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-１１／１５

３．18　ホストリモートコマンド(S2F41)関連関数

３．19　拡張リモートコマンド(S2F49)関連関数

２００９年6月

株式会社データマップ

文書番号　DSHGEM-LIB-09-3032B-00

[取り扱い注意]

- この資料ならびにソフトウェアの一部または全部を無断で使用、複製することはできません。
- 本説明書に記述されている内容は予告なしで変更される可能性があります。
- Windows は米国 Microsoft Corporation の登録商標です。
- ユーザーが本ソフトウェアの使用によって生じた遺失履歴、(株) データマップの予見の有無を問わず発生した特別損害、付随的損害、間接損害およびその他の拡大損害に対して責任を負いません。

【改訂履歴】

番号	改訂日付	項 目	概 略
1.	2009.6	改訂版	以前の DSHGEM-LIB-07-3032x-00 を全面改訂 .Net VB2008, C#2008 対応関数の説明を追加した。

目　次

(VOL－１２に続く)

３．18　ホストリモートコマンド(S2F41)関連関数

S2F41 ホストリモートコマンドメッセージ処理に使用できるライブラリ関数について説明します。

（1）送信 API 関数

	API 関数名	機能
1	GemSendS2F41()	ﾎｽﾄﾘﾓｰﾄｺﾏﾝﾄﾞ ﾒｯｾｰｼﾞ S2F41 を送信します。

（2）ライブラリ関数一覧

	ﾗｲﾌﾞﾗﾘ関数名	機能
1	DshDecodeS2F41 ()	S2F41 ﾒｯｾｰｼﾞ をﾃﾞｺｰﾄﾞ し TRCMD_INFO 構造体に情報を格納します。
2	DshEncodeS2F41 ()	TRCMD_INFO 構造体に格納されている情報を S2F41 ﾒｯｾｰｼﾞ にｴﾝｺｰﾄﾞ します。
3	DshFreeTRCMD_INFO()	DshDecodeS2F41()で使用した TRCMD_INFO 内で使用したﾒﾓﾘを開放します。
4	DshCopyTRCMD_INFO()	DshDecodeS2F41()で取得した TRCMD_INFO 構造体情報をｺﾋﾟｰします。
5	DshInitTRCMD_INFO()	S2F41 ｴﾝｺｰﾄﾞ で使用する TRCMD_INFO 情報構造体を初期化します。
6	DshAddTRCMD_INFO()	S2F41 ｴﾝｺｰﾄﾞ で使用する TRCMD_INFO 情報構造体にﾎｽﾄｺﾏﾝﾄﾞ 情報を加えます。
7	DshAddTRCMD_PARA()	S2F41 ｴﾝｺｰﾄﾞ で使用する TRCMD_INFO 情報構造体にﾊﾟﾗﾒｰﾀ情報を加えます。
8	DshInitTRCMD_ERR_INFO()	S2F42 応答情報構造体 TRCMD_ERR_INFO を初期化します。
9	DshPutTRCMD_ERR_PARA()	TRCMD_ERR_INFO 構造体にｴﾗｰﾊﾟﾗﾒｰﾀ情報を加えます。
10	DshFreeTRCMD_ERR_INFO()	TRCMD_ERR_INFO 構造体内で使用したﾒﾓﾘを開放します。
11	DshMakeS2F41Response()	S2F42 応答ﾒｯｾｰｼﾞ を TRCMD_ERR_INFO と TRCMD_INFO 構造体の内容に基づき生成します。

（3）ユーザ作成ライブラリ関数

	ﾗｲﾌﾞﾗﾘ関数名	機能
1	DshResponseS2F42	S2F42 ホストリモートコマンド応答メッセージ

３．18．１　使用する情報格納構造体

（１）S2F41 情報格納用

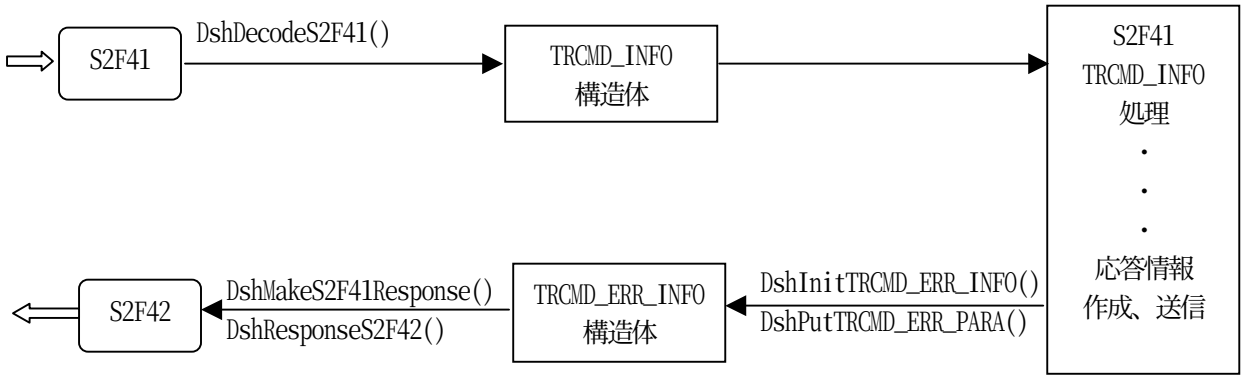
```
typedef struct{
        char        *rcmd;          // rcmd
        int         cp_count;       // parameter count
        TRCMD_PARA  **cp_list;      // paramete list
        }TRCMD_INFO;


typedef struct{
        char        *cpname;        // cpname
        int         cpval_fmt;      // cpval item fmt
        int         cpval_size;     // cpval data array size
        void        *cpval;         // cpval
        }TRCMD_PARA;
```

（２） S2F42 エラー情報

```
typedef struct{
        int         hcack;          // B
        int         err_count;
        int         *err_list;      // err position list (ｴﾗｰの在ったﾊﾟﾗﾒｰﾀの順位 0,1..)
        int         *cpack_list;    // err_list 内順位の cpack のﾘｽﾄ( 値はＢﾌｫｰﾏｯﾄです)
} TRCMD_ERR_INFO;
```

（３）ライブラリ関数との関係は次のようになります。



3．18．２　ホストコマンド送信関数

3．18．２．１　GemSendS2F41() - ホストコマンドメッセージ送信関数

（1）呼出書式

[c,C++]

```
API int APIX GemSendS2F41(
        int     eqid,                   // 通信対象装置 ID(0,1,2,...)
        TRCMD_INFO *info,               // ﾎｽﾄｺﾏﾝﾄﾞ情報格納領域のﾎﾟｲﾝﾀ
        TRCMD_ERR_INFO *erinfo,         // S4F42 応答情報格納用構造体のﾎﾟｲﾝﾀ
        int (WINAPI *callback)(),       // 実行終了時のｺｰﾙﾊﾞｯｸ関数
        ULONG  upara                    // callback 時のﾊﾟﾗﾒｰﾀ
);
```

[.NET VB]

```
Function GemSendS2F41 (
        ByVal eqid As Int32,
        ByRef info As dsh_info.TRCMD_INFO,
        ByRef erinfo As dsh_info.TRCMD_HERR_INFO,
        ByVal callback As vcallback.callback_S2F41,
        ByVal upara As Int32) As Int32
```

[.NET C#]

```
int GemSendS2F41(
        int eqid,
        ref TRCMD_INFO info,
        ref TRCMD_HERR_INFO erinfo,
        CallbackS2F41 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

送信したいホストコマンド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 S4F42 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) ﾌﾞﾛｯｸﾓｰﾄﾞ： 正常に送信できた。 erinfo に S4F42 応答情報が返却されます。 (2) 非ﾌﾞﾛｯｸﾓｰﾄﾞ：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

装置にホストコマンド情報の送信要求を行います。S2F41 メッセージで送信します。

要求を受けた DSHGEM-LIB は、info に格納されているホストコマンド情報を S2F41 メッセージにエンコードし、装置に送信します。

S4F42 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 S4F42 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)	S2F41 送信後、応答ﾒｯｾｰｼﾞ S4F42 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 S4F42 応答情報が返却されます。
あり	送信要求後、S2F41 の送信前に制御が戻されます。実行結果はｺｰﾙﾊﾞｯｸ関数で与えられます。 ｺｰﾙﾊﾞｯｸ関数の end_status が=0 ならば erinfo に S4F42 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 TRCMD_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、その構造体に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 DshFreeTRCMD_ERR_INFO()関数を使って行ってください。

```
DshFreeTRCMD_ERR_INFO (erinfo)
```

TRCMD_INFO 構造体へのホストコマンド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。

```
DshInitTRCMD_INFO(), DshAddTRCMD_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
        TRCMD_ERR_INFO *erinfo,         // S4F42 応答情報格納用構造体のﾎﾟｲﾝﾀ
        ULONG upara                     // 呼出時に指定したﾊﾟﾗﾒｰﾀ
);
```

[.NET VB]

```
Function callback_S2F41(ByVal eqid As Integer, ByVal end_status As Integer, ByRef erinfo As
dsh_info.TRCMD_HERR_INFO, ByVal upara As Integer) As Integer
```

[.NET C#]

```
int CallbackS2F41(int eqid, int end_status, ref TRCMD_HERR_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 S4F42 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。

3．18．3　S2F41、S2F42 メッセージ処理関連関数

3．18．3．1　DshDecodeS2F41 - S2F41 デコード関数

（1）呼出書式

[c,C++]

```
API int  APIX DshDecodeS2F41(
        DSHMSG *smsg,                   // SECS ﾒｯｾｰｼﾞ情報構造体のﾎﾟｲﾝﾀ
        TRCMD_INFO *pinfo               // ﾃﾞｺｰﾄﾞした情報を格納する構造体のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshDecodeS2F41 (
        ByRef smsg As dshdr2.DSHMSG,
        ByRef info As dsh_info.TRCMD_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS2F41(
        ref DSHMSG smsg,
        ref TRCMD_INFO info );
```

（2）引数

smsg

S2F41 の SECS ﾒｯｾｰｼﾞ情報が格納されている構造体のポインタです。

pinfo

デコードしたリモートコマンド情報を格納する構造体のポインタです。

（3）戻り値

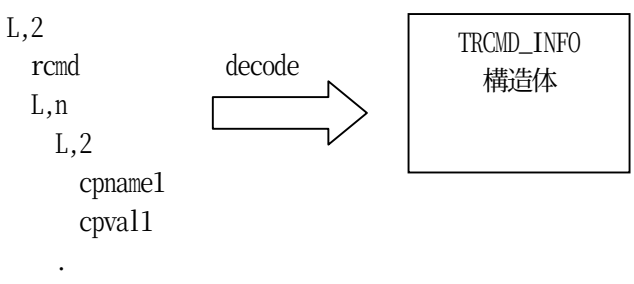
戻り値	意味
0	正常にﾃﾞｺｰﾄﾞできた。
(-1)	smsg を正しくﾃﾞｺｰﾄﾞできなかった。

（4）説明

S2F41 メッセージに含まれるリモートコマンド情報を、ユーザプログラムが処理しやすい TRCMD_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 DshFreeTRCMD_INFO()関数を使って開放してください。

smsg S2F41



3．18．3．2　DshEncodeS2F41() - ホストコマンド情報を S2F41 へエンコード

（1）呼出書式

[c,C++]

```
API int  APIX DshEncodeS2F41(
        DSHMSG *smsg,                   // SECS ﾒｯｾｰｼﾞ情報構造体のﾎﾟｲﾝﾀ
        BYTE   *buffer,                 // S2F41 を格納するﾊﾞｯﾌｧﾎﾟｲﾝﾀ
        int    bufflen,                 // buffer のﾊﾞｲﾄｻｲｽﾞ
        TRCMD_INFO *info                // ｴﾝｺｰﾄﾞしたいﾎｽﾄｺﾏﾝﾄﾞ情報格納構造体のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshEncodeS2F41 (
          ByRef smsg As dshdr2.DSHMSG, _
          ByRef buff As Byte, _
          ByVal bufflen As Integer, _
          ByRef info As dsh_info.TRCMD_INFO ) As Int32
```

[.NET C#]

```
int DshEncodeS2F41(
            ref DSHMSG smsg,
            byte() buffer,
            int bufflen,
            ref TRCMD_INFO info );
```

（2）引数

smsg

エンコードした S2F41 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 S2F41 のテキストを格納するバッファポインタです。

bufflen

buffer のバイトサイズです。

info

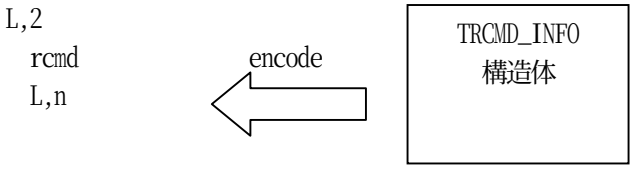
エンコードしたいホストコマンド情報が格納されている構造体のポインタです。

（3）戻り値

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

（4）説明

TRCMD_INFO 構造体に格納されているホストコマンド情報を、S2F41 の SECS メッセージにエンコードします。



```
L,2
  cpname1
  cpval1
.
```

TRCMD_INFO 構造体へのホストコマンド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。

DshInitTRCMD_INFO(), DshAddTRCMD_INFO()

３．18．３．３　DshFreeTRCMD_INFO() - リモートコマンド情報構造体メモリの開放

（1）呼出書式

[c,C++]

```
API void APIX DshFreeTRCMD_INFO(
      TRCMD_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ﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Sub DshFreeTRCMD_INFO (
        ByRef info As dsh_info.TRCMD_INFO)
```

[.NET C#]

```
void DshFreeTRCMD_INFO(
        ref TRCMD_INFO info );
```

（2）引数

pinfo

メモリを解放したいリモートコマンド情報構造体のポインタです。

（3）戻り値

なし。

（4）説明

TRCMD_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、TRCMD_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。

3．18．3．4　DshCopyTRCMD_INFO() - リモートコマンド情報構造体メモリのコピー

（1）呼出書式

[c,C++]

```
API int APIX   DshCopyTRCMD_INFO(
        TRCMD_INFO *dinfo,                  // ｺﾋﾟｰ先のﾎﾟｲﾝﾀ
        TRCMD_INFO *sinfo                   // ｺﾋﾟｰ元のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshCopyTRCMD_INFO (
          ByRef dinfo As dsh_info.TRCMD_INFO,
          ByRef sinfo As dsh_info.TRCMD_INFO) As Int32
```

[.NET C#]

```
int DshCopyTRCMD_INFO(
          ref TRCMD_INFO dinfo,
          ref TRCMD_INFO sinfo );
```

（2）引数

dinfo

リモートコマンド情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のリモートコマンド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	正常にｺﾋﾟｰできた。
(-1)	sinfo または dinfo の値が NULL だったのでｺﾋﾟｰできなかった。

（4）説明

sinfo が指す TRCMD_INFO 構造体内に格納されているリモートコマンド情報を dinfo が指定する TRCMD_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、DshFreeTRCMD_INFO()関数を使って開放してください。

3．18．3．5　DshInitTRCMD_INFO - ホストコマンド情報 TRCMD_INFO の初期設定

（1）呼出書式

[c,C++]

```
API void  APIX DshInitTRCMD_INFO(
        TRCMD_INFO     *info,              // ﾎｽﾄｺﾏﾝﾄﾞ情報 TRCMD_INFO 構造体のﾎﾟｲﾝﾀ
        char           *rcmd,              // ｺﾏﾝﾄﾞ名
        int            cp_count            // 設定できるｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀ数
);
```

[.NET VB]

```
Sub DshInitTRCMD_INFO (
        ByRef info As dsh_info.TRCMD_INFO,
        ByVal rcmd As String,
        ByVal cp_count As Int32)
```

[.NET C#]

```
void DshInitTRCMD_INFO(
        ref TRCMD_INFO info,
        byte[] rcmd,
        int cp_count );
```

（2）引数

info

ホストコマンド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。

rcmd

コマンド名が格納されているポインタです。

cp_count

ホストコマンド情報の中に設定できるコマンド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 TRCMD_INFO 構造体内に rcmd で指定されたコマンド名を設定します。また cp_count 分のコマンド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 DshAddTRCMD_INFO()関数を使用してください。

TRCMD_INFO 構造体の使用後は DshFreeTRCMD_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。

3．18．3．6　DshAddTRCMD_INFO() - ホストコマンドパラメータの追加

（1）呼出書式

[c,C++]

```
API int APIX DshAddTRCMD_INFO(
        TRCMD_INFO *info,               // ﾎｽﾄｺﾏﾝﾄﾞｺﾏﾝﾄﾞ情報構造体のﾎﾟｲﾝﾀ
        char       *cp_pname,           // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀ名
        int        fmt,                 // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀのﾌｫｰﾏｯﾄ( ICODE_A, ICODE_U1 etc )
        int        asize,               // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀﾃﾞｰﾀの配列ｻｲｽﾞ
        void       *cp_val              // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀﾃﾞｰﾀ格納ﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshAddTRCMD_INFO (
        ByRef info As dsh_info.TRCMD_INFO,
        ByVal cp_name As String,
        ByVal fmt As Int32,
        ByVal asize As Int32,
        ByVal cp_val As Int32) As Int32
```

[.NET C#]

```
int DshAddTRCMD_INFO(
        ref TRCMD_INFO info,
        byte[] cp_name,
        int fmt,
        int asize,
        byte[] cp_val );
```

（2）引数

info

ホストコマンド情報構造体のポインタです。

cp_name

加えたいコマンドパラメータ名が格納されているポインタです。

fmt

コマンドパラメータデータのフォーマットです。(ICODE_A,ICODE_U1 など)

asize

コマンドパラメータデータの配列サイズです。

cp_val

コマンド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)	ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀの数が既に指定数に達している。

（4）説明

先にDshInitTRCMD_INFO()で初期設定されたホストコマンド情報構造体 info にコマンドパラメータを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TRCMD_PARAパラメータ構造体のメモリを確保し、その中にcp_nameとcp_valを設定し、cp_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cp_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。
それから、パラメータがリスト構造になる場合がありますが、そのネスティングレベルは1とします。
その場合、fmt = ICODE_Lにして、asizeに次のレベルでのパラメータ数を指定してください。

```
L,n                 ﾊﾟﾗﾒｰﾀ数=n
  L,2
    cpname-1
    L,m             ﾃﾞｰﾀではなく、List 構造, ﾊﾟﾗﾒｰﾀが m 個
      L,2
        cpname-1-1
        cpval-1-1
      L,2
        .
```

下位レベルへのパラメータ追加はDshAddTRCMD_PARA()関数で設定することができます。

３．18．３．７　DshAddTRCMD_PARA() - ホストコマンドパラメータの追加

（1）呼出書式

[c,C++]

```
API int APIX DshAddTRCMD_PARA(
        TRCMD_INFO    *info,          // ﾎｽﾄｺﾏﾝﾄﾞｺﾏﾝﾄﾞ情報構造体のﾎﾟｲﾝﾀ
        int           order,          // ﾊﾟﾗﾒｰﾀ位置順位(0,1,2..)
        int           fmt,            // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀのﾌｫｰﾏｯﾄ( ICODE_A, ICODE_U1 etc )
        int           asize,          // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀﾃﾞｰﾀの配列ｻｲｽﾞ
        void          *cp_val         // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀﾃﾞｰﾀ格納ﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshAddTRCMD_PARA (
        ByRef info As dsh_info.TRCMD_INFO,
        ByVal order As Int32,
        ByVal fmt As Int32,
        ByVal asize As Int32,
        ByVal cp_val As Int32) As Int32
```

[.NET C#]

```
int DshAddTRCMD_PARA(
        ref TRCMD_INFO info,
        int order,
        int fmt,
        int asize,
        byte[] cp_val );
```

（2）引数

info

ホストコマンド情報構造体のポインタです。

order

加えたいコマンドパラメータを設定する info の cp_list リストの順位を指定します。

fmt

コマンドパラメータデータのフォーマットです。(ICODE_A,ICODE_U1 など)

asize

コマンドパラメータデータの配列サイズです。

cp_val

コマンド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)	order で指定された順位がﾘｽﾄｻｲｽﾞを超えていたまたは order 番目の format が ICODE_L ではない。または、ﾚﾍﾞﾙ-1 のｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀの数が既に指定数に達している。

（4）説明

ホストコマンド情報構造体 info の cp_list の order 番目のリストのパラメータ内のリストにネスティングしたコマンドを１個追加します。
設定は、新たに TRCMD_PARA パラメータ構造体のメモリを確保し、その中に cp_name と cp_val を設定し、cp_list[order]のパラメータ構造体の cpval をリストとみなしその中の空き位置にポインタを設定します。
正常設定の場合 0 を返却します。もし、以下の条件の場合は、(-1)を返却します。
①info 内の cp_count <= order である。　②order 番目のﾊﾟﾗﾒｰﾀの format が ICODE_L でない。
③コマンドパラメータ構造体 TRCMD_PARA のメンバーcpval（これがパラメータリストになる）が既に満杯であった場合。

```
cp_list[order] - cpname     order 番目のﾊﾟﾗﾒｰﾀ
                 cpval_fmt  ICODE_L
                 cpval_size これがﾘｽﾄｻｲｽﾞ                                    引数
                 cpval      これがﾊﾟﾗﾒｰﾀﾘｽﾄ[]になる。この空きﾘｽﾄに-cpname      <-- NULL
                                                          cpval_fmt  <-- fmt
                                                          cpval_size <-- asize
                                                          cpval      <-- cp_val
```

３．18．３．８　DshInitTRCMD_ERR_INFO () - ホストリモートコマンド応答情報の初期化

（1）呼出書式

[c,C++]

```
API void APIX   DshInitTRCMD_ERR_INFO(
        TRCMD_INFO *info,                   // S2F41 で得られたﾎｽﾄﾘﾓｰﾄｺﾏﾝﾄﾞ情報構造体ﾎﾟｲﾝﾀ
        TRCMD_ERR_INFO *erinfo,             // ｴﾗｰ情報格納構造体ﾘｽﾄのﾎﾟｲﾝﾀ
        int         hcack,                  // ackﾃﾞｰﾀ
        int         err_count               // ｴﾗｰ情報のﾘｽﾄｻｲｽﾞ（個数　0,1,2...）
);
```

[.NET VB]

```
Function DshInitTRCMD_ERR_INFO (
        ByRef info As dsh_info.TRCMD_INFO,
        ByRef erinfo As dsh_info.TRCMD_ERR_INFO,
        ByVal hcack As Int32,
        ByVal err_count As Int32) As Int32
```

[.NET C#]

```
int DshInitTRCMD_ERR_INFO(
        ref TRCMD_INFO info,
        ref TRCMD_ERR_INFO erinfo,
        int hcack,
        int err_count );
```

（2）引数

info

S2F41 で得られた TRCMD_INFO 情報構造体のポインタです。

erinfo

TRCMD_ERR_INFO 応答情報構造体のポインタです。

hcack

hcack - 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、ホストリモート関連応答メッセージ TRCMD_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 hcack メンバーに引数 hcack 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、cpname_list と cpack_list に err_count 分だけのエラー情報のための領域を設けます。

3．18．3．9　DshPutTRCMD_ERR_PARA () - ホストリモートコマンドエラー情報の設定

（1）呼出書式

[c,C++]

```
API void APIX   DshPutTRCMD_ERR_PARA (
        TRCMD_INFO *info,                   // S2F41 で得られたﾎｽﾄﾘﾓｰﾄｺﾏﾝﾄﾞ情報構造体ﾎﾟｲﾝﾀ
        TRCMD_ERR_INFO *erinfo,             // ｴﾗｰ情報格納構造体ﾘｽﾄのﾎﾟｲﾝﾀ
        int         order,                  // ﾊﾟﾗﾒｰﾀ位置順位(0,1,..)
        int         cpack                   // ﾊﾟﾗﾒｰﾀの ack
);
```

[.NET VB]

```
Function DshPutTRCMD_ERR_PARA (
        ByRef info As dsh_info.TRCMD_INFO,
        ByRef erinfo As dsh_info.TRCMD_ERR_INFO,
        ByVal order As Int32,
        ByVal cpack As Int32) As Int32
```

[.NET C#]

```
int DshPutTRCMD_ERR_PARA(
        ref TRCMD_INFO info,
        ref TRCMD_ERR_INFO erinfo,
        int order,
        int cpack );
```

（2）引数

info

S2F41 で得られた TRCMD_INFO 情報構造体のポインタです。

errinfo

ホストリモートコマンドエラー情報構造体のポインタです。

order

info 内のパラメーターリストの位置順位です。

cpack

設定するパラメータの ACK 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。
空きリストがあれば、その空きリストに order と cpack 値を設定します。

DshMakeS2F41Response()関数が、S2F42 メッセージのエンコードを行いますが、info と errinfo が引数として使用されます。

３．18. ３．10 DshFreeTRCMD_ERR_INFO() - ホストコマンド応答情報メモリの開放

（1）呼出書式

[c,C++]

```
API void APIX DshFreeTRCMD_ERR_INFO(
      TRCMD_ERR_INFO *erinfo            // ﾒﾓﾘを開放したい応答情報格納構造体のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Sub DshFreeTRCMD_ERR_INFO (
        ByRef info As dsh_info.TRCMD_ERR_INFO)
```

[.NET C#]

```
void DshFreeTRCMD_ERR_INFO(
        ref TRCMD_ERR_INFO info );
```

（2）引数

erinfo

メモリを解放したいホストコマンド応答情報構造体のポインタです。

（3）戻り値

なし。

（4）説明

TRCMD_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。

3．18. 3．11 DshMakeS2F41Response() - S2F41 の応答メッセージの生成

（1）呼出書式

[c,C++]

```
API int APIX DshMakeS2F41Response(
        TRCMD_INFO *info,               // ﾎｽﾄﾘﾓｰﾄｺﾏﾝﾄﾞ情報格納領域ﾎﾟｲﾝﾀ
        TRCMD_ERR_INFO *erinfo,         // S2F42 に設定する応答情報格納領域のﾎﾟｲﾝﾀ
        DSHMSG *smsg,                   // S2F42 ﾒｯｾｰｼﾞを格納するﾒｯｾｰｼﾞ構造体のﾎﾟｲﾝﾀ
        BYTE   *buff,                   // S2F42 のﾃｷｽﾄ格納ﾊﾞｯﾌｧﾎﾟｲﾝﾀ
        int    buff_size                // buff のﾊﾞｲﾄｻｲｽﾞ
);
```

[.NET VB]

```
Function DshMakeS2F41Response (
          ByRef info As dsh_info.TRCMD_INFO,
          ByRef erinfo As dsh_info.TRCMD_ERR_INFO,
          ByRef smsg As dshdr2.DSHMSG,
          ByRef buff As Byte,
          ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS2F41Response(
          ref TRCMD_INFO info,
          ref TRCMD_ERR_INFO erinfo,
          ref DSHMSG smsg,
          byte[] buff,
          int buff_size );
```

（2）引数

info

ホストリモートコマンド情報が格納されている領域のポインタです。

erinfo

S2F42 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

S2F42 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

S2F42 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

S2F41 に対する S2F42 応答メッセージを info に含まれるホストリモートコマンド情報と応答情報に従って作成します。

応答情報内の、hcack を S2F42 の HCACK として設定します。
HCACK はユーザが S2F41 ホストリモートコマンドメッセージを評価した結果です。

erinfo の情報の設定は、DshInitTRCMD_ERR_INFO()と DshPutTRCMD_ERR_PARA()関数を使って行うことができます。

3．18．４　ユーザ作成ライブラリ関数

3．18．４．１　DshResponseS2F42() – S2F42 ホストリモートコマンド応答メッセージ

（1）呼出書式

[c,C++]

```
API int APIX DshResponseS2F42(
        int     eqid,                   // 通信対象装置 ID(0,16,...)
        ID_TR   trid,                   // DSHDR2 のﾄﾗﾝｻﾞｸｼｮﾝ ID
        TRCMD_INFO *info,               // ホストﾘﾓｰﾄｺﾏﾝﾄﾞﾒｯｾｰｼﾞ情報格納領域のﾎﾟｲﾝﾀ
        TRCMD_ERR_INFO *erinfo          // S2F42 応答情報格納用構造体のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshResponseS2F42 (
        ByVal eqid As Int32,
        ByVal trid As Int32,
        ByRef info As dsh_info.TRCMD_INFO,
        ByRef erinfo As dsh_info.TRCMD_ERR_INFO) As Int32
```

[.NET C#]

```
int DshResponseS2F42(
        int eqid,
        uint trid,
        ref TRCMD_INFO info,
        ref TRCMD_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

S2F41 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

ホストリモートコマン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 S2F42 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

ホストリモートコマンドメッセージ S2F41 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 TRCMD_ERR_INFO 構造体に含まれている情報から S2F42 メッセージを組み立て、その後、S2F42 メッセージを送信します。

送信が終ったら、TRCMD_ERR_INFO の構造体に使用されたメモリを DshFreeTRCMD_ERR_INFO ()関数を使って開放します。

なお、S2F42 メッセージの組み立てに、DshMakeS2F42Response()関数を使用できます。

3．19　拡張リモートコマンド(S2F49)関連関数

S2F49 ホスト拡張リモートコマンドメッセージ処理に使用できる API、ライブラリ関数について説明します。

（1）送信 API 関数

	API 関数名	機能
1	GemSendS2F49()	拡張ﾘﾓｰﾄｺﾏﾝﾄﾞ ﾒｯｾｰｼﾞ S2F41 を送信します。

（2）ライブラリ関数一覧

	ﾗｲﾌﾞﾗﾘ関数名	機能
1	DshDecodeS2F49 ()	S2F49 ﾒｯｾｰｼﾞ をﾃﾞｺｰﾄﾞ し TERCMD_INFO 構造体に情報を格納します。
2	DshEncodeS2F49()	TTCMD_INFO 構造体内のﾎｽﾄ拡張ｺﾏﾝﾄﾞ 情報から S2F49 ﾒｯｾｰｼﾞ をﾃﾞｺｰﾄﾞ します。
3	DshFreeTERCMD_INFO()	TERCMD_INFO 構造体内に使用したﾒﾓﾘを開放します。
4	DshCopyTERCMD_INFO()	TERCMD_INFO 構造体のﾎｽﾄ拡張ｺﾏﾝﾄﾞ 情報をｺﾋﾟｰします。
5	DshInitTERCMD_INFO()	S2F49 情報構造体 TERCMD_INFO を初期化します。
6	DshPutTERCMD_PARA()	S2F49 情報構造体 TERCMD_INFO にﾊﾟﾗﾒｰﾀ情報を加えます。
7	DshInitTERCMD_ERR_INFO()	S2F50 応答情報構造体 TERCMD_ERR_INFO を初期化します。
8	DshPutTERCMD_ERR_PARA()	S2F50 応答情報構造体 TERCMD_ERR_INFO にｴﾗｰﾊﾟﾗﾒｰﾀ情報を加えます。
9	DshFreeTERCMD_ERR_INFO()	TERCMD_ERR_INFO 構造体内に使用されたﾒﾓﾘを開放します。
10	DshMakeS2F49Response()	S2F50 応答ﾒｯｾｰｼﾞ を TERCMD_ERR_INFO 構造体内のい情報に基づき生成します。

（3）ユーザ作成ライブラリ関数

	ﾗｲﾌﾞﾗﾘ関数名	機能
1	DshResponseS2F50()	S2F50 拡張リモートコマンド応答メッセージ

3．19．１　使用する情報格納構造体

（1）S2F49 情報格納用

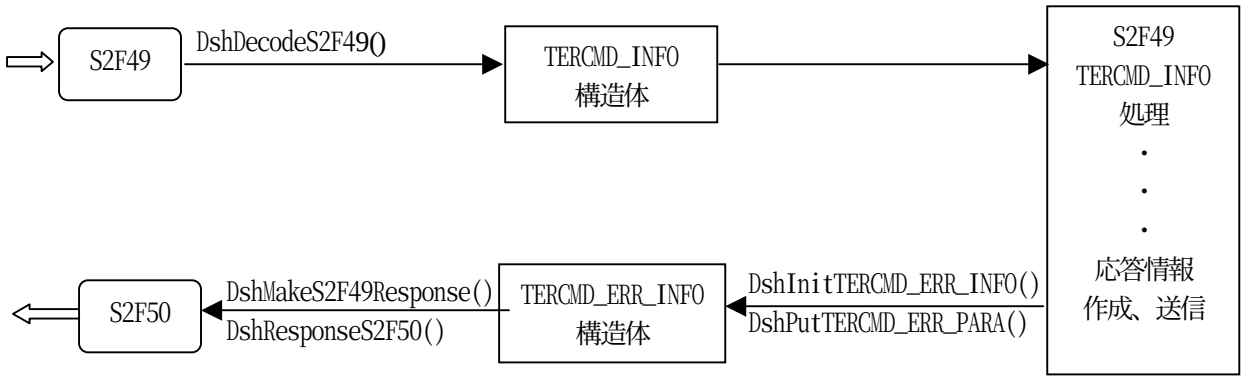
```
typedef struct{
        char        *objspec;           // object spec
        char        *rcmd;              // rcmd
        int         cp_count;           // parameter count
        TERCMD_PARA  **cpx_list;        // paramete list
        }TERCMD_INFO;


typedef struct{
        char        *cpname;            // cpname
        int         cpx_count;          // nest ﾊﾟﾗﾒｰﾀの数
        struct tercmd_para **cpx_list;  // nestｹｰｽの parameter list
        int         cpval_fmt;          // cpval item fmt
        int         cpval_size;         // cpval data array size
        void        *cpval;             // cpval
        }TERCMD_PARA;
```

（2） S2F50 エラー情報

```
typedef struct{
        int         hcack;              // B
        int         err_count;
        int         *err_list;          // err position list (ｴﾗｰの在ったﾊﾟﾗﾒｰﾀの順位 0,1..)
        int         *cepack_list;       // err_list 内順位の cpack のﾘｽﾄ( 値はBﾌｫｰﾏｯﾄです)
} TERCMD_ERR_INFO;
```

（3）ライブラリ関数との関係は次のようになります。



3．19．2　拡張リモートコマンド送信関数

3．19．2．1　GemSendS2F49() - 拡張リモートコマンドメッセージ送信関数

（1）呼出書式

[c,C++]

```
API int APIX GemSendS2F49(
        int     eqid,                   // 通信対象装置 ID(0,1,2,...)
        TERCMD_INFO *info,              // 拡張ﾘﾓｰﾄｺﾏﾝﾄﾞ情報格納領域のﾎﾟｲﾝﾀ
        TERCMD_ERR_INFO *erinfo,        // S4F50 応答情報格納用構造体のﾎﾟｲﾝﾀ
        int (WINAPI *callback)(),       // 実行終了時のｺｰﾙﾊﾞｯｸ関数
        ULONG  upara                    // callback 時のﾊﾟﾗﾒｰﾀ
);
```

[.NET VB]

```
int GemSendS2F49(
        int eqid,
        ref TERCMD_INFO info,
        ref TERCMD_HERR_INFO erinfo,
        CallbackS2F49 callback,
        uint upara );
```

[.NET C#]

```
Function GemSendS2F49 (
        ByVal eqid As Int32,
        ByRef info As dsh_info.TERCMD_INFO,
        ByRef erinfo As dsh_info.TERCMD_HERR_INFO,
        ByVal callback As vcallback.callback_S2F49,
        ByVal upara As Int32) As Int32
```

（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

送信したい拡張リモートコマンド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 S4F50 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) ﾌﾞﾛｯｸﾓｰﾄﾞ： 正常に送信できた。 erinfo に S4F50 応答情報が返却されます。 (2) 非ﾌﾞﾛｯｸﾓｰﾄﾞ：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

装置に拡張リモートコマンド情報の送信要求を行います。S2F49 メッセージで送信します。

要求を受けた DSHGEM-LIB は、info に格納されている拡張リモートコマンド情報を S2F49 メッセージにエンコードし、装置に送信します。

S4F50 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 S4F50 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)	S2F49 送信後、応答ﾒｯｾｰｼﾞ S4F50 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 S4F50 応答情報が返却されます。
あり	送信要求後、S2F49 の送信前に制御が戻されます。実行結果はｺｰﾙﾊﾞｯｸ関数で与えられます。 ｺｰﾙﾊﾞｯｸ関数の end_status が=0 ならば erinfo に S4F50 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 TERCMD_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、その構造体に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 DshFreeTERCMD_ERR_INFO()関数を使って行ってください。

```
DshFreeTERCMD_ERR_INFO (erinfo)
```

TERCMD_INFO 構造体への拡張リモートコマンド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。

```
DshInitTERCMD_INFO(), DshAddTERCMD_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
        TERCMD_ERR_INFO *erinfo,        // S4F50 応答情報格納用構造体のﾎﾟｲﾝﾀ
        ULONG upara                     // 呼出時に指定したﾊﾟﾗﾒｰﾀ
);
```

[.NET VB]

```
Function callback_S2F49(ByVal eqid As Integer, ByVal end_status As Integer, ByRef erinfo As
dsh_info.TERCMD_HERR_INFO, ByVal upara As Integer) As Integer
```

[.NET C#]

```
int CallbackS2F49(int eqid, int end_status, ref TERCMD_HERR_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 S4F50 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。

3．19．3　S2F49、S2F50 メッセージ処理関連関数

3．19．3．1　DshDecodeS2F49 - S2F49 デコード関数

（1）呼出書式

[c,C++]

```
API int  APIX DshDecodeS2F49(
        DSHMSG *smsg,                   // SECS ﾒｯｾｰｼﾞ情報構造体のﾎﾟｲﾝﾀ
        TERCMD_INFO *pinfo              // ﾃﾞｺｰﾄﾞした情報を格納する構造体のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshDecodeS2F49 (
        ByRef smsg As dshdr2.DSHMSG,
        ByRef info As dsh_info.TERCMD_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS2F49(
        ref DSHMSG smsg,
        ref TERCMD_INFO info );
```

（2）引数

smsg

S2F49 の SECS ﾒｯｾｰｼﾞ情報が格納されている構造体のポインタです。

pinfo

デコードした拡張リモートコマンド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	正常にﾃﾞｺｰﾄﾞできた。
(-1)	smsg を正しくﾃﾞｺｰﾄﾞできなかった。

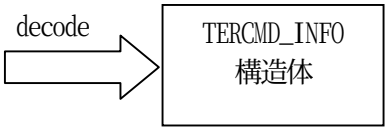
（4）説明

S2F49 メッセージに含まれる拡張リモートコマンド情報を、ユーザプログラムが処理しやすい TERCMD_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 DshFreeTERCMD_INFO()関数を使って開放してください。

smsg S2F49

```
L,4
  dataid
  objspec
  rcmd
  Ln
    L,2
      cpname1
      cepval1
```



3．19．3．2　DshEncodeS2F49() - ホスト拡張リモートコマンド情報を S2F49 へエンコード

（1）呼出書式

[c,C++]

```
API int  APIX DshEncodeS2F49(
        int     eqid,                   // 装置 ID(0,1..)
        DSHMSG *smsg,                   // SECS ﾒｯｾｰｼﾞ情報構造体のﾎﾟｲﾝﾀ
        BYTE   *buffer,                 // S2F49 を格納するﾊﾞｯﾌｧﾎﾟｲﾝﾀ
        int     bufflen,                // buffer のﾊﾞｲﾄｻｲｽﾞ
        TERCMD_INFO *info               // ｴﾝｺｰﾄﾞしたいﾎｽﾄ拡張ｺﾏﾝﾄﾞ情報格納構造体のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshEncodeS2F49 (
        ByVal eqid As Int32,
        ByRef smsg As dshdr2.DSHMSG,
        ByRef buff As Byte,
        ByVal bufflen As Int32,
        ByRef info As dsh_info.TERCMD_INFO) As Int32
```

[.NET C#]

```
int DshEncodeS2F49(
        int eqid,
        ref DSHMSG smsg,
        byte[] buff,
        int bufflen,
        ref TERCMD_INFO info);
```

（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。

smsg

エンコードした S2F49 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 S2F49 のテキストを格納するバッファポインタです。

bufflen

buffer のバイトサイズです。

info

エンコードしたいホスト拡張リモートコマンド情報が格納されている構造体のポインタです。

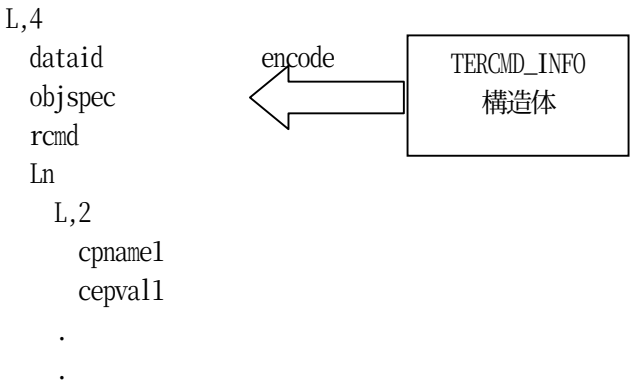
（3）戻り値

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

（4）説明

TERCMD_INFO 構造体に格納されているホスト拡張リモートコマンド情報を、S2F49 の SECS メッセージにエ

ンコードします。



TERCMD_INFO 構造体へのホストコマンド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。

DshInitTERCMD_INFO(), DshAddTERCMD_INFO()

３．19．３．３　DshFreeTERCMD_INFO() - 拡張リモートコマンド情報構造体メモリの開放

（1）呼出書式

[c,C++]

```
API void APIX DshFreeTERCMD_INFO(
      TERCMD_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ﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Sub DshFreeTERCMD_INFO (
        ByRef pinfo As dsh_info.TERCMD_INFO)
```

[.NET C#]

```
void DshFreeTERCMD_INFO(
        ref TERCMD_INFO pinfo );
```

（2）引数

pinfo

メモリを解放したい拡張リモートコマンド情報構造体のポインタです。

（3）戻り値

なし。

（4）説明

TERCMD_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、TERCMD_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。

３．19．３．４　DshCopyTERCMD_INFO() - 拡張リモートコマンド情報構造体メモリのコピー

（1）呼出書式

[c,C++]

```
API int APIX   DshCopyTERCMD_INFO(
        TERCMD_INFO *dinfo,                  // ｺﾋﾟｰ先のﾎﾟｲﾝﾀ
        TERCMD_INFO *sinfo                   // ｺﾋﾟｰ元のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshCopyTERCMD_INFO (
          ByRef dinfo As dsh_info.TERCMD_INFO,
          ByRef sinfo As dsh_info.TERCMD_INFO) As Int32
```

[.NET C#]

```
int DshCopyTERCMD_INFO(
          ref TERCMD_INFO dinfo,
          ref TERCMD_INFO sinfo );
```

（2）引数

dinfo

拡張リモートコマンド情報のコピー先構造体メモリのポインタです。

sinfo

コピー元の拡張リモートコマンド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	正常にｺﾋﾟｰできた。
(-1)	sinfo または dinfo の値が NULL だったのでｺﾋﾟｰできなかった。

（4）説明

sinfo が指す TERCMD_INFO 構造体内に格納されている拡張リモートコマンド情報を dinfo が指定する TERCMD_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、DshFreeTERCMD_INFO()関数を使って開放してください。

３．19．３．５　DshInitTERCMD_INFO - 拡張リモートコマンド情報 TERCMD_INFO の初期設定

（1）呼出書式

[c,C++]

```
API void  APIX DshInitTERCMD_INFO(
        TERCMD_INFO    *info,              // 拡張ﾘﾓｰﾄｺﾏﾝﾄﾞ情報 TERCMD_INFO 構造体のﾎﾟｲﾝﾀ
        char           *objspec,           // ｵﾌﾞｼﾞｪｸﾄ名
        char           *rcmd,              // ｺﾏﾝﾄﾞ名
        int            cp_count            // 設定できるｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀ数
);
```

[.NET VB]

```
Sub DshInitTERCMD_INFO (
        ByRef info As dsh_info.TERCMD_INFO,
        ByVal objspec As String,
        ByVal rcmd As String,
        ByVal cp_count As Int32)
```

[.NET C#]

```
void DshInitTERCMD_INFO(
        ref TERCMD_INFO info,
        byte[] objspec,
        byte[] rcmd,
        int cp_count);
```

（2）引数

info

　　拡張リモートコマンド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。

objspec

　　オブジェクト名が格納されているポインタです。

rcmd

　　コマンド名が格納されているポインタです。

cp_count

　　拡張リモートコマンド情報の中に設定できるコマンド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 TERCMD_INFO 構造体内に objspec のオブジェクト名と rcmd で指定されたコマンド名を設定します。また cp_count 分のコマンド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 DshAddTERCMD_INFO()関数を使用してください。

TERCMD_INFO 構造体の使用後は DshFreeTERCMD_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。

3．19．3．6　DshAddTERCMD_INFO() - 拡張リモートコマンドパラメータの追加

（1）呼出書式

[c,C++]

```
API int APIX DshAddTERCMD_INFO(
        TERCMD_INFO *info,              //拡張ﾘﾓｰﾄｺﾏﾝﾄﾞ情報構造体のﾎﾟｲﾝﾀ
        char        *cp_pname,          // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀ名
        int         fmt,                // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀのﾌｫｰﾏｯﾄ( ICODE_A, ICODE_U1 etc )
        int         asize,              // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀﾃﾞｰﾀの配列ｻｲｽﾞ
        void        *cp_val             // ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀﾃﾞｰﾀ格納ﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshAddTERCMD_INFO_PARA (
          ByRef info As dsh_info.TERCMD_INFO,
          ByVal cp_name As String,
          ByVal fmt As Int32,
          ByVal asize As Int32,
          ByRef cp_val As Int32) As Int32
```

[.NET C#]

```
int DshAddTERCMD_INFO_PARA(
          ref TERCMD_INFO info,
          byte[] cp_name,
          int fmt,
          int asize,
          byte[] cp_val);
```

（2）引数

info

拡張リモートコマンド情報構造体のポインタです。

cp_name

加えたいコマンドパラメータ名が格納されているポインタです。

fmt

コマンドパラメータデータのフォーマットです。(ICODE_A,ICODE_U1 など)

asize

コマンドパラメータデータの配列サイズです。

cp_val

コマンド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)	ｺﾏﾝﾄﾞﾊﾟﾗﾒｰﾀの数が既に指定数に達している。

（4）説明

先にDshInitTERCMD_INFO()で初期設定された拡張リモートコマンド情報構造体infoにコマンドパラメータを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TERCMD_PARAパラメータ構造体のメモリを確保し、その中にcp_nameとcp_valを設定し、cp_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cp_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。
それから、パラメータがリスト構造になる場合がありますが、そのネスティングレベルは1とします。
その場合、fmt = ICODE_Lにして、asizeに次のレベルでのパラメータ数を指定してください。

```
L,n                 ﾊﾟﾗﾒｰﾀ数=n
  L,2
    cpname-1
    L,m             ﾃﾞｰﾀではなく、List構造, ﾊﾟﾗﾒｰﾀが m 個
      L,2
        cpname-1-1
        cpval-1-1
      L,2
        .
```

下位レベルへのパラメータ追加はDshAddTERCMD_PARA()関数で設定することができます。

3．19．3．7　DshInitTERCMD_ERR_INFO () - ホスト拡張リモートコマンド応答情報の初期化

（1）呼出書式

[c,C++]

```
API void APIX   DshInitTERCMD_ERR_INFO(
        TERCMD_INFO *info,                  // S2F49 で得られたﾎｽﾄ拡張ﾘﾓｰﾄｺﾏﾝﾄﾞ情報構造体ﾎﾟｲﾝﾀ
        TERCMD_ERR_INFO *erinfo,            // ｴﾗｰ情報格納構造体ﾘｽﾄのﾎﾟｲﾝﾀ
        int         hcack,                  // ackﾃﾞｰﾀ
        int         err_count               // ｴﾗｰ情報のﾘｽﾄｻｲｽﾞ（個数　0,1,2...）
);
```

[.NET VB]

```
Function DshInitTERCMD_ERR_INFO (
        ByRef info As dsh_info.TERCMD_INFO,
        ByRef erinfo As dsh_info.TERCMD_ERR_INFO,
        ByVal hcack As Int32,
        ByVal err_count As Int32) As Int32
```

[.NET C#]

```
int DshInitTERCMD_ERR_INFO(
        ref TERCMD_INFO info,
        ref TERCMD_ERR_INFO erinfo,
        int hcack,
        int err_count );
```

（2）引数

info

S2F49 で得られた TERCMD_INFO 情報構造体のポインタです。

erinfo

TERCMD_ERR_INFO 応答情報構造体のポインタです。

hcack

hcack - 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、ホスト拡張リモートコマンド関連応答メッセージ TERCMD_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 hcack メンバーに引数 hcack 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、cpname_list と cpack_list に err_count 分だけのエラー情報のための領域を設けます。

3．19．3．8　DshPutTERCMD_ERR_PARA () - ホスト拡張リモートコマンドエラー情報の設定

（1）呼出書式

[c,C++]

```
API void APIX   DshPutTERCMD_ERR_PARA (
        TERCMD_INFO *info,                  // S2F49 で得られたﾎｽﾄ拡張ﾘﾓｰﾄｺﾏﾝﾄﾞ情報構造体ﾎﾟｲﾝﾀ
        TERCMD_ERR_INFO *erinfo,            // ｴﾗｰ情報格納構造体ﾘｽﾄのﾎﾟｲﾝﾀ
        int         order,                  // ﾊﾟﾗﾒｰﾀ位置順位(0,1,..)
        int         cpack                   // ﾊﾟﾗﾒｰﾀの ack
);
```

[.NET VB]

```
Function DshPutTERCMD_ERR_PARA (
        ByRef info As dsh_info.TERCMD_INFO,
        ByRef erinfo As dsh_info.TERCMD_ERR_INFO,
        ByVal order As Int32,
        ByVal cepack As Int32) As Int32
```

[.NET C#]

```
int DshPutTERCMD_ERR_PARA(
        ref TERCMD_INFO info,
        ref TERCMD_ERR_INFO erinfo,
        int order,
        int cepack );
```

（2）引数

info

S2F49 で得られた TERCMD_INFO 情報構造体のポインタです。

errinfo

ホスト拡張リモートコマンドエラー情報構造体のポインタです。

order

info 内のパラメーターリストの位置順位です。

cpack

設定するパラメータの ACK 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。
空きリストがあれば、その空きリストに order と cpack 値を設定します。

DshMakeS2F49Response()関数が、S2F50 メッセージのエンコードを行いますが、info と errinfo が引数として使用されます。

3．19．3．9　DshFreeTERCMD_ERR_INFO() - ホスト拡張コマンド応答情報メモリの開放

（1）呼出書式

[c,C++]

```
API void APIX DshFreeTERCMD_ERR_INFO(
      TERCMD_ERR_INFO *erinfo            // ﾒﾓﾘを開放したい応答情報格納構造体のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Sub DshFreeTERCMD_ERR_INFO (
        ByRef erinfo As dsh_info.TERCMD_ERR_INFO)
```

[.NET C#]

```
void DshFreeTERCMD_ERR_INFO(
        ref TERCMD_ERR_INFO erinfo );
```

（2）引数

erinfo

メモリを解放したいホスト拡張コマンド応答情報構造体のポインタです。

（3）戻り値

なし。

（4）説明

TERCMD_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。

3．19．3．10 DshMakeS2F49Response() - S2F49 の応答メッセージの生成

（1）呼出書式

[c,C++]

```
API int APIX DshMakeS2F49Response(
        TERCMD_INFO *info,              // ﾎｽﾄ拡張ﾘﾓｰﾄｺﾏﾝﾄﾞ情報格納領域ﾎﾟｲﾝﾀ
        TERCMD_ERR_INFO *erinfo,        // S2F50 に設定する応答情報格納領域のﾎﾟｲﾝﾀ
        DSHMSG *smsg,                   // S2F50 ﾒｯｾｰｼﾞを格納するﾒｯｾｰｼﾞ構造体のﾎﾟｲﾝﾀ
        BYTE    *buff,                  // S2F50 のﾃｷｽﾄ格納ﾊﾞｯﾌｧﾎﾟｲﾝﾀ
        int     buff_size               // buff のﾊﾞｲﾄｻｲｽﾞ
);
```

[.NET VB]

```
Function DshMakeS2F49Response (
          ByRef info As dsh_info.TERCMD_INFO,
          ByRef erinfo As dsh_info.TERCMD_ERR_INFO,
          ByRef smsg As dshdr2.DSHMSG,
          ByRef buff As Byte,
          ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS2F49Response(
          ref TERCMD_INFO info,
          ref TERCMD_ERR_INFO erinfo,
          ref DSHMSG smsg,
          byte[] buff,
          int buff_size );
```

（2）引数

info

ホスト拡張リモートコマンド情報が格納されている領域のポインタです。

erinfo

S2F50 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

S2F50 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

S2F50 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

S2F49 に対する S2F50 応答メッセージを info に含まれるホスト拡張リモートコマンド情報と応答情報に従って作成します。

応答情報内の、hcack を S2F50 の HCACK として設定します。
HCACK はユーザが S2F49 ホスト拡張リモートコマンドメッセージを評価した結果です。

erinfo の情報の設定は、DshInitTERCMD_ERR_INFO()と DshPutTERCMD_ERR_PARA()関数を使って行うことができます。

3．19．４　ユーザ作成ライブラリ関数

3．19．４．１　DshResponseS2F50() – S2F50 拡張リモートコマンド応答メッセージ

（1）呼出書式

[c,C++]

```
API int APIX DshResponseS2F50(
        int     eqid,                   // 通信対象装置 ID(0,16,...)
        ID_TR   trid,                   // DSHDR2 のﾄﾗﾝｻﾞｸｼｮﾝ ID
        TERCMD_INFO *info,              // 拡張ﾘﾓｰﾄｺﾏﾝﾄﾞﾒｯｾｰｼﾞ情報格納領域のﾎﾟｲﾝﾀ
        TERCMD_ERR_INFO *erinfo         // S2F50 応答情報格納用構造体のﾎﾟｲﾝﾀ
);
```

[.NET VB]

```
Function DshResponseS2F50 (
        ByVal eqid As Int32,
        ByVal trid As Int32,
        ByRef info As dsh_info.TERCMD_INFO,
        ByRef erinfo As dsh_info.TERCMD_ERR_INFO) As Int32
```

[.NET C#]

```
int DshResponseS2F50(
        int eqid,
        uint trid,
        ref TERCMD_INFO info,
        ref TERCMD_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

S2F49 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

拡張リモートコマン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 S2F50 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

拡張リモートコマンドメッセージ S2F49 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 TERCMD_ERR_INFO 構造体に含まれている情報から S2F50 メッセージを組み立て、その後、S2F50 メッセージを送信します。

送信が終ったら、TERCMD_ERR_INFO の構造体に使用されたメモリを DshFreeTERCMD_ERR_INFO ()関数を使って開放します。

なお、S2F50 メッセージの組み立てに、DshMakeS2F50Response()関数を使用できます。