

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 4 / 1 5

- 3 . 2 2 レチクル制御(S14F19,S14F21)サービス関数
- 3 . 2 3 レチクル搬送ジョブ要求(S3F35)関連関数

2 0 0 9年6月

株式会社データマップ

[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。

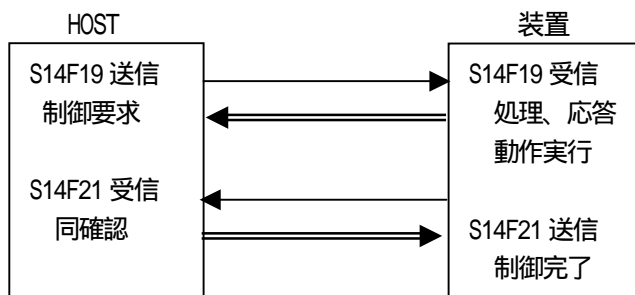
3.22 レチクル制御(S14F19,S14F21)サービス関数	1
3.22.1 使用する情報格納構造体.....	3
3.22.1.1 S14F19 – サービス要求メッセージ関連.....	3
3.22.1.2 S14F21 – サービス実行完了通知メッセージ関連.....	4
3.22.2 レチクル制御メッセージ送信関数.....	5
3.22.2.1 GemSendS14F19() – レチクル制御要求メッセージ送信関数.....	5
3.22.2.2 GemSendS14F21() – レチクル制御完了通知メッセージ送信関数.....	8
3.22.3 レチクル制御要求/完了通知ライブラリ関数.....	11
3.22.3.1 DshDecodeS14F19() - S14F19 をレチクル制御要求情報にデコード.....	11
3.22.3.2 DshEncodeS14F19() - レチクル制御要求情報を S14F19 へエンコード.....	12
3.22.3.3 DshFreeTS14F19_INFO() - レチクル制御要求情報構造体メモリの開放.....	14
3.22.3.4 DshCopyTS14F19_INFO() – レチクル制御情報構造体メモリのコピー.....	15
3.22.3.5 DshInitTS14F19_INFO – レチクル制御 TS14F19_INFO の初期設定.....	16
3.22.3.6 DshPutTS14F19_SP_INFO() – レチクル制御サービスパラメータの設定.....	18
3.22.3.7 DshDecodeS14F20() - S14F20 をレチクル制御要求応答情報にデコード.....	20
3.22.3.8 DshMakeS14F19Response() - S14F19 の応答メッセージの生成.....	21
3.22.3.9 DshFreeTS14F20_ERR_INFO() - レチクル制御応答情報構造体メモリの開放.....	22
3.22.3.10 DshCopyTS14F20_ERR_INFO() – レチクル制御応答情報構造体メモリのコピー.....	23
3.22.3.11 DshInitTS14F20_ERR_INFO – レチクル制御尾応答情報構造体の初期設定.....	24
3.22.3.12 DshPutTS14F20_SP_INFO() – レチクル制御サービスパラメータの設定.....	26
3.22.3.13 DshPutTS14F20_ERR_INFO () – エラー情報の設定.....	28
3.22.3.14 DshDecodeS14F21() - S14F21 をレチクル制御完了通知情報にデコード.....	29
3.22.3.15 DshEncodeS14F21() - レチクル制御完了通知情報を S14F21 へエンコード.....	30
3.22.3.16 DshFreeTS14F21_INFO() - レチクル制御完了通知情報構造体メモリの開放.....	32
3.22.3.17 DshCopyTS14F21_INFO() – レチクル制御完了通知情報構造体メモリのコピー.....	33
3.22.3.18 DshInitTS14F21_INFO – レチクル制御 TS14F21_INFO の初期設定.....	34
3.22.3.19 DshPutTS14F21_SP_INFO() – レチクル制御サービスパラメータの設定.....	36
3.22.3.20 DshPutTS14F21_ERR_INFO () – エラー情報の設定.....	38
3.22.3.21 DshMakeS14F21Response() - S14F21 の応答メッセージの生成.....	39
3.22.4 ユーザ作成ライブラリ関数.....	40
3.22.4.1 DshResponseS14F20() – S14F20 レチクル制御要求応答関数.....	40
3.22.4.2 DshResponseS14F22() – S14F22 レチクル制御完了通知応答関数.....	42
3.23 レチクル搬送ジョブ要求(S3F35)関連関数	44
3.23.1 使用する情報格納構造体.....	46
3.23.1.1 S3F35 – レチクル搬送ジョブ要求メッセージ関連.....	46
3.23.2 レチクル搬送ジョブ要求メッセージ送信関数.....	47
3.23.2.1 GemSendS3F35() – レチクル搬送ジョブ要求メッセージ送信関数.....	47
3.23.3 レチクル搬送ジョブ要求関連ライブラリ関数.....	50
3.23.3.1 DshDecodeS3F35() - S3F35 をレチクル搬送ジョブ要求情報にデコード.....	50
3.23.3.2 DshEncodeS3F35() - レチクル搬送ジョブ要求情報を S3F35 へエンコード.....	51
3.23.3.3 DshFreeTS3F35_INFO() - レチクル搬送ジョブ要求情報構造体メモリの開放.....	53
3.23.3.4 DshCopyTS3F35_INFO() – レチクル搬送ジョブ要求情報構造体メモリのコピー.....	54
3.23.3.5 DshInitTS3F35_INFO – レチクル搬送ジョブ要求 TS3F35_INFO の初期設定.....	55
3.23.3.6 DshAddTS3F35_INFO_ATTR_INFO () – レチクル搬送ジョブ要求属性の設定.....	57
3.23.3.7 DshSetTRET_ATTR_INFO () – レチクル搬送ジョブ要求属性情報の設定.....	58

3 . 23 . 3 . 8	DshFreeTRET_ATTR_INFO() – レチクル属性情報構造体メモリの開放.....	60
3 . 23 . 3 . 9	DshInitTRET_REMOVE_INST_INFO – レチクル抜き出し情報構造体の初期設定....	61
3 . 23 . 3 . 10	DshAddTRET_REMOVE_INST_ATTR – レチクル抜き出し情報に属性情報を追加..	62
3 . 23 . 3 . 11	DshAddTRET_REMOVE_INST_INFO () – 抜き出し情報の設定.....	63
3 . 23 . 3 . 12	DshFreeTRET_REMOVE_INST_INFO() – レチクル抜き出し情報構造体メモリの開放	64
3 . 23 . 3 . 13	DshCopyTRET_REMOVE_INST_INFO_INFO() – レチクル取り出し情報構造体のコピー	65
3 . 23 . 3 . 14	DshAddTRET_PLACE_INST_INFO – レチクル投入情報構造体の追加設定.....	66
3 . 23 . 3 . 15	DshDecodeS3F36() - S3F36 をレチクル搬送ジョブ要求応答情報にデコード.....	67
3 . 23 . 3 . 16	DshMakeS3F35Response() - S3F35 の応答メッセージの生成.....	68
3 . 23 . 3 . 17	DshFreeTS3F36_ERR_INFO() - レチクル搬送ジョブ要求応答情報メモリの開放.....	69
3 . 23 . 3 . 18	DshCopyTS3F36_ERR_INFO() – レチクル搬送ジョブ要求応答情報メモリのコピー .	70
3 . 23 . 3 . 19	DshInitTS3F36_ERR_INFO – レチクル搬送ジョブ要求尾応答情報構造体の初期設定	71
3 . 23 . 3 . 20	DshAddTS3F36_ERR_INFO() – エラー情報の設定.....	72
3 . 23 . 4	ユーザ作成ライブラリ関数.....	73
3 . 23 . 4 . 1	DshResponseS3F36() – S3F36 レチクル搬送ジョブ要求応答関数.....	73

(VOL - 15 に続く)

3.22 レチクル制御(S14F19,S14F21)サービス関数

レチクル管理に関する S14F19, S14F21 メッセージに関連する API 関数について説明します。
 (SEMI E109.1-0704 レチクル及び POD 管理(RPMS)に対する SECS-II プロトコルの暫定仕様 参照)



準備されている関数は、次のようになります。

(1) 送信 API 関数

	API 関数名	機能
1	GemSendS14F19()	S14F19 レチクルコントロール要求メッセージを送信します。(Host 側送信)
2	GemSendS14F21()	S14F21 レチクルコントロール完了メッセージを送信します。(装置側送信)

(2) ライブラリ関数

	ライブラリ関数名	機能
1	DshDecodeS14F19()	S14F19 メッセージをデコードし TS14F19_INFO 構造体に情報を格納します。
2	DshEncodeS14F19()	TS14F19_INFO 構造体の情報を S14F19 メッセージにエンコードします。
3	DshFreeTS14F19_INFO()	TS14F19_INFO 構造体内に使用したメモリを開放します。
4	DshCopyTS14F19_INFO()	TS14F19_INFO の内容を別の領域の構造体内にコピーします。
5	DshInitTS14F19_INFO()	S14F19 情報を格納する TS14F19_INFO 構造体の内容を初期化します。
6	DshPutTS14F19_SP_INFO()	TS14F19_INFO 構造体の中にサービスパラメータ 1 個だけ設定します。
7	DshDecodeS14F20()	S14F20 メッセージをデコードし TS14F20_ERR_INFO 構造体に情報を格納します。
8	DshMakeS14F19Response()	TS14F20_ERR_INFO 構造体の内容に従って S14F20 メッセージを作成します。
9	DshFreeTS14F20_ERR_INFO()	TS14F20_ERR_INFO 構造体内に使用したメモリを開放します。
10	DshCopyTS14F20_ERR_INFO()	TS14F20_ERR_INFO の内容を別の領域の構造体内にコピーします。
11	DshInitTS14F20_ERR_INFO()	S14F20 情報を格納する TS14F20_ERR_INF 構造体の内容を初期化します。

12	DshPutTS14F20_SP_INFO()	TS14F20_INFO 構造体の中にサービスパラメータ 1 個だけ設定します。
13	DshPutTS14F20_ERR_INFO()	TS14F20_INFO 構造体の中にエラー情報を 1 個だけ設定します。
14	DshDecodeS14F21()	S14F21 メッセージ をデコード し TS14F21_INFO 構造体に情報を格納します。
15	DshEncodeS14F21()	TS14F21_INFO 構造体の情報を S14F21 メッセージ にエンコード します。
16	DshFreeTS14F21_INFO()	TS14F21_INFO 構造体内に使用したメモリを開放します。
17	DshCopyTS14F21_INFO()	TS14F21_INFO の内容を別の領域の構造体内にコピー します。
18	DshInitTS14F21_INFO()	S14F19 情報を格納する TS14F19_INFO 構造体の内容を初期化します。
19	DshPutTS14F21_SP_INFO()	TS14F21_INFO 構造体の中にサービスパラメータ 1 個だけ設定します。
20	DshPutTS14F21_ERR_INFO()	TS14F21_INFO 構造体の中にエラー情報 1 個だけ設定します。
21	DshMakeS14F21Response()	応答 S14F22 メッセージ を作成します。

(3) ユーザ作成ライブラリ関数

	ライブラリ関数名	機能
1	DshResponseS14F20()	S14F20 レチクル制御要求応答関数
2	DshResponseS14F22()	S14F22 レチクル制御完了通知応答関数

3.22.1 使用する情報格納構造体

3.22.1.1 S14F19 サービス要求メッセージ関連

(1) S14F19 のエンコード、デコードに使用する構造体

```
typedef struct{
    ULONG      opid;           // OPID
    char       *objspec;      // object spec
    char       *svcname;      // svcname
    int        sp_count;      // parameter count
    TGSV_PARA **sp_list;      // parameter list
}TS14F19_INFO;

typedef struct tgsv_para{
    char       *spname;       // spname
    int        spval_fmt;     // spval item fmt
    int        spval_size;    // spval data array size
    void       *spval;        // spval
}TGSV_PARA;
```

svcname に対するインデクス値として以下のマクロを使用します。

```
#define RTC_CancelMoveReticle    0
#define RTC_MoveReticle          1
#define RTC_OktoUseReticle       2
#define RTC_RejectReticle        3
#define RTC_Re_qualifyReticle    4
```

spname に対するインデクス値として以下のマクロを使用します。

```
#define RTCPN_ReticleID          0
#define RTCPN_PortID            1
#define RTCPN_DestinationLocation 2
#define RTCPN_SourceLocation     3
#define RTCPN_QualificationIntervalTime 4
#define RTCPN_Status             5
```

(2) S14F20 のエンコード、デコードに使用する構造体

```
typedef struct{
    int        svcack;         // Bin
    ULONG      linkid;        // Link ID
    int        sp_count;      // service para count
    TGSV_PARA **sp_list;      // parameter list
    int        err_count;     // # of err
    TERR_INFO **err_list;     // error info list
}TS14F20_ERR_INFO;
```

3 . 22 . 1 . 2 S14F21 サービス実行完了通知メッセージ関連

(1) S14F21

```
typedef struct{
    ULONG      opid;           // OPID
    ULONG      linkid;        // Link ID
    int        sp_count;      // parameter count
    TGSV_PARA  **sp_list;     // paramete list
    int        svcack;        // Bin
    int        err_count;     // error count
    TERR_INFO  **err_list;    // error info
}TS14F21_INFO;
```


3.22.2 レチクル制御メッセージ送信関数

3.22.2.1 GemSendS14F19() レチクル制御要求メッセージ送信関数

(1) 呼出書式

[C, C++]

```
API int WINAPI GemSendS14F19(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    TS14F19_INFO *info, // レチクル制御要求情報格納領域のポインタ
    TS14F20_ERR_INFO *erinfo, // S14F20 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 GemSendS14F19 (
    ByVal eqid As Int32,
    ByRef info As dsh_info.TS14F19_INFO,
    ByRef erinfo As dsh_info.TS14F20_ERR_INFO,
    ByVal callback As vcallback.callback_S14F19,
    ByVal upara As Int32) As Int32
```

[.NET C#]

```
int GemSendS14F19(
    int eqid,
    ref TS14F19_INFO info,
    ref TS14F20_ERR_INFO erinfo,
    CallbackDefault 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

送信したいレチクル制御要求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 S14F20 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 S14F20 応答情報が返却されます。 (2) 非ブロケット：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 S14F19 レチクル制御要求メッセージを送信します。

要求を受けた DSHGEM-LIB は、info に格納されているレチクル制御要求情報を S14F19 メッセージにエンコードした上で装置に送信します。

S14F20 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 S14F20 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)	S14F19 送信後、応答メッセージ S14F20 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 S14F20 応答情報が返却されます。
あり	送信要求後、S14F19 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 S14F20 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 TS14F19_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 DshFreeTS14F20_ERR_INFO() 関数を使って行ってください。

```
DshFreeTS14F20_ERR_INFO (erinfo)
```

TS14F19 構造体へのレチクル制御要求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。

```
DshInitTS14F19_INFO (), DshPutTS14F19_SP_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
    TS14F20_ERR_INFO *erinfo, // S14F20 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_S14F19(ByVal eqid As Integer, ByVal end_status As Integer, ByRef erinfo As
```

dsh_info.TS14F20_ERR_INFO, ByVal upara As Integer) As Integer

[.NET C#]

```
int CallbackS14F19(int eqid, int end_status, ref TOBJ_S14F20_ERR_INFO errinfo, 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 S14F20 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。

3.22.2.2 GemSendS14F21() レチクル制御完了通知メッセージ送信関数

(1) 呼出書式

[C, C++]

```
API int APIX GemSendS14F21(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    TS14F21 *info, // レチクル制御完了情報格納領域のポインタ
    int *dataack, // S14F22 の dataack 格納ポインタ
    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 GemSendS14F21 (
    ByVal eqid As Int32,
    ByRef info As dsh_info.TS14F21_INFO,
    ByRef dataack As Int32,
    ByVal callback As vcallback.callback_S14F21,
    ByVal upara As Int32) As Int32
```

[.NET C#]

```
int GemSendS14F21(
    int eqid,
    ref TS14F21_INFO info,
    ref int dataack,
    CallbackS14F21 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

送信したいレチクル制御完了通知情報が格納されている構造体のポインタです。

dataack

受信した応答メッセージ S14F22 に含まれる dataack 情報を格納するための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 S14F22 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

ホストに S14F21 レチクル制御完了通知メッセージを送信します。

要求を受けた DSHGEM-LIB は、info に格納されているレチクル制御完了通知情報を S14F21 メッセージにエンコードした上でホストに送信します。

S14F22 応答メッセージ受信で得られた ACK 情報は dataack で指定された領域に返却されます。

送信要求から S14F22 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)	S14F21 送信後、応答メッセージ S14F22 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 S14F22 応答情報が返却されます。
あり	送信要求後、S14F21 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 S14F22 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、dataack で指定した領域に ACK 情報が設定返却されます。
=0 が正常に

TS14F21 構造体へのレチクル制御完了通知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。

```
DshInitTS14F21_INFO (), DshPutTS14F21_SP_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
    int *dataack,       // S14F22 の ACK、dataack の格納領域
    ULONG upara        // 呼出時に指定したパラメータ
);
```

[.NET VB]

```
Function callback_S14F21(ByVal eqid As Integer, ByVal end_status As Integer, ByRef dataack As Integer, ByVal upara As Integer) As Integer
```

[.NET C#]

```
int CallbackS14F21(int eqid, int end_status, ref int dataack, 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。dataack に応答 ACK 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。

3.22.3 レチクル制御要求 / 完了通知ライブラリ関数

3.22.3.1 DshDecodeS14F19() - S14F19 をレチクル制御要求情報にデコード

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS14F19(
    DSHMSG *smsg,           // SECS メッセージ 情報構造体のポインタ
    TS14F19_INFO *pinfo     // デコードしたレチクル制御要求情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS14F19 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef info As dsh_pinfo.TS14F19_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS14F19(
    ref DSHMSG smsg,
    ref TS14F19_INFO pinfo );
```

(2) 引数

smsg

S14F19 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

デコードしたレチクル制御要求情報を格納するための構造体ポインタです。

(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。

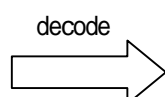
(4) 説明

S14F19 メッセージに含まれるレチクル制御要求情報を、ユーザプログラムが処理しやすい TS14F19_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 DshFreeTS14F19_INFO() 関数を使って開放してください。

smsg S14F19

```
L,5
dataid
opid
objspec
svcname
L,m
```



3.22.3.2 DshEncodeS14F19() - レチクル制御要求情報を S14F19 へエンコード

(1) 呼出書式

[C, C++]

```
API int APIX DshEncodeS14F19(
    int eqid, // 装置 ID
    DSHMSG *smsg, // SECS メッセージ情報構造体のポインタ
    BYTE *buffer, // S14F19 を格納するバッファポインタ
    int buflen, // buffer のバイトサイズ
    TS14F19_INFO *pinfo // エンコードしたいレチクル制御要求情報格納構造体のポインタ
);
```

[.NET VB]

```
Function DshEncodeS14F19 (
    ByVal eqid As Int32,
    ByRef smsg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buflen As Int32,
    ByRef info As dsh_info.TS14F19_INFO) As Int32
```

[.NET C#]

```
int DshEncodeS14F19(
    int eqid,
    ref DSHMSG smsg,
    byte* buff,
    int buflen,
    ref TS14F19_INFO info );
```

(2) 引数

smsg

エンコードした S14F19 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 S14F19 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

エンコードしたいレチクル制御要求情報が格納されている構造体リストのポインタです。

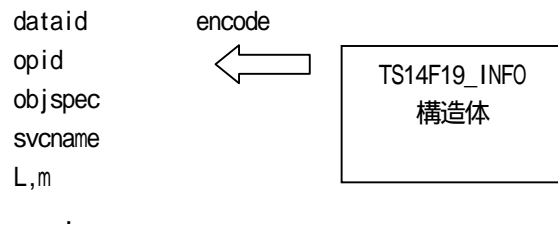
(3) 戻り値

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

TS14F19_INFO 構造体に格納されているレチクル制御要求情報を、S14F19 の SECS メッセージにエンコードします。

smsg S14F19
L,5



3.22.3.3 DshFreeTS14F19_INFO() - レチクル制御要求情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTS14F19_INFO(
    TS14F19_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 DshFreeTS14F19_INFO (
    ByRef info As dsh_info.TS14F19_INFO)
```

[.NET C#]

```
void DshFreeTS14F19_INFO(
    ref TS14F19_INFO info );
```

(2) 引数

pinfo

メモリを解放したいレチクル制御要求情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TS14F19_INFO 構造体内でレチクル制御要求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、TS14F19_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。

3.22.3.4 DshCopyTS14F19_INFO() レチクル制御情報構造体メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTS14F19_INFO(
    TS14F19_INFO *dinfo,           // 北°-先のポインタ
    TS14F19_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTS14F19_INFO (
    ByRef dinfo As dsh_info.TS14F19_INFO,
    ByRef sinfo As dsh_info.TS14F19_INFO) As Int32
```

[.NET C#]

```
void DshCopyTS14F19_INFO(
    ref TS14F19_INFO dinfo,
    ref TS14F19_INFO sinfo );
```

(2) 引数

dinfo

レチクル制御情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のレチクル制御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 TS14F19_INFO 構造体内に格納されているレチクル制御情報を dinfo が指定する TS14F19_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、DshFreeTS14F19_INFO()関数を使って開放してください。

3.22.3.5 DshInitTS14F19_INFO レチクル制御 TS14F19_INFO の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTS14F19_INFO(
    TS14F19_INFO *info,           // TS14F19_INFO 構造体のポインタ
    ULONG        opid,           // オペレーション ID
    char         *objspec,       // object spec
    char         *svcname,       // サービス名
    int          spcount        // サービスパラメータ数
);
```

[.NET VB]

```
Sub DshInitTS14F19_INFO (
    ByRef info As dsh_info.TS14F19_INFO,
    ByVal opid As Int32,
    ByVal objspec As String,
    ByVal svcname As String,
    ByVal spcount As Int32)
```

[.NET C#]

```
void DshInitTS14F19_INFO(
    ref TS14F19_INFO info,
    uint opid,
    byte* objspec,
    byte* svcname,
    int spcount );
```

(2) 引数

info

TS14F19_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。この中に S14F19 生成情報を詰めます。

opid

オペレーション ID です。

objspec

Object Spec 文字列です。NULL でも構いません。

svcname

サービス名称です。SEMI-E109.1-0704 の暫定仕様では以下のものがあります。

```
"CancelMoveReticle", // 0
"MoveReticle",       // 1
"OktoUseReticle",    // 2
"RejectReticle",     // 3
"Re-qualifyReticle", // 4
```

spcount

付属されるサービスパラメータの数です。

(3) 戻り値

なし。

(4) 説明

S14F19メッセージをエンコードするときに使用する情報をTS14F19_INFO構造体内に生成するために使用します。

本関数は、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。メモリが必要な構造体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。

サービスパラメータ情報の設定にはDshPutTS14F19_SP_INFO()関数を使用してください。

TS14F19_INFO 構造体の使用後はDshFreeTS14F19_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。

3.22.3.6 DshPutTS14F19_SP_INFO() レチクル制御サービスパラメータの設定

(1) 呼出書式

[C, C++]

```
API int APIX DshPutTS14F19_SP_INFO(
    TS14F19_INFO *info,           // TS14F19_INFO 情報構造体のポインタ
    char          *spname,        // サービスパラメータ名称
    int           fmt,           // サービスパラメータ値のフォーマット
    int           size,          // サービスパラメータ値のデータ数
    void          *spval         // サービスパラメータ値格納領域のポインタ
);
```

[.NET VB]

```
Function DshPutTS14F19_SP_INFO (
    ByRef info As dsh_info.TS14F19_INFO,
    ByVal spname As String,
    ByVal fmt As Int32,
    ByVal size As Int32,
    ByVal spval As Int32) As Int32
```

[.NET C#]

```
int DshPutTS14F19_SP_INFO(
    ref TS14F19_INFO info,
    byte[] spname,
    int fmt,
    int size,
    byte* spval );
```

(2) 引数

info

TS14F19_INFO 情報構造体のポインタです。この中に S14F19 のサービスパラメータ生成情報を詰めます。

spname

設定サービスパラメータ名です。SEMI-E109.1-0704 の暫定仕様では以下のものがあります。

```
"ReticleID",           // 0
"PortID",              // 1
"DestinationLocation", // 2
"SourceLocation",     // 3
"QualificationIntervalTime", // 4
"Status",              // 5
```

fmt

サービスパラメータデータのフォーマットです。DSHDR2 で使用するデータフォーマットで指定します。(ICODE_?? ??はB, A など)

size

サービスパラメータデータの個数です。文字列の場合は、文字列数(バイト)になります。

spval

サービスパラメータデータの値が格納されている領域のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	パラメータ情報の数が指定数を超えている。

(4) 説明

先に `DshInitTS14F19_Info()` で初期設定された `info` 構造体の中の `sp_list` にサービスパラメータ情報を 1 個加えます。

本関数が実行される順に属性情報リスト `sp_list` 上に、情報を順に設定していきます。正常に設定できた場合は 0 を返却します。

もし、`info` 内の `sp_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。

3.22.3.7 DshDecodeS14F20() - S14F20 をレチクル制御要求応答情報にデコード

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS14F20(
    DSHMSG *msg,           // SECSメッセージ情報構造体のポインタ
    TS14F20_ERR_INFO *pinfo // デコードしたCJ生成応答情報を格納する構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS14F20 (
    ByRef msg As dshdr2.DSHMSG,
    ByRef erinfo As dsh_info.TS14F20_ERR_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS14F20(
    ref DSHMSG msg,
    ref TS14F20_ERR_INFO erinfo );
```

(2) 引数

msg

S14F20 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

デコードしたレチクル制御要求応答情報を格納するための構造体ポインタです。

(3) 戻り値

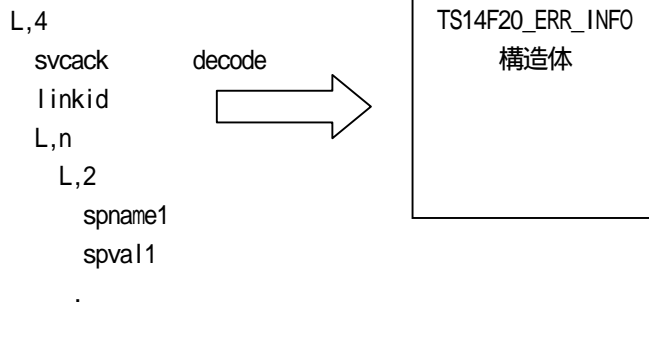
戻り値	意味
0	正常にデコードできた。
(-1)	msg を正しくデコードできなかった。

(4) 説明

S14F20 メッセージに含まれるレチクル制御要求応答情報を、ユーザプログラムが処理しやすい TS14F20_ERR_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 DshFreeTS14F20_ERR_INFO()関数を使って開放してください。

msg S14F20



3.22.3.8 DshMakeS14F19Response() - S14F19 の応答メッセージの生成

(1) 呼出書式

[C, C++]

```
API int APIX DshMakeS14F19Response(
    TS14F20_ERR_INFO *erinfo,      // S14F20 に設定する応答情報格納領域のポインタ
    DSHMSG *smsg,                  // S14F20 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,                    // S14F20 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 DshMakeS14F19Response (
    ByRef erinfo As dsh_info.TS14F20_ERR_INFO,
    ByRef smsg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS14F19Response(
    ref TS14F20_ERR_INFO erinfo,
    ref DSHMSG smsg,
    byte* buff,
    int buff_size );
```

(2) 引数

erinfo

S14F20 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

S14F20 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

S14F20 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

S14F19 に対する S14F20 応答メッセージを erinfo 内に設定されている応答情報に従って作成します。erinfo の情報の設定は、DshInitTS14F20_ERR_INFO() と DshPutTS14F20_SP_INFO() 関数を使って行うことができます。

3.22.3.9 DshFreeTS14F20_ERR_INFO() - レチクル制御応答情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTS14F20_ERR_INFO(
    TS14F20_ERR_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 DshFreeTS14F20_ERR_INFO (
    ByRef info As dsh_info.TS14F20_ERR_INFO)
```

[.NET C#]

```
void DshFreeTS14F20_ERR_INFO(
    ref TS14F20_ERR_INFO info );
```

(2) 引数

pinfo

メモリを解放したいレチクル制御応答情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TS14F20_ERR_INFO 構造体内で応答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、TS14F20_ERR_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。

3.22.3.10 DshCopyTS14F20_ERR_INFO() レチクル制御応答情報構造体メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTS14F20_ERR_INFO(
    TS14F20_ERR_INFO *dinfo,           // 北°-先のポインタ
    TS14F20_ERR_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTS14F20_ERR_INFO (
    ByRef dinfo As dsh_info.TS14F20_ERR_INFO,
    ByRef sinfo As dsh_info.TS14F20_ERR_INFO) As Int32
```

[.NET C#]

```
int DshCopyTS14F20_ERR_INFO(
    ref TS14F20_ERR_INFO dinfo,
    ref TS14F20_ERR_INFO sinfo );
```

(2) 引数

dinfo

レチクル制御応答情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のレチクル制御応答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 TS14F20_ERR_INFO 構造体内に格納されているレチクル制御応答情報を dinfo が指定する TS14F20_ERR_INFO 構造体に構造体ごとコピーします。

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、DshFreeTS14F20_ERR_INFO()関数を使って開放してください。

3.22.3.11 DshInitTS14F20_ERR_INFO レチクル制御尾応答情報構造体の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTS1420_ERR_Info(
    TS14F20_ERR_INFO *erinfo,      // TS14F20_ERR_INFO 構造体のポインタ
    int svckack,                    // S14F19 に対する ACK
    ULONG linkid,                   // リンク ID
    int spcount,                    // サービスパラメータ数
    int errcount                     // エラー情報数
);
```

[.NET VB]

```
Sub DshInitTS14F20_ERR_INFO (
    ByRef erinfo As dsh_info.TS14F20_ERR_INFO,
    ByVal svckack As Int32,
    ByVal linkid As Int32,
    ByVal sp_count As Int32,
    ByVal err_count As Int32)
```

[.NET C#]

```
void DshInitTS14F20_ERR_INFO(
    ref TS14F20_ERR_INFO erinfo,
    int svckack,
    uint linkid,
    int sp_count,
    int err_count );
```

(2) 引数

info

TS14F20_ERR_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。この中に S14F19 生成情報を詰めます。

svckack

制御要求を受け付けたかどうかを示す ACK です。(Binary)

linkid

S14F19 要求と S14F21 完了通知との間の対応をとるための ID です。

spcount

付属されるサービスパラメータの数です。

errcount

付属されるエラー情報の数です。

(3) 戻り値

なし。

(4) 説明

S14F20 メッセージをエンコードするときに使用する情報を TS14F20_ERR_INFO 構造体内に生成するために使用します。

本関数は、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。
メモリが必要な構造体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。

サービスパラメータ情報の設定には DshPutTS14F20_SP_INFO() 関数を使用してください。

エラー情報の設定には DshPutTS14F20_ERR_INFO () 関数を使用してください。

TS14F20_ERR_INFO 構造体の使用後は DshFreeTS14F20_ERR_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。

3.22.3.12 DshPutTS14F20_SP_INFO() レチクル制御サービスパラメータの設定

(1) 呼出書式

[C, C++]

```
API int APIX DshPutTS14F20_SP_INFO (
    TS14F20_ERR_INFO *errinfo, // TS14F20_ERR_INFO 情報構造体のポインタ
    char *spname, // サービスパラメータ名称
    int fmt, // サービスパラメータ値のフォーマット
    int size, // サービスパラメータ値のデータ数
    void *spval // サービスパラメータ値格納領域ポインタ
);
```

[.NET VB]

```
Function DshPutTS14F20_SP_INFO (
    ByRef erinfo As dsh_info.TS14F20_ERR_INFO,
    ByVal spname As String,
    ByVal fmt As Int32,
    ByVal size As Int32,
    ByVal spval As Int32) As Int32
```

[.NET C#]

```
int DshPutTS14F20_SP_INFO(
    ref TS14F20_ERR_INFO erinfo,
    byte* spname,
    int fmt,
    int size,
    byte* spval );
```

(2) 引数

errinfo

TS14F20_ERR_INFO 情報構造体のポインタです。この中に S14F19 のサービスパラメータ生成情報を詰めます。

spname

設定サービスパラメータ名です。SEMI-E109.1-0704 の暫定仕様では以下のものがあります。

```
"ReticleID", // 0
"PortID", // 1
"DestinationLocation", // 2
"SourceLocation", // 3
"QualificationIntervalTime", // 4
"Status", // 5
```

fmt

サービスパラメータデータのフォーマットです。DSHDR2 で使用するデータフォーマットで指定します。(ICODE_?? ??はB, A など)

size

サービスパラメータデータの個数です。文字列の場合は、文字列数(バイト)になります。

spval

サービスパラメータデータの値が格納されている領域のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	パラメータ情報の数が指定数を超えている。

(4) 説明

先に `DshInitTS14F20_ERR_Info()` で初期設定された `errinfo` 構造体の中の `sp_list` にサービスパラメータ情報を 1 個加えます。

本関数が実行される順に属性情報リスト `sp_list` 上に、情報を順に設定していきます。正常に設定できた場合は 0 を返却します。

もし、`errinfo` 内の `sp_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。

3.22.3.13 DshPutTS14F20_ERR_INFO () エラー情報の設定

(1) 呼出書式

[C, C++]

```
API int APIX DshPutTS14F20_ERR_INFO (
    TS14F20_ERR_INFO *erinfo,          // TS14F20_ERR_INFO 情報格納構造体のポインタ
    int errcode,                        // 設定するエラーコード
    char *errtext                       // 設定するエラーテキストのポインタ
);
```

[.NET VB]

```
Function DshPutTS14F20_ERR_INFO (
    ByRef erinfo As dsh_info.TS14F20_ERR_INFO,
    ByVal errcode As Int32,
    ByVal errtext As String) As Int32
```

[.NET C#]

```
int DshPutTS14F20_ERR_INFO(
    ref TS14F20_ERR_INFO erinfo,
    int errcode,
    byte* errtext );
```

(2) 引数

erinfo

TS14F20_ERR_INFO 情報構造体のポインタです。

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

erinfo で指定された TS14F20_ERR_INFO 構造体内の err_list 上に新たにエラー情報 ercode と errtext を設定します。

正常に設定できた場合は 0 を返却します。

erinfo 内の err_list 上のエラー数が err_count の数に達していた場合は (-1)を返却します。

本関数を最初に実行する前に、DshInitTS14F20_ERR_INFO()関数を使ってリストの初期設定を行ってください。

3.22.3.14 DshDecodeS14F21() - S14F21 をレチクル制御完了通知情報にデコード

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS14F21(
    DSHMSG *smsg,           // SECS メッセージ 情報構造体のポインタ
    TS14F21_INFO *pinfo    // デコードしたレチクル制御完了通知情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS14F21 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef info As dsh_info.TS14F21_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS14F21(
    ref DSHMSG smsg,
    ref TS14F21_INFO info );
```

(2) 引数

smsg

S14F21 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

デコードしたレチクル制御完了通知情報を格納するための構造体ポインタです。

(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

S14F21 メッセージに含まれるレチクル制御完了通知情報を、ユーザプログラムが処理しやすい TS14F21_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 DshFreeTS14F21_INFO()関数を使って開放してください。

smsg S14F21

L,5

dataid

opid

linkid

L,n

L,2

.

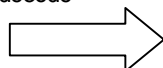
L,2

svcack

L,2

.

decode



3.22.3.15 DshEncodeS14F21() - レチクル制御完了通知情報を S14F21 へエンコード

(1) 呼出書式

[C, C++]

```
API int APIX DshEncodeS14F21(
    int eqid, // 装置 ID
    DSHMSG *smsg, // SECS メッセージ 情報構造体のポインタ
    BYTE *buffer, // S14F21 を格納するバッファポインタ
    int buflen, // buffer のバイトサイズ
    TS14F21_INFO *pinfo // レチクル制御完了通知情報格納構造体のポインタ
);
```

[.NET VB]

```
Function DshEncodeTS14F21 (
    ByVal eqid As Int32,
    ByRef smsg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buflen As Int32,
    ByRef pinfo As dsh_info.TS14F21_INFO) As Int32
```

[.NET C#]

```
int DshEncodeS14F21(
    int eqid,
    ref DSHMSG smsg,
    byte[] buff,
    int buflen,
    ref TS14F21_INFO pinfo );
```

(2) 引数

smsg

エンコードした S14F21 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 S14F21 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

エンコードしたいレチクル制御完了通知情報が格納されている構造体リストのポインタです。

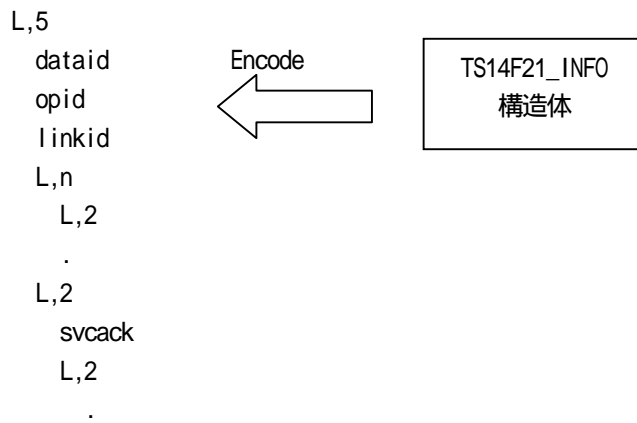
(3) 戻り値

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

TS14F21_INFO 構造体に格納されているレチクル制御完了通知情報を、S14F21 の SECS メッセージにエンコードします。

smsg S14F21



3.22.3.16 DshFreeTS14F21_INFO() - レチクル制御完了通知情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTS14F21_INFO(
    TS14F21_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 DshFreeTS14F21_INFO (
    ByRef pinfo As dsh_info.TS14F21_INFO)
```

[.NET C#]

```
void DshFreeTS14F21_INFO(
    ref TS14F21_INFO pinfo );
```

(2) 引数

pinfo

メモリを解放したいレチクル制御完了通知情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TS14F21_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、TS14F21_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。

3.22.3.17 DshCopyTS14F21_INFO() レチクル制御完了通知情報構造体メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTS14F21_INFO(
    TS14F21_INFO *dinfo,           // 北°-先のポインタ
    TS14F21_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTS14F21_INFO (
    ByRef dinfo As dsh_info.TS14F21_INFO,
    ByRef sinfo As dsh_info.TS14F21_INFO) As Int32
```

[.NET C#]

```
int DshCopyTS14F21_INFO(
    ref TS14F21_INFO dinfo,
    ref TS14F21_INFO sinfo );
```

(2) 引数

dinfo

レチクル制御情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のレチクル制御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 TS14F21_INFO 構造体内に格納されているレチクル制御完了通知情報を dinfo が指定する TS14F21_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、DshFreeTS14F21_INFO()関数を使って開放してください。

3.22.3.18 DshInitTS14F21_INFO レチクル制御 TS14F21_INFO の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTS14F21_INFO(
    TS14F21_INFO *info,           // TS14F21_INFO 構造体のポインタ
    ULONG        opid,           // オペレーション ID
    char         *objspec,       // object spec
    char         *svcname,       // サービス名
    int          spcount         // サービスパラメータ数
);
```

[.NET VB]

```
Sub DshInitTS14F21_INFO (
    ByRef info As dsh_info.TS14F21_INFO,
    ByVal opid As Int32,
    ByVal linkid As Int32,
    ByVal spcount As Int32,
    ByVal svcack As Int32,
    ByVal errcount As Int32)
```

[.NET C#]

```
void DshInitTS14F21_INFO(
    ref TS14F21_INFO info,
    uint opid,
    uint linkid,
    int spcount,
    int svcack,
    int errcount );
```

(2) 引数

info

TS14F21_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。この中に S14F21 生成情報を詰めます。

opid

オペレーション ID です。

objspec

Object Spec 文字列です。NULL でも構いません。

svcname

サービス名称です。SEMI-E109.1-0704 の暫定仕様では以下のものがあります。

```
"CancelMoveReticle", // 0
"MoveReticle",       // 1
"OktoUseReticle",    // 2
"RejectReticle",     // 3
"Re-qualifyReticle", // 4
```

spcount

付属されるサービスパラメータの数です。

(3) 戻り値
なし。

(4) 説明

S14F21 メッセージをエンコードするときに使用する情報を TS14F21_INFO 構造体内に生成するために使用します。

本関数は、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。メモリが必要な構造体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。

サービスパラメータ情報の設定には DshPutTS14F21_SP_INFO() 関数を使用してください。

TS14F21_INFO 構造体の使用後は DshFreeTS14F21_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。

3.22.3.19 DshPutTS14F21_SP_INFO() レチクル制御サービスパラメータの設定

(1) 呼出書式

[C, C++]

```
API int APIX DshPutTS14F21_SP_INFO(
    TS14F21_INFO *info,           // TS14F21_INFO 情報構造体のポインタ
    char *spname,                // サービスパラメータ名称
    int fmt,                     // サービスパラメータ値のフォーマット
    int size,                    // サービスパラメータ値のデータ数
    void *spval                  // サービスパラメータ値格納領域ポインタ
);
```

[.NET VB]

```
Function DshPutTS14F21_SP_INFO (
    ByRef info As dsh_info.TS14F21_INFO,
    ByVal spname As String,
    ByVal fmt As Int32,
    ByVal size As Int32,
    ByVal spval As Int32) As Int32
```

[.NET C#]

```
int DshPutTS14F21_SP_INFO(
    ref TS14F21_INFO info,
    byte[] spname,
    int fmt,
    int size,
    byte[] spval );
```

(2) 引数

info

TS14F21_INFO 情報構造体のポインタです。この中に S14F21 のサービスパラメータ生成情報を詰めます。

spname

設定サービスパラメータ名です。SEMI-E109.1-0704 の暫定仕様では以下のものがあります。

```
"ReticleID",           // 0
"PortID",              // 1
"DestinationLocation", // 2
"SourceLocation",      // 3
"QualificationIntervalTime", // 4
"Status",              // 5
```

fmt

サービスパラメータデータのフォーマットです。DSHDR2 で使用するデータフォーマットで指定します。(ICODE_?? ??はB, A など)

size

サービスパラメータデータの個数です。文字列の場合は、文字列数(バイト)になります。

spval

サービスパラメータデータの値が格納されている領域のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	パラメータ情報の数が指定数を超えている。

(4) 説明

先に `DshInitTS14F21_Info()` で初期設定された `info` 構造体の中の `sp_list` にサービスパラメータ情報を1個加えます。

本関数が実行される順に属性情報リスト `sp_list` 上に、情報を順に設定していきます。正常に設定できた場合は 0 を返却します。

もし、`info` 内の `sp_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。

3.22.3.20 DshPutTS14F21_ERR_INFO () エラー情報の設定

(1) 呼出書式

[C, C++]

```
API int APIX DshPutTS14F21_ERR_INFO (
    TS14F21_INFO *erinfo,          // TS14F21_INFO 情報格納構造体のポインタ
    int errcode,                   // 設定するエラーコード
    char *errtext                  // 設定するエラーテキストのポインタ
);
```

[.NET VB]

```
Function DshPutTS14F21_ERR_INFO (
    ByRef info As dsh_info.TS14F21_INFO,
    ByVal errcode As Int32,
    ByVal errtext As String) As Int32
```

[.NET C#]

```
int DshPutTS14F21_ERR_INFO(
    ref TS14F21_INFO info,
    int errcode,
    byte[] errtext );
```

(2) 引数

erinfo

TS14F21_INFO 情報構造体のポインタです。

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

erinfo で指定された TS14F21_INFO 構造体内の err_list 上に新たにエラー情報 errcode と errtext を設定します。

正常に設定できた場合は 0 を返却します。

erinfo 内の err_list 上のエラー数が err_count の数に達していた場合は (-1)を返却します。

本関数を最初に実行する前に、DshInitTS14F21_INFO()関数を使ってリストの初期設定を行ってください。

3.22.3.21 DshMakeS14F21Response() - S14F21 の応答メッセージの生成

(1) 呼出書式

[C, C++]

```
API int APIX DshMakeS14F21Response(
    int dataack,           // S14F22 に設定する dataack の値
    DSHMSG *msg,          // S14F22 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,           // S14F22 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 DshMakeS14F21Response (
    ByVal dataack As Int32,
    ByRef msg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS14F21Response(
    int dataack,
    ref DSHMSG msg,
    byte[] buff,
    int buff_size );
```

(2) 引数

dataack

S14F22 メッセージに設定する dataack です。

msg

S14F22 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

S14F22 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

S14F21 に対する S14F22 応答メッセージを msg, buff 内に作成します。

S14F22 の ACK として dataack の値を設定します。

3.22.4 ユーザ作成ライブラリ関数

3.22.4.1 DshResponseS14F20() – S14F20 レチクル制御要求応答関数

(1) 呼出書式

[c, C++]

```
API int APIX DshResponseS14F20(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    ID_TR trid, // DSHDR2 のトランザクション ID
    TS14F19_INFO *info, // レチクル制御要求情報格納領域のポインタ
    TS14F20_ERR_INFO *erinfo // S14F20 応答情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshResponseS14F20 (
    ByVal eqid As Int32,
    ByVal trid As Int32,
    ByRef info As dsh_info.TS14F19_INFO,
    ByRef erinfo As dsh_info.TS14F20_ERR_INFO) As Int32
```

[.NET C#]

```
int DshResponseS14F20(
    int eqid,
    uint trid,
    ref TS14F19_INFO info,
    ref TS14F20_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

S14F19 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

レチクル制御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 S14F20 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

レチクル制御要求メッセージ S14F19 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 TS14F20_ERR_INFO 構造体に含まれている情報から S14F20 メッセージを組み立て、その後、S14F20 メッセージを送信します。

送信が終わったら、TS14F20_ERR_INFO の構造体で使用されたメモリを DshFreeTS14F20_ERR_INFO() 関数を使って開放します。

なお、S14F20 メッセージの組み立てに、DshMakeS14F19Response() 関数を使用できます。

3.22.4.2 DshResponseS14F22() S14F22 レチクル制御完了通知応答関数

(1) 呼出書式

[C, C++]

```
API int APIX DshResponseS14F22(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    ID_TR trid, // DSHDR2 のトランザクション ID
    TS14F21_INFO *info, // レチクル制御要求情報格納領域のポインタ
    TS14F22_ERR_INFO *erinfo // S14F22 応答情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshResponseS14F22 (
    ByVal eqid As Int32,
    ByVal trid As Int32,
    ByRef info As dsh_info.TS14F21_INFO,
    ByVal dataack As Int32) As Int32
```

[.NET C#]

```
int DshResponseS14F22(
    int eqid,
    ID_TR trid,
    ref TS14F21_INFO info,
    int dataack);
```

(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

S14F21 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

レチクル制御完了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 S14F22 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

レチクル制御完了通知メッセージ S14F21 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)

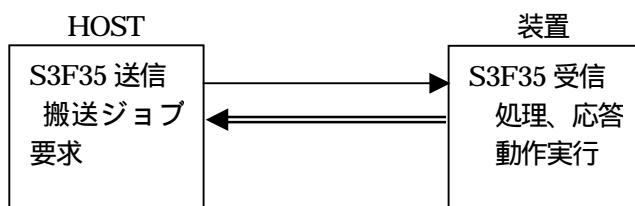
引数に指定されている TS14F22_ERR_INFO 構造体に含まれている情報から S14F22 メッセージを組み立て、その後、S14F22 メッセージを送信します。

送信が終わったら、TS14F22_ERR_INFO の構造体で使用されたメモリを DshFreeTS14F22_ERR_INFO()関数を使って開放します。

なお、S14F22 メッセージの組み立てに、DshMakeS14F21Response()関数を使用できます。

3.23 レチクル搬送ジョブ要求(S3F35)関連関数

レチクル搬送に関する S3F35 メッセージに関連する API 関数について説明します。



準備されている関数は、次のようになります。

(1) 送信 API 関数

	API 関数名	機能
1	GemSendS3F35()	S3F35 レチクルコントロール要求メッセージを送信します。(Host 側送信)

(2) ライブラリ関数

	ライブラリ関数名	機能
1	DshDecodeS3F35()	S3F35 メッセージをデコードし TS3F35_INFO 構造体に情報を格納します。
2	DshEncodeS3F35()	TS3F35_INFO 構造体の情報を S3F35 メッセージにエンコードします。
3	DshFreeTS3F35_INFO()	TS3F35_INFO 構造体内に使用したメモリを開放します。
4	DshCopyTS3F35_INFO()	TS3F35_INFO の内容を別の領域の構造体内にコピーします。
5	DshInitTS3F35_INFO()	S3F35 情報を格納する TS3F35_INFO 構造体の内容を初期化します。
6	DshAddTS3F35_INFO_ATTR_INFO()	TS3F35_INFO 構造体の中に属性情報を 1 個だけ追加設定します。
7	DshSetTRET_ATTR_INFO()	レチクル属性情報体に属性データを設定します。
8	DshFree TRET_ATTR_INFO()	TRET_ATTR_INFO 構造体内に使用したメモリを開放します。
9	DshInitTRET_REMOVE_INST_INFO()	レチクル抜取り情報 TRET_REMOVE_INST_INFO() 構造体の初期設定を行います。
10	DshAddTRET_REMOVE_INST_ATTR()	レチクル抜取り情報に属性情報を加えます。
11	DshAddTRET_REMOVE_INST_INFO()	レチクル抜取り情報 TRET_REMOVE_INST_INFO を TS3F35_INFO に追加します。
12	DshFreeTRET_REMOVE_INST_INFO()	TRET_REMOVE_INST_INFO 構造体内に使用したメモリを開放します。
13	DshCopyTRET_REMOVE_INST_INFO	抜取り情報構造体の内容を別の構造体にコピーします。

14	DshAddTRET_PLACE_INST_INFO()	リクル投入情報構造体を TS3F35_INFO 構造体に追加します。
15	DshDecodeS3F36()	S3F36 メッセージをデコードし TS3F36_ERR_INFO 構造体に情報を格納します。
16	DshMakeS3F35Response()	TS3F36_ERR_INFO 構造体の内容に従って S3F36 メッセージを作成します。
17	DshFreeTS3F36_ERR_INFO()	TS3F36_ERR_INFO 構造体内に使用したメモリを開放します。
18	DshCopyTS3F36_ERR_INFO()	TS3F36_ERR_INFO の内容を別の領域の構造体内にコピーします。
19	DshInitTS3F36_ERR_INFO()	S3F36 情報を格納する TS3F36_ERR_INF 構造体の内容を初期化します。
20	DshAddTS3F36_ERR_INFO()	TS3F36_INFO 構造体の中にエラー情報を 1 個だけ設定します。

(3) ユーザ作成ライブラリ関数

	ライブラリ関数名	機能
1	DshResponseS3F36()	S3F36 レチクル搬送ジョブ要求応答関数

3.23.1 使用する情報格納構造体

3.23.1.1 S3F35 レチクル搬送ジョブ要求メッセージ関連

(1) S3F35 のエンコード、デコードに使用する構造体

```

typedef struct{
    char      *attrid;                // attrid
    int       fmt;
    int       size;
    BYTE      *attrdata;             // data
} TRET_ATTR_INFO;
//-----
typedef struct{
    char      *ret_id;                // reticle id
    char      *inst;                 // instruction
} TRET_PLACE_INST_INFO;
//-----
typedef struct{
    char      *ret_id;                // ret inst
    char      *inst;
    int       attr_count;
    TRET_ATTR_INFO **attr_list;
} TRET_REMOVE_INST_INFO;
//-----
typedef struct{
    char      *jobaction;             // action
    char      *podid;
    int       in_ptn;
    int       out_ptn;
    int       attr_count;
    TRET_ATTR_INFO **attr_list;
    int       remove_inst_count;
    TRET_REMOVE_INST_INFO **remove_inst_list;
    int       place_inst_count;
    TRET_PLACE_INST_INFO **place_inst_list;
} TS3F35_INFO;

```

(2) S3F36 のエンコード、デコードに使用する構造体

```

//----- S3F36 information -----
typedef struct{
    int       rpmack;
    int       err_count;
    TERR_INFO **err_list;
} TS3F36_ERR_INFO;

```

3.23.2 レチクル搬送ジョブ要求メッセージ送信関数

3.23.2.1 GemSendS3F35() レチクル搬送ジョブ要求メッセージ送信関数

(1) 呼出書式

[C, C++]

```
API int WINAPI GemSendS3F35(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    TS3F35_INFO *info, // レチクル搬送ジョブ要求情報格納領域のポインタ
    TS3F36_ERR_INFO *erinfo, // S3F36 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 GemSendS3F35 (
    ByVal eqid As Int32,
    ByRef info As dsh_info.TS3F35_INFO,
    ByRef erinfo As dsh_info.TS3F36_ERR_INFO,
    ByVal callback As vcallback.callback_S3F35,
    ByVal upara As Int32) As Int32
```

[.NET C#]

```
int GemSendS3F35(
    int eqid,
    ref TS3F35_INFO info,
    ref TS3F36_ERR_INFO erinfo,
    CallbackS3F35 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

送信したいレチクル搬送ジョブ要求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 S3F36 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 S3F36 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

装置に S3F35 レチクル搬送ジョブ要求メッセージを送信します。

要求を受けた DSHGEM-LIB は、info に格納されているレチクル搬送ジョブ要求情報を S3F35 メッセージにエンコードした上で装置に送信します。

S3F36 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 S3F36 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)	S3F35 送信後、応答メッセージ S3F36 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 S3F36 応答情報が返却されます。
あり	送信要求後、S3F35 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 S3F36 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 TS3F35_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 DshFreeTS3F36_ERR_INFO()関数を使って行ってください。

```
DshFreeTS3F36_ERR_INFO (erinfo)
```

TS3F35 構造体へのレチクル搬送ジョブ要求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。

```
DshInitTS3F35_INFO (),DshAddTS3F35_INFO_ATTR_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
    TS3F36_ERR_INFO *erinfo, // S3F36 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_S3F35(ByVal eqid As Integer, ByVal end_status As Integer, ByRef erinfo As  
dsh_info.TS3F36_ERR_INFO, ByVal upara As Integer) As Integer
```

[.NET C#]

```
int CallbackS3F35(int eqid, int end_status, ref TS3F36_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 S3F36 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。

3.23.3 レチクル搬送ジョブ要求関連ライブラリ関数

3.23.3.1 DshDecodeS3F35() - S3F35 をレチクル搬送ジョブ要求情報にデコード

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS3F35(
    DSHMSG *msg,           // SECS メッセージ情報構造体のポインタ
    TS3F35_INFO *pinfo     // デコードしたレチクル搬送ジョブ要求情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS3F35 (
    ByRef msg As dshdr2.DSHMSG,
    ByRef pinfo As dsh_info.TS3F35_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS3F35(
    ref DSHMSG msg,
    ref TS3F35_INFO info );
```

(2) 引数

msg

S3F35 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

デコードしたレチクル搬送ジョブ要求情報を格納するための構造体ポインタです。

(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。

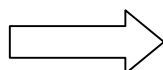
(4) 説明

S3F35 メッセージに含まれるレチクル搬送ジョブ要求情報を、ユーザプログラムが処理しやすい TS3F35_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 DshFreeTS3F35_INFO() 関数を使って開放してください。

msg S3F35

```
L,7
jobaction;
podid;
in_ptn;
out_ptn;
```



decode



3.23.3.2 DshEncodeS3F35() - レチクル搬送ジョブ要求情報を S3F35 へエンコード

(1) 呼出書式

[C, C++]

```
API int APIX DshEncodeS3F35(
    DSHMSG *smsg,          // SECS メッセージ 情報構造体のポインタ
    BYTE *buffer,         // S3F35 を格納するバッファポインタ
    int buflen,           // buffer のバイトサイズ
    TS3F35_INFO *pinfo    // エンコードしたいレチクル搬送ジョブ要求情報格納構造体のポインタ
);
```

[.NET VB]

```
Function DshEncodeS3F35 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buflen As Int32,
    ByRef info As dsh_info.TS3F35_INFO) As Int32
```

[.NET C#]

```
int DshEncodeS3F35(
    ref DSHMSG smsg,
    byte[] buff,
    int buflen,
    ref TS3F35_INFO info );
```

(2) 引数

smsg
エンコードした S3F35 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 S3F35 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
エンコードしたいレチクル搬送ジョブ要求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。

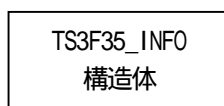
(4) 説明

TS3F35_INFO 構造体に格納されているレチクル搬送ジョブ要求情報を、S3F35 の SECS メッセージにエンコードします。

smsg S3F35

L,7

jobaction;
podid;



```
in_ptn;  
out_ptn;      encode  
.
```


3.23.3.3 DshFreeTS3F35_INFO() - レチクル搬送ジョブ要求情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTS3F35_INFO(
    TS3F35_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 DshFreeTS3F35_INFO (
    ByRef info As dsh_info.TS3F35_INFO)
```

[.NET C#]

```
void DshFreeTS3F35_INFO(
    ref TS3F35_INFO info );
```

(2) 引数

pinfo

メモリを解放したいレチクル搬送ジョブ要求情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TS3F35_INFO 構造体内でレチクル搬送ジョブ要求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、TS3F35_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。

3.23.3.4 DshCopyTS3F35_INFO() レチクル搬送ジョブ要求情報構造体メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTS3F35_INFO(
    TS3F35_INFO *dinfo,           // 北°-先のポインタ
    TS3F35_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTS3F35_INFO (
    ByRef dinfo As dsh_info.TS3F35_INFO,
    ByRef sinfo As dsh_info.TS3F35_INFO) As Int32
```

[.NET C#]

```
int DshCopyTS3F35_INFO(
    ref TS3F35_INFO dinfo
    ref TS3F35_INFO sinfo);
```

(2) 引数

dinfo

レチクル搬送ジョブ要求情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のレチクル搬送ジョブ要求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 TS3F35_INFO 構造体内に格納されているレチクル搬送ジョブ要求情報を dinfo が指定する TS3F35_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、DshFreeTS3F35_INFO()関数を使って開放してください。

3.23.3.5 DshInitTS3F35_INFO レチクル搬送ジョブ要求 TS3F35_INFO の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTS3F35_INFO(
    TS3F35_INFO *info,           // TS3F35_INFO 構造体のポインタ
    char *jobaction,
    char *podid,
    int in_ptn,
    int out_ptn,
    int attr_count,
    int remove_inst_count,
    int place_inst_count
);
```

[.NET VB]

```
Sub DshInitTS3F35_INFO (
    ByRef info As dsh_info.TS3F35_INFO,
    ByVal jobaction As String,
    ByVal podid As String,
    ByVal in_ptn As Int32,
    ByVal out_ptn As Int32,
    ByVal attr_count As Int32,
    ByVal remove_inst_count As Int32,
    ByVal place_inst_count As Int32)
End Sub
```

[.NET C#]

```
void DshInitTS3F35_INFO(
    ref TS3F35_INFO info,
    byte[] jobaction,
    byte[] podid,
    int in_ptn,
    int out_ptn,
    int attr_count,
    int remove_inst_count,
    int place_inst_count);
```

(2) 引数

info

TS3F35_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。この中に S3F35 生成情報を詰めます。

jobaction

jobaction 文字列です。

podid

podid 文字列です。

in_ptn

input port です。

attr_count

attr 情報数です。
remove_inst_count
 抜取り指示数です。
place_inst_count
 投入指示数です。

(3) 戻り値

なし。

(4) 説明

S3F35 メッセージをエンコードするときに使用する情報を TS3F35_INFO 構造体内に生成するために使用します。

本関数は、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。メモリが必要な構造体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。

属性情報の設定には、以下の関数を使用します。

```
DshAddTS3F35_INFO_ATTR_INFO( )  
DshSetTRET_ATTR_INFO( )
```

それから TS3F35_INFO 内の情報の中に、取出し指示情報と投入指示情報がありますが、それぞれ以下の関数を使用できます。

取出し指示情報

```
DshInitTRET_REMOVE_INST_INFO()  
DshAddTRET_REMOVE_INST_INFO()  
DshAddTS3F35_INFO_REMOVE_INST_INFO()
```

投入指示情報

```
DshAddTRET_PLACE_INST_INFO()
```

TS3F35_INFO 構造体の使用後は DshFreeTS3F35_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。

3.23.3.6 DshAddTS3F35_INFO_ATTR_INFO () レチクル搬送ジョブ要求属性の設定

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTS3F35_INFO_ATTR_INFO(
    TS3F35_INFO *info,           // TS3F35_INFO 情報構造体のポインタ
    TRET_ATTR_INFO *ainfo       // 属性情報が格納されている構造体のポインタ
);
```

[.NET VB]

```
Function DshAddTS3F35_INFO_ATTR_INFO (
    ByRef info As dsh_info.TS3F35_INFO,
    ByRef ainfo As dsh_info.TRET_ATTR_INFO) As Int32
```

[.NET C#]

```
int DshAddTS3F35_INFO_ATTR_INFO(
    ref TS3F35_INFO info,
    ref TRET_ATTR_INFO ainfo );
```

(2) 引数

info

TS3F35_INFO 情報構造体のポインタです。この中に S3F35 の属性情報を詰めます。

ainfo

属性情報が格納されている構造体のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	属性情報の数が指定数を超えている。

(4) 説明

先に DshInitTS3F35_Info() で初期設定された info 構造体の中の attr_list に属性情報を 1 個加えます。本関数が実行される順に属性情報リスト attr_list 上に、順番に情報を設定していきます。正常に設定できた場合は 0 を返却します。

もし、info 内の attr_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。

3.23.3.7 DshSetTRET_ATTR_INFO () レチクル搬送ジョブ要求属性情報の設定

(1) 呼出書式

[C, C++]

```
API void APIX DshSetTRET_ATTR_INFO(
    TRET_ATTR_INFO *info           // 属性情報が格納されている構造体のポインタ
    char *attrid,                  // 属性 ID
    int fmt,                       // 属性データのフォーマット
    int size,                      // size データ数
    byte *attrdata                 // 属性データ格納番地
);
```

[.NET VB]

```
Sub DshSetTRET_ATTR_INFO (
    ByRef info As dsh_info.TRET_ATTR_INFO,
    ByVal attrid As String,
    ByVal fmt As Int32,
    ByVal size As Int32,
    ByRef attrdata As Byte)
```

[.NET C#]

```
void DshSetTRET_ATTR_INFO(
    ref TRET_ATTR_INFO info,
    byte *attrid,
    int fmt,
    int size,
    byte *attrdata );
```

(2) 引数

info

TRET_ATTR_INFO 属性情報構造体のポインタです。この中に attrid 以降の引数の情報を設定します。

attrid

属性 ID の文字が格納されているポインタです。

fmt

属性データのフォーマットです。DSHDR2 ドライバーのフォーマットを指定します。
ICODE_A, ICODE_U4 など

size

属性データのデータ数です。ICODE_A の場合は文字列長になります。

attrdata

属性データが格納されているメモリの先頭番地です。

(3) 戻り値

ありません。

(4) 説明

先に TRET_ATTR_INFO 構造体に属性 ID と属性データを設定します。

設定した後、DshAddTS3F35_INFO_ATTR_INFO()関数を使って TS3F35_INFO 構造体に追加します。

DshAddTS3F35_INFO_ATTR_INFO()関数で追加した後、DshTRET_ATTR_INFO()を使って、info 内に情報格納用に獲得使用されたメモリを開放してください。

3.23.3.8 DshFreeTRET_ATTR_INFO() レチクル属性情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTRET_ATTR_INFO(
    TRET_ATTR_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 DshFreeTRET_ATTR_INFO (
    ByRef info As dsh_info.TRET_ATTR_INFO)
```

[.NET C#]

```
void DshFreeTRET_ATTR_INFO(
    ref TRET_ATTR_INFO info );
```

(2) 引数

pinfo

メモリを解放したいレチクル属性情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TRET_ATTR_INFO 構造体内でレチクル属性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、TRET_ATTR_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。

3.23.3.9 DshInitTRET_REMOVE_INST_INFO レチクル抜き出し情報構造体の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTRET_REMOVE_INST_INFO(
    TRET_REMOVE_INST_INFO *info,          // TRET_REMOVE_INST_INFO 構造体のポインタ
    char *ret_id,
    char *inst,
    int count
);
```

[.NET VB]

```
Sub DshInitTRET_REMOVE_INST_INFO (
    ByRef info As dsh_info.TRET_REMOVE_INST_INFO,
    ByVal ret_id As String,
    ByVal inst As String,
    ByVal count As Int32)
End Sub
```

[.NET C#]

```
void DshInitTRET_REMOVE_INST_INFO(
    ref TRET_REMOVE_INST_INFO info,
    byte[] ret_id,
    byte[] inst,
    int count);
```

(2) 引数

info

TRET_REMOVE_INST_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。この中に抜き出し指令情報を詰めます。

ret_id

レチクル ID (文字列) です。

inst

指示情報 (文字列) です。

count

抜き出し情報に加える属性情報の数を指定します。

(3) 戻り値

なし。

(4) 説明

TS3F35_INFO 内に含める TRET_REMOVE_INST_INFO 構造体に対し初期設定をします。

本関数は、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。メモリが必要な構造体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。

3.23.3.10 DshAddTRET_REMOVE_INST_ATTR レチクル抜き出し情報に属性情報を追加

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTRET_REMOVE_INST_ATTR(
    TRET_REMOVE_INST_ATTR *info, // TRET_REMOVE_INST_ATTR 構造体のポインタ
    TRET_ATTR_INFO *ainfo // 属性情報構造体のポインタ
);
```

[.NET VB]

```
Function DshAddTRET_REMOVE_INST_ATTR (
    ByRef info As dsh_info.TRET_REMOVE_INST_INFO,
    ByRef ainfo As dsh_info.TRET_ATTR_INFO) As Int32
```

[.NET C#]

```
int DshAddTRET_REMOVE_INST_ATTR(
    ref TRET_REMOVE_INST_INFO info,
    ref TRET_ATTR_INFO ainfo);
```

(2) 引数

info

TRET_REMOVE_INST_ATTR 構造体のポインタです。

ainfo

レチクル属性情報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)	属性情報の数が指定数を超えている。

(4) 説明

TRET_REMOVE_INST_ATTR 内に属性情報を追加します。

ainfo を使用後は、DshFreeTRET_ATTR_INFO()関数を使って ainfo 内に使用したメモリを開放します。

3.23.3.11 DshAddTRET_REMOVE_INST_INFO () 抜き出し情報の設定

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTRET_REMOVE_INST_INFO(
    TS3F35_INFO *info,           // TS3F35_INFO 情報構造体のポインタ
    TRET_REMOVE_INST_INFO *ainfo // 抜き出し情報が格納されている構造体のポインタ
);
```

[.NET VB]

```
Function DshAddTRET_REMOVE_INST_INFO (
    ByRef info As dsh_info.TS3F35_INFO,
    ByRef ainfo As dsh_info.TRET_REMOVE_INST_INFO) As Int32
```

[.NET C#]

```
int DshAddTRET_REMOVE_INST_INFO(
    ref TS3F35_INFO info,
    ref TRET_REMOVE_INST_INFO ainfo );
```

(2) 引数

info

TS3F35_INFO 情報構造体のポインタです。この中に S3F35 の抜き出し情報を詰めます。

ainfo

抜き出し情報が格納されている構造体のポインタを指定します。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	抜き出し情報の数が指定数を超えている。

(4) 説明

先に DshInitTS3F35_Info() で初期設定された info 構造体の中の remove_inst_list に抜き出し情報を 1 個加えます。

本関数が実行される順に抜き出し情報リスト remove_inst_list 上に、順番に情報を設定していきます。正常に設定できた場合は 0 を返却します。

もし、info 内の remove_inst_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。

3.23.3.12 DshFreeTRET_REMOVE_INST_INFO() レチクル抜き出し情報構造体メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTRET_REMOVE_INST_INFO(  
    TRET_REMOVE_INST_INFO *pinfo // メモリを開放したい情報の格納ポインタ  
);
```

[.NET VB]

```
Sub DshFreeTRET_REMOVE_INST_INFO (  
    ByRef info As dsh_info.TRET_REMOVE_INST_INFO)
```

[.NET C#]

```
void DshFreeTRET_REMOVE_INST_INFO(  
    ref TRET_REMOVE_INST_INFO info);
```

(2) 引数

pinfo

メモリを解放したいレチクル抜き出し情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TRET_REMOVE_INST_INFO 構造体内でレチクル抜き出し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、TRET_REMOVE_INST_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。

3.23.3.13 DshCopyTRET_REMOVE_INST_INFO_INFO() レチクル取出し情報構造体のコピー

(1) 呼出書式

[C, C++]

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

[.NET VB]

```
Function DshCopyTRET_REMOVE_INST_INFO (
    ByRef dinfo As dsh_info.TRET_REMOVE_INST_INFO,
    ByRef sinfo As dsh_info.TRET_REMOVE_INST_INFO) as Int32
```

[.NET C#]

```
int DshCopyTRET_PLACE_INST_INFO(
    ref TRET_PLACE_INST_INFO dinfo,
    ref TRET_PLACE_INST_INFO sinfo );
```

(2) 引数

dinfo

レチクル取出し情報構造体のコピー先構造体メモリのポインタです。

sinfo

コピー元のレチクル取出し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 TRET_REMOVE_INST_INFO_INFO 構造体内に格納されているレチクル取出し情報を dinfo が指定する TRET_REMOVE_INST_INFO_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、DshFreeTRET_REMOVE_INST_INFO_INFO()関数を使って開放してください。

3.23.3.14 DshAddTRET_PLACE_INST_INFO レチクル投入情報構造体の追加設定

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTRET_PLACE_INST_INFO(
    TS3F35_INFO *info,          // TRET_PLACE_INST_INFO 構造体のポインタ
    char *ret_id,
    char *inst,
);
```

[.NET VB]

```
Function DshAddTRET_PLACE_INST_INFO (
    ByRef info As dsh_info.TS3F35_INFO,
    ByVal ret_id As String,
    ByVal inst As String) As Int32
```

[.NET C#]

```
int DshAddTRET_PLACE_INST_INFO(
    ref TS3F35_INFO info,
    byte[] ret_id,
    byte[] inst );
```

(2) 引数

info

TS3F35 _INFO 構造体のポインタです。この構造体に投入指令情報を追加設定します。

ret_id

レチクル ID (文字列) です。

inst

指示情報 (文字列) です。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	投入情報の数が指定数を超えている。

(4) 説明

TS3F35_INFO 内の place_inst_list に RET_PLACE_INST_INFO 構造体を作り、その中に対しレチクル投入情報を追加設定します。(RET_PLCAE_INST_INFO 構造体は DSHGEMLIB によって準備されます。)

追加できる数は、DshInitTS3F35_INFO()関数で指定した place_inst_count の数までです。

3.23.3.15 DshDecodeS3F36() - S3F36 をレチクル搬送ジョブ要求応答情報にデコード

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS3F36(
    DSHMSG *msg,           // SECS メッセージ 情報構造体のポインタ
    TS3F36_ERR_INFO *pinfo // デコードした CJ 生成応答情報を格納する構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS3F36 (
    ByRef msg As dshdr2.DSHMSG,
    ByRef erinfo As dsh_info.TS3F36_ERR_INFO) As Int32
```

[.NET C#]

```
int DshDecodeS3F36(
    ref DSHMSG msg,
    ref TS3F36_ERR_INFO erinfo );
```

(2) 引数

msg

S3F36 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

デコードしたレチクル搬送ジョブ要求応答情報を格納するための構造体ポインタです。

(3) 戻り値

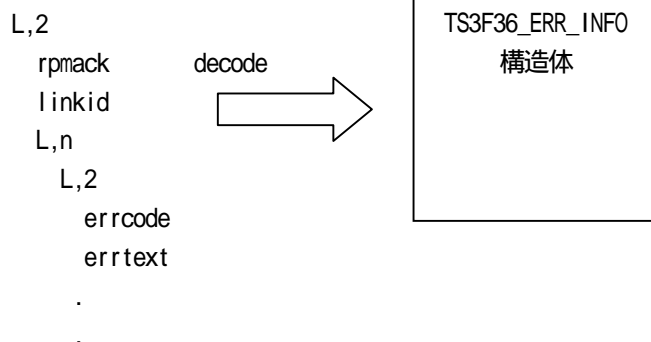
戻り値	意味
0	正常にデコードできた。
(-1)	msg を正しくデコードできなかった。

(4) 説明

S3F36 メッセージに含まれるレチクル搬送ジョブ要求応答情報を、ユーザプログラムが処理しやすい TS3F36_ERR_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 DshFreeTS3F36_ERR_INFO()関数を使って開放してください。

msg S3F36



3.23.3.16 DshMakeS3F35Response() - S3F35 の応答メッセージの生成

(1) 呼出書式

[C, C++]

```
API int APIX DshMakeS3F35Response(
    TS3F36_ERR_INFO *erinfo,      // S3F36 に設定する応答情報格納領域のポインタ
    DSHMSG *msg,                  // S3F36 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,                   // S3F36 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 DshMakeS3F35Response (
    ByRef erinfo As dsh_info.TS3F36_ERR_INFO,
    ByRef msg As dshdr2.DSHMSG,
    ByRef buff As Byte,
    ByVal buff_size As Int32) As Int32
```

[.NET C#]

```
int DshMakeS3F35Response(
    ref TS3F36_ERR_INFO erinfo,
    ref DSHMSG msg,
    byte[] buff,
    int buff_size );
```

(2) 引数

erinfo

S3F36 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

S3F36 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

S3F36 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

S3F35 に対する S3F36 応答メッセージを erinfo 内に設定されている応答情報に従って作成します。

erinfo の情報の設定は、DshInitTS3F36_ERR_INFO() と DshPutTS3F36_SP_INFO() 関数を使って行うことができます。

3.23.3.17 DshFreeTS3F36_ERR_INFO() - レチクル搬送ジョブ要求応答情報メモリの開放

(1) 呼出書式

[C, C++]

```
API void APIX DshFreeTS3F36_ERR_INFO(  
    TS3F36_ERR_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 DshFreeTS3F36_ERR_INFO (  
    ByRef pinfo As dsh_info.TS3F36_ERR_INFO)
```

[.NET C#]

```
void DshFreeTS3F36_ERR_INFO(  
    ref TS3F36_ERR_INFO pinfo );
```

(2) 引数

pinfo

メモリを解放したいレチクル搬送ジョブ要求応答情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TS3F36_ERR_INFO 構造体内で応答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、TS3F36_ERR_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。

3.23.3.18 DshCopyTS3F36_ERR_INFO() レチクル搬送ジョブ要求応答情報メモリのコピー

(1) 呼出書式

[C, C++]

```
API int APIX DshCopyTS3F36_ERR_INFO(
    TS3F36_ERR_INFO *dinfo,           // 北°-先のポインタ
    TS3F36_ERR_INFO *sinfo           // 北°-元のポインタ
);
```

[.NET VB]

```
Function DshCopyTS3F36_ERR_INFO (
    ByRef dinfo As dsh_info.TS3F36_ERR_INFO,
    ByRef sinfo As dsh_info.TS3F36_ERR_INFO) As Int32
```

[.NET C#]

```
int DshCopyTS3F36_ERR_INFO(
    ref TS3F36_ERR_INFO dinfo,
    ref TS3F36_ERR_INFO sinfo);
```

(2) 引数

dinfo

レチクル搬送ジョブ要求応答情報のコピー先構造体メモリのポインタです。

sinfo

コピー元のレチクル搬送ジョブ要求応答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 TS3F36_ERR_INFO 構造体内に格納されているレチクル搬送ジョブ要求応答情報を dinfo が指定する TS3F36_ERR_INFO 構造体に構造体ごとコピーします。

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、DshFreeTS3F36_ERR_INFO() 関数を使って開放してください。

3.23.3.19 DshInitTS3F36_ERR_INFO レチクル搬送ジョブ要求尾応答情報構造体の初期設定

(1) 呼出書式

[C, C++]

```
API void APIX DshInitTS3F36_ERR_Info(
    TS14209_ERR_INFO *erinfo,      // TS3F36_ERR_INFO 構造体のポインタ
    int rpmack,                    // S3F35 に対する ACK
    int errcount                   // エラー情報数
);
```

[.NET VB]

```
Sub DshInitTS3F36_ERR_INFO (
    ByRef info As dsh_info.TS3F36_ERR_INFO,
    ByVal ack As Int32,
    ByVal err_count As Int32)
```

[.NET C#]

```
void DshInitTS3F36_ERR_INFO(
    ref TS3F36_ERR_INFO info,
    int ack,
    int err_count );
```

(2) 引数

info

TS3F36_ERR_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。この中に S3F35 生成情報を詰めます。

rpmack

搬送ジョブ要求を受け付けたかどうかを示す ACK です。(Binary)

errcount

付属されるエラー情報の数です。

(3) 戻り値

なし。

(4) 説明

S3F36 メッセージをエンコードするときに使用する情報を TS3F36_ERR_INFO 構造体内に生成するために使用します。

本関数は、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。メモリが必要な構造体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。

属性情報の設定には DshPutTS3F36_SP_INFO()関数を使用してください。

エラー情報の設定には DshPutTS3F36_ERR_INFO ()関数を使用してください。

TS3F36_ERR_INFO 構造体の使用後は DshFreeTS3F36_ERR_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。

3.23.3.20 DshAddTS3F36_ERR_INFO() エラー情報の設定

(1) 呼出書式

[C, C++]

```
API int APIX DshAddTS3F36_ERR_INFO (
    TS3F36_ERR_INFO *erinfo,      // TS3F36_ERR_INFO 情報格納構造体のポインタ
    int errcode,                  // 設定するエラーコード
    char *errtext                 // 設定するエラーテキストのポインタ
);
```

[.NET VB]

```
Function DshAddTS3F36_ERR_INFO (
    ByRef info As dsh_info.TS3F36_ERR_INFO,
    ByVal err_code As Int32,
    ByVal err_text As String) As Int32
```

[.NET C#]

```
int DshAddTS3F36_ERR_INFO(
    ref TS3F36_ERR_INFO info,
    int err_code,
    byte[] err_text );
```

(2) 引数

erinfo

TS3F36_ERR_INFO 情報構造体のポインタです。

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

erinfo で指定された TS3F36_ERR_INFO 構造体内の err_list 上に新たにエラー情報 errcode と errtext を設定します。

正常に設定できた場合は 0 を返却します。

erinfo 内の err_list 上のエラー数が err_count の数に達していた場合は (-1)を返却します。

本関数を最初に実行する前に、DshInitTS3F36_ERR_INFO()関数を使ってリストの初期設定を行ってください。

3.23.4 ユーザ作成ライブラリ関数

3.23.4.1 DshResponseS3F36() S3F36 レチクル搬送ジョブ要求応答関数

(1) 呼出書式

[C, C++]

```
API int APIX DshResponseS3F36(
    int eqid, // 通信対象装置 ID(0,1,2,...)
    ID_TR trid, // DSHDR2 のトランザクション ID
    TS3F35_INFO *info, // レチクル搬送ジョブ要求情報格納領域のポインタ
    TS3F36_ERR_INFO *erinfo // S3F36 応答情報格納用構造体のポインタ
);
```

[.NET VB]

```
Function DshResponseS3F36 (
    ByVal eqid As Int32,
    ByVal trid As Int32,
    ByRef erinfo As dsh_info.TS3F36_ERR_INFO) As Int32
```

[.NET C#]

```
int DshResponseS3F36(
    int eqid,
    uint trid,
    ref TS3F36_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

S3F35 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

レチクル搬送ジョブ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 S3F36 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

レチクル搬送ジョブ要求メッセージ S3F35 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 TS3F36_ERR_INFO 構造体に含まれている情報から S3F36 メッセージを組み立て、その後、S3F36 メッセージを送信します。

送信が完了したら、TS3F36_ERR_INFO の構造体に使用されたメモリを DshFreeTS3F36_ERR_INFO()関数を使って開放します。

なお、S3F36 メッセージの組み立てに、DshMakeS3F35Response()関数を使用できます。