

DSHEng4 装置通信エンジン (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- 4 / 15

- 3 . 6 CE 収集イベント情報アクセス、関連メッセージ送信関数
- 3 . 7 Report レポート情報アクセス、関連メッセージ送信関数
- 3 . 8 アラーム情報アクセス、関連メッセージ送信関数

2009年7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	初版	

## 目 次

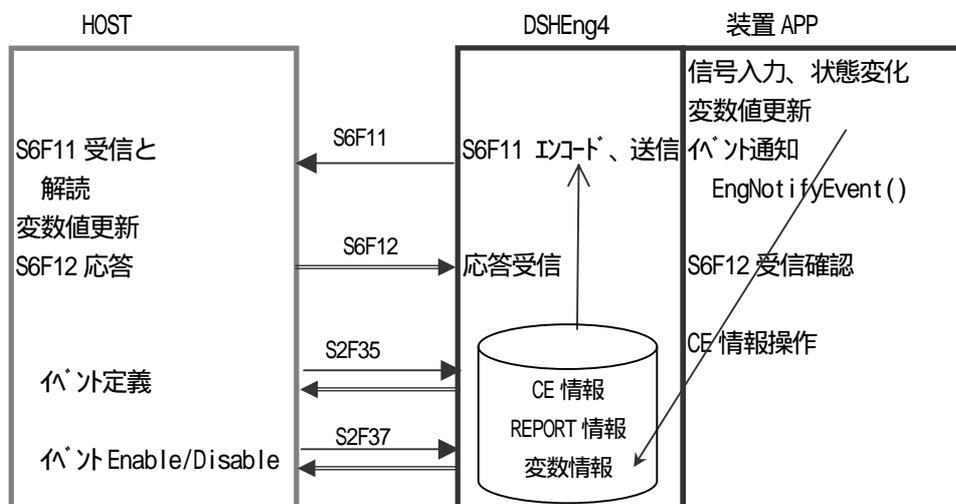
3.6	CE 収集イベント情報アクセスと通知関数.....	3
3.6.1	使用する構造体.....	5
3.6.2	CE 収集イベント情報アクセスと通知関数.....	6
3.6.2.1	EngGetCeName() – イベント名取得関数.....	6
3.6.2.2	EngGetCeEnabled() – イベント有効状態取得関数.....	7
3.6.2.3	EngSetCeEnabled() – イベント有効状態設定関数.....	8
3.6.2.4	EngGetCeRpCount() – イベントにリンクされているレポート数取得関数.....	9
3.6.2.5	EngGetCeRpid() – イベントにリンクされている指定順位のレポートID取得関数.....	10
3.6.2.6	EngGetCeAllRpid – イベントにリンクされている全レポートID取得関数.....	11
3.6.2.7	EngGetCeRpName() – イベントにリンクされている指定順位のレポート名取得関数.....	12
3.6.2.8	EngSetCeRpLink () – イベントのレポートリンクの設定関数.....	13
3.6.2.9	EngGetReservedCeid – 予約収集イベントのCEID取得関数.....	14
3.6.2.10	EngGetCeList() – 全登録CEID取得関数.....	16
3.6.2.11	EngNotifyEvent() – 収集イベント通知要求.....	17
3.6.2.12	EngAnNotifyEvent() – 注釈付き収集イベント通知要求.....	19
3.6.3	CE 収集イベント関連ライブラリ関数.....	20
3.6.3.1	DshDecodeS2F35() – S2F35 をイベント定義情報ヘデコード.....	20
3.6.3.2	DshDecodeS2F37() – S2F37 のイベント有効/無効情報のデコード.....	22
3.7	Report レポート情報アクセス関数.....	24
3.7.1	使用する構造体.....	24
3.7.2	レポート情報アクセスと通知関数.....	25
3.7.2.1	EngGetRpName() – レポート名取得関数.....	25
3.7.2.2	EngGetRpVCount() – レポートにリンクされている変数の数の取得関数.....	26
3.7.2.3	EngGetRpVid() – レポートにリンクされている指定順位の変数ID取得関数.....	27
3.7.2.4	EngGetRpAllVid - RP最小(Min)値取得関数.....	28
3.7.2.5	EngGetRpVName() – レポートにリンクされている指定順位の変数名取得関数.....	29
3.7.2.6	Eng SetRpVLink () – レポートの変数リンクの設定関数.....	30
3.7.2.7	EngGetRpList() – 全登録レポートID取得関数.....	31
3.7.3	レポート関連ライブラリ関数.....	32
3.7.3.1	DshDecodeS2F33() – S2F33 をレポート定義情報ヘデコード.....	32
3.8	Alarm アラーム情報アクセスと通知関数.....	34
3.8.1	使用する構造体.....	35
3.8.2	Alarm アラーム情報アクセス関数.....	36
3.8.2.1	EngGetAlName() – アラーム名取得関数.....	36
3.8.2.2	EngGetAlEnabled() – アラーム有効状態取得関数.....	37
3.8.2.3	EngSetAlEnabled() – アラーム有効状態設定関数.....	38
3.8.2.4	EngGetAlcd() – アラームのALCD取得関数.....	39
3.8.2.5	EngGetAltx() – アラームのALTX取得関数.....	40
3.8.2.6	EngGetAlCeOn() – アラーム発生時のリンクイベントID取得関数.....	41
3.8.2.7	EngGetAlCeOff() – アラーム復旧時のリンクイベントID取得関数.....	42
3.8.2.8	EngGetAlList() – 全登録アラームID取得関数.....	43
3.8.2.9	EngNotifyAlarm() – アラーム通知要求.....	44

(VOL - 5 に続く)



### 3.6 CE 収集イベント情報アクセスと通知関数

CE 収集イベント情報は、DSHEng4 が管理します。従って、APP はこれらの情報をアクセスするために以下の DSHEng4 API 関数を使用します。



#### (1) 情報アクセスと送信 API 関数

収集イベント情報のアクセスと装置へのメッセージ送信に関連するサービスのための API 関数名は次の一覧表のとおりです。

	API 関数名	機能
1	EngGetCeName()	指定された CEID の収集イベント名を取得します。
2	EngGetCeEnabled()	指定された CEID の Enable 状態を取得します。
3	EngSetCeEnabled()	指定された CEID の Enable 状態を設定します。
4	EngGetCeRpCount()	指定された CEID にリンクされているポート ID の個数を取得します。
5	EngGetCeRpid()	指定された CEID にリンクされているポート ID を取得します。
6	EngGetCeAllRpid()	指定された CEID にリンクされている全ポート ID を取得します。
7	EngSetReservedCeid()	指定されたインデックスの予約 CEID を設定します。
8	EngGetReservedCeid()	指定されたインデックスの予約 CEID を取得します。
9	EngGetCeList()	全登録 CEID の ID リストを取得する。
10	EngNotifyEvent()	S6F11 収集イベント通知を送信します。
11	EngAnNotifyEvent()	S6F13 注釈付き収集イベント通知を送信します。

## (2) イベント関連ライブラリ関数

APP が使用できるイベント情報処理関連ライブラリ関数として、以下の関数があります。

	API 関数名	機能
1	DshDecodeS2F35()	S2F35 メッセージの内容を TCE_LIST 構造体内にデコードします。
2	DshDecodeS2F37()	S2F37 メッセージ内の CEID と CEED(有効/無効)をデコードします。

S2F35 (Link Event Report), S2F37 ( Enable/Disable Event Report)メッセージの受信に対して、DSHEng4 は自動的に DSHEng4 内で処理をし、処理結果に基づき応答メッセージを返します。  
従って、APP はこれらの関数を使って処理することはありません。

S2F35 については、含まれているリンク情報に従って、DSHEng4 が管理している CEID に対するレポート ID の再リンクを行います。

S2F37 については、同様に、含まれている CEID, Enable/Disable 情報に従って、イベントの有効・無効の設定を行います。(無効の場合には、APP からその CEID のイベント通知要求があってもイベント通知メッセージを送信しません。)

### 3.6.1 使用する構造体

(1) S6F11, S6F13 メッセージのデコード結果を格納する構造体です。

```
typedef struct{
    TCEID      ceid;          // 収集イベント ID
    int        rp_count;     // リンクされているレポート数
    TRP_CONTENT **rp_list;   // リンクレポート情報のリスト
    int        next_rp;     // (関数内部使用)
    int        next_v;     // (関数内部使用)
} TCE_CONTENT;

typedef struct{
    TRPID      rpid;        // リンクされているレポート ID
    int        v_count;     // レポートにリンクされている変数の数
    TV_CONTENT **v_list;    // リンク変数情報のリスト
} TRP_CONTENT;

typedef struct{
    TVID       vid;        // リンクされている変数 ID
    int        format;     // 変数のフォーマット
    int        asize;      // 変数データの配列サイズ
    void       *dptr;      // 変数データのポインタ (fmt L ならば nesting ->TV_CONTENT)
} TV_CONTENT;
```

(2) S2F35 イベントレポート設定、S2F33 レポート設定メッセージ処理に使用する構造体です。

```
typedef struct{
    int        count;       // 本構造体を含む CE の数
    TCE_LINK  **ce_list;    // CE リンク情報構造体のリストポインタ
} TCE_LIST;

typedef struct{
    TCEID      ceid;       // CEID
    int        count;     // リンクされているレポートの数
    TRPID      *rpid_list; // リンクされているレポート ID のリスト
} TCE_LINK;

typedef struct{
    int        count;       // 本構造体に含まれるレポートの数
    TRP_LINK  **rp_list;   // レポートリンク構造体のリストポインタ
} TRP_LIST;

typedef struct{
    TRPID      rpid;       // RPID
    int        count;     // リンクされている変数の数
    TVID       *vid_list;  // リンクされている VID のリスト
} TRP_LINK;
```

## 3.6.2 CE 収集イベント情報アクセスと通知関数

### 3.6.2.1 EngGetCeName() イベント名取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetCeName(
    TCEID ceid,                // CEID
    char *name,                // 取得した名前の格納領域ポインタ
    int *bytesize             // 名前の文字列長格納用
);
```

##### [.NET VB]

```
Function EngGetCeName (
    ByVal ceid As Int32,
    ByVal name As String) As Int32
```

##### [.NET C#]

```
int EngGetCeName(
    uint ceid,
    byte[] name );
```

#### (2) 引数

ceid

イベント ID です。

装置管理情報定義ファイルに登録されているイベントの ID でなければなりません。

name

取得した名前 (文字列) を格納する領域のポインタです。名前格納に十分な領域を準備してください。

bytesize

取得された名前の文字列長です。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	ceid が正しくなかった。

#### (4) 説明

ceid で指定された CE (イベント) の名前を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得された名前のバイト長は bytesize で指定された領域に返却されます。

### 3.6.2.2 EngGetCeEnabled() イベント有効状態取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetCenabled(
    TCEID ceid // CEID
);
```

##### [.NET VB]

```
Function EngGetCeEnabled (
    ByVal ceid As Int32) As Int32
```

##### [.NET C#]

```
int EngGetCeEnabled(
    uint ceid );
```

#### (2) 引数

ceid

イベント ID です。

装置管理情報定義ファイルに登録されているイベントの ID でなければなりません。

#### (3) 戻り値

戻り値	意味
0	有効状態 0=無効、1=有効
(-1)	ceid の値が正しくなかった。

#### (4) 説明

ceid で指定された CE(イベント)の有効状態を取得します。

戻り値が =1 の場合、有効状態です。EngNotifyEvent()によってイベント通知できる状態です。

戻り値が =0 の場合、無効状態です。EngNotifyEvent()によってイベント通知できない状態です。

装置に対し CE の送信有効状態の通知 (送信) は EngSendS2F37()関数を使って行うことができます。

### 3.6.2.3 EngSetCeEnabled() イベント有効状態設定関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngStCenabled(
    TCEID ceid,           // CEID
    int on_off           // 有効 On/Off
);
```

##### [.NET VB]

```
Function EngSetCeEnabled (
    ByVal ceid As Int32,
    ByVal on_off As Int32) As Int32
```

##### [.NET C#]

```
int EngSetCeEnabled(
    uint ceid,
    int on_off );
```

#### (2) 引数

ceid

イベント ID です。

装置管理情報定義ファイルに登録されているイベントの ID でなければなりません。

on\_off

有効 / 無効の設定したい状態を指定します。

=1 の場合、有効状態にします。EngNotifyEvent()によってイベント通知できる状態です。

=0 の場合、無効状態にします。EngNotifyEvent()によってイベント通知できない状態です。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	ceid の値が正しくなかった。

#### (4) 説明

ceid で指定された CE (イベント) の有効状態を on\_off 値で設定します。

### 3.6.2.4 EngGetCeRpCount() イベントにリンクされているレポート数取得関数

(1) 呼出書式

**[C, C++]**

```
API int APIX EngGetCeRpCount(
    TCEID ceid // CEID
);
```

**[.NET VB]**

```
Function EngGetCeRpCount (
    ByVal ceid As Int32) As Int32
```

**[.NET C#]**

```
int EngGetCeRpCount(
    uint ceid );
```

(2) 引数

ceid

イベント ID です。

装置管理情報定義ファイルに登録されているイベントの ID でなければなりません。

(3) 戻り値

戻り値	意味
>= 0	リンクされているレポート数。
(-1)	ceid の値が正しくなかった。

(4) 説明

ceid で指定された CE (イベント) にリンクされているレポート ID の数を取得します。

=0 はリンクされているイベントがないことを意味します。

### 3.6.2.5 EngGetCeRpid() イベントにリンクされている指定順位のレポート ID 取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetCeRpid(
    TCEID ceid,           // CEID
    int order,           // 順位(0,1,2.. )
    TRPID *rpid          // 取得したレポート ID 格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetCeRpid (
    ByVal ceid As Int32,
    ByVal order As Int32,
    ByRef rpid As Int32) As Int32
```

##### [.NET C#]

```
int EngGetCeRpid(
    uint ceid,
    int order,
    ref uint rpid );
```

#### (2) 引数

ceid

イベント ID です。  
装置管理情報定義ファイルに登録されているイベントの ID でなければなりません。

order

何番目のレポート ID を取得するかを指定します。先頭の指定は=0 です。

rpid

取得したレポート ID を格納する領域のポインタです。

#### (3) 戻り値

戻り値	意味
= 0	正常に取得できた。
(-1)	ceid の値が正しくなかった。または order 番目のレポートは無かった。

#### (4) 説明

ceid で指定された CE (イベント) にリンクされている order 番目のレポート ID を取得します。

### 3.6.2.6 EngGetCeAllRpid イベントにリンクされている全レポート ID 取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetCeMin(
    TCEID ceid, // CEID
    TRPID *rpid // リンクされている rpid を格納する領域のポインタ
);
```

##### [.NET VB]

```
Function EngGetCeAllRpid (
    ByVal ceid As Int32,
    ByRef rpid As Int32) As Int32
```

##### [.NET C#]

```
int EngGetCeAllRpid(
    uint ceid,
    ref uint rpid );
```

#### (2) 引数

ceid

イベント ID です。  
装置管理情報定義ファイルに登録されているイベントの ID でなければなりません。

rpid

リンクされている全レポート ID を格納するための領域ポインタです。  
リンクされている可能性のある最大数文の領域を準備してください。

#### (3) 戻り値

戻り値	意味
>=0	取得できたレポート ID 数
(-1)	ceid の値が正しくなかった。

#### (4) 説明

ceid で指定された CE (イベント) にリンクされている全レポート ID を取得します。

戻り値 = 0 の場合、リンクされているレポートがないことを意味します。

戻り値 > 0 の場合、戻り値がリンクされているレポート数で、それらの ID が rpid の領域に設定されます。

### 3.6.2.7 EngGetCeRpName() イベントにリンクされている指定順位のレポート名取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetCeRpName(
    TCEID ceid,           // CEID
    int order,           // 順位(0,1,2.. )
    char *rpname         // 取得したレポート名格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetCeRpName (
    ByVal ceid As Int32,
    ByVal order As Int32,
    ByVal rpname As String) As Int32
```

##### [.NET C#]

```
int EngGetCeRpName(
    uint ceid,
    int order,
    byte[] rpname );
```

#### (2) 引数

ceid

イベント ID です。

装置管理情報定義ファイルに登録されているイベントの ID でなければなりません。

order

何番目のレポートの名前を取得するかを指定します。先頭の指定は=0 です。

rpname

取得したレポート名を格納する領域のポインタです。

#### (3) 戻り値

戻り値	意味
= 0	正常に取得できた。
(-1)	ceid の値が正しくなかった。または order 番目のレポートは無かった。

#### (4) 説明

ceid で指定された CE (イベント) にリンクされている order 番目のレポートの名前を取得します。

### 3.6.2.8 EngSetCeRpLink () イベントのレポートリンクの設定関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX Eng SetCeRpLink (
    TCE_LIST *list           // リンク作成したい情報が格納されているリスト
);
```

##### [.NET VB]

```
Function EngSetCeRpLink (
    ByRef list As dsh_info.TCE_LIST) As Int32
```

##### [.NET C#]

```
int EngSetCeRpLink(
    ref TCE_LIST list );
```

#### (2) 引数

list

リンク対象のCEID とリンクしたいレポート ID が含まれているリストのポインタです。

#### (3) 戻り値

戻り値	意味
= 0	正常に取得できた。
(-1)	ceidの値が正しくなかった。またはレポートIDは無かった。

#### (4) 説明

list に含まれる 1 個以上の CE リンク情報に従って CE に対するレポートのリンクを設定しなおします。

list には、リストに含まれる TCE\_LINK 構造体の数とポインタリストが含まれます。

TCE\_LINK 構造体には、1 個の CEID と、それにリンクされるレポートの数とレポート ID のリストが含まれます。

### 3.6.2.9 EngGetReservedCeid 予約収集イベントのCEID 取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetReservedCeid(
    int index,           // 予約CEIDのindex
    TCEID *ceid         // 取得したCEID格納用
);
```

##### [.NET VB]

```
Function EngGetReservedCeid (
    ByVal index As Int32,
    ByRef ceid As Int32) As Int32
```

##### [.NET C#]

```
int EngGetReservedCeid(
    int index,
    ref uint ceid );
```

#### (2) 引数

index

予約されている収集イベントに割り当てられたインデクスです。  
インデクスと予約CEの対応はシステムで規定されています。(C言語ではマクロ定義されています。)  
インデクス値の範囲は、0~31 でなければなりません。

ceid

予約されたインデクスで指定された収集イベントのCEIDを格納するための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)	インデクス値が正しくなかったまたはCEIDが設定されていなかった。

#### (4) 説明

あるイベント、例えば、装置との通信確立状態になったタイミングで通知するイベントなど、DSHEng4内で自動的に送信しなければならない収集イベントに予約されたCEIDを取得するための関数です。

indexで指定された収集イベントに対し予約されているCEIDを、ceidで指定された領域に格納し返却します。

ユーザは、このインデクスを使って、CEIDを取得し、イベント通知するために使用することができます。

indexはデフォルトで次のように予約されています。

インデクス値	C言語マクロ	適用収集イベント
0	CEX_RSV_COMMUNICATING	通信確立時に通知するイベント
1	CEX_RSV_SPOOL_END	スプール送信終了時に送信するイベント
2		
3		

~		
31		

### 3.6.2.10 EngGetCeList() 全登録 CEID 取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetCeList(
    TBIN_DLIST **list           // 取得リスト格納ポインタの格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetCeList (
    ByRef list As IntPtr) As Int32
```

##### [.NET C#]

```
int EngGetCeList(
    IntPtr list );
```

#### (2) 引数

list

取得できた CEID が格納されている TBIN\_DLIST 構造体のポインタを格納する領域のポインタです。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	取得できなかった。

#### (4) 説明

システムに登録されている全 CEID とその名前を TBIN\_DLIST 構造体に取り出すための関数です。

取出す名前は、装置管理情報定義ファイルで CE 定義時に与えられた名前です。

取得した情報の処理が終了した後、DshFreeTBIN\_DLIST()関数で list 内部の情報格納用に使用されているメモリを開放してください。

TBIN\_DLIST 構造体は次のとおりです。

```
typedef struct{
    int          count;           // 取得できた ID 数
    ULONG       *id_list;        // 取得できた ID 格納用配列
    char        **name_list;     // 取得できた名前格納ポインタ配列
}TBIN_DLIST;
```

### 3.6.2.11 EngNotifyEvent() 収集イベント通知要求

#### (1) 呼出書式

##### [C, C++]

```
API int WINAPI EngNotifyEvent(
    TCEID      ceid,           // 通知 ceid
    int (WINAPI *EventCallback)(), // 実行終了時の callback 関数
    ULONG      upara          // callback 時のパラメータ
);
```

##### [.NET VB]

```
Function EngNotifyEvent (
    ByVal ceid As Int32,
    ByVal callback As vcallback.callback_notifyEvent,
    ByVal upara As Int32) As Int32
```

##### [.NET C#]

```
int EngNotifyEvent(
    uint ceid,
    CallbackNotifyEvent NEventCallback,
    uint upara );
```

#### (2) 引数

ceid

ホストに通知するための CEID(収集イベント ID)を指定します。  
S6F11 の ceid 値になります。

EventCallback

DSHEng4 によるイベント通知処理が終了したときに呼出される callback 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。

本パラメータとして = 0 が指定された場合は、コールバック関数が無く、ブロックモードとみなされ、関数で指定されたメッセージの通信トランザクションが終了するまで制御は戻ってきません。

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することができます。

#### (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, ackc6=0 であった。 (2) 非ブロックモード : 要求が受け付けられた。
9	キャンセルされた。
(-1)	要求が受け付けられなかったか送信エラーを検出した。
(-14)	T3タイムアウトを検出した。
その他 > 0	ブロックモードの場合, ackc6 の値が返却されます。

#### (4) 説明

ホストに収集イベントメッセージ (S6F11) を送信するための関数です。

DSHEng4 は指定された ceid のメッセージを自動的に組立てた上でそれをホストに送信します。

ceid にリンクされているレポート、そしてそのレポートにリンクされている変数データ情報を S6F11 メッセージに設定展開した上で送信します。

定義されていない ceid が指定された場合、list=0 の S6F11 が送信されます。

ブロックモードならびにスプール関連については 3.2.4 の EngSendPrimary() 関数の説明を参照願います。

#### (5) コールバック関数

##### [c,C++]

```
API int APIX EventCallback(
    int end_status,           // 実行結果
    TCEID ceid,              // 通知した CEID
    ULONG upara              // 呼出時に指定したパラメータ
);
```

##### [.NET VB]

```
Function callback_NotifyEvent(ByVal end_status As Integer, ByVal ceid As Integer, ByVal upara As Integer) As Integer
```

##### [.NET C#]

```
int CallbackNotifyEvent(int end_status, uint id, 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。
9	スプールされた。
(-1)	送信エラーを検出した。
(-14)	T3タイムアウトを検出した。
その他 > 0	ackc6 の値が返却されます。

### 3.6.2.12 EngAnNotifyEvent() 注釈付き収集イベント通知要求

#### (1) 呼出書式

##### [C, C++]

```
API int WINAPI EngAnNotifyEvent(  
    TCEID ceid, // 通知 ceid  
    int (WINAPI *EventCallback)(), // 実行終了時の callback 関数  
    ULONG upara // callback 時のパラメータ  
);
```

##### [.NET VB]

```
Function EngNotifyAnEvent (  
    ByVal ceid As Int32,  
    ByVal callback As vcallback.callback_NotifyAnEvent,  
    ByVal upara As Int32) As Int32
```

##### [.NET C#]

```
int EngNotifyAnEvent(  
    uint ceid,  
    CallbackNotifyEvent DshEndNotifyEvent,  
    uint upara );
```

#### (2) 引数

EngNotifyEvent()関数とまったく同じです。EngNotifyEvent()関数を参照ください。

#### (3) 戻り値

EngNotifyEvent()関数とまったく同じです。EngNotifyEvent()関数を参照ください。

#### (4) 説明

EngNotifyEvent()関数とまったく同じです。EngNotifyEvent()関数を参照ください。

#### (5) コールバック関数

EngNotifyEvent()関数とまったく同じです。EngNotifyEvent()関数を参照ください。

### 3.6.3 CE 収集イベント関連ライブラリ関数

#### 3.6.3.1 DshDecodeS2F35() S2F35 をイベント定義情報ヘデコード

(1) 呼出書式

[c, C++]

```
API int APIX DshDecodeS2F35(
    DSHMSG *smsg, // S2F35 が格納されている SECS メッセージ 情報構造体のポインタ
    TCE_LIST *list // デコードしたイベント ID 定義情報リスト格納構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS2F35 (
    ByRef smsg As dshdr2.DSHMSG,
    ByRef list As dsh_info.TCE_LIST) As Int32
```

[.NET C#]

```
int DshDecodeS2F35(
    ref DSHMSG smsg,
    ref TCE_LIST list );
```

(2) 引数

smsg

S2F35 メッセージが格納されているメッセージ情報構造体のポインタです。

list

デコードしたイベント ID リストの結果を格納するための構造体のポインタです。

(3) 戻り値

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

(4) 説明

S2F35 の SECS メッセージに含まれるイベント定義情報を list で指定される TCE\_LIST 構造体にデコードし格納します。

smsg S2F35

L,2

dataid

L,a

L,2

CEID

L,c

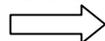
RPTID1

RPTID2

.

RPTIDc

decode



TCE\_LIST  
構造体

得られた list 情報の処理が終わった後は、DshFreeTCE\_LIST()関数で list 内部に割付使用されたメモリを開

放することができます。

### 3.6.3.2 DshDecodeS2F37() S2F37 のイベント有効/無効情報のデコード

#### (1) 呼出書式

##### [C, C++]

```
API int APIX DshDecodeS2F37(
    DSHMSG *msg, // S2F37 が格納されている SECS メッセージ 情報構造体のポインタ
    TCEID *list // デコードしたイベント ID 定義情報リスト格納用領域のポインタ
    int *count, // メッセージに含まれていたイベント ID 数の格納用
    int *ceed // 有効/無効(1/0)格納用
);
```

##### [.NET VB]

```
Function DshDecodeS2F37 (
    ByRef msg As dshdr2.DSHMSG,
    ByRef list As Int32,
    ByRef count As Int32,
    ByRef ceed As Int32) As Int32
```

##### [.NET C#]

```
int DshDecodeS2F37(
    ref DSHMSG msg,
    ref uint list,
    ref int count,
    ref int ceed );
```

#### (2) 引数

msg

デコードした S2F37 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。

list

デコードしたイベント ID リストの結果を格納するための構造体のポインタです。

count

S2F37 メッセージに含まれるイベント ID の数を格納します。

ceed

S2F37 メッセージで指定される CEED を格納します。

#### (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。

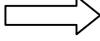
#### (4) 説明

S2F37 の SECS メッセージをデコードし、含まれるイベント ID を list で指定される領域に、イベント ID の数を count に、そして CEED (有効/無効) を ceed に格納します。

list 領域のメモリは S2F37 に含まれる可能性のある最大限の数分を準備してください。

smg S2F37

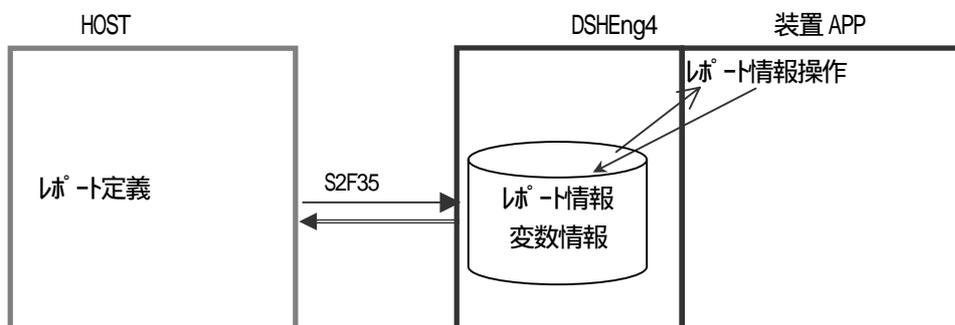
L,2  
ceed  
L,n  
CEID1  
CEID2  
.  
CEIDn

decode  




### 3.7 Report レポート情報アクセス関数

レポート情報は収集イベントにリンクされる情報であり、DSHEng4 が管理します。従って、APP はこれらの情報をアクセスするために以下の DSHEng4 API 関数を使用します。



#### (1) 情報アクセスと送信 API 関数

レポート情報のアクセスと装置へのメッセージ送信に関連するサービスのための API 関数名は次の一覧表のとおりです。

	API 関数名	機能
1	EngGetRpName()	指定された REPORT ID のレポート名を取得します。
2	EngGetRpVCount()	指定された REPORT ID の Enable 状態を取得します。
3	EngGetRpVid()	指定された REPORT ID にリンクされている変数 ID を取得します。
4	EngGetRpAllVid()	指定された REPORT ID にリンクされている全変数 ID を取得します。
5	EngGetRpList()	システムに登録されているレポート ID 名の一覧表を取得します。

#### (2) レポート関連ライブラリ関数

APP が使用できるレポート情報処理関連ライブラリ関数として、以下の関数があります。

	API 関数名	機能
1	DshDecodeS2F33()	S2F33 メッセージをデコードしレポート ID リストに設定します。

#### 3.7.1 使用する構造体

先に説明した 3.4.1 CE 収集イベント情報アクセス関数が使用する構造体の項を参照ください。

## 3.7.2 レポート情報アクセスと通知関数

### 3.7.2.1 EngGetRpName() レポート名取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetRpName(
    TRPID rpid,           // RPID
    char *name           // 取得した名前の格納領域ポインタ
);
```

##### [.NET VB]

```
Function EngGetRpName (
    ByVal rpid As Int32,
    ByVal name As String) As Int32
```

##### [.NET C#]

```
int EngGetRpName(
    uint rpid,
    byte[] name );
```

#### (2) 引数

rpId

レポート ID です。

装置管理情報定義ファイルに登録されているレポートの ID でなければなりません。

name

取得した名前 (文字列) を格納する領域のポインタです。名前格納に十分な領域を準備してください。

bytesize

取得された名前の文字列長です。

#### (3) 戻り値

戻り値	意味
>= 0	正常に取得できた。
(-1)	rpId が正しくなかった。

#### (4) 説明

rpId で指定された RP (レポート) の名前を取得します。

正常に取得できた場合は、関数の戻り値は 取得できた名前のバイト長 になります。

rpId が未定義の場合は (-1) が戻ります。

### 3.7.2.2 EngGetRpVCount() レポートにリンクされている変数の数の取得関数

(1) 呼出書式

**[C, C++]**

```
API int APIX EngGetRpVCount(
    TRPID rpid // RPID
);
```

**[.NET VB]**

```
Function EngGetRpVCount (
    ByVal rpid As Int32) As Int32
```

**[.NET C#]**

```
int EngGetRpVCount(
    uint rpid );
```

(2) 引数

rpId

レポート ID です。

装置管理情報定義ファイルに登録されているレポートの ID でなければなりません。

(3) 戻り値

戻り値	意味
>= 0	リンクされているデータ変数の数。
(-1)	rpIdの値が正しくなかった。

(4) 説明

rpIdで指定されたRP(レポート)にリンクされている変数 ID の数を取得します。

=0 はリンクされている変数がないことを意味します。

### 3.7.2.3 EngGetRpVid() レポートにリンクされている指定順位の変数 ID 取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetRpVid(
    TRPID rpid,           // RPID
    int order,           // 順位(0,1,2.. )
    TRPID *vid           // 取得した変数 ID 格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetRpVid (
    ByVal rpid As Int32,
    ByVal order As Int32,
    ByRef vid As Int32) As Int32
```

##### [.NET C#]

```
int EngGetRpVid(
    uint rpid,
    int order,
    ref uint vid );
```

#### (2) 引数

rpId

レポート ID です。

装置管理情報定義ファイルに登録されているレポートの ID でなければなりません。

order

何番目のレポート ID を取得するかを指定します。先頭の指定は=0 です。

vid

取得した変数 ID(VID)を格納する領域のポインタです。

#### (3) 戻り値

戻り値	意味
= 0	正常に取得できた。
(-1)	rpId の値が正しくなかった。または order 番目のレポートは無かった。

#### (4) 説明

rpId で指定された RP(レポート)にリンクされている order 番目の変数 ID を取得します。

### 3.7.2.4 EngGetRpAllVid - RP 最小(Min)値取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetRpAllVid(
    TRPID rpid,                // RPID
    TRPID *vid                 // リンクされている変数 vid を格納する領域のポインタ
);
```

##### [.NET VB]

```
Function EngGetRpAllVid (
    ByVal rpid As Int32,
    ByRef vid As Int32) As Int32
```

##### [.NET C#]

```
int EngGetRpAllVid(
    uint rpid,
    ref uint vid );
```

#### (2) 引数

rpId

レポート ID です。

装置管理情報定義ファイルに登録されているレポートの ID でなければなりません。

vid

リンクされている全変数 ID を格納するための領域ポインタです。

リンクされている可能性のある最大数分の領域を準備してください。

#### (3) 戻り値

戻り値	意味
>=0	取得できた変数 ID 数
(-1)	rpId の値が正しくなかった。

#### (4) 説明

rpId で指定された RP(レポート)にリンクされている全変数 ID を取得します。

戻り値 = 0 の場合、リンクされている変数がないことを意味します。

戻り値 > 0 の場合、戻り値がリンクされている変数の数で、それらの ID が vid の領域に設定されます。

### 3.7.2.5 EngGetRpVName() レポートにリンクされている指定順位の変数名取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetRpVName(
    TRPID rpid,           // RPID
    int order,           // 順位(0,1,2.. )
    char *vname          // 取得した変数名格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetRpVName (
    ByVal rpid As Int32,
    ByVal order As Int32,
    ByVal vname As String) As Int32
```

##### [.NET C#]

```
int EngGetRpVName(
    uint rpid,
    int order,
    byte[] vname );
```

#### (2) 引数

rpId

レポート ID です。

装置管理情報定義ファイルに登録されているレポートの ID でなければなりません。

order

何番目の変数の名前を取得するかを指定します。先頭の指定は=0 です。

vname

取得した変数名を格納する領域のポインタです。

#### (3) 戻り値

戻り値	意味
= 0	正常に取得できた。
(-1)	rpId の値が正しくなかった。または order 番目の変数は無かった。

#### (4) 説明

rpId で指定された RP(レポート)にリンクされている order 番目の変数の名前を取得します。

### 3.7.2.6 Eng SetRpVLink () レポートの変数リンクの設定関数

(1) 呼出書式

**[C, C++]**

```
API int APIX EngSetRpVLink (
    TRP_LIST *list // リンク作成したい情報が格納されているリスト
);
```

**[.NET VB]**

```
Function EngSetRpVLink (
    ByRef list As dsh_info.TRP_LIST) As Int32
```

**[.NET C#]**

```
int EngSetRpVLink(
    ref TRP_LIST list );
```

(2) 引数

list

リンク対象のRPIDとリンクしたい変数IDが含まれているリストのポインタです。

(3) 戻り値

戻り値	意味
= 0	正常に取得できた。
(-1)	rpидの値が正しくなかった。または変数IDは無かった。

(4) 説明

listに含まれる1個以上のRPリンク情報に従ってRPに対する変数のリンクを設定しなおします。

listには、リストに含まれるTRP\_LINK構造体の数とポインタリストが含まれます。

TRP\_LINK構造体には、1個のRPIDと、それにリンクされる変数の数と変数IDのリストが含まれます。

### 3.7.2.7 EngGetRpList() 全登録レポート ID 取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetRpList(
    TBIN_DLIST **list           // 取得リスト格納ポインタの格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetRpList (
    ByRef list As IntPtr) As Int32
```

##### [.NET C#]

```
int EngGetRpList(
    IntPtr list );
```

#### (2) 引数

list

取得できた RPID が格納されている TBIN\_DLIST 構造体のポインタを格納する領域のポインタです。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	取得できなかった。

#### (4) 説明

システムに登録されている全レポート ID とその名前を TBIN\_DLIST 構造体に取り出すための関数です。

取出す名前は、装置管理情報定義ファイルでレポート ID 定義時に与えられた名前です。

取得した情報の処理が終了した後、DshFreeTBIN\_DLIST()関数で list 内部の情報格納用に使用されているメモリを開放してください。

TBIN\_DLIST 構造体は次のとおりです。

```
typedef struct{
    int          count;           // 取得できた ID 数
    ULONG       *id_list;       // 取得できた ID 格納用配列
    char        **name_list;    // 取得できた名前格納ポインタ配列
}TBIN_DLIST;
```

### 3.7.3 レポート関連ライブラリ関数

#### 3.7.3.1 DshDecodeS2F33() S2F33 をレポート定義情報ヘデコード

(1) 呼出書式

[C, C++]

```
API int APIX DshDecodeS2F33(
    DSHMSG *msg, // S2F33 が格納されている SECS メッセージ 情報構造体のポインタ
    TRP_LIST *list // デコードしたレポート ID 定義情報リスト格納構造体のポインタ
);
```

[.NET VB]

```
Function DshDecodeS2F33 (
    ByRef msg As dshdr2.DSHMSG,
    ByRef list As dsh_info.TRP_LIST) As Int32
```

[.NET C#]

```
int DshDecodeS2F33(
    ref DSHMSG msg,
    ref TRP_LIST list );
```

(2) 引数

msg

S2F33 が格納されているメッセージ情報構造体のポインタです。

list

デコードしたレポート ID リストの結果を格納するための構造体のポインタです。

(3) 戻り値

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

(4) 説明

S2F33 の SECS メッセージに含まれるレポート定義情報を list で指定される TRP\_LIST 構造体にデコードし格納します。

msg S2F33

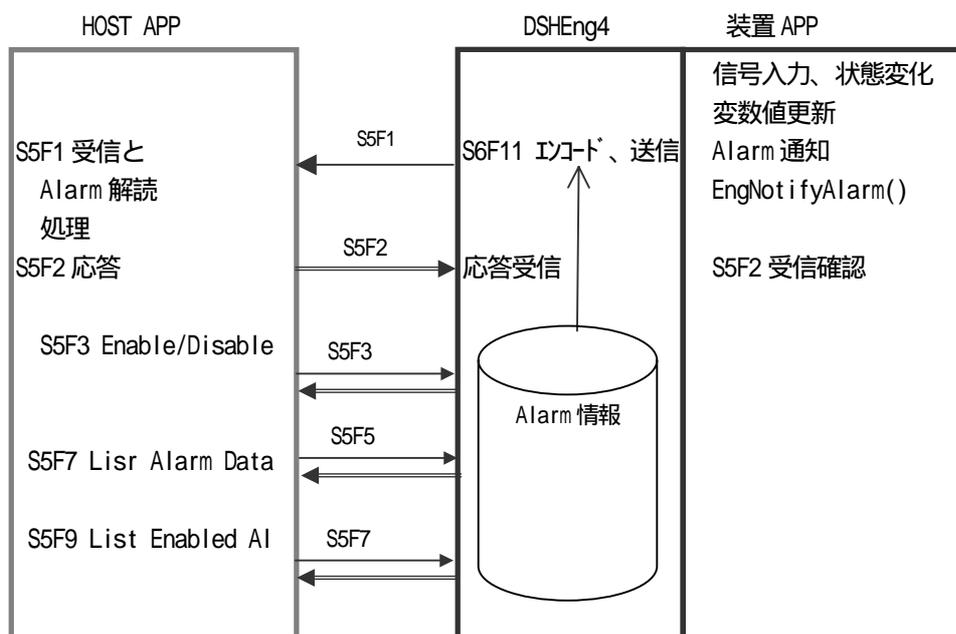
```
L,2
dataid
L,a
L,2
RPTID
L,c
VID1
VID2
.
VIDc
```

decode  
→



得られた list 情報の処理が終わった後は、DshFreeTRP\_LIST()関数で list 内部に割付使用されたメモリを開放することができます。

### 3.8 Alarm アラーム情報アクセスと通知関数



#### (1) 情報アクセスと送信 API 関数

アラーム情報のアクセスと装置へのメッセージ送信に関連するサービスのための API 関数名は一覧表のとおりです。

	API 関数名	機能
1	EngGetAlName()	指定された ALID の収集バット名を取得します。
2	EngGetAlEnabled()	指定された ALID の Enable 状態を取得します。
3	EngSetAlEnabled()	指定された ALID の Enable 状態を設定します。
4	EngGetAlcd()	指定された ALID に設定されている ALCD を取得します。
5	EngGetAltx()	指定された ALID に設定されている ALTX を取得します。
6	EngGetAlCeOn()	指定された ALID のアラーム発生時にリンクされている CEID を取得します。
7	EngGetAlCeOff()	指定された ALID のアラーム復旧時にリンクされている CEID を取得します。
8	EngNotifyAlarm()	指定された ALID のアラーム通知(S5F1)をします。

### 3.8.1 使用する構造体

(1) S5F1 アラーム報告メッセージのデコード結果を格納する構造体です。

```
typedef struct{
    int      on_off;          // 有効/無効(1/0)
    TALID    alid;           // ALID
    TALCD    alcd;           // ALCD
    char     *al tx;         // ALTX
} TAL_S5F1_INFO;
```

(2) S5F6 メッセージのデコード結果を格納する構造体です。

```
typedef struct{
    int      count;          // 含まれるアラーム ID 数
    TAL_S5F1_INFO **al_list; // アラーム情報のリスト
} TAL_S5F6_LIST;
```

## 3.8.2 Alarm アラーム情報アクセス関数

### 3.8.2.1 EngGetAlName() アラーム名取得関数

#### (1) 呼出書式

**A[C,C++]**

```
PI int APIX EngGetAlName(
    TALID alid,           // ALID
    char *name           // 取得した名前の格納領域ポインタ
);
```

**[.NET VB]**

```
Function EngGetAlName (
    ByVal alid As Int32,
    ByVal name As String) As Int32
```

**[.NET C#]**

```
int EngGetAlName(
    uint alid,
    byte[] name );
```

#### (2) 引数

alid

アラーム ID です。

装置管理情報定義ファイルに登録されているアラームの ID でなければなりません。

name

取得した名前（文字列）を格納する領域のポインタです。名前格納に十分な領域を準備してください。

#### (3) 戻り値

戻り値	意味
>= 0	正常に取得できた。
(-1)	alid が正しくなかった。

#### (4) 説明

alid で指定された AL(アラーム)の名前を取得します。

正常に取得できた場合は、関数の戻り値は 得られた名前のバイト長 になります。

alid が未定義の場合は(-1)が戻ります。

### 3.8.2.2 EngGetAIEnabled() アラーム有効状態取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetAIEnabled(
    TALID alid // ALID
);
```

##### [.NET VB]

```
Function EngGetAIEnabled (
    ByVal alid As Int32) As Int32
```

##### [.NET C#]

```
int EngGetAIEnabled(
    uint alid );
```

#### (2) 引数

alid

アラーム ID です。

装置管理情報定義ファイルに登録されているアラームの ID でなければなりません。

#### (3) 戻り値

戻り値	意味
0	有効状態 0=無効、1=有効
(-1)	alid の値が正しくなかった。

#### (4) 説明

alid で指定された AL(アラーム)の有効状態を取得します。

戻り値が =1 の場合、有効状態です。装置がアラーム通知できる状態です。

戻り値が =0 の場合、無効状態です。装置がアラーム通知できない状態です。

### 3.8.2.3 EngSetAlEnabled() アラーム有効状態設定関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngStAlnabled(
    TALID alid,           // ALID
    int on_off           // 有効 On/Off
);
```

##### [.NET VB]

```
Function EngSetAlEnabled (
    ByVal alid As Int32,
    ByVal on_off As Int32) As Int32
```

##### [.NET C#]

```
int EngSetAlEnabled(
    uint alid,
    int on_off );
```

#### (2) 引数

alid

アラーム ID です。

装置管理情報定義ファイルに登録されているアラームの ID でなければなりません。

on\_off

有効 / 無効の設定したい状態を指定します。

=1 の場合、有効状態にします。EngNotifyAlarm()によってアラーム通知できる状態です。

=0 の場合、無効状態にします。EngNotifyAlarm()によってアラーム通知できない状態です。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	alid の値が正しくなかった。

#### (4) 説明

alid で指定された AL (アラーム) の有効状態を on\_off 値で設定します。

### 3.8.2.4 EngGetAlcd() アラームのALCD取得関数

(1) 呼出書式

**[C, C++]**

```
API TALCD APIX EngGetAlcd(
    TALID alid // ALID
);
```

**[.NET VB]**

```
Function EngGetAlcd (
    ByVal alid As Int32) As Int32
```

**[.NET C#]**

```
byte EngGetAlcd(
    uint alid );
```

(2) 引数

alid

アラーム ID です。

装置管理情報定義ファイルに登録されているアラームの ID でなければなりません。

(3) 戻り値

戻り値	意味
>= 0	指定された alid が有する ALCD アラームコード
(-1)	alid の値が正しくなかった。

(4) 説明

alid で指定された AL(アラーム)のアラームコード(ALCD)を取得します。

### 3.8.2.5 EngGetAltx() アラームのALTX取得関数

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngGetAltx(
    TALID alid           // ALID
    char *altx
);
```

##### [.NET VB]

```
Function EngGetAltx (
    ByVal alid As Int32,
    ByVal altx As String) As Int32
```

##### [.NET C#]

```
int EngGetAltx(
    uint alid,
    byte[] altx );
```

#### (2) 引数

alid

アラーム ID です。  
装置管理情報定義ファイルに登録されているアラームの ID でなければなりません。

al tx

取得したアラームテキストを格納するバッファのポインタです。  
41 バイト以上のサイズのバッファが必要です。

#### (3) 戻り値

戻り値	意味
= 0	正常に取得できた。
(-1)	alid の値が正しくなかった。

#### (4) 説明

alid で指定された AL(アラーム)のアラームテキスト(ALTX)を al tx のバッファに取得します。

### 3.8.2.6 EngGetAlCeOn() アラーム発生時のリンクイベント ID 取得関数

#### (1) 呼出書式

##### [C, C++]

```
API TCEID APIX EngGetAlCeOn(
    TALID alid // ALID
);
```

##### [.NET VB]

```
Function EngGetAlCeOn (
    ByVal alid As Int32) As Int32
```

##### [.NET C#]

```
uint EngGetAlCeOn(
    uint alid );
```

#### (2) 引数

alid

アラーム ID です。

装置管理情報定義ファイルに登録されているアラームの ID でなければなりません。

#### (3) 戻り値

戻り値	意味
>= 0	取得したアラーム発生時のイベント ID
(-1)	alid の値が正しくなかった。またはリンクイベントは無かった。

#### (4) 説明

alid で指定された AL (アラーム) が発生したときに通知するためにリンクされているイベント ID (CEID) を取得します。

アラーム発生時のリンクイベントがあれば、戻り値に取得したイベント ID が返却されます。

リンクされていない場合は (-1) が返却されます。

### 3.8.2.7 EngGetAlCeOff() アラーム復旧時のリンクイベント ID 取得関数

(1) 呼出書式

**[C, C++]**

```
API TCEID APIX EngGetAlCeOff(
    TALID alid // ALID
);
```

**[.NET VB]**

```
Function EngGetAlCeOff (
    ByVal alid As Int32) As Int32
```

**[.NET C#]**

```
uint EngGetAlCeOff(
    uint alid );
```

(2) 引数

alid

アラーム ID です。  
装置管理情報定義ファイルに登録されているアラームの ID でなければなりません。

(3) 戻り値

戻り値	意味
>= 0	取得したアラーム復旧時のイベント ID
(-1)	alid の値が正しくなかった。またはリンクイベントは無かった。

(4) 説明

alid で指定された AL (アラーム) が復旧したときに通知するためにリンクされているイベント ID (CEID) を取得します。

アラーム復旧時のリンクイベントがあれば、戻り値に取得したイベント ID が返却されます。

リンクされていない場合は (-1) が返却されます。

### 3.8.2.8 EngGetAllList() 全登録アラーム ID 取得関数

(1) 呼出書式

**[C, C++]**

```
API int APIX EngGetAllList(
    TBIN_DLIST **list           // 取得リスト格納ポインタの格納ポインタ
);
```

**[.NET VB]**

```
Function EngGetAllList (
    ByRef list As IntPtr) As Int32
```

**[.NET C#]**

```
int EngGetAllList(
    IntPtr list );
```

(2) 引数

list

取得できた ALID が格納されている TBIN\_DLIST 構造体のポインタを格納する領域のポインタです。

(3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	取得できなかった。

(4) 説明

システムに登録されている全アラーム ID とその名前を TBIN\_DLIST 構造体に取り出すための関数です。

取出す名前は、装置管理情報定義ファイルでアラーム ID 定義時に与えられた名前です。

取得した情報の処理が終了した後、DshFreeTBIN\_DLIST()関数で list 内部の情報格納用に使用されているメモリを開放してください。

TBIN\_DLIST 構造体は次のとおりです。

```
typedef struct{
    int         count;           // 取得できた ID 数
    ULONG      *id_list;        // 取得できた ID 格納用配列
    char       **name_list;     // 取得できた名前格納ポインタ配列
}TBIN_DLIST;
```

### 3.8.2.9 EngNotifyAlarm() アラーム通知要求

#### (1) 呼出書式

##### [C, C++]

```
API int APIX EngNotifyAlarm(
    TALID    alid,           // 通知 alid(アラーム ID)
    int      on_off,        // 発生/復旧(1/0)の指定
    int (WINAPI *AlarmCallback)(), // 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 EngNotifyAlarm (
    ByVal alid As Int32,
    ByVal on_off As Int32,
    ByVal callback As vcallback.callback_NotifyAlarm,
    ByVal upara As Int32) As Int32
```

##### [.NET C#]

```
int EngNotifyAlarm(
    uint alid,
    int on_off,
    CallbackNotifyAlarm AlarmCallback,
    uint upara );
```

#### (2) 引数

alid

ホストに通知するための ALID (アラーム ID) を指定します。  
S5F1 の ALID になります。

on\_off

アラーム発生か復旧かを指定します。 発生の場合は=1、復旧の場合は=0 を指定してください。  
S5F1 の ALCD の最上位ビットに反映されます。

AlarmCallback

DSHEng4 によるアラーム通知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することができます。

#### (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。 msg に応答メッセージが格納されているポインタが返却される。 (2) 非ブロックモード : 要求が受け付けられた。
1	alid は定義されているが Enable になっていなかった。

9	スプールされた。
(-1)	alid が定義されていないか、送信エラーを検出した。
(-14)	T3タイムアウトを検出した。
その他 > 0	ブロックモードの場合、ackc5 の値が返却されます。

#### (4) 説明

ホストにアラーム (S5F1) メッセージを送信するための関数です。

DSHEng4 は指定された alid のメッセージを自動的に組立てた上でホストに送信します。

alid がシステム内に定義されており、報告状態が有効であるとき、ALCD, ALTX を S5F1 に設定展開した上で送信します。

また、指定 alid に CEID (イベント ID) がリンクされている場合、同時にその CEID の S6F11 メッセージも送信します。

AlarmCallback として =0 が指定されている場合は、ブロックモードになり、アラームメッセージの送信が完了してから制御が戻ってきます。

AlarmCallback が指定されている場合、要求を受けられた後、ただちに制御が戻ってきます。そして、アラームメッセージの送信が完了したら、コールバック関数によって完了が通知されます。

#### (5) コールバック関数

##### [C, C++]

```
API int APIX AlarmCallback(
    int end_status,           // 実行結果
    TALID alid,              // 通知した ALID
    ULONG upara              // 呼出時に指定したパラメータ
);
```

##### [.NET VB]

```
Function callback_NotifyAlarm(ByVal end_status As Integer, ByVal alid As Integer, ByVal upara As Integer) As Integer
```

##### [.NET C#]

```
int CallbackNotifyAlarm(int end_status, uint id, 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。
1	alid は定義されているが Enable になっていなかった。
9	スプールされた。
(-1)	alid が定義されていないか、送信エラーを検出した。
(-14)	T3タイムアウトを検出した。
その他 >0	ackc5 の値が返却されます。