Contents
<目次>

1. 利用概要

・　現在、SDTP関数はデータ蓄積用関数群とSIRIUS用関数群の2種類があり、取得先を切り替えたい場合には、関数を使い分けなければならず、汎用性が低かった。

・　新しいSDTP関数は、1つの関数群で取得先を意識することなく、データ蓄積とSIRIUSへの接続が可能となっている。

・　実際には、SIRIUS用SDTP関数をベースに、現2種類のSDTP関数の引数を統合し、引数に受信時刻で取得するか生成時刻で取得するかを指定することで、データ蓄積に接続するか、SIRIUSに接続するか判定することになる。

なお、新しいSDTP関数は、Solaris、Linux、Windows上で稼働させることが可能である。

SDTP関数の利用イメージを図 1-1に示す。

また、SDTP関数で使用する用語の説明を「付録1　用語説明」に示す。

The new SDTP function can be run on Solaris, Linux, Windows. The usage image of the SDTP function is shown in Figure 1-1.

Also explanation of the terms used in the SDTP function is shown in Appendix 1 Glossary.

Usage summary

・ Currently, there are two kinds of SDTP functions, a data storage function group and a SIRIUS function group, and when you want to change the acquisition destination, you have to select the function properly, and the versatility is low.

・ With the new SDTP function, data can be accumulated and connected to SIRIUS without being conscious of one function acquisition destination.

・ In fact, based on the SDTP function for SIRIUS, we integrate the arguments of SDTP functions and designate whether to acquire at the reception time or at the generation time as the argument to connect to the data storage. It is determined whether to connect to SIRIUS

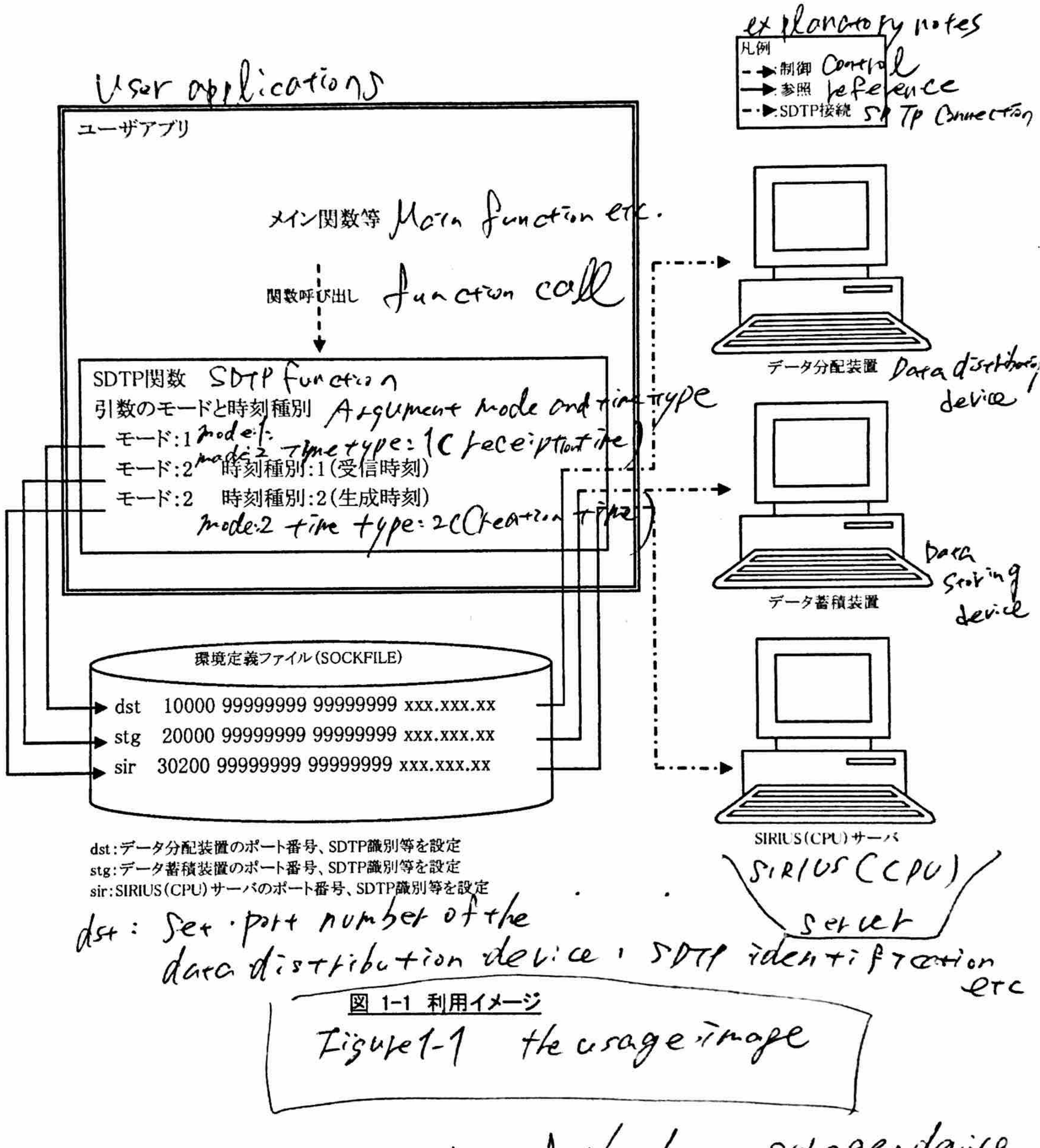
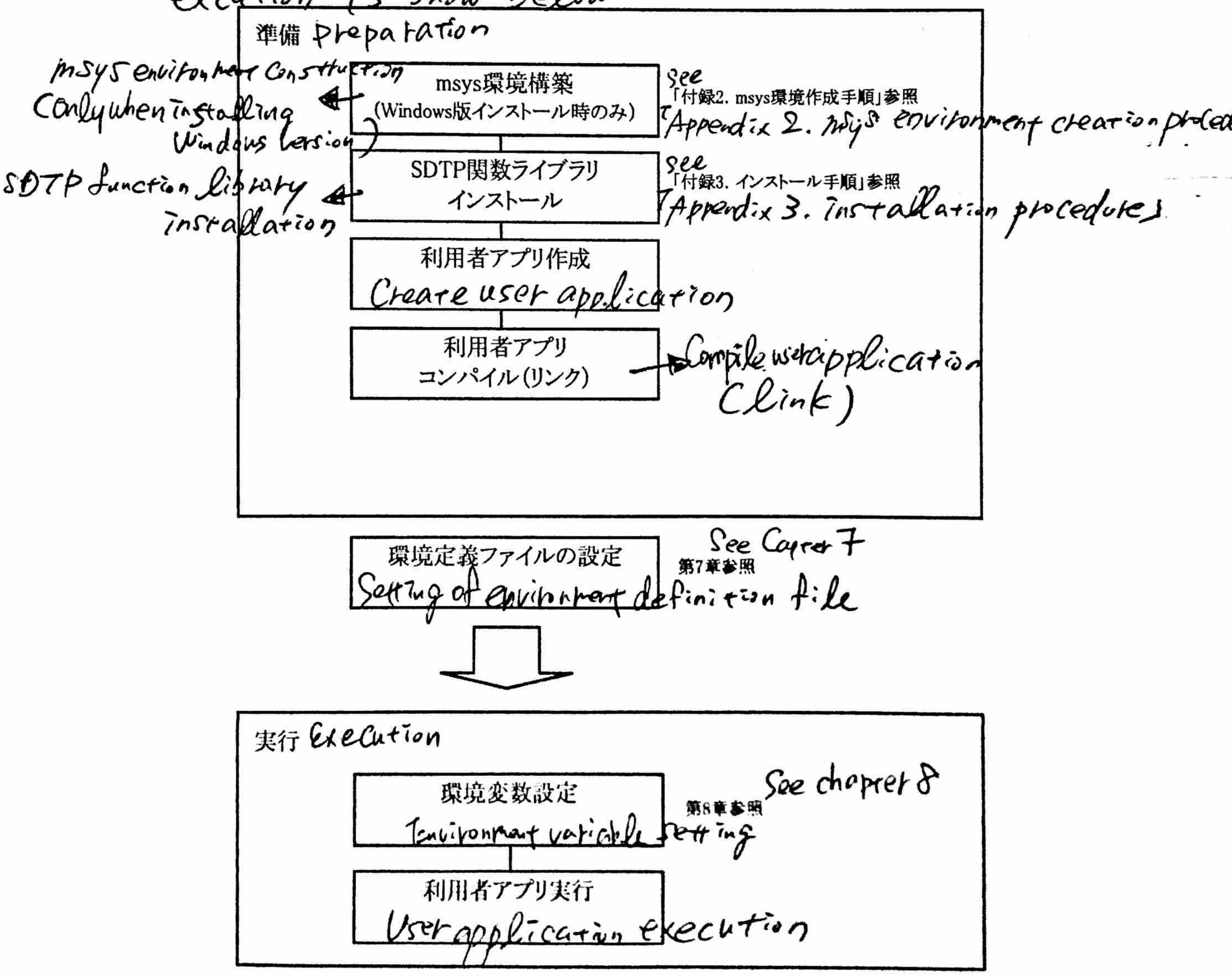

凡例
制御
参照
SDTP接続
ユーザアプリ
メイン関数等
関数呼び出し
SDTP関数
引数のモードと時刻種別
モード:1　時刻種別:1(受信時刻)
モード:2　時刻種別:1(受信時刻)
モード:2　時刻種別:2(生成時刻)
環境定義ファイル(SOCKFILE)
dst　10000 99999999 99999999 xxx.xxx.xx
stg　20000 99999999 99999999 xxx.xxx.xx
sir　30200 99999999 99999999 xxx.xxx.xx
データ分配装置
データ蓄積装置
SIRIUS(CPU)サーバ
dst:データ分配装置のポート番号、SDTP識別等を設定
stg:データ蓄積装置のポート番号、SDTP識別等を設定
sir:SIRIUS(CPU)サーバのポート番号、SDTP識別等を設定

図 1-1 利用イメージ

4. 環境構築から実行までの流れ Flow from environment construction to execution

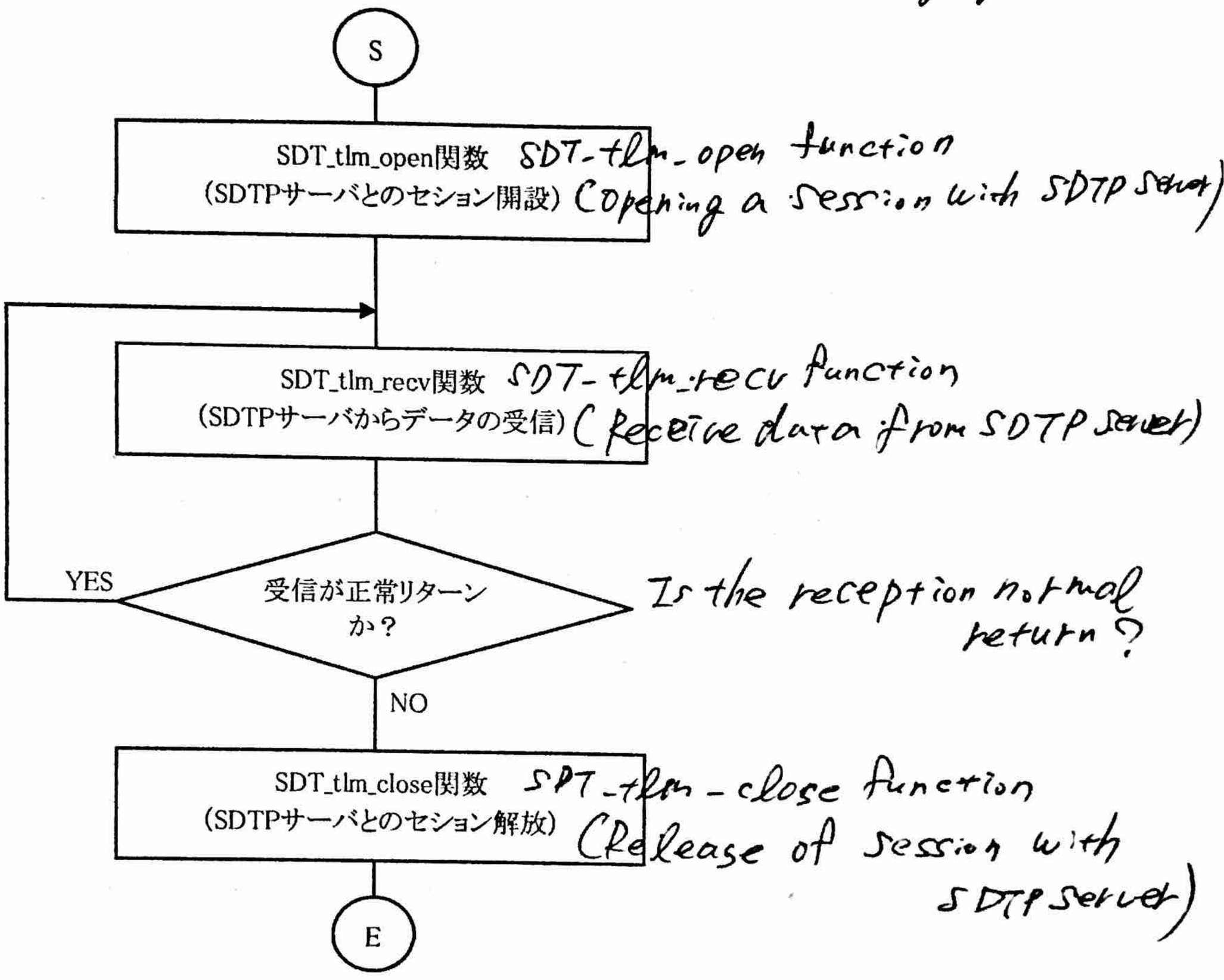
4.1. 環境構築・実行 Environment Construction • execution

以下に環境構築から利用者アプリ実行までの流れを示す。
The flow from environment construction to user application-excution is show below



5. SDTP関数処理フロー SDTP Function processing flow

SDTP関数の使用例を以下のフローに示す。An example of using the SDTP function is shown in the following flow.



※詳細は、ソースファイルに同梱のサンプルプログラム参照
For details, refer to the sample program included in the source file

6. SDTP関数説明 SDTP function description

6.1. SDTP関数詳細 SDTP function detail

SDTP関数の詳細を以降に示す。
The details of the SDTP function are shown bela

(1) 現SDT_tlm_openとSDT_tlm_open_siriusの引数の統合

(2) 引数に受信時刻でデータ取得をするのか、生成時刻でデータ取得をするのかを指定できる項目を「時刻種別」として追加

(3) 入力引数に対するconst指定とプラットフォーム非依存型の対応

(4) エラーコードの統一と見直し

↓

(1) Argument integration of current SDT_tlm_open and SDT_tlm_open_sirius

(2) "time_kind" "Time type" is added as an item that can specify whether to acquire reception time data or argument of data acquisition at generation time.

(3) Correspondence of const designation and platform-independent type to input arguments

(4) Unification and review of error codes.

モジュール略称名	モジュール名	適用言語	備考
SDTPテレメトリ開設処理	SDT_tlm_open	C	1／2

【機能概要】
SDTPクライアントからテレメトリデータを取得するためのセッションを開設する処理を行う。

【記述形式】
```
#include "SDT_all.h"     /* SDTP情報関連定義ヘッダファイル */
#include "SDT_tlm.h"     /* SDTP情報関連定義ヘッダファイル */

int SDT_tlm_open(const int mode, const int type, const int sat_no, const int ant_band[][2],
                 const int cpn_type, const int cpn_cnt, const Cpn_Data cpn_data[],
                 const int blk_no, const int time_kind, const char *start_time,
                 const char *end_time, const char *passno );
const int mode;                 /* モード */
const int type;                 /* タイプ */
const int sat_no;               /* 衛星番号 */
const int ant_band[][2];        /* 受信アンテナ番号 受信バンド帯 */
const int cpn_type;             /* CPNサービス種 */
const int cpn_cnt;              /* CPNサービス個数 */
const Cpn_Data cpn_data[];      /* CPNサービス情報 cpn_cnt分*/
const int blk_no;               /* ブロック化係数 */
const int time_kind;            /* 時刻種別 */
const char *start_time;         /* 開始時刻 */
const char *end_time;           /* 終了時刻 */
const char *passno;             /* パス番号 */
```

【パラメタ説明】
mode　　:(IN)モード(1:リアル、2:レートバッファ)
type　　:(IN)タイプ(1:パス番号指定、2:時刻指定。モードがリアルの場合は無効)
sat_no　　:(IN)衛星番号
ant_band　　:(IN)ant_bnd[][0]:受信アンテナ番号、ant_bnd[][1]:受信バンド帯(指定数分、最大8)
指定数が8以外の場合は、指定数の終端判定用に指定数(n)の次の受信バンド帯の要素(ant_bnd[n][1])にゼロを入れること。
SIRIUSから旧フレーム形式衛星のデータを取得する場合は、アンテナ番号に群番号を指定する。
cpn_type　　:(IN)CPNサービス種別
(1:パケット個別指定　2:VCDU個別指定　8:トランスファフレーム指定)
(非CCSDS　1:同期フレーム指定　2:非同期　3:無効)
cpn_cnt　　:(IN)CPNサービス個数
cpn_data　　:(IN)CPNサービス情報構造体の先頭ポインタ(以下の構造体のCPNサービス個数分定義)

```
typedef struct{
    uint8_t   vc_mask;     /* バーチャルチャネルIDのマスク値 */
    uint8_t   vc_ch;       /* バーチャルチャネルIDの値 */
    uint8_t   pc_mask[2]; /* パケットID のマスク値。ネットワークバイトオーダで指定 */
    uint8_t   pc_id[2];    /* パケットID (AP-ID)の値。ネットワークバイトオーダで指定 */
}Cpn_Data;
```

(バーチャルチャネルID/パケットID指定の設定データ内容については「DIOSAインタフェース仕様 宇宙データ転送プロトコル(SDTP) OSO 501-Xを参照。
http://c-soda.isas.jaxa.jp/sog_document_document.html に掲載。)
blk_no　　:(IN)ブロック化係数(1読込み当りのデータ件数、ブロック数が大きい程転送速度が早いが、1ブロック65536バイトは超えない。モードがリアルの場合は無効)
time_kind　　:(IN)時刻種別(1:受信時刻、2:生成時刻)
(モードがリアルの場合は無効。タイプがパス番号指定、時刻指定に関わらず指定する。)

[Parameter Description]

mode : (IN) mode (1: real, 2: Rate buffer)

type : (IN) type (1: pass number specification, 2: time specification
Invalid when the mode is real.

Sat_no : (IN) Satellite number

ant_band : (IN) ant_bnd[][0] : Receiving antenna number,
ant_bnd[][1] : Receiving band (Specified quantity, Up to 8)
If the specified number is other than 8 1
put zero in the element (ant_bnd[n][1]) of the receiving
band next to the specified number (n) for
judgment of the end of the specified number.
When acquiring old frame type satellite data
from SIRIUS, specify the group number as
the antenna number.

cpn_type : (IN) CPN service type
(1 : Individual specification of packet
2 : Individual specification of VCDU
8 : Transfer frame specification
(Non CCSDS · 1: Sync frame specification
2: asynchronous 3: Disabled (Inoperative?)

cpn_cnt : (IN) CPN service number

cpn_data : (IN) CPN head pointer of the service information
structure (defined as the number of CPN services
of the following structure)

```
typedef struct{
    uint8_t  vcmask; /* Mask value of virtual channel ID*/
    uint8_t  vc_ch; /* Value of virtual channel ID*/
    uint8_t  pcmask[2]; /* Mask value of packet ID
                           Specify by network byte order*/
    uint8_t  pc_id[2]; /* Value of packet ID (AP-ID)
                          Specify by network byte order*/
```

} Cpn_Data;
(for setting Contents of virtual channel ID / Packet ID
specification date, refer to "D/OSA interface
specification. Space data transfer Protocol (SDTP)
050-50/-X.
posted on http://c-sda.isas.jaxa.jp/sdg/document/
document.html)

blk: no :(IN) Blocking factor
(The larger the number of data per read and
the larger the number of blacks per read.
is, the faster the transfer rate
but not more than 65536 bytes per block.
Invalid when the mode is real.)

time_kind :(IN) Time type (1. Reception time, 2: Creation time)
(time kind)
(Invalid when the mode is real.
Type is specified regardless of
pass number specification or time specification)

モジュール略称名	モジュール名	適用言語	備考
SDTPテレメトリ開設処理	SDT_tlm_open	C	2/2

start_time : (IN)タイプが時刻指定の場合、開始時刻(YYYYMMDDhhmmss)
(モードがリアルの場合は無効)
end_time : (IN)タイプが時刻指定の場合、終了時刻(YYYYMMDDhhmmss)
(モードがリアルの場合は無効)
Passno : (IN)タイプがパス番号指定の場合、パス番号(YYMMDD9999)
(モードがリアルの場合は無効)

【復帰値】

復帰値	意味	処置
0	正常終了	
-20001	入力パラメタエラー	入力パラメタを見直す。
-20003	開設否定応答受信	入力パラメタ、環境定義ファイルを見直す。
-20004	タイムアウト	クローズする。
-20005	SOCKET生成エラー	環境定義ファイルを見直す。
-20007	connectエラー	環境定義ファイルを見直す。
-20009	サーバ側CLOSE	クローズする。
-20010	SDTP初期化エラー	入力パラメタを見直す。
-20011	環境変数取得エラー	環境変数設定を見直す。
-20012	ファイルアクセスエラー	環境定義ファイルを見直す。
-20013	環境定義ファイル内容異常	環境定義ファイルを見直す。
-20014	環境変数設定エラー	クローズする。
上記以外	付録4のエラーコードを負にした値(詳細は、付録4を参照)	クローズする。

【注意事項】

関数の引数mode値およびtime_kind値から接続先およびSDTPの動作が決定し、環境定義ファイルの接続先識別の対応情報を使用して接続を行う。対応関係を以下に示す。

SDT_tlm_openの引数		SDTP関数の動作	使用する環境定義ファイルの接続先識別
mode	time_kind		
1(リアル)	−	データ分配装置に接続し、リアルタイムに受信を行う。	dsi
2(レートバッファ)	1(受信時刻)	データ蓄積装置に接続し、レートバッファで蓄積データを受信する。	stg
2(レートバッファ)	2(生成時刻)	SIRIUSに接続し、レートバッファで蓄積データを受信する。	sir

SDTP telemetry opening process

Module Abbreviation name モジュール略称名	Module name モジュール名	Application language 適用言語	Remarks 備考
SDTPテレメトリ受信処理	SDT_tlm_recv	C	

【機能概要】Functional overview
SDTPサーバからテレメトリデータを取得する。Acquire telemetry data from SDTP server

【記述形式】Description format

```
#include "SDT_all.h"        /* SDTP情報関連定義ヘッダファイル */
#include "SDT_tlm.h"        /* SDTP情報関連定義ヘッダファイル */

int    SDT_tlm_recv(int *pdu_type, uint8_t *recv_data, int *recv_len);

int      *pdu_type;     /* 受信したPDUのPDU種別 */
uint8_t  *recv_data;    /* 受信データ部格納領域 */
int      *recv_len;     /* 受信データ長 */
```

SDTP information related definition header file
PDU type of the received PDU
Stage area of receive data part.
Receive data length
PDU type of the received PDU

【パラメタ説明】Description of parameters

pdu_type : (OUT)受信したPDUのPDU種別
recv_data : (OUT)受信データ部格納領域
recv_len : (OUT)受信データ長

refer to 「DIOSAインタフェース仕様 宇宙データ転送プロトコル(SDTP) OSO 501-Xを参照。
http://c-soda.isas.jaxa.jp/sog/document/document.html に掲載。 Posted on

DIOSA interface specification space data transfer protocol (SDTP) OSO-501-X

Receive data length
Stage area of receive data part.

【復帰値】Return value

復帰値 Return value	意味 meaning	処置 treatment
0	正常終了(受信データ有り) Successful completion (With received data)	
1	正常終了(受信データ無し) Successful completion (No received data)	
2	通信解放依頼PDU受信	クローズする。to close
-20002	通信シーケンスエラー	クローズする。to close
-20004	タイムアウト time out	クローズする。to close
-20006	データ抜け発生 Data missing occurred (シーケンスカウンタエラー) (Sequence counter error)	処理続行可能 Processing can continue
-20009	サーバ側CLOSE Server side close	クローズする。to close
上記以外 other than above	付録4のエラーコードを負にした値(詳細は、付録4を参照) Value with negative error code in Appendix 4 (For detail, refer to Appendix 4)	クローズする。to close

Receive Communication release request PDU
Communication sequence error

【注意事項】Note
なし None

When changing from the old STD_tlm_open to the new SDT_tlm_open, change the ant_id and band set in the existing processing to the two-dimensional array of ant_band and designate the two dimensional array as the new SDT_tlm_open.
Also add time_kind between blk_no and start_time.

How to change from an existing fonction

6.2. 既存関数からの変更方法

Change from old STD_tlm_open to new SDT_tlm_open

(1) 旧SDT_tlm_openから新SDT_tlm_openへの変更

旧SDT_tlm_openから新SDT_tlm_openに変更する場合、既存処理で設定していたant_idとbandをant_bandの2次元配列に設定するように変更し、2次元配列を新SDT_tlm_openに指定する。また、blk_noとstart_timeの間にtime_kindを追加する。

SDT_tlm_open(mode, type, sat_no, ant_id, band, cpn_type, cpn_cnt, cpn_data, blk_no,start_time, end_time, pathno)

SDT_tlm_open(mode, type, sat_no, ant_band, cpn_type, cpn_cnt, cpn_data[], blk_no, time_kind, start_time, end_time,passno)

change from SDT_tlm_open_sirius to new SDT_tlm_open

(2) SDT_tlm_open_siriusから新SDT_tlm_openへの変更

SDT_tlm_open_siriusから新SDT_tlm_openに変更する場合、blk_noとstart_timeの間にtime_kindを追加する。

When changing from SDT_tlm_open_sirius to new SDT_tlm_open, add time_kind between blk_no and start_time

SDT_tlm_open_sirius(mode,type,sat_no, ant_ band, cpn_type, cpn_cnt, cpn_data, blk_no, start_time, end_time,pathno)

SDT_tlm_open(mode, type, sat_no, ant_band, cpn_type, cpn_cnt, cpn_data[], blk_no, time_kind, start_time, end_time, passno)

7. 環境定義ファイル

・SDTP関数を使用する際には、以下の環境定義ファイルが必要であるため、利用者が任意のディレクトリ配下に設定しておく。環境定義ファイルのファイル名は任意(ファイル名に空白があるものは不可)であるが、SOCKFILEを推奨する。

7.1. ファイル形式

本ファイルは、情報行とコメント行で構成されるASCIIテキストファイルである。

情報行は、改行コードを除いてASCII文字から構成され、情報行内の各項目は、1文字以上の「スペース(0x20)」で区切られる。

コメント行は、先頭位置に「#(0x23)」を設定した行とし、#より後の文字は、改行コードを除けばどの文字コードでも良い。

改行コードは、OSの文字コードに依存するため、本SDTP関数を動作させる文字コードに合致した改行コードを設定する。(例:Shift-JIS:CR+LF、EUC:LF、UTF-8:LF等)

なお、空行は不可とする。

7.2. ファイル項目

本ファイルの項目一覧を表 7-1に示す。

表 7-1 ファイル項目一覧

No.	項目名	項目説明	備考
1.	接続先識別	接続先装置の識別 dst:データ分配装置 stg:データ蓄積装置 sir:SIRIUS(CPU)サーバ	半角英小文字とする。
2.	ポート番号	接続先装置に接続するポート番号。ポート番号は以下の通りとする。 10000:データ分配 20000:データ蓄積 30200:SIRIUS(CPU)サーバ	数値のみ
3.	送信元識別子	利用者側システム装置のSDTPインタフェース識別番号	*1
4.	送信先識別子	データ発生元のSDTPインタフェース識別番号	*1
5.	接続先ホスト名(IPアドレス)	接続先装置のホスト名または、IPアドレス	*2

*1:送信元識別子、送信先識別子は「DIOSAインタフェース仕様 宇宙データ伝送プロトコル(SDTP) OSO 501-X (個別規定) http://c-soda.isas.jaxa.jp/sog/document/documen[illegible]」を参照

*2:接続先装置のホスト名(IPアドレス)
UNIX系は、/etc/hosts、Windows系は、C:¥WINDOWS¥system32¥drivers¥etc¥hostsファイルに定義しておくこと。(IPアドレスの場合は定義不要。)

7.3. ファイルイメージ File Image

本ファイルのイメージを図 7-1に示す。The image of this file is shown in Figure 7-1

#接続先識別	ポート番号	送信元識別子	送信先識別子	接続先ホスト名
dst	10000	999999999999	999999999999	xxxxxxx
stg	20000	999999999999	999999999999	xxx.xxx.xxx.xxx
sir	30200	999999999999	999999999999	xxxxxxx

※未使用機器は定義不要 Unused devices need not be defined

図 7-1 ファイルイメージ

Figure 7-1 File Image

7.4. 既存ファイルからの変更

旧環境定義ファイルを新環境定義ファイルに変更する場合の対応を以下に示す。

①は①'、②は②'、③は③'に対応し、変更箇所は接続先識別のみである。

- 旧環境定義ファイル(データ分配・データ蓄積アクセス用)

	#モード	ポート番号	送信元識別子	送信先識別子	接続先ホスト名
①	1	10000	999999999999	999999999999	xxxxxxx
②	2	20000	999999999999	999999999999	xxx.xxx.xxx.xxx

- 旧環境定義ファイル(データ分配・SIRIUSアクセス用)

	#モード	ポート番号	送信元識別子	送信先識別子	接続先ホスト名
①	1	10000	999999999999	999999999999	xxxxxxx
③	2	30200	999999999999	999999999999	xxxxxxx

- 新環境定義ファイル

	#接続先識別	ポート番号	送信元識別子	送信先識別子	接続先ホスト名
①'	dst	10000	999999999999	999999999999	xxxxxxx
②'	stg	20000	999999999999	999999999999	xxx.xxx.xxx.xxx
③'	sir	30200	999999999999	999999999999	xxxxxxx