rank, uint16 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	uint16 *	input	1	データ格納バッファアドレス

データセット設定(uint32型)				
HDFアクセスラベルを指定してuint32型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_uint32(hid_t HDF_file_id, char *label, int *dims, int rank, uint32 *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセスラベル
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	uint32 *	input	1	データ格納バッファアドレス

データセット設定(float型)				
HDFアクセラブルを指定してfloat型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGK_set_dataset_float(hid_t HDF_file_id, char *label, int kind, int *dims, int rank, float *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
kind	int	input	1	データ種別 輝度(SGTK_KIND_RADIANCE=0) 反射率(SGTK_KIND_REFLECTANCE=1) その他(SGTK_KIND_OTHER=3)
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	float *	input	1	データ格納バッファアドレス

データセット設定(float32型)				
HDFアクセラブルを指定してfloat32型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGK_set_dataset_float32(hid_t HDF_file_id, char *label, int kind, int *dims, int rank, float *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
kind	int	input	1	データ種別 輝度(SGTK_KIND_RADIANCE=0) 反射率(SGTK_KIND_REFLECTANCE=1) その他(SGTK_KIND_OTHER=3)
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	float *	input	1	データ格納バッファアドレス

データセット設定(float64型)				
HDFアクセラブルを指定してfloat64型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_float64(hid_t HDF_file_id, char *label, int kind, int *dims, int rank, double *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label	char *	input	1	データセットアクセラブル
kind	int	input	1	データ種別 輝度(SGTK_KIND_RADIANCE=0) 反射率(SGTK_KIND_REFLECTANCE=1) その他(SGTK_KIND_OTHER=3)
dims	int *	input	1	次元情報
rank	int	input	1	次元数
value	double *	input	1	データ格納バッファアドレス

緯度経度データ設定				
緯度経度データを指定されたスキャン番号に書き込みます。 - スキャン番号は、開始スキャン番号と終了スキャン番号を指定します。 終了スキャン番号が、開始スキャン番号より小さい場合は、エラーが返ります。 終了スキャン番号が指定されたHDF データセットに格納されているスキャン番号より大きい場合は、HDF データセットに格納されているデータまでを設定します。 - 設定するデータのメモリエリアは、関数を使用するユーザが確保します。				
int SGTK_set_dataset_latlon(hid_t HDF_file_id, char *label_lat, char *label_lon, int kind, int start_no, int end_no, int set_num, float *latlon)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
label_lat	char *	input	1	データセットアクセスラベル(緯度)
label_lon	char *	input	1	データセットアクセスラベル(経度)
kind	int	input	1	データ種別 輝度(SGTK_KIND_RADIANCE=0) 反射率(SGTK_KIND_REFLECTANCE=1) その他(SGTK_KIND_OTHER=3)
start_no	int	input	1	取得開始スキャン番号
end_no	int	input	1	取得終了スキャン番号
set_num	int	input	1	要素数(緯度の数×経度の数)
latlon	float **	input	1	データを格納するメモリエリア

5.1.4 その他の関数

グループ作成				
空のグループを作成します。				
int SGK_make_group(hid_t HDF_file_id, char *group)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
group	char *	input	1	グループ名

時刻変換(UTCからTAI93)				
UTCからTAI93に時刻を変換します。				
int SGK_utc_to_tai93(int year, int month, int day, int hour, int min, int sec, int msec, double *tai93time)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
year	int	input	1	年
month	int	input	1	月
day	int	input	1	日
hour	int	input	1	時
min	int	input	1	分
sec	int	input	1	秒
msec	int	input	1	ミリ秒
tai93time	double *	output	1	TAI93時間

時刻変換(TAI93からUTC)				
TAI93からUTCに時刻を変換します。				
int SGK_tai93_to_utc(double tai93time, int *year, int *month, int *day, int *hour, int *min, int *sec, int *msec)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
tai93time	double	input	1	TAI93時刻
year	int *	output	1	年
month	int *	output	1	月
day	int *	output	1	日
hour	int *	output	1	時
min	int *	output	1	分
sec	int *	output	1	秒
msec	int *	output	1	ミリ秒

メモリ確保				
指定された次元情報からメモリを確保します。				
void *SGTK_malloc(int *dims, int rank, int size)				
名前	型	入出力区別	サイズ	説明
戻り値				
void*	void *	output	1	確保したメモリアドレス
パラメータ				
dims	int *	input	1	次元情報
rank	int	input	1	次元数
size	int	input	1	次元情報サイズ

メモリ解放				
指定されたアドレスのメモリを解放します。				
int SGK_free(void *buf)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
buf	void *	input	1	データ格納領域のバッファアドレス

緯度経度情報取得				
指定されたグラニューールIDから緯度経度を取得し、バッファに格納します。				
int GTK_get_global_latlon (hid_t HDF_file_id, float *latlon)				
名前	型	入出力区別	サイズ	説明
戻り値				
data_num	int	output	1	正常時:取得要件件数(緯度数×経度数) 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	hid_t	input	1	HDFアクセスfile id
latlon	float	output	1	緯度経度情報

設定ファイル反映				
指定されたGTK設定ファイルの内容を反映します。				
int GTK_load_config(char *file_name)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
file_name	char *	input	1	GTK設定ファイルパス

5.2 Fortran 言語

5.2.1 共通関数

(1) L1A、L1B、L2、L3 入力機能

アクセラブルが定義されたパラメータファイルの読み込み処理				
アクセラブル定義パラメータファイルを読み込み、パラメータテーブルを作成します。				
INTEGER SGTK_make_parameter_table (parameter_file)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:負の値が返ります。
パラメータ				
parameter_file	CHARACTER	input	1	パラメータファイル名

プロダクトファイルオープン(読み込み専用)				
HDFプロダクトファイルをオープンします。				
INTEGER SGTK_open_file(file_name)				
名前	型	入出力区別	サイズ	説明
戻り値				
HDF_file_id	INTEGER	output	1	正常時:HDF access file idが返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
file_name	CHARACTER	input	N	SGLI HDFファイル名

プロダクトファイルクローズ				
HDFプロダクトファイルをクローズします。				
INTEGER SGTK_close_file(HDF_file_id)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id

5.2.2 データ取得関数

(1) L1A、L1B、L2、L3 データ取得機能

データセットの次元情報取得				
プロダクトから次元情報を取得します。 ー 複数のデータセットを対象とするアクセララベルが指定された場合は、エラーを返します。				
INTEGER SGKTK_get_dims(HDF_file_id, label, size, num)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:配列次元数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	「7.1.1 データセットアクセララベル」を参照してください。
size	INTEGER	output	1	次元情報サイズ (1～6次元の情報)
num	INTEGER	output	1	次元情報 (1～6次元の情報)

メタデータ取得(メタデータ名)(CHARACTER 型)				
プロダクトのメタデータ名でメタデータ(CHARACTER型)を取得します。 ー メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGKTK_get_meta_string(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセララベル
meta	CHARACTER	input	N	メタデータ名
value	CHARACTER	output	N	メタデータ値

メタデータ取得(メタデータ名)(int 型)				
プロダクトのメタデータ名でメタデータ(int型)を取得します。 ー メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGKTK_get_meta_int(HDF_file_id, label, meta, p_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセララベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	output	1	メタデータ値

メタデータ取得(メタデータ名)(int8 型)				
プロダクトのメタデータ名でメタデータ(INTEGER型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_int8(HDF_file_id, label, meta, p_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
Status	int	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	output	1	メタデータ値

メタデータ取得(メタデータ名)(int16 型)				
プロダクトのメタデータ名でメタデータ(INTEGER型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_int16(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	output	2	メタデータ値

メタデータ取得(メタデータ名)(int32 型)				
プロダクトのメタデータ名でメタデータ(INTEGER型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_int32(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	output	4	メタデータ値

メタデータ取得(メタデータ名)(uint 型)				
プロダクトのメタデータ名でメタデータ(INTEGER型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_uint(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	int	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	output	1	メタデータ値

メタデータ取得(メタデータ名)(uint8 型)				
プロダクトのメタデータ名でメタデータ(INTEGER型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_uint8(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	output	1	メタデータ値

メタデータ取得(メタデータ名)(uint16 型)				
プロダクトのメタデータ名でメタデータ(INTEGER型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_uint16(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	output	2	メタデータ値

メタデータ取得(メタデータ名)(uint32 型)				
プロダクトのメタデータ名でメタデータ(INTEGER型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_uint32(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	output	4	メタデータ値

メタデータ取得(メタデータ名)(float 型)				
プロダクトのメタデータ名でメタデータ(REAL型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_float(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	REAL	output	1	メタデータ値

メタデータ取得(メタデータ名)(float32 型)				
プロダクトのメタデータ名でメタデータ(REAL型)を取得します。 — メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_float32(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	REAL	output	4	メタデータ値

メタデータ取得(メタデータ名)(float64 型)				
プロダクトのメタデータ名でメタデータ(REAL型)を取得します。 - メタデータ値を設定するメモリ領域がNULLの場合は、エラーを返します。				
INTEGER SGTK_get_meta_float64(int *HDF_file_id, char *label, char *meta, double *value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:メタデータ文字数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	REAL	output	8	メタデータ値

整数型データ取得(int 型)				
HDFアクセスラベルを指定してINTEGER型のデータセットの値を取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_get_dataset_int(HDF_file_id, label, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
get_value	INTEGER	output	1	データを格納するメモリエリア

整数型データ取得(int8 型)				
HDFアクセスラベルを指定してINTEGER型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_get_dataset_int8(HDF_file_id, label, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
get_value	INTEGER	output	1	データを格納するメモリエリア

整数型データ取得(int16 型)				
HDFアクセラブルを指定してINTEGER型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_get_dataset_int16(HDF_file_id, label, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセラブル
get_value	INTEGER	output	2	データを格納するメモリエリア

整数型データ取得(int32 型)				
HDFアクセラブルを指定してINTEGER型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_get_dataset_int32(HDF_file_id, label, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセラブル
get_value	INTEGER	output	4	データを格納するメモリエリア

整数型データ取得(uint 型)				
HDFアクセラブルを指定してINTEGER型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_get_dataset_uint(HDF_file_id, label, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセラブル
get_value	INTEGER	output	1	データを格納するメモリエリア

整数型データ取得(uint8 型)				
HDFアクセラブルを指定してINTEGER型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_get_dataset_uint8(HDF_file_id, label, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセラブル
get_value	INTEGER	output	1	データを格納するメモリエリア

整数型データ取得(uint16 型)				
HDFアクセラブルを指定してINTEGER型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_get_dataset_uint16(HDF_file_id, label, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセラブル
get_value	INTEGER	output	2	データを格納するメモリエリア

整数型データ取得(uint32 型)				
HDFアクセラブルを指定してINTEGER型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_get_dataset_uint32(HDF_file_id, label, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセラブル
get_value	INTEGER	output	4	データを格納するメモリエリア

整数型データ取得(float 型)				
HDFアクセラブルを指定してREAL型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGK_get_dataset_float(HDF_file_id, label, kind, get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセラブル
kind	INTEGER	input	1	データ種別 輝度(SGTK_KIND_RADIANCE=0) 反射率(SGTK_KIND_REFLECTANCE=1) その他(SGTK_KIND_OTHER=3)
get_value	REAL	output	1	データを格納するメモリエリア

整数型データ取得(float32 型)				
HDFアクセラブルを指定してREAL型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGK_get_dataset_float32(HDF_file_id, label, p_kind, p_get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセラブル
p_kind	INTEGER	input	1	データ種別 輝度(SGTK_KIND_RADIANCE=0) 反射率(SGTK_KIND_REFLECTANCE=1) その他(SGTK_KIND_OTHER=3)
get_value	REAL	output	4	データを格納するメモリエリア

整数型データ取得(float64 型)				
HDFアクセスラベルを指定してREAL型のデータを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGK_get_dataset_float64(HDF_file_id, label, p_kind, p_get_value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
p_kind	INTEGER	input	1	データ種別 輝度(SGTK_KIND_RADIANCE=0) 反射率(SGTK_KIND_REFLECTANCE=1) その他(SGTK_KIND_OTHER=3)
p_get_value	REAL	output	8	データを格納するメモリエリア

緯度経度データ取得				
指定されたスキャン番号の範囲の緯度経度データを取得します。 - スキャン番号は、開始スキャン番号と終了スキャン番号を指定します。 終了スキャン番号が、開始スキャン番号より小さい場合は、エラーが返ります。 終了スキャン番号が指定されたHDF データセットに格納されているスキャン番号より大きい場合は、HDF データセットに格納されているデータまでを取得します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGK_get_dataset_latlon(HDF_file_id, latitude, longitude, kind, start_no, end_no, latlon)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
latitude	CHARACTER	input	N	データセットアクセスラベル(緯度)
Longitude	CHARACTER	input	N	データセットアクセスラベル(経度)
kind	INTEGER	input	1	データ種別 輝度(SGTK_KIND_RADIANCE=0) 反射率(SGTK_KIND_REFLECTANCE=1) その他(SGTK_KIND_OTHER=3)
start_no	INTEGER	input	1	取得開始スキャン番号
end_no	INTEGER	input	1	取得終了スキャン番号
latlon	REAL	output	1	データを格納するメモリエリア

5.2.3 データ設定関数

(1) L1A、L1B、L2、L3 データ設定機能

HDF データセット作成				
プロダクトヘデータディメンジョンサイズのHDF データセットを作成します。 ー 生成する際、HDF データセットを欠損値で初期化します。				
INTEGER SGTK_set_dims(HDF_file_id, label, p_dims, rank, dat_type, dim_lbl)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元要素数
rank	INTEGER	input	1	次元数
dat_type	INTEGER	input	1	HDF5 データタイプ
dim_lbl	CHARACTER	input	N	ディメンジョンラベル

メタデータ設定(string型)				
HDFプロダクトのメタデータ名でCHARACTER型のメタデータを設定します。				
INTEGER SGTK_set_meta_string(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
Label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	CHARACTER	input	N	メタデータ値格納バッファアドレス

メタデータ設定(int型)				
HDFプロダクトのメタデータ名でINTEGER型のメタデータを設定します。				
INTEGER SGTK_set_meta_int(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	input	1	メタデータ値格納バッファアドレス

メタデータ設定(int8型)				
HDFプロダクトのメタデータ名でINTEGER型のメタデータを設定します。				
INTEGER SGTK_set_meta_int8(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	input	1	メタデータ値格納バッファアドレス

メタデータ設定(int16型)				
HDFプロダクトのメタデータ名でINTEGER型のメタデータを設定します。				
INTEGER SGTK_set_meta_int16(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	input	2	メタデータ値格納バッファアドレス

メタデータ設定(int32型)				
HDFプロダクトのメタデータ名でINTEGER型のメタデータを設定します。				
INTEGER SGTK_set_meta_int32(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	Output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	Input	1	HDFアクセスfile id
label	CHARACTER	Input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	input	4	メタデータ値格納バッファアドレス

メタデータ設定(uint型)				
HDFプロダクトのメタデータ名でINTEGER型のメタデータを設定します。				
INTEGER SGTK_set_meta_uint(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	input	1	メタデータ値格納バッファアドレス

メタデータ設定(uint8型)				
HDFプロダクトのメタデータ名でINTEGER型のメタデータを設定します。				
INTEGER SGTK_set_meta_uint8(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	input	1	メタデータ値格納バッファアドレス

メタデータ設定(uint16型)				
HDFプロダクトのメタデータ名でINTEGER型のメタデータを設定します。				
INTEGER SGTK_set_meta_uint16(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	input	2	メタデータ値格納バッファアドレス

メタデータ設定(uint32型)				
HDFプロダクトのメタデータ名でINTEGER型のメタデータを設定します。				
INTEGER SGK_set_meta_uint32(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	INTEGER	input	1	メタデータ値格納バッファアドレス

メタデータ設定(float型)				
HDFプロダクトのメタデータ名でREAL型のメタデータを設定します。				
INTEGER SGK_set_meta_float(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	REAL	input	1	メタデータ値格納バッファアドレス

メタデータ設定(float32型)				
HDFプロダクトのメタデータ名でREAL型のメタデータを設定します。				
INTEGER SGK_set_meta_float32(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	REAL	input	4	メタデータ値格納バッファアドレス

メタデータ設定(float64型)				
HDFプロダクトのメタデータ名でREAL型のメタデータを設定します。				
INTEGER SFTK_set_meta_float64(HDF_file_id, label, meta, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
meta	CHARACTER	input	N	メタデータ名
value	REAL	input	8	メタデータ値格納バッファアドレス

データセット設定(int型)				
HDFアクセスラベルを指定してINTEGER型のデータを設定します。				
- データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_int(HDF_file_id, label, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	INTEGER	input	1	データ格納バッファアドレス

データセット設定(int8型)				
HDFアクセスラベルを指定してINTEGER型のデータを設定します。				
- データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_int8(HDF_file_id, label, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	INTEGER	input	1	データ格納バッファアドレス

データセット設定(int16型)				
HDFアクセスラベルを指定してINTEGER型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_int16(HDF_file_id, label, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	INTEGER	input	2	データ格納バッファアドレス

データセット設定(int32型)				
HDFアクセスラベルを指定してINTEGER型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_int32(HDF_file_id, label, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	INTEGER	input	4	データ格納バッファアドレス

データセット設定(uint型)				
HDFアクセスラベルを指定してINTEGER型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_uint(HDF_file_id, label, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	INTEGER	input	1	データ格納バッファアドレス

データセット設定(uint8型)				
HDFアクセスラベルを指定してINTEGER型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_uint8(HDF_file_id, label, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	INTEGER	input	1	データ格納バッファアドレス

データセット設定(uint16型)				
HDFアクセスラベルを指定してINTEGER型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_uint16(HDF_file_id, label, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	INTEGER	input	2	データ格納バッファアドレス

データセット設定(uint32型)				
HDFアクセスラベルを指定してINTEGER型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_uint32(HDF_file_id, label, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	INTEGER	input	4	データ格納バッファアドレス

データセット設定(float型)				
HDFアクセスラベルを指定してREAL型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_float(HDF_file_id, label, kind, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	1	データセットアクセスラベル
kind	INTEGER	input	1	データ種別(radiance/reflectance/other)
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	REAL	input	1	データ格納バッファアドレス

データセット設定(float32型)				
HDFアクセスラベルを指定してfloat32型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_float32(HDF_file_id, label, kind, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
kind	INTEGER	input	1	データ種別(radiance/reflectance/other)
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	REAL	input	4	データ格納バッファアドレス

データセット設定(float64型)				
HDFアクセスラベルを指定してfloat64型のデータを設定します。 - データを格納するメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGTK_set_dataset_float64(HDF_file_id, label, kind, dims, rank, value)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label	CHARACTER	input	N	データセットアクセスラベル
kind	INTEGER	input	1	データ種別(radiance/reflectance/other)
dims	INTEGER	input	1	次元情報
rank	INTEGER	input	1	次元数
value	REAL	input	8	データ格納バッファアドレス

緯度経度データ設定				
緯度経度データを指定されたスキャン番号に書き込みます。 - スキャン番号は、開始スキャン番号と終了スキャン番号を指定します。 終了スキャン番号が、開始スキャン番号より小さい場合は、エラーが返ります。 終了スキャン番号が指定されたHDF データセットに格納されているスキャン番号より大きい場合は、HDF データセットに格納されているデータまでを設定します。 - 設定するデータのメモリエリアは、関数を使用するユーザが確保します。				
INTEGER SGK_set_dataset_latlon(HDF_file_id, label_lat, label_lon, kind, start_no, end_no, set_num, latlon)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:取得スキャン数が返ります。 異常時:取得できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
label_lat	CHARACTER	input	N	データセットアクセスラベル(緯度)
label_lon	CHARACTER	input	N	データセットアクセスラベル(経度)
kind	INTEGER	input	1	輝度(SGTK_KIND_RADIANCE=0)/反射率(SGTK_KIND_REFLECTANCE=1)/その他(SGTK_KIND_OTHER=3) 識別
start_no	INTEGER	input	1	取得開始スキャン番号
end_no	INTEGER	input	1	取得終了スキャン番号
set_num	INTEGER	input	1	要素数(緯度の数×経度の数)
latlon	REAL	input	1	データを格納するメモリエリア

5.2.4 その他の関数

グループ作成				
空のグループを作成します。				
INTEGER SGK_make_group(HDF_file_id, group)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	input	1	HDFアクセスfile id
group	CHARACTER	input	N	グループ名

時刻変換(UTCからTAI93)				
UTCからTAI93に時刻を変換します。				
INTEGER SGK_utc_to_tai93(year, month, day, hour, min, sec, msec, tai93time)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
year	INTEGER	input	1	年
month	INTEGER	input	1	月
day	INTEGER	input	1	日
hour	INTEGER	input	1	時
min	INTEGER	input	1	分
sec	INTEGER	input	1	秒
msec	INTEGER	input	1	ミリ秒
tai93time	REAL	output	8	TAI93時間

時刻変換(TAI93からUTC)				
TAI93からUTCに時刻を変換します。				
INTEGER SGK_tai93_to_utc(tai93time, year, month, day, hour, min, sec, msec)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	Output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
tai93time	REAL	Input	8	TAI93時刻
year	INTEGER	output	1	年
month	INTEGER	output	1	月
day	INTEGER	output	1	日
hour	INTEGER	output	1	時
min	INTEGER	output	1	分
sec	INTEGER	output	1	秒
msec	INTEGER	output	1	ミリ秒

緯度経度情報取得				
指定されたグラニューIDから緯度経度を取得し、バッファに格納します。				
INTEGER SGK_get_global_latlon(HDF_file_id, latlon)				
名前	型	入出力区別	サイズ	説明
戻り値				
data_num	INTEGER	output	1	正常時:取得要件件数(緯度数×経度数) 異常時:設定できない場合は、負の値が返ります。
パラメータ				
HDF_file_id	INTEGER	Input	1	HDFアクセスfile id
latlon	REAL	output	4	緯度経度情報

設定ファイル反映				
指定されたSGTK設定ファイルの内容を反映します。				
INTEGER SGK_load_config(file_name)				
名前	型	入出力区別	サイズ	説明
戻り値				
status	INTEGER	output	1	正常時:0 異常時:設定できない場合は、負の値が返ります。
パラメータ				
file_name	CHARACTER	input	N	SGTK設定ファイルパス

6 ファイル定義

6.1 設定ファイル

SGTKの動作を定義するための設定ファイルです。デフォルトの設定ファイルはSGTKに添付されていますが、内容は任意で変更することが可能です。デフォルトの設定から変更する場合には、設定ファイル反映関数（5.1.4または5.2.4 参照）を使用して、SGTKに変更の内容を反映させる必要があります。

形式はテキストファイルであり、文字コード：ASCII, 改行コード：UNIX (LF) としています。

No.	キー（物理名）	内容	値
1	DATATYPE_COMPRESS_TYPE	プロダクトの圧縮種別	空または gzip とする。 デフォルト： gzip
2	DATATYPE_COMPRESS_LEVEL	プロダクトの圧縮レベル	レベル 1～9 までの整数値。圧縮種別が空でないときに有効となる。 デフォルト：4
3	DATATYPE_BYTE_ORDER	プロダクトのバイトオーダー	LE または BE とする。 デフォルト：LE
4	DATATYPE_CHUNK	プロダクトのチャンクサイズ	1 以上の整数値。 デフォルト：256
5	LEAP_SECOND_FILE_PATH	うるう秒ファイルのパス	デフォルト：.././params/leapsec.dat (サンプルプログラムからの相対パスを記述)
6	CORRECT_SLOPE_ATTRIBUTE_NAME	メタデータによる補正機能の Slope を示す HDF5 Attribute 名	複数の場合は“,”で区切る。デフォルト：Slope
7	CORRECT_OFFSET_ATTRIBUTE_NAME	メタデータによる補正機能の Offset を示す HDF5 Attribute 名	複数の場合は“,”で区切る。デフォルト：“Offset
8	CORRECT_BASE_ATTRIBUTE_NAME	メタデータによる補正機能の Base を示す HDF5 Attribute 名	複数の場合は“,”で区切る。デフォルト：Base
9	CORRECT_SLOPE_REF_ATTRIBUTE_NAME	メタデータによる補正機能の Slope を示す HDF5 Attribute 名	複数の場合は“,”で区切る。 デフォルト：Slope_reflectance
10	CORRECT_OFFSET_REF_ATTRIBUTE_NAME	メタデータによる補正機能の Offset を示す HDF5 Attribute 名	複数の場合は“,”で区切る。 デフォルト：Offset_reflectance
11	RANGE_MIN_ATTRIBUTE_NAME	メタデータによる範囲チェック機能の最小 値を示す HDF5 Attribute 名	複数の場合は“,”で区切る。 デフォルト：Minimum_valid_DN
12	RANGE_MAX_ATTRIBUTE_NAME	メタデータによる範囲チェック機能の最大 値を示す HDF5 Attribute 名	複数の場合は“,”で区切る。 デフォルト：Maximum_valid_DN
13	RANGE_ERROR_VALUE	メタデータによる範囲チェック機能の異常 値を示す HDF5 Attribute 名	複数の場合は“,”で区切る。 デフォルト：Error_DN

6.2 うるう秒ファイル

IERS (International Earth Rotation and Reference Systems Service) が公開する、TAI-UTCを定義したうるう秒ファイルです。うるう秒ファイルのファイルパスは設定ファイル (6.1 参照) で変更することができます。

①提供元 : IERS (<ftp://maia.usno.navy.mil/ser7/leapsec.dat>)

URIは2013年6月時点のものです。

②サンプル

```
Checked(unchanged): 2013-06-04T22:26:05Z using USNO tai-utc.dat file of Jun 4 18:26
1961 JAN 1 =JD 2437300.5 TAI-UTC= 1.4228180 S + (MJD - 37300.) X 0.0012960 S ACTUAL
1961 AUG 1 =JD 2437512.5 TAI-UTC= 1.3728180 S + (MJD - 37300.) X 0.0012960 S ACTUAL
...
2006 JAN 1 =JD 2453736.5 TAI-UTC= 33.0000000 S + (MJD - 41317.) X 0.0000000 S ACTUAL
2009 JAN 1 =JD 2454832.5 TAI-UTC= 34.0000000 S + (MJD - 41317.) X 0.0000000 S ACTUAL
2012 JUL 1 =JD 2456109.5 TAI-UTC= 35.0000000 S + (MJD - 41317.) X 0.0000000 S ACTUAL
```

③フォーマット

- File format : Text
- Encoding : ASCII
- Return code : LF(UNIX)
- ヘッダ部 (1行目) :

```
Checked(unchanged): 2013-06-04T22:26:05Z using USNO tai-utc.dat file of Jun 4 18:26
```

- データ部 (2行目以降) :

```
2009 JAN 1 =JD 2454832.5 TAI-UTC=34.0000000 S + (MJD - 41317.) X 0.0000000 S ACTUAL
          *1          *2          *3
```

④使用方法

- ヘッダ部は更新情報であり、参照する必要はありません。データ部の *n を使用します。
- 1971年以前は、JD(*1)以降にTAI-UTC(*2 + *3)が挿入されます。
1971年以前のうるう秒を求める場合は、(*3)の加算が必要になります。
 - JD:2437300.5 (UTC:1961-01-01T00:00:00) 以降 => うるう秒:1.422818
 - JD:2437512.5 (UTC:1961-08-01T00:00:00) 以降 => うるう秒:1.69757
- 1972年以降は、JD(*1)以降にTAI-UTC(*2)が挿入されます。
1972年以降のうるう秒を求める場合は、(*3)は使用しません。
 - JD:2454832.5 (UTC:2009-01-01T00:00:00) 以降 => うるう秒:34
 - JD:2456109.5 (UTC:2012-07-01T00:00:00) 以降 => うるう秒:35

6.3 アクセラベルパラメータファイル

ユーザがプロダクトのグループ・データセット・アトリビュート領域にアクセスするために使用するデータセットアクセラベル (7.1.1 参照) を定義したパラメータファイルです。アクセラベルパラメータファイルはプロダクトファイル単位で作成します。SGLIプロダクトを読み込む際には、プロダクトに対応するパラメータファイルを読み込み処理関数 (5.1.1または5.2.1 参照) で設定する必要があります。

アクセラベルパラメータファイルの記述は、

「データセットアクセラベル」 = 「HDFパス」 のフォーマットで記載します。

アクセラベルパラメータファイルのフォーマットのサンプルを図 6-1に示します。

```
↓
/** Group Access Labels.↓
*/↓
GC1_GRP_RESERVED=/Reserved↓
GC1_GRP_RAW_DATA_AUX_PACKET=/Raw_data/AUX_packet↓
GC1_GRP_RAW_DATA=/Raw_data↓
↓
/** Dataset Access Labels.↓
*/↓
GC1_DTS_RESERVED_UNUSED_PACKET_RIGHT=/Reserved/Unused_packet_right↓
GC1_DTS_RESERVED_UNUSED_PACKET_NADIR=/Reserved/Unused_packet_nadir↓
GC1_DTS_RESERVED_UNUSED_PACKET_LEFT=/Reserved/Unused_packet_left↓
GC1_DTS_RAW_DATA_VN11=/Raw_data/VN11↓
GC1_DTS_RAW_DATA_VN10=/Raw_data/VN10↓
GC1_DTS_RAW_DATA_VN09=/Raw_data/VN09↓
GC1_DTS_RAW_DATA_VN08=/Raw_data/VN08↓
GC1_DTS_RAW_DATA_VN07=/Raw_data/VN07↓
GC1_DTS_RAW_DATA_VN06=/Raw_data/VN06↓
GC1_DTS_RAW_DATA_VN05=/Raw_data/VN05↓
GC1_DTS_RAW_DATA_VN04=/Raw_data/VN04↓
GC1_DTS_RAW_DATA_VN03=/Raw_data/VN03↓
GC1_DTS_RAW_DATA_VN02=/Raw_data/VN02↓
GC1_DTS_RAW_DATA_VN01=/Raw_data/VN01↓
↓
/** Attribute Access Labels.↓
*/↓
GC1_ATR_PRODUCT_FILE_NAME=Product_file_name↓
GC1_ATR_WORST_ORBIT_SOURCE_DATA_DESCRIPTION=Worst_orbit_source_data_description↓
GC1_ATR_WORST_ORBIT_SOURCE=Worst_orbit_source↓
GC1_ATR_WORST_ATTITUDE_SOURCE_DATA_DESCRIPTION=Worst_attitude_source_data_description↓
GC1_ATR_WORST_ATTITUDE_SOURCE=Worst_attitude_source↓
GC1_ATR_UPPER_RIGHT_LONGITUDE=Upper_right_longitude↓
GC1_ATR_UPPER_RIGHT_LATITUDE=Upper_right_latitude↓
GC1_ATR_UPPER_LEFT_LONGITUDE=Upper_left_longitude↓
GC1_ATR_UPPER_LEFT_LATITUDE=Upper_left_latitude↓
[EOF]
```

図 6-1 アクセラベルパラメータファイルのフォーマット

アクセラベルパラメータファイル格納場所は下記の通りです。

SGTK_SGLI¥sample¥bin¥L1	: L1用アクセラベルパラメータファイル格納場所
SGTK_SGLI¥sample¥bin¥L2	: L2用アクセラベルパラメータファイル格納場所
SGTK_SGLI¥sample¥bin¥L3	: L3用アクセラベルパラメータファイル格納場所

7 入出力データ

7.1 データ定義

7.1.1 データセットアクセスラベル

SGTK は、SGLI レベル 1～レベル 3 プロダクトに対して入出力を行います。グループ・データセット・アトリビュートの入出力機能に関してはレベル 1～3 で共通であり、SGTK としてはその差異を意識しない設計として、全てのプロダクトはグループ、データセット、アトリビュートの集合という意味で等価であると見なします。

従って、ユーザプログラムにおいてグループ・データセット・アトリビュートを入出力するとき、入出力関数にレベル 1～3 の各要素を識別するデータセットアクセスラベルを指定しますが、これはユーザプログラムにて任意で指定するものであり、SGTK としてはレベル別の処理を行うことはありません。

SGLI プロダクトのグループ・データセット・アトリビュートの仕様が変更される場合は、単純にデータセットアクセスラベルを変更することで対応可能です。

データセットアクセスラベルはアクセスラベルパラメータファイル (6.3 参照) 内にキー&バリューとして記載しています。

デフォルトの設定ファイルは SGTK に添付されていますが、内容は任意で変更することが可能です。

8 エラー番号

SGTKの関数で異常時のエラー番号を表 8-1に示します。

番号は、マイナス値で示し、以下のカテゴリで番号が付与されています。

-100番台	プログラム実行環境のエラー
-200番台	SGTKを使用するユーザプログラムバグに起因するエラー
-500番台	HDFファイルに関するエラー
-600番台	設定データに関するエラー

表 8-1 エラー番号一覧

番号	説明	備考
-100	うるう秒ファイルなどのファイルのオープンができません。	
-101	メモリ領域の確保ができません。	
-102	環境変数の取得ができません。	
-103	うるう秒の取得ができません。	
-109	物理量定義ファイルを取得できません。ファイルが存在しない、もしくはファイルの定義内容が誤っています。	
-201	スキャン時刻の指定年が 1993 年以前です。	
-202	スキャン時刻の指定月が範囲外です。	月の指定は、1 から 12 です。
-203	スキャン時刻の指定日が範囲外です。	日の指定は、1 から 31 です。
-204	スキャン時刻の指定時が範囲外です。	時の指定は、1 から 24 です。
-205	スキャン時刻の指定分が範囲外です。	分の指定は、0 から 59 です。
-206	スキャン時刻の指定秒が範囲外です。	秒の指定は、0 から 60 です。
-207	スキャン時刻の指定ミリ秒が範囲外です。	ミリ秒の指定は、0 から 999 です。
-208	スキャン時刻の指定が範囲外です。	
-209	指定された通算秒がマイナス値です。	
-210	ファイル名の指定がありませんでした。	
-211	終了スキャン番号より開始スキャン番号の方が大きな値です。	
-213	HDF ファイル ID が 0 かマイナス値です。	
-214	メタデータのインデックス値がマイナス値です。	
-215	指定されたディメンジョンサイズが不正です。	
-216	書き込みデータの指定がありません。	
-217	アクセスラベルが 0 で指定されました。	
-218	指定アクセスラベルが書込対象外のものです。	
-219	指定アクセスラベルが読込対象外です。	
-220	メモリ領域を取得しようとした際にデータサイズが 0 です。	
-221	パラメータで指定されたデータのアドレスが NULL です。	
-238	HDF ファイル書き込みモードの指定が誤っています。	
-239	アクセスラベルと対応する関数のデータ型が異なっています。	
-501	HDF ファイルオープンができません。	

番号	説明	備考
-502	HDF ファイルクローズができません。	
-503	ディメンジョンサイズの設定ができません。	
-504	データセットアクセスに失敗しました。	
-505	データセットのオープンができません。	
-506	データセットのスペース取得ができません。	
-507	データセット配列次元数の取得ができません。	
-508	データセットのディメンジョンサイズ取得ができません。	
-509	データセットの作成ができません。	
-510	データセットへ書き込みができません。	
-511	データセットの読み込みができません。	
-520	メタデータ名のオープンができません。	
-521	メタデータ名のデータ種別の取得ができません。	
-522	メタデータ値の読み込ができません。	
-523	メタデータ名のインデックス指定で、オープンができません。	
-524	メタデータ名の作成ができません。	
-525	メタデータ値の書き込みができません。	
-526	データ種別クラス取得ができません。	
-527	データ種別のコピーができません。	
-528	データ種別のサイズ設定ができません。	
-529	データ種別の取得ができません。	
-540	データ種別のスペース取得ができません。	
-541	データセット読み書き位置指定ができません。	
-550	メタデータ "ProductName" オープンエラーのため、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル
-551	メタデータ "ProductName" の読み込み時に HDF ライブラリ内部エラーが発生し、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル
-552	メタデータ "OverlapScans" オープンエラーのため、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル
-553	メタデータ "OverlapScans" の読み込み時に HDF ライブラリ内部エラーが発生し、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル
-554	メタデータ "GeophysicalName" オープンエラーのため、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル
-555	メタデータ "GeophysicalName" の読み込み時に HDF ライブラリ内部エラーが発生し、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル
-556	メタデータ "GranuleID" オープンエラーのため、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル
-557	メタデータ "GranuleID" の読み込み時に HDF ライブラリ内部エラーが発生し、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル
-558	メタデータ "CoRegistrationParameterA1" オープンエラーのため、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセスラベル

番号	説明	備考
-559	メタデータ "CoRegistrationParameterA1" の読み込み時に HDF ライブラリ内部エラーが発生し、データ セットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセス ラベル
-560	メタデータ "CoRegistrationParameterA2" オープンエラーの ため、データセットのアクセスに失敗しまし た。	参考: 7.1.1 データセットアクセス ラベル
-561	メタデータ "CoRegistrationParameterA2" の読み込み時に HDF ライブラリ内部エラーが発生し、データ セットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセス ラベル
-562	メタデータ "Projection" オープンエラーのため、 データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセス ラベル
-563	メタデータ "Projection" の読み込み時に HDF ラ イブラリ内部エラーが発生し、データセットの アクセスに失敗しました。	参考: 7.1.1 データセットアクセス ラベル
-564	メタデータ "OrbitDirection" オープンエラーのた め、データセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセス ラベル
-565	メタデータ "OrbitDirection" の読み込み時に HDF ライブラリ内部エラーが発生し、データ セットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセス ラベル
-566	メタデータ "ProductionDateTime" オープンエ ラーのため、データセットのアクセスに失敗し ました。	参考: 7.1.1 データセットアクセス ラベル
-567	メタデータ "ProductionDateTime" の読み込み時 に HDF ライブラリ内部エラーが発生し、デー タセットのアクセスに失敗しました。	参考: 7.1.1 データセットアクセス ラベル
-601	読み書きするデータ種別がデータセットの データ種別と一致していません。	
-602	データ書き込み時の最大最小値が異常です。	
9999	書き込みデータが最大値を超えました。	

9 付録

9.1 サンプルプログラム (sample_01) について

9.1.1 Linux 環境の場合

インストールディレクトリは下記の値で記述されています。
この例と違う場合は `makefile` や以下のコマンドのパス名を変更してください。

HDF5 インストールディレクトリ : `/HDF5/shared`

SGTK のインストールディレクトリ : `/SGTK_SGLI`

※ライブラリ `sgtk` は同一環境で作成されたものを使用して下さい。

```
$ cd /SGTK_SGLI/sample/linux      Linux 版サンプル生成のためにディレクトリ移動
$ make                             make します。
$ cd ../bin                        実行プログラムが格納されたディレクトリへ移動
                                   ↓ライブラリのパスを環境変数に設定
$ export LD_LIBRARY_PATH=/HDF5/shared/lib:/SGTK_SGLI/lib:$LD_LIBRARY_PATH
$ sample_01                        サンプルプログラムを実行
SGTK_make_parameter_table : make parameter table
pata_ret=0
map_size=408
SGTK_load_config : load configuration file
SGTK_open_file : open product file
SGTK_set_dims : set dimensions
SGTK_malloc : allocate memory to set data
SGTK_set_dataset_int : set dataset value
SGTK_set_meta_string : set string attribute value
SGTK_set_meta_float : set numeric attribute value
SGTK_malloc : allocate memory to get data
SGTK_get_dataset_int : get dataset value
SGTK_get_meta_string : get string attribute value
SGTK_get_meta_float : get numeric attribute value
SGTK_close_file : close product file
SGTK_free : release alloc'd memory
sample program normal end

$
```

9.1.2 Windows 環境の場合

C:\¥SGTK_SGLI¥sample¥msvc¥sample_01.sln をダブルクリックして VisualStudio2015 でビルドします。ビルドされたプログラムは下記ディレクトリにコピーされますので、それを実行します。

```
C:\¥SGTK_SGLI¥sample¥bin>
```

```
C:\¥SGTK_SGLI¥sample¥bin>下記のコマンドでライブラリディレクトリ指定  
set PATH=%PATH%;C:\¥CMake-hdf5-1.8.21¥build¥bin¥Release;C:\¥SGTK_SGLI¥lib
```

```
C:\¥SGTK_SGLI¥sample¥bin> sample_01  
SGTK_make_parameter_table : make parameter table  
pata_ret=0  
map_size=408  
SGTK_load_config : load configuration file  
SGTK_open_file : open product file  
SGTK_set_dims : set dimensions  
SGTK_malloc : allocate memory to set data  
SGTK_set_dataset_int : set dataset value  
SGTK_set_meta_string : set string attribute value  
SGTK_set_meta_float : set numeric attribute value  
SGTK_malloc : allocate memory to get data  
SGTK_get_dataset_int : get dataset value  
SGTK_get_meta_string : get string attribute value  
SGTK_get_meta_float : get numeric attribute value  
SGTK_close_file : close product file  
SGTK_free : release alloc'd memory  
sample program normal end
```

この取扱説明書と違うディレクトリに HDF5、SGTK をインストールした場合は、それに合わせて VisualStudio2015 の中で sample_01 の include ディレクトリ、ライブラリディレクトリのプロパティを変更してください。