

DSHGEM-LIB 通信エンジンライブラリ(GEM+GEM300)
ソフトウェア・パッケージ

APP インタフェース ライブラリ関数説明書

(C, C++, .Net-Vb,C#)

VOL- 1 0 / 1 5

- 3 . 16 キャリアアクション(S3F17)関連関数
- 3 . 17 ポートアクション、アクセスモード(S3F23,S3F25,S3F27)関連関数

2 0 0 9年6月

株式会社データマップ

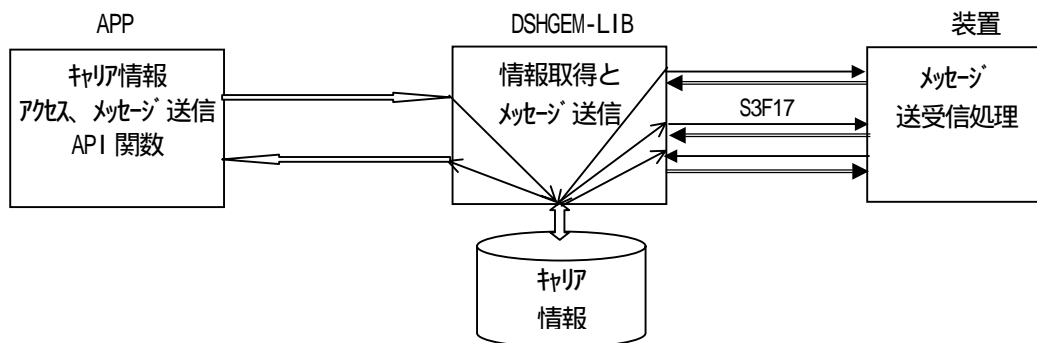
目次

3.16	キャリアアクションメッセージ関連関数	1
3.16.1	使用する情報格納構造体	3
3.16.2	キャリアアクション関連メッセージ送信関数	4
3.16.2.1	GemSendS3F17() - キャリアアクションメッセージ送信関数	4
3.16.3	キャリアアクション関連ライブラリ関数	7
3.16.3.1	DshEncodeS3F17() - キャリア・アクション情報を S3F17 へエンコード	7
3.16.3.2	DshDecodeS3F17() - キャリア・アクション情報を S3F17 へデコード	9
3.16.3.3	DshFreeTCACT_INFO() - キャリア・アクション情報構造体メモリの開放	10
3.16.3.4	DshCopyTCACT_INFO() - キャリア・アクション情報構造体メモリのコピー	11
3.16.3.5	DshInitTCACT_INFO() - TCACT_INFO キャリアアクション情報の初期設定	12
3.16.3.6	DshAddTCACT_INFO() - TCACT_INFO キャリアアクション情報のパラメータ設定	14
3.16.3.7	DshMakeTCACT_PARA() - TCACT_PARA キャリアアクション情報の初期設定	15
3.16.3.8	DshFreeTCACT_CONTENT() - キャリアコンテンツ情報メモリ開放	17
3.16.3.9	DshPutTCACT_CONTENT() - TCACT_CONTENT キャリアアクション情報の設定	18
3.16.3.10	DshPutTCACT_SLOT_INFO() - スロットマップデータの設定	19
3.16.3.11	DshGetCaActionIndex() - キャリアアクションサービスのインデクス取得	20
3.16.3.12	DshGetCaActionName() - キャリアアクションインデクスからアクション名取得	21
3.16.3.13	DshGetCaAttrIndex() - キャリアアクション属性のインデクス取得	22
3.16.3.14	DshGetCaAttrName() - キャリアアクション属性インデクスから属性名取得	23
3.16.3.15	DshInitTCACT_ERR_INFO() - キャリアアクション要求応答情報の初期化	24
3.16.3.16	DshFreeTCACT_ERR_INFO() - キャリアアクション応答情報メモリの開放	25
3.16.3.17	DshMakeS3F17Response() - S3F17 の応答メッセージの生成	26
3.16.4	ユーザ作成ライブラリ関数	28
3.16.4.1	DshResponseS3F18() - S3F18 キャリアアクション要求応答メッセージ	28
3.17	ポートアクション、アクセスモード(S3F23,S3F25,S3F27)関連関数	30
3.17.1	使用する情報格納構造体	33
3.17.2	ポートアクション関連メッセージ送信関数	35
3.17.2.1	GemSendS3F23() - ポートグループアクションメッセージ送信関数	35
3.17.2.2	GemSendS3F25() - ポートアクションメッセージ送信関数	38
3.17.2.3	GemSendS3F27() - アクセス変更メッセージ送信関数	41
3.17.3	ポート関連情報処理ライブラリ関数	44
3.17.3.1	DshDecodeS3F23 - S3F23 デコード関数	44
3.17.3.2	DshEncodeS3F23() - ポートグループアクション情報を S3F23 へエンコード	45
3.17.3.3	DshFreeTPORTG_INFO() - ポートグループアクション情報構造体メモリの開放	47
3.17.3.4	DshCopyTPORTG_INFO() - ポートグループアクション情報構造体メモリのコピー	48
3.17.3.5	DshInitTPORTG_INFO - ポートグループアクション情報 TPORTG_INFO の初期設定	49
3.17.3.6	DshAddTPORTG_INFO() - ポートグループアクションパラメータの追加	50
3.17.3.7	DshMakeS3F23Response() - S3F23 の応答メッセージの生成	52
3.17.3.8	DshDecodeS3F25 - S3F25 デコード関数	54
3.17.3.9	DshEncodeS3F25() - ポートアクション情報を S3F25 へエンコード	55
3.17.3.10	DshFreeTPORTG_INFO() - ポートアクション情報構造体メモリの開放	57
3.17.3.11	DshCopyTPORTG_INFO() - ポートアクション情報構造体メモリのコピー	58

3 . 17 . 3 . 12	DshInitTPORT_INFO - ポートアクション情報 TPORT_INFO の初期設定.....	59
3 . 17 . 3 . 13	DshAddTPORT_INFO() - ポートアクションパラメータの追加.....	60
3 . 17 . 3 . 14	DshMakeS3F25Response() - S3F25 の応答メッセージの生成.....	62
3 . 17 . 3 . 15	DshDecodeS3F27 - S3F27 デコード関数.....	64
3 . 17 . 3 . 16	DshEncodeS3F27() - ポートアクセス情報を S3F27 ヘエンコード.....	65
3 . 17 . 3 . 17	DshFreeTACCESS_INFO() - ポートアクセス変更情報構造体メモリの開放.....	67
3 . 17 . 3 . 18	DshCopyTACCESS_INFO() - ポートアクセス変更情報構造体メモリのコピー.....	68
3 . 17 . 3 . 19	DshInitTACCESS_INFO - ポートアクセス変更情報 TACCESS_INFO の初期設定	69
3 . 17 . 3 . 20	DshAddTACCESS_INFO() - アクセス変更ポート番号の追加.....	70
3 . 17 . 3 . 21	DshInitTACCESS_ERR_INFO () - ポートアクセス応答情報の初期化.....	71
3 . 17 . 3 . 22	DshPutTACCESS_ERR_INFO () - ポートアクセスエラー情報の設定.....	73
3 . 17 . 3 . 23	DshFreeTACCESS_ERR_INFO() - ポートアクセス応答情報メモリの開放.....	75
3 . 17 . 3 . 24	DshMakeS3F27Response() - S3F27 の応答メッセージの生成.....	76
3 . 17 . 4	ユーザ作成ライブラリ関数.....	78
3 . 17 . 4 . 1	DshResponseS3F24) - S3F24 グループポートアクション応答メッセージ.....	78
3 . 17 . 4 . 2	DshResponseS3F26) - S3F26 ポートアクション応答メッセージ.....	80
3 . 17 . 4 . 3	DshResponseS3F28) - S3F28 ポートアクセスモード応答メッセージ.....	82

(VOL - 1 1 に続く)

3.16 キャリアアクションメッセージ関連関数



(1) 情報アクセスと送信 API 関数

キャリアアクション送信に関連するサービスのための API 関数名は一覧表のとおりです。

	API 関数名	機能
1	GemSendS3F17()	S3F17 キャリアアクション要求メッセージを送信します。

(2) ライブラリ関数

他に APP が使用できるキャリアアクション処理用 API 関数として、以下の関数があります。

	API 関数名	機能
1	DshEncodeS3F17()	TCACT_INFO 構造体のキャリアアクション情報を S3F17 メッセージ にエンコード するための関数です。
2	DshDecodeS3F18()	S3F17 の応答メッセージ S3F18 を TCACT_ERR_INFO 構造体にデコード します。
3	DshFreeTCACT_INFO()	キャリアアクション情報が格納されている TCACT_INFO 構造体の内部で使用されているメモリを開放するための関数です。
4	DshCopyTCACT_INFO()	TCACT_INFO のキャリアアクション情報を別の構造体にコピー します。
5	DshInitTCACT_INFO()	S3F17 メッセージ 用の TCACT_INFO キャリアアクション情報構造体の初期設定を行います。
6	DshAddTCACT_INFO()	S3F17 メッセージ 用の TCACT_INFO キャリアアクション情報構造体に 1 個の属性情報を加えます。
7	DshMakeTCACT_PARA()	S3F17 メッセージ 用の TCACT_PARA キャリアアクション属性情報を生成します。 TCACT_PARA は TCACT_INFO に設定されるパラメータ情報です。
8	DshFreeTCACT_CONTENT	TCACT_CONTENT 構造体の内部で使用されているメモリを開放するための関数です。
9	DshPutTCACT_CONTENT()	S3F17 メッセージ 用の ContentMap 属性情報を TCACT_PARA 構造体に設定します。
10	DshPutTCACT_SLOT_INFO()	S3F17 メッセージ 用の Slotmap データを TCACT_PARA 構造体内に設定します。
11	DshGetCaActionIndex ()	キャリアアクションサービス名のインデックス取得
12	DshGetCaActionName()	キャリアアクションサービス名のインデックスからアクション名を取得する。
13	DshGetCaAttrIndex()	キャリアアクションの属性名から属性インデックスを取得する。

14	DshGetCaAttrName()	キャリアアクションの属性インデックスから属性インデックス名を取得する。
15	DshInitTCACT_ERR_INFO()	TCACT_ERR_INFO 構造体内の情報を初期設定します。 (S3F18 の応答情報)
16	DshFreeTCACT_ERR_INFO()	TCACT_ERR_INFO S3F18 応答情報内のメモリを開放します。
17	DshMakeS3F17Response()	S3F18 メッセージを生成します。

(3) ユーザ作成ライブラリ関数

	ライブラリ関数名	機能
1	DshResponseS3F18()	S3F17 キャリアアクション要求応答

3.16.1 使用する情報格納構造体

キャリア情報を操作する関数は、キャリア情報の格納のために TCAR_INFO 構造体を使用します。キャリア情報とそのアクションなどに関連する構造体は下記のとおりです。

(1) TCACT_INFO – Carrier Action Information - S3F17

```
typedef struct{
    TDATAID    dataid;
    char       *caction;          // car action cmd
    int        action_index;      // caction の index
    char       *carspec;          // carrier spec ( carid )
    int        ptn;               // port no.
    int        cp_count;          // parameter count
    TCACT_PARA **cp_list;         // paramete list
}TCACT_INFO;
```

(2) TCACT_PARA – Carrier Action Parameter Information

```
typedef struct{
    char       *cattrid;          // cattrid
    int        attr_index;        // cattrid のインデックス
    void       *cattrdata;        // cattrdata
}TCACT_PARA;
```

(3) TCACT_ERR_INFO – Carrier Action Response Information – S3F18

```
typedef struct{
    int        caack;             // CAACK
    int        err_count;         // 含まれているエラー情報の数
    TERR_INFO **err_list;        // エラー情報リスト
} TCACT_ERR_INFO;
```

(4) TERR_INFO – Object 応答エラー情報

```
typedef struct{
    int        errcode;           // エラーコード (U1)
    char       *errtext;         // エラーテキスト
} TERR_INFO;
```

3.16.2 キャリアアクション関連メッセージ送信関数

3.16.2.1 GemSendS3F17() キャリアアクションメッセージ送信関数

(1) 呼出書式

[c,C++]

```
API int APIX GemSendS3F17(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    TCACT_INFO *info, // キャリアアクション情報格納領域のポインタ
    TCACT_ERR_INFO *erinfo, // S3F18 応答情報格納用構造体のポインタ
    int (WINAPI *callback)(), // 実行終了時のコールバック関数
    ULONG upara // callback 時のパラメータ
);
```

[.NET VB]

```
Function GemSendS3F17 (
    ByVal eqid As Int32,
    ByRef info As dsh_info.TCACT_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO,
    ByVal callback As vcallback.callback_S3F17,
    ByVal upara As Int32) As Int32
```

[.NET C#]

```
int GemSendS3F17(
    int eqid,
    ref TCACT_INFO info,
    ref TCACT_ERR_INFO erinfo,
    CallbackS3F17 callback,
    uint upara );
```

(2) 引数

eqid

通信エンジンが通信する対象装置 ID を指定します。装置 ID は 0 から始まる番号です。

info

送信したいキャリアアクション情報が格納されている構造体のポインタです。

erinfo

受信した応答メッセージ S3F18 に含まれる情報を格納するための構造体領域のポインタを指定します。

callback

DSHGEM-LIB によるメッセージ送信処理が終了したときに呼出されるコールバック関数を指定します。ユーザは任意の関数名を指定できます。コールバックの指定が=0 の場合はブロックモードになります。

upara

コールバックされたときに引数で渡してもらうためのパラメータです。関数実行終了時にコールバックされた際、何かの判別情報として使用したい値、構造体ポインタなどを設定します。

(3) 戻り値

戻り値	意味
-----	----

0	(1) プロトコルモード : 正常に送信できた。 erinfo に S3F18 応答情報が返却されます。 (2) 非プロトコルモード : 要求が受け付けられた。
(-1)	送信できなかった。
(-14)	T3タイムアウトを検出した。

(4) 説明

ホストにキャリアアクション情報の送信要求を行います。S3F17 メッセージで送信します。

要求を受けた DSHGEM-LIB は、info に格納されているキャリアアクション情報を S3F17 メッセージにエンコードし、ホストに送信します。

S3F18 応答メッセージ受信で得られた情報はデコードされて erinfo で指定された構造体領域に返却されません。

送信要求から S3F18 応答メッセージ受信までの制御は引数のコールバック関数指定の有無によって次のようになります。

callback の指定	制御の流れ
なし (=0)	S3F17 送信後、応答メッセージ S3F18 が受信されるか、またはエラーを検出するまで制御が要求元に戻ってきません。 正常終了であれば、erinfo に S3F18 応答情報が返却されます。
あり	送信要求後、S3F17 の送信前に制御が戻されます。実行結果はコールバック関数で与えられます。 コールバック関数の end_status が=0 ならば erinfo に S3F18 応答情報が返却されません。 エラーが検出された場合、(-1)が end_status にセットされます。

正常に応答メッセージを受信した場合、その中に含まれている応答情報がデコードされ erinfo で指定された TCACT_ERR_INFO 構造体の中に格納され返却されます。ユーザ側で erinfo 内の情報の処理を終えた後、その構造体で使用されているメモリを解放してください。

解放は次のように DshFreeTCACT_ERR_INFO()関数を使って行ってください。

```
DshFreeTCACT_ERR_INFO (erinfo)
```

TCACT_INFO 構造体へのキャリアアクション情報の設定には以下のライブラリ関数を使用することができます。

```
DshInitTCACT_INFO(), DshAddTCACT_INFO()
DshmakeTCACT_PAR(), DshPutCACT_CONTENT(), DshPutTCACT_SLOT_INFO()
```

(5) コールバック関数

[c,C++]

```
API int APIX callback(
    int eqid,           // 装置 ID
    int end_status,    // 実行結果
    TCACT_ERR_INFO *erinfo, // S3F18 応答情報格納用構造体のポインタ
    ULONG upara        // 呼出時に指定したパラメータ
```

);

[.NET VB]

Function callback_S3F17(ByVal eqid As Integer, ByVal end_status As Integer, ByRef erinfo As dsh_info.TCACT_ERR_INFO, ByVal upara As Integer) As Integer

[.NET C#]

int CallbackS3F17(int eqid, int end_status, ref TCACT_ERR_INFO erinfo, uint upara);

end_status には以下の値が設定されます。

結果	意味
0	正常に送信できた。erinfo に S3F18 応答情報が返却されます。
(-1)	送信できなかった。
(-14)	T3タイムアウトを検出した。

3.16.3 キャリアアクション関連ライブラリ関数

3.16.3.1 DshEncodeS3F17() - キャリア・アクション情報を S3F17 へエンコード

(1) 呼出書式

[C, C++]

```
API int APIX DshEncodeS3F17(
    int eqid, // 装置 ID
    SHMSG *smsg, // SECS メッセージ情報構造体のポインタ
    BYTE *buffer, // S3F17 を格納するバッファポインタ
    int buflen, // buffer のバイトサイズ
    TCACT_INFO *pinfo // エンコードしたいアクション情報が格納されている構造体のポインタ
);
```

[.NET VB]

```
function DshEncodeS3F17 (
    ByVal eqid As Int32,
    ByRef smsg As dshdr2.DSHMSG,
    ByRef buff As byte,
    ByVal buflen As Int32, _
    ByRef info As dsh_info.TCACT_INFO ) As Int32
```

[.NET C#]

```
int DshEncodeS3F17(
    int eqid,
    ref DSHMSG smsg,
    byte[] buff,
    int buflen,
    ref TCACT_INFO info);
```

(2) 引数

eqid
通信する相手の装置 ID(0,1,2...)です。

smsg
エンコードした S3F17 メッセージを格納するメッセージ情報構造体のポインタです。

buffer
エンコードした S3F17 のテキストを格納するバッファポインタです。

buflen
buffer のバイトサイズです。

pinfo
エンコードしたいキャリア・アクション情報が格納されている構造体のポインタです。

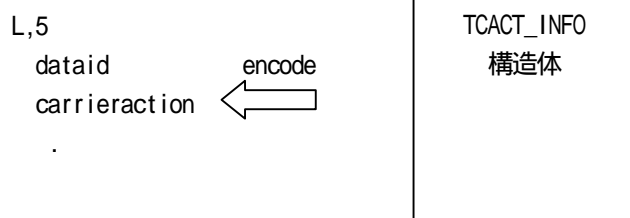
(3) 戻り値

戻り値	意味
0	正常にエンコードできた。
(-1)	smsg を正しくエンコードできなかった。

(4) 説明

TCACT_INFO 構造体に格納されているキャリア・アクション情報を、S3F17 の SECS メッセージにエンコードします。

msg S3F17



3.16.3.2 DshDecodeS3F17() - キャリア・アクション情報を S3F17 ヘデコード

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS3F17(
    DSHMSG *smsg,           // SECS メッセージ 情報構造体のポインタ
    TCACT_INFO *pinfo      // デコードしたいアクション情報が格納されている構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS3F17 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef info As dsh_info.TCACT_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS3F17(
    ref DSHMSG smsg,
    ref TCACT_INFO info );
```

(2) 引数

smsg

デコードした S3F17 メッセージが格納されているメッセージ情報構造体のポインタです。

pinfo

デコードしたキャリア・アクション情報を格納するための構造体のポインタです。

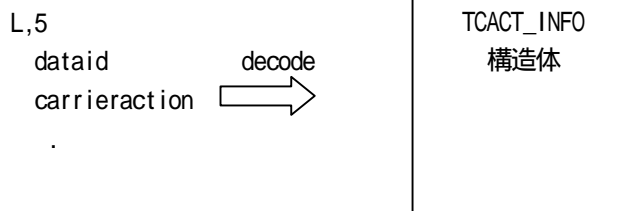
(3) 戻り値

戻り値	意味
0	正常にデコードできた。
(-1)	smsg を正しくデコードできなかった。

(4) 説明

smsg の S3F17 メッセージの中に含まれるキャリアアクション情報をデコードし TCACT_INFO 構造体に格納します。

smsg S3F17



3.16.3.3 DshFreeTCACT_INFO() - キャリア・アクション情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTCAR_INFO(  
    TCACT_INFO *pinfo // メリを開放したいアクション情報が格納されている構造体のポインタ  
);
```

[.NET VB]

```
Sub DshFreeTCACT_INFO (  
    ByRef info As dsh_info.TCACT_INFO)
```

[.NET C#]

```
void DshFreeTCACT_INFO(  
    ref TCACT_INFO info );
```

(2) 引数

pinfo

メモリを解放したいキャリア・アクション情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TCACT_INFO 構造体内で情報格納用に使用されているメモリを全て解放します。

開放した後、TCACT_INFO の内容を全て 0 で初期設定します。

pinfo が NULL ならば、何も処理しません。

3.16.3.4 DshCopyTCACT_INFO() - キャリア・アクション情報構造体メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTCAR_INFO(
    TCACT_INFO *dinfo,           // 北°-先のポインタ
    TCACT_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTCACT_INFO (
    ByRef dinfo As dsh_info.TCACT_INFO,
    ByRef sinfo As dsh_info.TCACT_INFO) As Int32
```

[.NET C#]

```
int DshCopyTCACT_INFO(
    ref TCACT_INFO dinfo,
    ref TCACT_INFO sinfo );
```

(2) 引数

dinfo

キャリア・アクション情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のキャリア・アクション情報が格納されている構造体メモリのポインタです。

(3) 戻り値

戻り値	意味
0	正常に北°-できた。
(-1)	sinfo または dinfo の値が NULL だったので北°-できなかった。

(4) 説明

sinfo が指す TCACT_INFO 構造体内に格納されているキャリア・アクション情報を dinfo が指定する TCACT_INFO 構造体にコピーします。

dinfo 内のメンバーで新しいメモリが必要なものは本関数が取得します。

dinfo 内メンバーで確保されたメモリは、使用後、DshFreeTCACT_INFO()関数を使って開放してください。

3.16.3.5 DshInitTCACT_INFO() TCACT_INFO キャリアアクション情報の初期設定

(1) 呼出書式

[C, C++]

```
API int APIX DshInitTCACT_INFO(
    TCACT_INFO *info,           // TCACT_INFO 構造体のポインタ
    int action_x,              // アクション名インデックス
    char *carid,               // キャリア ID
    int ptn,                   // ポート番号
    int attr_count             // 設定属性データの数
);
```

[.NET VB]

```
Sub DshInitTCACT_INFO (
    ByRef info As dsh_info.TCACT_INFO,
    ByVal action_x As Int32,
    ByVal carid As String,
    ByVal ptn As Int32,
    ByVal attr_count As Int32)
```

[.NET C#]

```
void DshInitTCACT_INFO(
    ref TCACT_INFO info,
    int action_x,
    byte[] carid,
    int ptn,
    int attr_count );
```

(2) 引数

info

TCACT_INFO 構造体のポインタです。このメンバーを初期設定します。

action_x

キャリアアクション名に対するインデックス値です。(4)の説明を参照してください。

carid

設定するキャリア ID (文字列) です。

ptn

ロードポート番号です。

attr_count

action_x で指定されたアクションに付属する属性データの数です。

(3) 戻り値

なし。

(4) 説明

本関数は APP が S3F17 キャリアアクション要求メッセージ送信時に GemSendS3F17()関数の引数に与えるキャリアアクション(CARRIERACTION)情報を TCACT_INFO 構造体の初期設定を行います。

最初に info 内をクリアします。そして、引数で指定された情報を info 内に設定します。

メモリが必要なメンバーについてはメモリを確保し情報をコピーします。

TCACT_INFO 構造体への属性データの設定には DshAddTCACT_INFO()関数を使用します。

使用が済んだあとの TCACT_INFO 構造体内で使用されているメモリの開放は DshFreeTCACT_INFO()関数を使用します。

action_x は S3F17 の CARRIERACTION データアイテムに設定する文字列を数値で指定します。次表を参照ください。

action_x 値	マクロ名	CARRIERACTION 文字列
0	CA_Bind	Bind
1	CA_CancelBind	CancelBind
2	CA_CancelAllCarrierOut	CancelAllCarrierOut
3	CA_CancelCarrier	CancelCarrier
4	CA_CancelCarrierAtPort	CancelCarrierAtPort
5	CA_CancelCarrierNotification	CancelCarrierNotification
6	CA_CancelCarrierOut	CancelCarrierOut
7	CA_CarrierIn	CarrierIn
8	CA_CarrierNotification	CarrierNotification
9	CA_CarrierOut	CarrierOut
10	CA_CarrierReCreate	CarrierReCreate
11	CA_CarrierRelease	CarrierRelease
12	CA_ProceedWithCarrier	ProceedWithCarrier

3.16.3.6 DshAddTCACT_INFO() TCACT_INFO キャリアアクション情報のパラメータ設定

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTCACT_INFO(
    TCACT_INFO *info,           // TCACT_INFO 構造体のポインタ
    TCACT_PARA *attrdata       // 設定属性データ格納構造体のポインタ
);
```

[.NET VB]

```
Function DshAddTCACT_INFO (
    ByRef info As dsh_info.TCACT_INFO,
    ByRef attrdata As dsh_info.TCACT_PARA) As Int32
```

[.NET C#]

```
int DshAddTCACT_INFO(
    ref TCACT_INFO info,
    TCACT_PARA *attrdata );
```

(2) 引数

info

TCACT_INFO 構造体のポインタです。この構造体メンバーに情報を設定します。

attrdata

アクション情報に加える属性データが格納されている構造体のポインタです。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	予約されているカウンタを超えていたので設定できなかった。

(4) 説明

本関数は APP が S3F17 キャリアアクション要求メッセージのアクション情報に付属する属性データを info で指定された TCACT_INFO 構造体の属性データリストの中に追加します。

attrdata で指定される TCACT_PARA 構造体の中身は DshMakeTCACT_PARA() と DshPutTCACT_PARA() 関数を使って属性情報 (パラメータ) を設定されていなければなりません。

info 内への属性情報の設定は attrdata が使用しているメモリをそのまま使用します。

3.16.3.7 DshMakeTCACT_PARA() TCACT_PARA キャリアアクション情報の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshMakeTCACT_PARA(
    TCACT_PARA **info,           // TCACT_PARA 構造体のポインタのポインタ
    int attr_x,                 // 属性指定インデクス
    void *attrdata             // 設定属性データ格納構造体のポインタ
);
```

[.NET VB]

```
Sub DshMakeTCACT_PARA (
    ByRef info As IntPtr,
    ByVal attr_x As Int32,
    ByVal attrdata As Int32)
```

[.NET C#]

```
void DshMakeTCACT_PARA(
    ref IntPtr info,
    int attr_x,
    byte[] attrdata );
```

(2) 引数

info

本関数で確保された TCACT_PARA 構造体メモリのポインタを格納するポインタです。

attr_x

属性名を表すインデクス値です。(4)を参照ください。

attrdata

アクション情報に加える属性データが格納されている構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

本関数は APP が S3F17 キャリアアクション要求メッセージ情報に指定属性に基づいて属性データを生成しそれを info に設定します。

attr_x は S3F17 のキャリアアクション属性文字列を示す次のインデクス値で指定します。

attr_x 値	マクロ名	ACTION 属性文字列	データ型
0	CA_ObjId	ObjId	文字列
1	CA_Capacity	Capacity	整数
2	CA_CarrierAccessingStatus	CarrierAccessingStatus	整数
3	CA_CarrierIDStatus	CarrierIDStatus	整数
4	CA_ContentMap	ContentMap	構造体
5	CA_LocationID	LocationID	文字列
6	CA_SlotMap	SlotMap	数値リスト
7	CA_SlotMapStatus	SlotMapStatus	整数

8	CA_SubStrateCount	SubStrateCount	整数
9	CA_Usage	Usage	文字列

最初に TCACT_PARA 構造体のメモリを準備します。そして、attr_x に対応した情報を設定するための属性データ領域を確保します。info に確保したメモリのポインタを設定します。

属性データのタイプによって以下のように TCACT_PARA 構造体メンバー attrdata に情報を設定します。

- 文字列 : メモリを確保し設定する。
- 数値 : attrdata に直接値を設定します。
- 構造体とリスト : 構造体またはリスト用メモリを確保します。

ContentMap の構造体への情報設定は DshPutTCACT_CONTENT() 関数を使用します。また、SlotMap についてはリストへの設定は DshPutTCACT_SLOT_INFO() 関数を使用します。

3.16.3.8 DshFreeTCACT_CONTENT() キャリアコンテンツ情報メモリ開放

(1) 呼出書式

[C, C++]

```
API int APIX DshFreeTCACT_CONTENT(  
    TCACT_CONTENT *info,                // キャリアコンテンツ情報構造体のポインタ
```

[.NET VB]

```
Sub DshFreeTCACT_CONTENT (  
    ByRef info As dsh_info.TCACT_CONTENT)
```

[.NET C#]

```
void DshFreeTCACT_CONTENT(  
    ref TCACT_CONTENT info );
```

(2) 引数

info

キャリアコンテンツ情報が格納されている構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

info で指定される TCACT_CONTENT 構造体内に使用されているメモリを開放します。
開放した後、TCACT_CONTENT の内容を全て 0 で初期設定します。
info が NULL ならば、何も処理しません。

3.16.3.9 DshPutTCACT_CONTENT() TCACT_CONTENT キャリアアクション情報の設定

(1) 呼出書式

[C, C++]

```
API int APIX DshPutTCACT_CONTENT(
    TCACT_PARA *info,           // TCACT_PARA 属性情報構造体のポインタ
    int order,                 // TCACT_CONTENT 内のリストの順位
    char *lotid,              // ロット ID
    cjar *substid             // 基板 ID
);
```

[.NET VB]

```
Function DshPutTCACT_CONTENT (
    ByRef info As dsh_info.TCACT_PARA,
    ByVal order As Int32,
    ByVal lotid As String,
    ByVal substid As String) As Int32
```

[.NET C#]

```
int DshPutTCACT_CONTENT(
    ref TCACT_PARA info,
    int order,
    byte[] lotid,
    byte[] substid );
```

(2) 引数

info

TCACT_PARA 構造体のポインタです。

order

情報を設定する TCACT_CONTENT 内の lotid, substid メンバーリストの配列位置(0,1,)を指定します。

lotid

ロット ID 文字列が格納されている領域のポインタです。

substid

基板 ID 文字列が格納されている領域のポインタです。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	予約されているカウント数を超過していたので設定できなかった。 (TCACT_CONTENT の count 値を超える値)

(4) 説明

本関数は APP が S3F17 キャリアアクション要求メッセージのアクション情報に付属する info で指定された "ContentMap" 属性データの中に lotid と substid の情報を設定します。

具体的には、info 内の attrdata に格納されているポインタが TCACT_CONTENT 構造体のポインタになります。

TCACT_CONTENT 構造体の中の lotid, substid リストメンバーの order 番目の配列位置に設定します。

3.16.3.10 DshPutTCACT_SLOT_INFO() スロットマップデータの設定

(1) 呼出書式

[C, C++]

```
API int APIX DshPutTCACT_CONTENT(
    TCACT_PARA *info,           // TCACT_PARA 属性情報構造体のポインタ
    int order,                 // TCACT_CONTENT 内のリストの順位
    int mapdata                // スロットマップデータ
);
```

[.NET VB]

```
Function DshPutTCACT_SLOT_INFO (
    ByRef info As dsh_info.TCACT_PARA,
    ByVal order As Int32,
    ByVal mapdata As Int32) As Int32
```

[.NET C#]

```
int DshPutTCACT_SLOT_INFO(
    ref TCACT_PARA info,
    int order,
    int mapdata );
```

(2) 引数

info

TCACT_PARA 構造体のポインタです。

order

情報を設定する TCACT_SLOT_INFO 内の slot_list メンバーリストの配列位置を指定します。
(0,1,2,...)

mapdata

設定するスロットマップデータの値です。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	予約されている桁外数を超えていたので設定できなかった。 (TCACT_SLOT_INFO の count 値を超える値)

(4) 説明

本関数は APP が S3F17 キャリアアクション要求メッセージのアクション情報に付属する info で指定された “SlotMap” 属性データの中の order 番目のスロットリストに mapdata の値を設定します。

具体的には、info 内の attrdata に格納されているポインタが TCACT_SLOT_INFO 構造体のポインタになります。

TCACT_SLOT_INFO 構造体の中の slot_list メンバーの order 番目の配列位置に mapdata の値を設定します。

3.16.3.11 DshGetCaActionIndex () キャリアアクションサービスのインデクス取得

(1) 呼出書式

[C, C++]

```
API int APIX DshGetCaActionIndex(
    char *action // アクションサービス文字列
);
```

[.NET VB]

```
Function DshGetCaActionIndex (
    ByVal ca_action As String) As Int32
```

[.NET C#]

```
int DshGetCaActionIndex(
    byte[] ca_action );
```

(2) 引数

action

キャリアアクションサービス名文字列のポインタです。

(3) 戻り値

戻り値	意味
>= 0	正常にインデクス値を取得できた。
(-1)	未定義であった。

(4) 説明

action 領域に格納されているアクションサービス名 (に対応するアクションインデクスを取得する。アクションサービス名とアクションインデクスの関係は下表の通りです。

action インデクス値	マクロ名	ACTION サービス文字列
0	CA_Bind	Bind
1	CA_CancelBind	CancelBind
2	CA_CancelAllCarrierOut	CancelAllCarrierOut
3	CA_CancelCarrier	CancelCarrier
4	CA_CancelCarrierAtPort	CancelCarrierAtPort
5	CA_CancelCarrierNotification	CancelCarrierNotification
6	CA_CancelCarrierOut	CancelCarrierOut
7	CA_CarrierIn	CarrierIn
8	CA_CarrierNotification	CarrierNotification
9	CA_CarrierOut	CarrierOut
10	CA_CarrierReCreate	CarrierReCreate
11	CA_CarrierRelease	CarrierRelease
12	CA_ProceedWithCarrier	ProceedWithCarrier

3.16.3.12 DshGetCaActionName () キャリアアクションインデクスからアクション名取得

(1) 呼出書式

[C, C++]

```
API char* APIX DshGetCaActionName(
    int index // アクションインデクス
);
```

[.NET VB]

N/A

[.NET C#]

N/A

(2) 引数

index

キャリアアクションのインデクスです。

(3) 戻り値

戻り値	意味
!= NULL	正常にアクション名を取得できた。
NULL	未定義であった。

(4) 説明

index で指定されたキャリアアクションインデクスに対するアクション名のポインタを取得する関数です。該当するアクションが存在しない場合はNULL を返却します。

存在する場合はアクション名が格納されているメモリのポインタが返却されます。この場合、ユーザは与えられたポインタのメモリ領域にデータを書き込んではいけません。

キャリアアクション名とアクションインデクスの関係は下表の通りです。

action インデクス値	マクロ名	ACTION サービス文字列
0	CA_Bind	Bind
1	CA_CancelBind	CancelBind
2	CA_CancelAllCarrierOut	CancelAllCarrierOut
3	CA_CancelCarrier	CancelCarrier
4	CA_CancelCarrierAtPort	CancelCarrierAtPort
5	CA_CancelCarrierNotification	CancelCarrierNotification
6	CA_CancelCarrierOut	CancelCarrierOut
7	CA_CarrierIn	CarrierIn
8	CA_CarrierNotification	CarrierNotification
9	CA_CarrierOut	CarrierOut
10	CA_CarrierReCreate	CarrierReCreate
11	CA_CarrierRelease	CarrierRelease
12	CA_ProceedWithCarrier	ProceedWithCarrier

3.16.3.13 DshGetCaAttrIndex () キャリアアクション属性のインデクス取得

(1) 呼出書式

[C, C++]

```
API int APIX DshGetCaAttrIndex(
    char *attrid // アクション属性文字列
);
```

[.NET VB]

```
function DshGetCaAttrIdIndex (
    ByVal attrid As string ) As Int32
```

[.NET C#]

```
int DshGetCaAttrIdIndex(
    byte[] attrid );
```

(2) 引数

attrid

キャリアアクション属性名文字列のポインタです。

(3) 戻り値

戻り値	意味
>= 0	正常にインデクス値を取得できた。
(-1)	未定義であった。

(4) 説明

attrid領域に格納されているアクション属性名(に対応するアクション属性インデクスを取得する。アクション属性名とアクション属性インデクスの関係は下表の通りです。

インデクス値	マクロ名	ACTION 属性文字列
0	CA_ObjId	ObjId
1	CA_Capacity	Capacity
2	CA_CarrierAccessingStatus	CarrierAccessingStatus
3	CA_CarrierIDStatus	CarrierIDStatus
4	CA_ContentMap	ContentMap
5	CA_LocationID	LocationID
6	CA_SlotMap	SlotMap
7	CA_SlotMapStatus	SlotMapStatus
8	CA_SubStrateCount	SubStrateCount
9	CA_Usage	Usage

3.16.3.14 DshGetCaAttrName () キャリアアクション属性インデクスから属性名取得

(1) 呼出書式

[C, C++]

```
API char* APIX DshGetCaAttrName(
    int index // アクション属性インデクス
);
```

[.NET VB]

N/A

[.NET C#]

N/A

(2) 引数

index

キャリアアクション属性インデクスです。

(3) 戻り値

戻り値	意味
!= NULL	正常に属性名を取得できた。
NULL	未定義であった。

(4) 説明

index で指定されたアクション属性インデクスに対する属性名のポインタを取得する関数です。該当する属性が存在しない場合は NULL を返却します。

存在する場合は属性名が格納されているメモリのポインタが返却されます。この場合、ユーザは与えられたポインタのメモリ領域にデータを書き込んではいけません。

アクション属性名とアクション属性インデクスの関係は下表の通りです。

インデクス値	マクロ名	ACTION 属性文字列
0	CA_ObjId	ObjId
1	CA_Capacity	Capacity
2	CA_CarrierAccessingStatus	CarrierAccessingStatus
3	CA_CarrierIDStatus	CarrierIDStatus
4	CA_ContentMap	ContentMap
5	CA_LocationID	LocationID
6	CA_SlotMap	SlotMap
7	CA_SlotMapStatus	SlotMapStatus
8	CA_SubstrateCount	SubstrateCount
9	CA_Usage	Usage

3.16.3.15 DshInitTCACT_ERR_INFO () キャリアアクション要求応答情報の初期化

(1) 呼出書式

A[C,C++]

```

PI void APIX DshInitTCACT_ERR_INFO(
    TCACT_ERR_INFO *erinfo,           // エラー情報格納構造体リストのポインタ
    int caack,                         // ackデータ
    int err_count                      // エラー情報のリストサイズ (個数 0,1,2...)
);

```

[.NET VB]

```

Sub DshInitTCACT_ERR_INFO (
    ByRef erinfo As dsh_info.TCACT_ERR_INFO,
    ByVal caack As Int32,
    ByVal errcount As Int32)

```

[.NET C#]

```

void DshInitTCACT_ERR_INFO(
    ref TCACT_ERR_INFO erinfo,
    int caack,
    int errcount );

```

(2) 引数

erinfo
TCACT_ERR_INFO 応答情報構造体のポインタです。

hcack
caack - ACK の値です。

err_count
エラー情報構造体の数です。 = 0 の場合はエラー情報がないことになります。

(3) 戻り値

なし。

(4) 説明

本関数は、キャリアアクション要求関連応答メッセージ TCACT_ERR_INFO 構造体に初期設定を行います。erinfo で指定された構造体の caack メンバーに引数 caack の値を設定し、err_count メンバーにも引数 err_count の値を設定します。もし、err_count > 0 の場合は、err_list に err_count 分だけのエラー情報のための領域を設けます。

erinfo 内へのエラー情報 (エラーコード、エラーテキスト) の設定には、3.21.2.2 DshPutTOBJ_ERR_INFO ()関数を使用します。

3.16.3.16 DshFreeTCACT_ERR_INFO() - キャリアアクション応答情報メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTCACT_ERR_INFO(  
    TCACT_ERR_INFO *erinfo           // xリを開放したい応答情報が格納されている構造体のポインタ  
);
```

[.NET VB]

```
Sub DshFreeTCACT_ERR_INFO (  
    ByRef info As dsh_info.TCACT_ERR_INFO)
```

[.NET C#]

```
void DshFreeTCACT_ERR_INFO(  
    ref TCACT_ERR_INFO info );
```

(2) 引数

erinfo

メモリを解放したいキャリアアクション応答情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TCACT_ERR_INFO 構造体内で情報格納用に使用されているメモリを全て解放します。

3 . 16 . 3 . 17 DshMakeS3F17Response() - S3F17 の応答メッセージの生成

(1) 呼出書式

[C, C++]

```
API int APIX DshMakeS3F17Response(
    TCACT_INFO *info,           // キャリアアクション要求情報格納領域のポインタ
    TCACT_ERR_INFO *erinfo,    // S3F18 に設定する応答情報格納領域のポインタ
    DSHMSG *msg,               // S3F18 メッセージを格納するメッセージ構造体のポインタ
    BYTE *buff,                // S3F18 のテキスト格納バッファポインタ
    int buff_size              // buff のバイトサイズ
);
```

[.NET VB]

```
Function DshMakeS3F17Response (
    ByRef info As dsh_info.TCACT_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO,
    ByRef msg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS3F17Response(
    ref TCACT_INFO info,
    ref TCACT_ERR_INFO erinfo,
    ref DSHMSG msg,
    byte[] buff,
    int buff_size );
```

(2) 引数

info

キャリアアクション要求情報が格納されている領域のポインタです。

erinfo

S3F18 メッセージに設定する応答情報が格納されている領域のポインタです。

msg

S3F18 応答メッセージ情報を格納するためのメッセージ構造体のポインタです。

buff

S3F18 応答メッセージのテキストを格納するためのバッファポインタです。

buff_size

buff のバイトサイズです。

(3) 戻り値

戻り値	意味
0	正常に生成できた。
(-1)	生成できなかった。(buff 領域不足)

(4) 説明

S3F17 に対する S3F18 応答メッセージを info に含まれるキャリアアクション要求情報と応答情報に従って作成します。

応答情報内の、caack を S3F18 の CAACK として設定します。
CAACK はユーザが S3F17 キャリアアクション要求メッセージを評価した結果です。

erinfo の情報生成、設定には、DshInitTCACT_ERR_INFO ()、DshPutTOBJ_ERR_INFO()関数を使うことができます。

3.16.4 ユーザ作成ライブラリ関数

3.16.4.1 DshResponseS3F18() S3F18 キャリアアクション要求応答メッセージ

(1) 呼出書式

[c, C++]

```
API int APIX DshResponseS3F18(
    int eqid, // 通信対象装置 ID(0,16,...)
    ID_TR trid, // DSHDR2 のトランザクション ID
    TCACT_INFO *info, // キャリアアクションメッセージ 情報格納領域のポインタ TCACT_ERR_INFO *erinfo
    // S3F18 応答情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshResponseS3F18 (
    ByVal eqid As Int32,
    ByVal trid As Int32,
    ByRef info As dsh_info.TCACT_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO) As Int32
```

[.NET C#]

```
int DshResponseS3F18(
    int eqid,
    uint trid,
    ref TCACT_INFO info,
    ref TCACT_ERR_INFO erinfo );
```

(2) 引数

eqid

通信エンジンが通信する対象装置 ID を指定します。装置 ID は 0 から始まる番号です。

trid

S3F17 受信時に DSHGEMLIB から渡される DSHDR2 通信ドライバーのトランザクション管理のための ID です。

info

キャリアアクション要求情報が格納されている構造体のポインタです。

erinfo

送信する応答メッセージ S3F18 に含まれる情報を格納するための構造体領域のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に送信できた。
(-1)	送信できなかった。

(4) 説明

キャリアアクション要求メッセージ S3F17 に対する応答メッセージを送信します。



本関数はユーザ作成ライブラリ DLL (dsh_ulib.dll) に含まれる関数ですが、ここでは DSHGEMLIB パッケージに標準的な関数として付属されているものです。(ユーザ独自による作成も可能です)

引数に指定されている TCACT_ERR_INFO 構造体に含まれている情報から S3F18 メッセージを組み立て、その後、S3F18 メッセージを送信します。

送信が終わったら、TCACT_ERR_INFO の構造体で使用されたメモリを DshFreeTCACT_ERR_INFO () 関数を使って開放します。

なお、S3F18 メッセージの組み立てに、DshMakeS3F18Response() 関数を使用できます。

3.17 ポートアクション、アクセスモード(S3F23,S3F25,S3F27)関連関数

搬入、搬出ポートのアクションならびにロードポートのアクセスモードの変更に関するメッセージ送信処理のために使用できる API 関数とライブラリ関数について説明します。

(1) 送信 API 関数

ポートアクションとアクセスモード指令メッセージ送信に関連するサービスのための API 関数名は一覧表のとおりです。

	API 関数名	機能
1	GemSendS3F23()	S3F23 ポートグループアクション要求メッセージを送信します。
2	GemSendS3F25()	S3F25 ポートアクション要求メッセージを送信します。
3	GemSendS3F27()	S3F27 ポートアクセス変更メッセージを送信します。

(2) ライブラリ関数一覧

	ライブラリ関数名	機能
1	DshDecodeS3F23 ()	S3F23 メッセージ をデコード しポートグループ アクション情報 TPORTG_INFO 構造体に情報を格納します。
2	DshEncodeS3F23()	TPORTG_INFO 構造体に設定されているポートグループ アクション情報を S3F23 メッセージ にエンコード します。
3	DshFreeTPORTG_INFO()	DshDecodeS3F23() で使用したメモリを開放します。
4	DshCopyTPORTG_INFO()	DshDecodeS3F23() で取得した構造体情報をコピー します。
5	DshInitTPORTG_INFO()	ポートグループ アクション情報 TPORTG_INFO 構造体を初期設定 します。
6	DshAddTPORTG_INFO()	ポートグループ アクション情報 TPORTG_INFO 構造体にポート情報を追加 します。
7	DshMakeS3F23Response()	S3F24 応答メッセージ を TCACT_ERR_INFO 構造体内の情報から生成 します。
8	DshDecodeS3F25 ()	S3F25 メッセージ をポートアクション情報 TPORT_INFO 構造体内にデコード します。
9	DshEncodeS3F25()	TPORT_INFO 構造体に設定されているポートアクション情報を S3F25 メッセージ にエンコード します。
10	DshFreeTPORT_INFO()	DshDecodeS3F25() で使用したメモリを開放 します。
11	DshCopyTPORT_INFO()	DshDecodeS3F25() で取得した構造体情報をコピー します。
12	DshInitTPORT_INFO()	ポートアクション情報 TPORT_INFO 構造体を初期設定 します。
13	DshAddTPORT_INFO()	ポートアクション情報 TPORT_INFO 構造体にポート情報を追加 します。
14	DshMakeS3F25Response()	S3F26 応答メッセージ を生成 します。
15	DshDecodeS3F27 ()	S3F27 メッセージ をポートアクセス変更情報構造体内にデコード します。
16	DshEncodeS3F27()	TACCESS_INFO 構造体に設定されているポートアクセス変更情報を S3F27 メッセージ にエンコード します。
17	DshFreeTACCESS_INFO()	DshDecodeS3F27() で使用した TACCESS_INFO 構造体内のメモリを開放 します。
18	DshCopyTACCESS_INFO()	DshDecodeS3F27() で取得した TACCESS_INFO 構造体情報をコピー します。
19	DshInitTACCESS_INFO()	ポートアクセス変更情報 TACCESS_INFO 構造体を初期設定 します。
20	DshAddTACCESS_INFO()	ポートアクセス変更情報 TACCESS_INFO 構造体にポート情報を追加 します。
21	DshInitTACCESS_ERR_INFO()	S3F28 応答用 TACCESS_ERR_INFO 情報構造体を初期化 します。
22	DshPutTACCESS_ERR_PARA()	S3F28 応答用 TACCESS_ERR_INFO 構造体にエラーパラメータ情報を加え ます。

23	DshFreeTACCESS_ERR_INFO()	S3F28 応答情報生成で使した TACCESS_ERR_INFO 構造体内のメモリを開放します。
24	DshMakeS3F27Response()	S3F28 応答メッセージを TACCESS_ERR_INFO と TACCESS_INFO 構造体内の情報から生成します。

(注) S3F23、S3F25 に対する応答メッセージのための応答情報の構造体は S3F17(キャリアアクション)で使用する TCACT_ERR_INFO 構造体と同じものを使用します。したがって TCACT_ERR_INFO の初期設定については S3F17 で説明した関数を参照ください。

(3) ユーザ作成ライブラリ関数

	ライブラリ関数名	機能
1	DshResponseS3F24()	S3F24 ポートグループアクション応答メッセージ
2	DshResponseS3F26()	S3F26 ポートアクション応答メッセージ
3	DshResponseS3F28()	S3F28 ポートアクセスモード応答メッセージ

3.17.1 使用する情報格納構造体

(1) S3F23、S3F24 情報格納用

S3F23 のポートグループアクション情報を以下の構造体に格納します。

```
typedef struct{
    char        *portgrpaction;    // group port action
    char        *portgrpname;     // port group name
    int         pn_count;         // parameter count
    TPORTG_PARA **pn_list;       // paramete list
}TPORTG_INFO;

typedef struct{
    char        *paramname;       // paramname
    int         pval_fmt;        // paramval item fmt
    int         pval_size;       // paramval data array size
    void        *paramval;       // paramval data
}TPORTG_PARA;
```

S3F24 応答情報は、前述の S3F18 で使用した TCACT_ERR_INFO と同じ構造体を使用します。

(2) S3F25 情報格納用

S3F25 のポートアクション情報を以下の構造体に格納します。

```
typedef struct{
    char        *portaction;      // port action
    int         ptn;
    int         pn_count;        // parameter count
    TPORT_PARA **pn_list;       // paramete list
}TPORT_INFO;

typedef struct{
    char        *paramname;       // paramname
    int         pval_fmt;        // paramval item fmt
    int         pval_size;       // paramval data array size
    void        *paramval;       // paramval data
}TPORT_PARA;
```

S3F26 応答情報は、前述の S3F18 で使用した TCACT_ERR_INFO と同じ構造体を使用します。

(3) S3F27 情報格納用

S3F27 のロードポートアクセスモード変更情報を以下の構造体に格納します。

```
typedef struct{
    int         accessmode;      // access mode 0/1
    int         port_count;      // no. of port
    int         *port_list;      // port no. list
}TACCESS_INFO;
```

S3F26 応答情報は、次の構造体を使用します。

```
typedef struct{
    int      caack;
    int      err_count;
    TACCESS_ERR_PORT **err_list;
} TACCESS_ERR_INFO;
```

```
typedef struct{
    int      port;           // port no.
    int      errcode;       // ok/ng - port
    char     *errtext;      // error text - port
} TACCESS_ERR_PORT;
```

3.17.2 ポートアクション関連メッセージ送信関数

3.17.2.1 GemSendS3F23() ポートグループアクションメッセージ送信関数

(1) 呼出書式

[c,C++]

```
API int APIX GemSendS3F23(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    TPORTG_INFO *info, // ポートグループアクション情報格納領域のポインタ
    TCACT_ERR_INFO *erinfo, // S3F24 応答情報格納用構造体のポインタ
    int (WINAPI *callback)(), // 実行終了時のコールバック関数
    ULONG upara // callback 時のパラメータ
);
```

[.NET VB]

```
Function GemSendS3F23 (
    ByVal eqid As Int32,
    ByRef info As dsh_info.TPORTG_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO,
    ByVal callback As vcallback.callback_S3F23,
    ByVal upara As Int32) As Int32
```

[.NET C#]

```
int GemSendS3F23(
    int eqid,
    ref TPORTG_INFO info,
    ref TCACT_ERR_INFO erinfo,
    CallbackS3F17 callback,
    uint upara );
```

(2) 引数

eqid

通信エンジンが通信する対象装置 ID を指定します。装置 ID は 0 から始まる番号です。

info

送信したいポートグループアクション情報が格納されている構造体のポインタです。

erinfo

受信した応答メッセージ S3F24 に含まれる情報を格納するための構造体領域のポインタを指定します。

callback

DSHGEM-LIB によるメッセージ送信処理が終了したときに呼出されるコールバック関数を指定します。ユーザは任意の関数名を指定できます。コールバックの指定が=0 の場合はブロックモードになります。

upara

コールバックされたときに引数で渡してもらうためのパラメータです。関数実行終了時にコールバックされた際、何かの判別情報として使用したい値、構造体ポインタなどを設定します。

(3) 戻り値

戻り値	意味
-----	----

0	(1) プロトコルド : 正常に送信できた。 erinfo に S3F24 応答情報が返却されます。 (2) 非プロトコルド : 要求が受け付けられた。
(-1)	送信できなかった。
(-14)	T3タイムアウトを検出した。

(4) 説明

装置にポートグループアクション情報の送信要求を行います。S3F23 メッセージで送信します。

要求を受けた DSHGEM-LIB は、info に格納されているポートグループアクション情報を S3F23 メッセージにエンコードし、装置に送信します。

S3F24 応答メッセージ受信で得られた情報はデコードされて erinfo で指定された構造体領域に返却されま
す。

送信要求から S3F24 応答メッセージ受信までの制御は引数のコールバック関数指定の有無によって次のよ
うになります。

callback の指定	制御の流れ
なし (=0)	S3F23 送信後、応答メッセージ S3F24 が受信されるか、またはエラーを検出するまで制 御が要求元に戻ってきません。 正常終了であれば、erinfo に S3F24 応答情報が返却されます。
あり	送信要求後、S3F23 の送信前に制御が戻されます。実行結果はコールバック関数で与 えられます。 コールバック関数の end_status が=0 ならば erinfo に S3F24 応答情報が返却されま す。 エラーが検出された場合、(-1)が end_status にセットされます。

正常に応答メッセージを受信した場合、その中に含まれている応答情報がデコードされ erinfo で指定され
た TCACT_ERR_INFO 構造体の中に格納され返却されます。ユーザ側で erinfo 内の情報の処理を終えた後、
その構造体で使用されているメモリを解放してください。

解放は次のように DshFreeTCACT_ERR_INFO()関数を使って行ってください。

```
DshFreeTCACT_ERR_INFO (erinfo)
```

TPORTG_INFO 構造体へのポートグループアクション情報の設定には以下のライブラリ関数を使用するこ
とができます。

```
DshInitTPORTG_INFO(), DshAddTPORTG_INFO()
```

(5) コールバック関数

[C,C++]

```
API int APIX callback(
    int eqid,                // 装置 ID
    int end_status,         // 実行結果
    TCACT_ERR_INFO *erinfo, // S3F24 応答情報格納用構造体のポインタ
    ULONG upara            // 呼出時に指定したパラメータ
);
```


[.NET VB]

```
Function callback_S3F23(ByVal eqid As Integer, ByVal end_status As Integer, ByRef erinfo As  
dsh_info.TCACT_ERR_INFO, ByVal upara As Integer) As Integer
```

[.NET C#]

```
int CallbackS3F23(int eqid, int status, ref TCACT_ERR_INFO erinfo, uint upara);
```

end_status には以下の値が設定されます。

結果	意味
0	正常に送信できた。erinfo に S3F24 応答情報が返却されます。
(-1)	送信できなかった。
(-14)	T3タイムアウトを検出した。

3.17.2.2 GemSendS3F25() ポートアクションメッセージ送信関数

(1) 呼出書式

[C, C++]

```
API int APIX GemSendS3F25(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    TPORT_INFO *info, // ポートアクション情報格納領域のポインタ
    TCACT_ERR_INFO *erinfo, // S3F26 応答情報格納用構造体のポインタ
    int (WINAPI *callback)(), // 実行終了時のコールバック関数
    ULONG upara // callback 時のパラメータ
);
```

[.NET VB]

```
Function GemSendS3F25 (
    ByVal eqid As Int32,
    ByRef info As dsh_info.TPORT_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO,
    ByVal callback As vcallback.callback_S3F25,
    ByVal upara As Int32) As Int32
```

[.NET C#]

```
int GemSendS3F25(
    int eqid,
    ref TPORT_INFO info,
    ref TCACT_ERR_INFO erinfo,
    CallbackS3F17 callback,
    uint upara );
```

(2) 引数

eqid

通信エンジンが通信する対象装置 ID を指定します。装置 ID は 0 から始まる番号です。

info

送信したいポートアクション情報が格納されている構造体のポインタです。

erinfo

受信した応答メッセージ S3F26 に含まれる情報を格納するための構造体領域のポインタを指定します。

callback

DSHGEM-LIB によるメッセージ送信処理が終了したときに呼出されるコールバック関数を指定します。ユーザは任意の関数名を指定できます。コールバックの指定が=0 の場合はブロックモードになります。

upara

コールバックされたときに引数で渡してもらうためのパラメータです。関数実行終了時にコールバックされた際、何かの判別情報として使用したい値、構造体ポインタなどを設定します。

(3) 戻り値

戻り値	意味
0	(1) ブロックモード：正常に送信できた。

	erinfo に S3F26 応答情報が返却されます。 (2) 非ブロックモード：要求が受け付けられた。
(-1)	送信できなかった。
(-14)	T3タイムアウトを検出した。

(4) 説明

装置にポートアクション情報の送信要求を行います。S3F25 メッセージで送信します。

要求を受けた DSHGEM-LIB は、info に格納されているポートアクション情報を S3F25 メッセージにエンコードし、装置に送信します。

S3F26 応答メッセージ受信で得られた情報はデコードされて erinfo で指定された構造体領域に返却されません。

送信要求から S3F26 応答メッセージ受信までの制御は引数のコールバック関数指定の有無によって次のようになります。

callback の指定	制御の流れ
なし (=0)	S3F25 送信後、応答メッセージ S3F26 が受信されるか、またはエラーを検出するまで制御が要求元に戻ってきません。 正常終了であれば、erinfo に S3F26 応答情報が返却されます。
あり	送信要求後、S3F25 の送信前に制御が戻されます。実行結果はコールバック関数で与えられます。 コールバック関数の end_status が=0 ならば erinfo に S3F26 応答情報が返却されません。 エラーが検出された場合、(-1) が end_status にセットされます。

正常に応答メッセージを受信した場合、その中に含まれている応答情報がデコードされ erinfo で指定された TCACT_ERR_INFO 構造体の中に格納され返却されます。ユーザ側で erinfo 内の情報の処理を終えた後、その構造体で使用されているメモリを解放してください。

解放は次のように DshFreeTCACT_ERR_INFO() 関数を使って行ってください。

```
DshFreeTCACT_ERR_INFO (erinfo)
```

TPOINT_INFO 構造体へのポートアクション情報の設定には以下のライブラリ関数を使用することができます。

```
DshInitTPOINT_INFO(), DshAddTPOINT_INFO()
```

(5) コールバック関数

[C, C++]

```
API int APIX callback(
    int eqid, // 装置 ID
    int end_status, // 実行結果
    TCACT_ERR_INFO *erinfo, // S3F26 応答情報格納用構造体のポインタ
    ULONG upara // 呼出時に指定したパラメータ
);
```

[.NET VB]

Function callback_S3F25(ByVal eqid As Integer, ByVal end_status As Integer, ByRef erinfo As dsh_info.TCACT_ERR_INFO, ByVal upara As Integer) As Integer

[.NET C#]

```
int CallbackS3F25(int eqid, int status, ref TCACT_ERR_INFO erinfo, uint upara);
```

end_status には以下の値が設定されます。

結果	意味
0	正常に送信できた。erinfo に S3F26 応答情報が返却されます。
(-1)	送信できなかった。
(-14)	T3タイムアウトを検出した。

3.17.2.3 GemSendS3F27() アクセス変更メッセージ送信関数

(1) 呼出書式

[C, C++]

```
API int APIX GemSendS3F27(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    TACCESS_INFO *info, // アクセス変更情報格納領域のポインタ
    TACCESS_ERR_INFO *erinfo, // S3F28 応答情報格納用構造体のポインタ
    int (WINAPI *callback)(), // 実行終了時のコールバック関数
    ULONG upara // callback 時のパラメータ
);
```

[.NET VB]

```
Function GemSendS3F27 (
    ByVal eqid As Int32,
    ByRef info As dsh_info.TACCESS_INFO,
    ByRef erinfo As dsh_info.TACCESS_ERR_INFO,
    ByVal callback As vcallback.callback_S3F27,
    ByVal upara As Int32) As Int32
```

[.NET C#]

```
int GemSendS3F27(
    int eqid,
    ref TACCESS_INFO info,
    ref TACCESS_ERR_INFO erinfo,
    CallbackS3F27 callback,
    uint upara );
```

(2) 引数

eqid

通信エンジンが通信する対象装置 ID を指定します。装置 ID は 0 から始まる番号です。

info

送信したいアクセス変更情報が格納されている構造体のポインタです。

erinfo

受信した応答メッセージ S3F28 に含まれる情報を格納するための構造体領域のポインタを指定します。

callback

DSHGEM-LIB によるメッセージ送信処理が終了したときに呼出されるコールバック関数を指定します。ユーザは任意の関数名を指定できます。コールバックの指定が=0 の場合はブロックモードになります。

upara

コールバックされたときに引数で渡してもらうためのパラメータです。関数実行終了時にコールバックされた際、何かの判別情報として使用したい値、構造体ポインタなどを設定します。

(3) 戻り値

戻り値	意味
0	(1) ブロックモード : 正常に送信できた。 erinfo に S3F28 応答情報が返却されます。

	(2) 非プロトコル: 要求が受け付けられた。
(-1)	送信できなかった。
(-14)	T3タイムアウトを検出した。

(4) 説明

装置にアクセス変更情報の送信要求を行います。S3F27 メッセージで送信します。

要求を受けた DSHGEM-LIB は、info に格納されているアクセス変更情報を S3F27 メッセージにエンコードし、装置に送信します。

S3F28 応答メッセージ受信で得られた情報はデコードされて erinfo で指定された構造体領域に返却されません。

送信要求から S3F28 応答メッセージ受信までの制御は引数のコールバック関数指定の有無によって次のようになります。

callback の指定	制御の流れ
なし (=0)	S3F27 送信後、応答メッセージ S3F28 が受信されるか、またはエラーを検出するまで制御が要求元に戻ってきません。 正常終了であれば、erinfo に S3F28 応答情報が返却されます。
あり	送信要求後、S3F27 の送信前に制御が戻されます。実行結果はコールバック関数で与えられます。 コールバック関数の end_status が=0 ならば erinfo に S3F28 応答情報が返却されません。 エラーが検出された場合、(-1) が end_status にセットされます。

正常に応答メッセージを受信した場合、その中に含まれている応答情報がデコードされ erinfo で指定された TACCESS_ERR_INFO 構造体の中に格納され返却されます。ユーザ側で erinfo 内の情報の処理を終えた後、その構造体で使用されているメモリを解放してください。

解放は次のように DshFreeTACCESS_ERR_INFO() 関数を使って行ってください。

```
DshFreeTACCESS_ERR_INFO (erinfo)
```

TACCESS_INFO 構造体へのアクセス変更情報の設定には以下のライブラリ関数を使用することができます。

```
DshInitTACCESS_INFO(), DshAddTACCESS_INFO()
```

(5) コールバック関数

[C, C++]

```
API int APIX callback(
    int eqid, // 装置 ID
    int end_status, // 実行結果
    TACCESS_ERR_INFO *erinfo, // S3F28 応答情報格納用構造体のポインタ
    ULONG upara // 呼出時に指定したパラメータ
);
```

[.NET VB]

```
Function callback_S3F27(ByVal eqid As Integer, ByVal end_status As Integer, ByRef erinfo As
```

dsh_info.TACCESS_ERR_INFO, ByVal upara As Integer) As Integer

[.NET C#]

```
int CallbackS3F27(int eqid, int end_status, ref TACCESS_ERR_INFO erinfo, uint upara);
```

end_status には以下の値が設定されます。

結果	意味
0	正常に送信できた。erinfo に S3F28 応答情報が返却されます。
(-1)	送信できなかった。
(-14)	T3タイムアウトを検出した。

3.17.3 ポート関連情報処理ライブラリ関数

3.17.3.1 DshDecodeS3F23 - S3F23 デコード関数

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS3F23(
    DSHMSG *smsg, // SECS メッセージ 情報構造体のポインタ
    TPORTG_INFO *pinfo // デコードした情報を格納する構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS3F23 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef info As dsh_info.TPORTG_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS3F23(
    ref DSHMSG smsg,
    ref TPORTG_INFO info );
```

(2) 引数

smsg

S3F23 の SECS メッセージ 情報が格納されている構造体のポインタです。

pinfo

デコードしたポートグループアクション情報を格納する構造体のポインタです。

(3) 戻り値

戻り値	意味
0	正常にデコードできた。
(-1)	smsg を正しくデコードできなかった。

(4) 説明

S3F23 メッセージに含まれるポートグループアクション情報を、ユーザプログラムが処理しやすい TPORTG_INFO 構造体の中にデコードします。

なお、構造体使用後は、構造体内部で使用されたメモリを DshFreeTPORTG_INFO() 関数を使って開放してください。

smsg S3F23

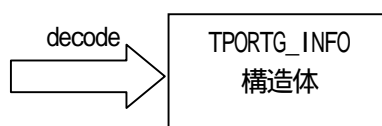
L,3

pgrpaction
portgrpname
Lm

L2

paramname
paramval

.



3.17.3.2 DshEncodeS3F23() - ポートグループアクション情報を S3F23 へエンコード

(1) 呼出書式

[C, C++]

```
API int APIX DshEncodeS3F23(
    DSHMSG *smsg,           // SECS メッセージ 情報構造体のポインタ
    BYTE *buffer,          // S3F23 を格納するバッファポインタ
    int buflen,            // buffer のバイトサイズ
    TPORTG_INFO *info      // エンコードしたいポートグループアクション情報格納構造体のポインタ
);
```

[.NET VB]

```
Function DshEncodeS3F23 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buflen As Int32,
    ByRef info As dsh_info.TPORTG_INFO) As Int32
```

[.NET C#]

```
int DshEncodeS3F23(
    ref DSHMSG smsg,
    byte[] buff,
    int buflen,
    ref TPORTG_INFO info);
```

(2) 引数

smsg

エンコードした S3F23 メッセージを格納するメッセージ情報構造体のポインタです。

buffer

エンコードした S3F23 のテキストを格納するバッファポインタです。

buflen

buffer のバイトサイズです。

info

エンコードしたいポートグループアクション情報が格納されている構造体のポインタです。

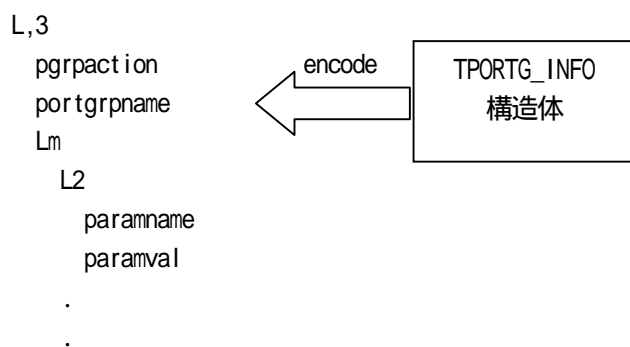
(3) 戻り値

戻り値	意味
0	正常にエンコードできた。
(-1)	smsg を正しくエンコードできなかった。 (buffer の容量不足)

(4) 説明

TPORTG_INFO 構造体に格納されているポートグループアクション情報を、S3F23 の SECS メッセージにエンコードします。

smmsg S3F23



TPORTG_INFO 構造体へのポートグループアクション情報の設定には以下のライブラリ関数を使用することができます。

DshInitTPORTG_INFO(), DshAddTPORTG_INFO()

3.17.3.3 DshFreeTPORTG_INFO() ポートグループアクション情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTPORTG_INFO(  
    TPORTG_INFO *pinfo // メリを開放したい情報が格納されている構造体のポインタ  
);
```

[.NET VB]

```
Sub DshFreeTPORTG_INFO (  
    ByRef info As dsh_info.TPORTG_INFO)
```

[.NET C#]

```
void DshFreeTPORTG_INFO(  
    ref TPORTG_INFO info );
```

(2) 引数

pinfo

メモリを解放したいポートグループアクション情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TPORTG_INFO 構造体内で情報格納用に使用されているメモリを全て解放します。

開放した後、TPORTG_INFO の内容を全て 0 で初期設定します。

pinfo が NULL ならば、何も処理しません。

3.17.3.4 DshCopyTPORTG_INFO() ポートグループアクション情報構造体メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTPORTG_INFO(
    TPORTG_INFO *dinfo,           // 北°-先のポインタ
    TPORTG_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTPORTG_INFO (
    ByRef dinfo As dsh_info.TPORTG_INFO,
    ByRef sinfo As dsh_info.TPORTG_INFO) As Int32
```

[.NET C#]

```
int DshCopyTPORTG_INFO(
    ref TPORTG_INFO dinfo,
    ref TPORTG_INFO sinfo );
```

(2) 引数

dinfo

ポートグループアクション情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のポートグループアクション情報が格納されている構造体メモリのポインタです。

(3) 戻り値

戻り値	意味
0	正常に北°-できた。
(-1)	sinfo または dinfo の値が NULL だったので北°-できなかった。

(4) 説明

sinfo が指す TPORTG_INFO 構造体内に格納されているポートグループアクション情報を dinfo が指定する TPORTG_INFO 構造体にコピーします。

dinfo 内のメンバーで新しいメモリが必要なものは本関数が取得します。

dinfo 内メンバーで確保されたメモリは、使用后、DshFreeTPORTG_INFO()関数を使って開放してください。

3.17.3.5 DshInitTPORTG_INFO ポートグループアクション情報 TPORTG_INFO の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTPORTG_INFO(
    TPORTG_INFO *info,           // ポートグループアクション情報 TPORTG_INFO 構造体のポインタ
    char *paction,              // アクション名
    char *portgrpname,         // ポートグループ名
    int para_count              // 設定できるパラメータ数
);
```

[.NET VB]

```
Sub DshInitTPORTG_INFO (
    ByRef info As dsh_info.TPORTG_INFO,
    ByVal paction As String,
    ByVal ptn As Int32,
    ByVal para_count As Int32)
```

[.NET C#]

```
void DshInitTPORTG_INFO(
    ref TPORTG_INFO info,
    byte[] paction,
    int ptn,
    int para_count );
```

(2) 引数

info
ポートグループアクション情報構造体のポインタです。このメンバーを初期設定します。

paction
アクション名が格納されているポインタです。

portgrpname
ポートグループ名が格納されているポインタです。

para_count
ポートグループアクション情報の中に設定できるアクションパラメータの数です。

(3) 戻り値

なし。

(4) 説明

info で指定された TPORTG_INFO 構造体内に paction で指定されたアクション名と portgrpname で指定されたグループ名を設定します。また para_count 分のアクションパラメータを格納できるリストを生成します。

info にパラメータを加えるためには DshAddTPORTG_INFO()関数を使用してください。

TPORTG_INFO 構造体の使用後は DshFreeTPORTG_INFO()関数を使って構造体内部で使用したメモリを開放してください。

3.17.3.6 DshAddTPORTG_INFO() ポートグループアクションパラメータの追加

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTPORTG_INFO(
    TPORTG_INFO *info,           // ポートグループアクション情報構造体のポインタ
    char *pname,                 // アクションパラメータ名
    int fmt,                     // アクションパラメータのフォーマット( ICODE_A, ICODE_U1 etc )
    int asize,                   // アクションパラメータデータの配列サイズ
    void *pval                   // アクションパラメータデータ格納ポインタ
);
```

[.NET VB]

```
Function DshAddTPORTG_INFO (
    ByRef info As dsh_info.TPORTG_INFO,
    ByVal pname As String,
    ByVal fmt As Int32,
    ByVal asize As Int32,
    ByVal pval As Int32) As Int32
```

[.NET C#]

```
int DshAddTPORTG_INFO(
    ref TPORTG_INFO info,
    byte[] pname,
    int fmt,
    int asize,
    byte[] pval );
```

(2) 引数

info

ポートグループアクション情報構造体のポインタです。

pname

加えたいアクションパラメータ名が格納されているポインタです。

fmt

アクションパラメータデータのフォーマットです。(ICODE_A, ICODE_U1 など)

asize

アクションパラメータデータの配列サイズです。

pval

アクションパラメータデータが格納されているポインタです。

(3) 戻り値

戻り値	意味
0	正常に追加できた。
(-1)	アクションパラメータの数が既に指定数に達している。

(4) 説明

先にDshInitTPORTG_INFO()で初期設定されたポートグループアクション情報構造体 info にアクションパラメータを1個追加します。追加はリストの空き位置に行います。

設定は、新たに TPORTG_PARA パラメータ構造体のメモリを確保し、その中に pname と pval を設定し、pn_list にパラメータ構造体のポインタを設定します。

設定後 0 を返却します。

もし、info 内の pn_count で指定された分の情報が既に設定済みであった場合は、(-1)を返却します。

3.17.3.7 DshMakeS3F23Response() - S3F23 の応答メッセージの生成

(1) 呼出書式

[C, C++]

```
API int APIX DshMakeS3F23Response(
    TPORTG_INFO *info,           // ポートグループアクション情報格納領域のポインタ
    TCACT_ERR_INFO *erinfo,     // S3F24 に設定する応答情報格納領域のポインタ
    DSHMSG *msg,                // S3F24 メッセージを格納するメッセージ構造体のポインタ
    BYTE *buff,                 // S3F24 のテキスト格納バッファポインタ
    int buff_size                // buff のバイトサイズ
);
```

[.NET VB]

```
Function DshMakeS3F23Response (
    ByRef info As dsh_info.TPORTG_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO,
    ByRef msg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS3F23Response(
    ref TPORTG_INFO info,
    ref TCACT_ERR_INFO erinfo,
    ref DSHMSG msg,
    byte[] buff,
    int buff_size );
```

(2) 引数

info

ポートグループアクション情報が格納されている領域のポインタです。

erinfo

S3F24 メッセージに設定する応答情報が格納されている領域のポインタです。

msg

S3F24 応答メッセージ情報を格納するためのメッセージ構造体のポインタです。

buff

S3F24 応答メッセージのテキストを格納するためのバッファポインタです。

buff_size

buff のバイトサイズです。

(3) 戻り値

戻り値	意味
0	正常に生成できた。
(-1)	生成できなかった。(buff 領域不足)

(4) 説明

S3F23 に対する S3F24 応答メッセージを info に含まれるポートグループアクション情報と応答情報に従って作成します。

応答情報内の、caack を S3F24 の CAACK として設定します。

CAACK はユーザが S3F23 ポートグループアクションメッセージを評価した結果です。

erinfo の情報生成には、DshInitTCACT_ERR_INFO()、DshPutTOBJ_ERR_INFO()関数を使うことができます。

3.17.3.8 DshDecodeS3F25 - S3F25 デコード関数

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS3F25(
    DSHMSG *msg, // SECS メッセージ 情報構造体のポインタ
    TPORT_INFO *pinfo // デコードした情報を格納する構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS3F25 (
    ByRef msg As dshdr2.DSHMSG,
    ByRef info As dsh_info.TPORT_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS3F25(
    ref DSHMSG msg,
    ref TPORT_INFO info );
```

(2) 引数

msg

S3F25 の SECS メッセージ 情報が格納されている構造体のポインタです。

pinfo

デコードしたポートアクション情報を格納する構造体のポインタです。

(3) 戻り値

戻り値	意味
0	正常にデコードできた。
(-1)	msg を正しくデコードできなかった。

(4) 説明

S3F25 メッセージに含まれるポートアクション情報を、ユーザプログラムが処理しやすい TPORT_INFO 構造体の中にデコードします。

なお、構造体使用後は、構造体内部で使用されたメモリを DshFreeTPORT_INFO() 関数を使って開放してください。

msg S3F25

L,3

portaction

ptn

Lm

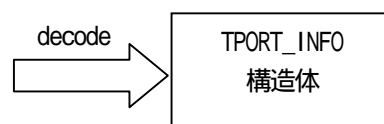
L2

paramname

paramval

.

.



3.17.3.9 DshEncodeS3F25() - ポートアクション情報を S3F25 へエンコード

(1) 呼出書式

[C, C++]

```
API int APIX DshEncodeS3F25(
    DSHMSG *smsg,           // SECS メッセージ 情報構造体のポインタ
    BYTE *buffer,          // S3F25 を格納するバッファポインタ
    int buflen,            // buffer のバイトサイズ
    TPORT_INFO *info       // エンコードしたいポートアクション情報格納構造体のポインタ
);
```

[.NET VB]

```
Function DshEncodeS3F25 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buflen As Int32,
    ByRef info As dsh_info.TPORT_INFO) As Int32
```

[.NET C#]

```
int DshEncodeS3F25(
    ref DSHMSG smsg,
    byte[] buff,
    int buflen,
    ref TPORT_INFO info );
```

(2) 引数

smsg

エンコードした S3F25 メッセージを格納するメッセージ情報構造体のポインタです。

buffer

エンコードした S3F25 のテキストを格納するバッファポインタです。

buflen

buffer のバイトサイズです。

info

エンコードしたいポートアクション情報が格納されている構造体のポインタです。

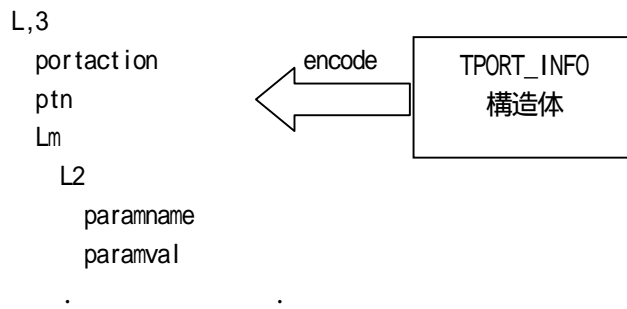
(3) 戻り値

戻り値	意味
0	正常にエンコードできた。
(-1)	smsg を正しくエンコードできなかった。 (buffer の容量不足)

(4) 説明

TPORT_INFO 構造体に格納されているポートアクション情報を、S3F25 の SECS メッセージにエンコードします。

smmsg S3F25



TPORT_INFO 構造体へのポートアクション情報の設定には以下のライブラリ関数を使用することができます。
DshInitTPORT_INFO(), DshAddTPORT_INFO()

3.17.3.10 DshFreeTPORT_INFO() ポートアクション情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTPORT_INFO(  
    TPORT_INFO *pinfo // メモリを開放したい情報が格納されている構造体のポインタ  
);
```

[.NET VB]

```
Sub DshFreeTPORT_INFO (  
    ByRef info As dsh_info.TPORT_INFO)
```

[.NET C#]

```
void DshFreeTPORT_INFO(  
    ref TPORT_INFO info );
```

(2) 引数

pinfo

メモリを解放したいポートアクション情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TPORT_INFO 構造体内で情報格納用に使用されているメモリを全て解放します。

開放した後、TPORT_INFO の内容を全て 0 で初期設定します。

pinfo が NULL ならば、何も処理しません。

3.17.3.11 DshCopyTPORT_INFO() ポートアクション情報構造体メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTPORT_INFO(
    TPORT_INFO *dinfo,           // 北°-先のポインタ
    TPORT_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTPORT_INFO (
    ByRef dinfo As dsh_info.TPORT_INFO,
    ByRef sinfo As dsh_info.TPORT_INFO) As Int32
```

[.NET C#]

```
int DshCopyTPORT_INFO(
    ref TPORT_INFO dinfo,
    ref TPORT_INFO sinfo );
```

(2) 引数

dinfo

ポートアクション情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のポートアクション情報が格納されている構造体メモリのポインタです。

(3) 戻り値

戻り値	意味
0	正常に北°-できた。
(-1)	sinfo または dinfo の値が NULL だったので北°-できなかった。

(4) 説明

sinfo が指す TPORT_INFO 構造体内に格納されているポートアクション情報を dinfo が指定する TPORT_INFO 構造体にコピーします。

dinfo 内のメンバーで新しいメモリが必要なものは本関数が取得します。

dinfo 内メンバーで確保されたメモリは、使用后、DshFreeTPORT_INFO()関数を使って開放してください。

3.17.3.12 DshInitTPORT_INFO ポートアクション情報 TPORT_INFO の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTPORT_INFO(
    TPORT_INFO *info,           // ポートアクション情報 TPORT_INFO 構造体のポインタ
    char *paction,             // アクション名
    int ptn,                   // ポート番号
    int para_count             // 設定できるパラメータ数
);
```

[.NET VB]

```
Sub DshInitTPORT_INFO (
    ByRef info As dsh_info.TPORT_INFO,
    ByVal paction As String,
    ByVal ptn As Int32,
    ByVal para_count As Int32)
```

[.NET C#]

```
void DshInitTPORT_INFO(
    ref TPORT_INFO info,
    byte[] paction,
    int ptn,
    int para_count );
```

(2) 引数

info
ポートアクション情報構造体のポインタです。このメンバーを初期設定します。

paction
アクション名が格納されているポインタです。

ptn
アクションを要求する対象のポート番号です。

para_count
ポートアクション情報の中に設定できるアクションパラメータの数です。

(3) 戻り値

なし。

(4) 説明

info で指定された TPORT_INFO 構造体内部に paction で指定されたアクション名と ptn で指定されたポート番号を設定します。また para_count 分のアクションパラメータを格納できるリストを生成します。

info にパラメータを加えるためには DshAddTPORT_INFO() 関数を使用してください。

TPORT_INFO 構造体の使用後は DshFreeTPORT_INFO() 関数を使って構造体内部で使用したメモリを開放してください。

3.17.3.13 DshAddTPORT_INFO() ポートアクションパラメータの追加

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTORT_INFO(
    TPORT_INFO *info,           // ポートアクション情報構造体のポインタ
    char *pname,                // アクションパラメータ名
    int fmt,                    // アクションパラメータのフォーマット( ICODE_A, ICODE_U1 etc )
    int asize,                  // アクションパラメータデータの配列サイズ
    void *pval                  // アクションパラメータデータ格納ポインタ
);
```

[.NET VB]

```
Function DshAddTPORT_INFO (
    ByRef info As dsh_info.TPORT_INFO,
    ByVal pname As String,
    ByVal fmt As Int32,
    ByVal asize As Int32,
    ByVal pval As Int32) As Int32
```

[.NET C#]

```
int DshAddTPORT_INFO(
    ref TPORT_INFO info,
    byte[] pname,
    int fmt,
    int asize,
    byte[] pval );
```

(2) 引数

info

ポートアクション情報構造体のポインタです。

pname

加えたいアクションパラメータ名が格納されているポインタです。

fmt

アクションパラメータデータのフォーマットです。(ICODE_A, ICODE_U1 など)

asize

アクションパラメータデータの配列サイズです。

pval

アクションパラメータデータが格納されているポインタです。

(3) 戻り値

戻り値	意味
0	正常に追加できた。
(-1)	アクションパラメータの数が既に指定数に達している。

(4) 説明

先に DshInitTPORT_INFO() で初期設定されたポートアクション情報構造体 info にアクションパラメータを 1 個追加します。追加はリストの空き位置に行います。

設定は、新たに TPORT_PARA パラメータ構造体のメモリを確保し、その中に pname と pval を設定し、pn_list にパラメータ構造体のポインタを設定します。

設定後 0 を返却します。

もし、info 内の pn_count で指定された分の情報が既に設定済みであった場合は、(-1)を返却します。

3.17.3.14 DshMakeS3F25Response() - S3F25 の応答メッセージの生成

(1) 呼出書式

[C, C++]

```
API int APIX DshMakeS3F25Response(
    TPORT_INFO *info,           // ポートアクション情報格納領域ポインタ
    TCACT_ERR_INFO *erinfo,    // S3F26 に設定する応答情報格納領域のポインタ
    DSHMSG *msg,               // S3F26 メッセージを格納するメッセージ構造体のポインタ
    BYTE *buff,                // S3F26 のテキスト格納バッファポインタ
    int buff_size               // buff のバイトサイズ
);
```

[.NET VB]

```
Function DshMakeS3F25Response (
    ByRef info As dsh_info.TPORT_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO,
    ByRef msg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS3F25Response(
    ref TPORT_INFO info,
    ref TCACT_ERR_INFO erinfo,
    ref DSHMSG msg,
    byte[] buff,
    int buff_size );
```

(2) 引数

info

ポートアクション情報が格納されている領域のポインタです。

erinfo

S3F26 メッセージに設定する応答情報が格納されている領域のポインタです。

msg

S3F26 応答メッセージ情報を格納するためのメッセージ構造体のポインタです。

buff

S3F26 応答メッセージのテキストを格納するためのバッファポインタです。

buff_size

buff のバイトサイズです。

(3) 戻り値

戻り値	意味
0	正常に生成できた。
(-1)	生成できなかった。(buff 領域不足)

(4) 説明

S3F25 に対する S3F26 応答メッセージを info に含まれるポートアクション情報と応答情報に従って作成します。

応答情報内の、caack を S3F26 の CAACK として設定します。
CAACK はユーザが S3F25 ポートアクションメッセージを評価した結果です。

erinfo の情報生成には、DshInitTCACT_ERR_INFO()、DshPutTOBJ_ERR_INFO()関数を使うことができます。

3.17.3.15 DshDecodeS3F27 - S3F27 デコード関数

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS3F27(
    DSHMSG *msg, // SECS メッセージ情報構造体のポインタ
    TACCESS_INFO *pinfo // デコードした情報を格納する構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS3F27 (
    ByRef msg As dshdr2.DSHMSG,
    ByRef pinfo As dsh_info.TACCESS_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS3F27(
    ref DSHMSG msg,
    ref TACCESS_INFO pinfo );
```

(2) 引数

msg

S3F27 の SECS メッセージ情報が格納されている構造体のポインタです。

pinfo

デコードしたポートアクセス変更情報を格納する構造体のポインタです。

(3) 戻り値

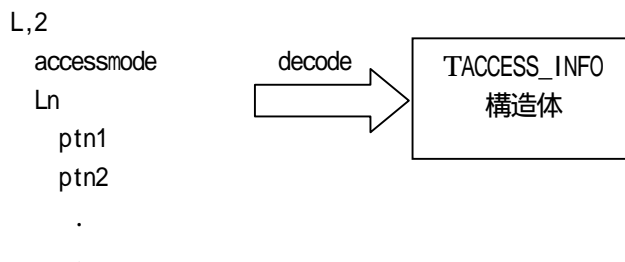
戻り値	意味
0	正常にデコードできた。
(-1)	msg を正しくデコードできなかった。

(4) 説明

S3F27 メッセージに含まれるポートアクセス変更情報を、ユーザプログラムが処理しやすい TACCESS_INFO 構造体の中にデコードします。

なお、構造体使用後は、構造体内部で使用されたメモリを DshFreeTACCESS_INFO() 関数を使って開放してください。

msg S3F27



3.17.3.16 DshEncodeS3F27() - ポートアクセス情報を S3F27 へエンコード

(1) 呼出書式

[C, C++]

```
API int APIX DshEncodeS3F27(
    DSHMSG *smsg,           // SECS メッセージ 情報構造体のポインタ
    BYTE *buffer,          // S3F27 を格納するバッファポインタ
    int buflen,            // buffer のバイトサイズ
    TACCESS_INFO *info     // エンコードしたいポートアクセス情報格納構造体のポインタ
);
```

[.NET VB]

```
Function DshEncodeS3F27 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buflen As Int32,
    ByRef info As dsh_info.TACCESS_INFO) As Int32
```

[.NET C#]

```
int DshEncodeS3F27(
    ref DSHMSG smsg,
    byte[] buff,
    int buflen,
    ref TACCESS_INFO info );
```

(2) 引数

smsg

エンコードした S3F27 メッセージを格納するメッセージ情報構造体のポインタです。

buffer

エンコードした S3F27 のテキストを格納するバッファポインタです。

buflen

buffer のバイトサイズです。

info

エンコードしたいポートアクセス情報が格納されている構造体のポインタです。

(3) 戻り値

戻り値	意味
0	正常にエンコードできた。
(-1)	smsg を正しくエンコードできなかった。 (buffer の容量不足)

(4) 説明

TACCESS_INFO 構造体に格納されているポートアクセス情報を、S3F27 の SECS メッセージにエンコードします。

msg S3F27

L,2

accessmode

L,n

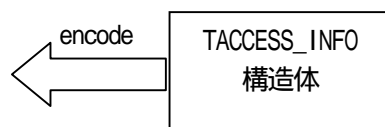
ptn1

ptn2

ptn3

.

.



TACCESS_INFO 構造体へのポートアクセス情報の設定には以下のライブラリ関数を使用することができます。

DshInitTACCESS_INFO(), DshAddTACCESS_INFO()

3.17.3.17 DshFreeTACCESS_INFO() ポートアクセス変更情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTACCESS_INFO(  
    TACCESS_INFO *pinfo // メモリを開放したい情報が格納されている構造体のポインタ  
);
```

[.NET VB]

```
Sub DshFreeTACCESS_INFO (  
    ByRef info As dsh_info.TACCESS_INFO)
```

[.NET C#]

```
void DshFreeTACCESS_INFO(  
    ref TACCESS_INFO info );
```

(2) 引数

pinfo

メモリを解放したいポートアクセス変更情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TACCESS_INFO 構造体内で情報格納用に使用されているメモリを全て解放します。

開放した後、TACCESS_INFO の内容を全て 0 で初期設定します。

pinfo が NULL ならば、何も処理しません。

3.17.3.18 DshCopyTACCESS_INFO() ポートアクセス変更情報構造体メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTACCESS_INFO(
    TACCESS_INFO *dinfo,           // 北°-先のポインタ
    TACCESS_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTACCESS_INFO (
    ByRef dinfo As dsh_info.TACCESS_INFO,
    ByRef sinfo As dsh_info.TACCESS_INFO) As Int32
```

[.NET C#]

```
int DshCopyTACCESS_INFO(
    ref TACCESS_INFO dinfo,
    ref TACCESS_INFO sinfo );
```

(2) 引数

dinfo

ポートアクセス変更情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のポートアクセス変更情報が格納されている構造体メモリのポインタです。

(3) 戻り値

戻り値	意味
0	正常に北°-できた。
(-1)	sinfo または dinfo の値が NULL だったので北°-できなかった。

(4) 説明

sinfo が指す TACCESS_INFO 構造体内に格納されているポートアクセス変更情報を dinfo が指定する TACCESS_INFO 構造体にコピーします。

dinfo 内のメンバーで新しいメモリが必要なものは本関数が取得します。

dinfo 内メンバーで確保されたメモリは、使用后、DshFreeTACCESS_INFO()関数を使って開放してください。

3.17.3.19 DshInitTACCESS_INFO ポートアクセス変更情報 TACCESS_INFO の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTACCESS_INFO(
    TACCESS_INFO *info,           // ポートアクセス情報 TACCESS_INFO 構造体のポインタ
    int accessmode,              // アクセスモード (0/1)
    int port_count               // 対象となるポート数
);
```

[.NET VB]

```
Sub DshInitTACCESS_INFO (
    ByRef info As dsh_info.TACCESS_INFO,
    ByVal accessmode As Int32,
    ByVal port_count As Int32)
```

[.NET C#]

```
void DshInitTACCESS_INFO(
    ref TACCESS_INFO info,
    int accessmode,
    int port_count );
```

(2) 引数

info

ポートアクセス情報構造体のポインタです。このメンバーを初期設定します。

accessmode

アクセスモードです。(0=手動、1=自動)

port_count

アクセス変更するポートの数です。

(3) 戻り値

なし。

(4) 説明

info で指定された TACCESS_INFO 構造体内に accessmode で指定されたアクセスモードと port_count で指定されたポート数を設定します。また port_count 分のポート番号を格納できるリストを生成します。

info にポート番号を設定するためには DshAddTACCESS_INFO()関数を使用してください。

TACCESS_INFO 構造体の使用後は DshFreeTACCESS_INFO()関数を使って構造体内部で使用したメモリを開放してください。

3.17.3.20 DshAddTACCESS_INFO() アクセス変更ポート番号の追加

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTACCESS_INFO(
    TACCESS_INFO *info,          // ポートアクセス情報構造体のポインタ
    int          ptn            // 加えるポート番号
);
```

[.NET VB]

```
Function DshAddTACCESS_INFO (
    ByRef info As dsh_info.TACCESS_INFO,
    ByVal ptn As Int32) As Int32
```

[.NET C#]

```
int DshAddTACCESS_INFO(
    ref TACCESS_INFO info,
    int ptn );
```

(2) 引数

info

ポートアクセス情報構造体のポインタです。

ptn

加えたいポート番号です。

(3) 戻り値

戻り値	意味
0	正常に追加できた。
(-1)	ポートの数が既に指定数に達している。

(4) 説明

先に DshInitTACCESS_INFO() で初期設定されたポートアクセス情報構造体 info にポート番号を 1 個追加します。追加はリストの空き位置に行います。

設定後 0 を返却します。

もし、info 内の port_count で指定された分のポートが既に設定済みであった場合は、(-1) を返却します。

3.17.3.21 DshInitTACCESS_ERR_INFO () ポートアクセス応答情報の初期化

(1) 呼出書式

[C,C++]

```
API int APIX DshInitTACCESS_ERR_INFO(
    TACCESS_ERR_INFO *errinfo, // エラ-情報格納構造体のポインタ
    TACCESS_INFO *info, // アクセス情報格納構造体のポインタ
    int caack, // ackデータ
    int err_count // エラ-情報のリストサイズ (個数 0,1,2...)
);
```

[.NET VB]

```
Function DshInitTACCESS_ERR_INFO (
    ByRef errinfo As dsh_info.TACCESS_ERR_INFO,
    ByRef info As dsh_info.TACCESS_INFO,
    ByVal caack As Int32,
    ByVal errcount As Int32) As Int32
```

[.NET C#]

```
int DshInitTACCESS_ERR_INFO(
    ref TACCESS_ERR_INFO errinfo,
    ref TACCESS_INFO info,
    int caack,
    int errcount );
```

(2) 引数

errinfo

TACCESS_ERR_INFO 応答情報構造体のポインタです。

info

TACCESS_INFO 応答情報構造体のポインタです。

caack

caack - ACK の値です。

err_count

errinfo 内部に設定するエラー情報構造体の数です。 = 0 の場合はエラー情報がないこととなります。

(3) 戻り値

戻り値	意味
0	正常に追加できた。
(-1)	info 内にエラーを検出した。

(4) 説明

本関数は、S3F27 に対する応答メッセージのための応答情報を TACCESS_ERR_INFO 構造体に初期設定するために使用します。

errinfo で指定された構造体の caack メンバーに引数 caack の値を設定し、err_count メンバーに引数 err_count の値を設定します。

そして、もし、err_count > 0 の場合は、err_list に err_count だけの TACCESS_ERR_PORT エラー情報構造

体のポインタリストを設けます。

err_info へのエラー情報の設定には DshPutTACCESS_ERR_INFO()関数を使用します。

3.17.3.22 DshPutTACCESS_ERR_INFO () ポートアクセスエラー情報の設定

(1) 呼出書式

[C, C++]

```
API int APIX DshPutTACCESS_ERR_INFO (
    TACCESS_ERR_INFO *errinfo, // エラ-情報格納構造体リストのポインタ
    TACCESS_INFO *info, // アクセス情報格納構造体のポインタ
    int order, // info 内の port_list 上の位置順位
    int errcode, // error code
    char *errtext // error text
);
```

[.NET VB]

```
Function DshPutTACCESS_ERR_INFO (
    ByRef errinfo As dsh_info.TACCESS_ERR_INFO,
    ByRef info As dsh_info.TACCESS_INFO,
    ByVal order As Int32,
    ByVal errcode As Int32,
    ByVal errtext As String) As Int32
```

[.NET C#]

```
int DshPutTACCESS_ERR_INFO(
    ref TACCESS_ERR_INFO errinfo,
    ref TACCESS_INFO info,
    int order,
    int errcode,
    byte[] errtext );
```

(2) 引数

errinfo

ポートアクセスエラー情報構造体のポインタです。

info

S3F27 から得られたポートアクセス情報 TACCESS_INFO 応答情報構造体のポインタです。

order

info 内の port_list 上に格納されているポート ID の格納位置順位

errcode

設定するエラーコードです。(メッセージ内のアイテムは U1(51)です。)

errtext

設定するエラーテキストが格納されている領域のポインタです。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	リストが満杯で設定できなかった。

(4) 説明

本関数は、errinfo 内の errlist リストの先頭から空きリストを探します。
もし、空きリストがなければ、(-1)を返却します。

もし、空きリストがあれば、その空きリストに 1 個 TACCESS_ERR_PORT 構造体領域を設け、その構造体に info 内のポートリスト上にある order 番目のポート ID を設定するとともに、rcode と errtext を格納し、0 を返却します。TERR_INFO と内部メンバーのメモリは本関数が取得します。

本関数の実行前に DshInitTACCESS_ERR_INFO()関数を使って errinfo を初期化しておく必要があります。

3.17.3.23 DshFreeTACCESS_ERR_INFO() - ポートアクセス応答情報メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTACCESS_ERR_INFO(  
    TACCESS_ERR_INFO *erinfo           // メリを開放したい応答情報格納構造体のポインタ  
);
```

[.NET VB]

```
Sub DshFreeTACCESS_ERR_INFO (  
    ByRef info As dsh_info.TACCESS_ERR_INFO)
```

[.NET C#]

```
void DshFreeTACCESS_ERR_INFO(  
    ref TACCESS_ERR_INFO info );
```

(2) 引数

erinfo

メモリを解放したいポートアクセス応答情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TACCESS_ERR_INFO 構造体内で情報格納用に使用されているメモリを全て解放します。

3.17.3.24 DshMakeS3F27Response() - S3F27 の応答メッセージの生成

(1) 呼出書式

[C, C++]

```
API int APIX DshMakeS3F27Response(
    TACCESS_INFO *info,           // アクセスモード 情報格納領域のポインタ
    TACCESS_ERR_INFO *erinfo,    // S3F28 に設定する応答情報格納領域のポインタ
    DSHMSG *msg,                 // S3F28 メッセージ を格納するメッセージ 構造体のポインタ
    BYTE *buff,                  // S3F28 のテキスト格納バッファポインタ
    int buff_size                 // buff のバイトサイズ
);
```

[.NET VB]

```
Function DshMakeS3F27Response (
    ByRef info As dsh_info.TACCESS_INFO,
    ByRef erinfo As dsh_info.TACCESS_ERR_INFO,
    ByRef msg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS3F27Response(
    ref TACCESS_INFO info,
    ref TACCESS_ERR_INFO erinfo,
    ref DSHMSG msg,
    byte[] buff,
    int buff_size );
```

(2) 引数

info

アクセスモード情報が格納されている領域のポインタです。

erinfo

S3F28 メッセージに設定する応答情報が格納されている領域のポインタです。

msg

S3F28 応答メッセージ情報を格納するためのメッセージ構造体のポインタです。

buff

S3F28 応答メッセージのテキストを格納するためのバッファポインタです。

buff_size

buff のバイトサイズです。

(3) 戻り値

戻り値	意味
0	正常に生成できた。
(-1)	生成できなかった。(buff 領域不足)

(4) 説明

S3F27 に対する S3F28 応答メッセージを info に含まれるアクセスモード情報と応答情報に従って作成します。

応答情報内の、caack を S3F28 の CAACK として設定します。
CAACK はユーザが S3F27 アクセスモードメッセージを評価した結果です。

erinfo の情報生成には、DshInitTACCESS_ERR_INFO ()、DshPutTACCESS_ERR_INFO ()関数を使うことができます。

3.17.4 ユーザ作成ライブラリ関数

3.17.4.1 DshResponseS3F24() S3F24 グループポートアクション応答メッセージ

(1) 呼出書式

[C, C++]

```
API int APIX DshResponseS3F24(
    int eqid, // 通信対象装置 ID(0,16,...)
    ID_TR trid, // DSHDR2 のトランザクション ID
    TPORTG_INFO *info, // グループポートアクションメッセージ 情報格納領域のポインタ TCACT_ERR_INFO
    *erinfo // S3F24 応答情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshResponseS3F24 (
    ByVal eqid As Int32,
    ByVal trid As Int32,
    ByRef info As dsh_info.TPORTG_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO) As Int32
```

[.NET C#]

```
int DshResponseS3F24(
    int eqid,
    uint trid,
    ref TPORTG_INFO info,
    ref TCACT_ERR_INFO erinfo );
```

(2) 引数

eqid

通信エンジンが通信する対象装置 ID を指定します。装置 ID は 0 から始まる番号です。

trid

S3F23 受信時に DSHGEMLIB から渡される DSHDR2 通信ドライバーのトランザクション管理のための ID です。

info

グループポートアクション情報が格納されている構造体のポインタです。

erinfo

送信する応答メッセージ S3F24 に含まれる情報を格納するための構造体領域のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に送信できた。
(-1)	送信できなかった。

(4) 説明

グループポートアクションメッセージ S3F23 に対する応答メッセージを送信します。



本関数はユーザ作成ライブラリ DLL (dsh_ulib.dll) に含まれる関数ですが、ここでは DSHGEMLIB パッケージに標準的な関数として付属されているものです。(ユーザ独自による作成も可能です)

引数に指定されている TCACT_ERR_INFO 構造体に含まれている情報から S3F24 メッセージを組み立て、その後、S3F24 メッセージを送信します。

送信が終わったら、TCACT_ERR_INFO の構造体で使用されたメモリを DshFreeTCACT_ERR_INFO () 関数を使って開放します。

なお、S3F24 メッセージの組み立てに、DshMakeS3F24Response() 関数を使用できます。

3.17.4.2 DshResponseS3F26() S3F26 ポートアクション応答メッセージ

(1) 呼出書式

[C, C++]

```
API int APIX DshResponseS3F26(
    int eqid,           // 通信対象装置 ID(0,16,...)
    ID_TR trid,        // DSHDR2 のトランザクション ID
    TPORT_INFO *info,  // ポートアクションメッセージ 情報格納領域のポインタ
    TCACT_ERR_INFO *erinfo // S3F26 応答情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshResponseS3F26 (
    ByVal eqid As Int32,
    ByVal trid As Int32,
    ByRef info As dsh_info.TPORT_INFO,
    ByRef erinfo As dsh_info.TCACT_ERR_INFO) As Int32
```

[.NET C#]

```
int DshResponseS3F26(
    int eqid,
    uint trid,
    ref TPORT_INFO info,
    ref TCACT_ERR_INFO erinfo );
```

(2) 引数

eqid

通信エンジンが通信する対象装置 ID を指定します。装置 ID は 0 から始まる番号です。

trid

S3F25 受信時に DSHGEMLIB から渡される DSHDR2 通信ドライバーのトランザクション管理のための ID です。

info

ポートアクション情報が格納されている構造体のポインタです。

erinfo

送信する応答メッセージ S3F26 に含まれる情報を格納するための構造体領域のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に送信できた。
(-1)	送信できなかった。

(4) 説明

ポートアクションメッセージ S3F25 に対する応答メッセージを送信します。

本関数はユーザ作成ライブラリ DLL(dsh_ulib.dll)に含まれる関数ですが、ここでは DSHGEMLIB パッケージに標準的な関数として付属されているものです。(ユーザ独自による作成も可能です)

引数に指定されている TCACT_ERR_INFO 構造体に含まれている情報から S3F26 メッセージを組み立て、その後、S3F26 メッセージを送信します。

送信が終わったら、TCACT_ERR_INFO の構造体で使用されたメモリを DshFreeTCACT_ERR_INFO ()関数を使って開放します。

なお、S3F26 メッセージの組み立てに、DshMakeS3F26Response()関数を使用できます。

3.17.4.3 DshResponseS3F28() S3F28 ポートアクセスモード応答メッセージ

(1) 呼出書式

[C, C++]

```
API int APIX DshResponseS3F28(
    int eqid,           // 通信対象装置 ID(0,16,...)
    ID_TR trid,        // DSHDR2 のトランザクション ID
    TACCESS_INFO *info, // ポートアクセスモードメッセージ情報格納領域のポインタ
    TACCESS_ERR_INFO *erinfo // S3F28 応答情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshResponseS3F28 (
    ByVal eqid As Int32,
    ByVal trid As Int32,
    ByRef info As dsh_info.TACCESS_INFO,
    ByRef erinfo As dsh_info.TACCESS_ERR_INFO) As Int32
```

[.NET C#]

```
int DshResponseS3F28(
    int eqid,
    uint trid,
    ref TACCESS_INFO info,
    ref TACCESS_ERR_INFO erinfo );
```

(2) 引数

eqid

通信エンジンが通信する対象装置 ID を指定します。装置 ID は 0 から始まる番号です。

trid

S3F27 受信時に DSHGEMLIB から渡される DSHDR2 通信ドライバーのトランザクション管理のための ID です。

info

ポートアクセスモードモード情報が格納されている構造体のポインタです。

erinfo

送信する応答メッセージ S3F28 に含まれる情報を格納するための構造体領域のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に送信できた。
(-1)	送信できなかった。

(4) 説明

ポートアクセスモードメッセージ S3F27 に対する応答メッセージを送信します。

本関数はユーザ作成ライブラリ DLL(dsh_ulib.dll)に含まれる関数ですが、ここでは DSHGEMLIB パッケージに標準的な関数として付属されているものです。(ユーザ独自による作成も可能です)

引数に指定されている TACCESS_ERR_INFO 構造体に含まれている情報から S3F28 メッセージを組み立て、その後、S3F28 メッセージを送信します。

送信が終わったら、TACCESS_ERR_INFO の構造体で使用されたメモリを DshFreeTACCESS_ERR_INFO ()関数を使って開放します。

なお、S3F28 メッセージの組み立てに、DshMakeS3F28Response()関数を使用できます。