

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- 2 / 1 5

3 . 3 変数 (EC, SV, DWAL) 情報アクセスと通信サービス

2 0 0 9 年 7 月

株式会社データマップ

目 次

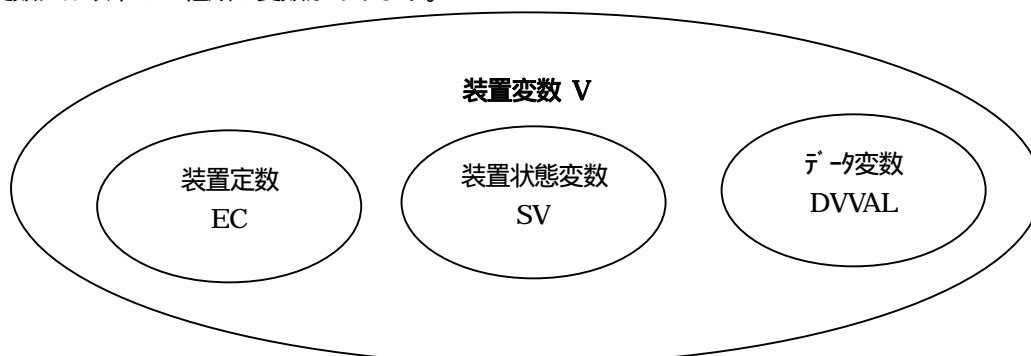
3.3	変数 (EC, SV, DVVAL) 情報アクセスと通信サービス	1
3.3.1	使用する情報構造体	3
3.3.2	装置変数情報アクセス関数	4
3.3.2.1	EngSetVVal装置変数() - 装置変数値設定関数	4
3.3.2.2	EngGetVVal() - 装置変数値取得関数	5
3.3.2.3	EngGetVName() - 装置変数名取得関数	6
	EngGetVNameA()	6
3.3.2.4	EngGetVUnits() - 装置変数単位名取得関数	8
	EngGetVUnitsA()	8
3.3.2.5	EngGetVFormat() - 装置変数フォーマットコード取得関数	10
3.3.2.6	EngGetVArraySize() - 装置変数値の配列サイズ取得関数	11
3.3.2.7	EngGetVMin() - 装置変数最小(Min)値取得関数	12
3.3.2.8	EngGetVMax() - 装置変数最大(Max)値取得関数	13
3.3.2.9	EngGetVNominal() - 装置変数初期(Nominal)値取得関数	14
3.3.2.10	EngCheckVVal() - 装置変数値Min,Maxチェック関数	15
3.3.2.11	EngGetVList() - 全登録装置変数ID取得関数	16
3.3.2.12	EngGetVSizeMinMax() - 装置変数値の最小、最大配列サイズ取得関数	17
3.3.3	EC 装置定数情報アクセスとメッセージ送信関数	18
3.3.3.1	EngSetEcVal() - EC値設定関数	18
3.3.3.2	EngGetEcVal() - EC値取得関数	20
3.3.3.3	EngGetEcName() - EC名取得関数	21
	EngGetEcNameA()	21
3.3.3.4	EngGetEcUnits() - EC単位名取得関数	23
	EngGetEcUnitsA()	23
3.3.3.5	EngGetEcFormat() - ECフォーマットコード取得関数	25
3.3.3.6	EngGetEcArraySize() - EC値の配列サイズ取得関数	26
3.3.3.7	EngGetEcMin() - EC最小(Min)値取得関数	27
3.3.3.8	EngGetEcMax() - EC最大(Max)値取得関数	28
3.3.3.9	EngGetEcNominal() - EC初期(Nominal)値取得関数	29
3.3.3.10	EngCheckEcVal() - EC値MinMaxチェック関数	30
3.3.3.11	EngGetEcList() - 全登録ECID取得関数	31
3.3.3.12	EngGetEcSizeMinMax() - EC値の最小、最大配列サイズ取得関数	32
3.3.4	SV 装置状態変数情報アクセスとメッセージ送信関数	33
3.3.4.1	EngSetSvVal() - SV値設定関数	33
3.3.4.2	EngGetSvVal() - SV値取得関数	35
3.3.4.3	EngGetSvName() - SV名取得関数	36
	EngGetSvNameA()	36
3.3.4.4	EngGetSvUnits() - SV単位名取得関数	38
	EngGetSvUnitsA()	38
3.3.4.5	EngGetSvFormat() - SVフォーマットコード取得関数	40
3.3.4.6	EngGetSvArraySize() - SV値の配列サイズ取得関数	41
3.3.4.7	EngGetSvMin() - SV最小(Min)値取得関数	42
3.3.4.8	EngGetSvMax() - SV最大(Max)値取得関数	43
3.3.4.9	EngGetSvNominal() - SV初期(Nominal)値取得関数	44
3.3.4.10	EngCheckSvVal() - SV値MinMaxチェック関数	45
3.3.4.11	EngGetSvList() - 全登録SVID取得関数	46

3.3.4.12 EngGetSvSizeMinMax()	- SV値の最小、最大配列サイズ取得関数	47
3.3.5 DVVAL	データ変数情報アクセスとメッセージ送信関数	48
3.3.5.1 EngSetDvVal()	- DVVAL値設定関数	48
3.3.5.2 EngGetDvVal()	- DVVAL値取得関数	50
3.3.5.3 EngGetDvName()	- DVVAL名取得関数	51
EngGetDvNameA()		51
3.3.5.4 EngGetDvUnits()	- DVVAL単位名取得関数	53
EngGetDvUnitsA()		53
3.3.5.5 EngGetDvFormat()	- DVVALフォーマットコード取得関数	55
3.3.5.6 EngGetDvArraySize()	- DVVAL値の配列サイズ取得関数	56
3.3.5.7 EngGetDvMin()	- DVVAL最小(Min)値取得関数	57
3.3.5.8 EngGetDvMax()	- DVVAL最大(Max)値取得関数	58
3.3.5.9 EngGetDvNominal()	- DVVAL初期(Nominal)値取得関数	59
3.3.5.10 EngCheckDvVal()	- DVVAL値Min,Maxチェック関数	60
3.3.5.11 EngGetDvList()	- 全登録DVVALID取得関数	61
3.3.5.12 EngGetDvSizeMinMax()	- DVVAL値の最小、最大配列サイズ取得関数	62

(VOL - 3に続く)

3.3 変数 (EC, SV, DWAL) 情報アクセスと通信サービス

装置変数には以下の3種類の変数があります。



ここで述べる情報は、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場合、以下の API 関数を呼出してアクセスすることになります。また、装置に対し変数値要求、定数変更などのメッセージの送信のための関数も準備されています。

(1) 情報アクセスと送信 API 関数

装置変数情報のアクセスとホスト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。

	情報名と API 関数名	機能
1	変数(EC, SV, DWAL)	EC, SV, DVVAL 全変数が対象です。
	(1) EngSetWval()	VID で指定された変数の値を設定変更します。
	(2) EngGetWval()	VID で指定された変数の値を取得します。
	(3) EngGetVName()	VID で指定された変数の V データ名を取得します。
	(4) EngGetVUnits()	VID で指定された変数の単位名を取得します。
	(5) EngGetVFormat()	VID で指定された変数の値の Format と単位データ単位長を取得します。
	(6) EngGetVArraySize()	VID で指定された変数の値のデータの配列サイズを取得します。
	(7) EngGetVMin()	VID で指定された変数の値の最小値を取得します。
	(8) EngGetVMax()	VID で指定された変数の値の最大値を取得します。
	(9) EngGetVNominal()	VID で指定された変数の値の初期設定値を取得します。
	(10) EngCheckVVal()	VID で指定された変数の値をチェックします。
	(11) EngGetVList()	全変数の一覧リストを取得します。
	(12) EngGetVSizeMinMax()	変数値の最小、最大配列サイズを取得します。
	(13) EngSetMulVLimit()	複数個の変数に対します。リット値を設定します。3.10.2.1 で説明します。
	(14) EngSetVLimit()	1 個の変数に対します。リット値を設定します。3.10.2.2 で説明します。
	(15) EngGetVLimit()	変数に対します。リット値を取得します。3.10.2.3 で説明します。
	(16) EngDelVLimit()	変数に対します。リット値を消去します。3.10.2.4 で説明します。
	(17) EngCheckVLimit()	変数値のリットチェックを行う。3.10.2.5 で説明します。
	(18) EngSendS2F45()	変数リット属性定義 S2F45 メッセージを送信します。3.10.2.6 で説明します。
	(19) EngSendS2F47()	変数リット属性要求 S2F47 メッセージを送信します。3.10.2.7 で説明します。
2	EC(装置定数)関連	対象は EC 変数だけです。
	(1) EngSetEcVal()	ECID で指定された定数の値を設定変更します。
	(2) EngGetEcVal()	ECID で指定された定数の値を取得します。
	(3) EngGetEcName()	ECID で指定された定数の EC 名を取得します。

	(4)	EngGetEcUnits()	ECID で指定された定数の単位名を取得します。
	(5)	EngGetEcFormat()	ECID で指定された定数の値の Format と単位 [°] 、 [′] 、 [″] 長を取得します。
	(6)	EngGetEcArraySize()	ECID で指定された定数の値の [°] 、 [′] 、 [″] 配列サイズを取得します。
	(7)	EngGetEcMin()	ECID で指定された定数の値の最小値を取得します。
	(8)	EngGetEcMax()	ECID で指定された定数の値の最大値を取得します。
	(9)	EngGetEcNominal()	ECID で指定された定数の値の初期設定値を取得します。
	(10)	EngCheckEcVal()	ECID で指定された定数の値をチェックします。
	(11)	EngGetEcList()	全 EC 変数の一覧リストを取得します。
	(12)	EngGetEcSizeMinMax()	装置定数値の最小、最大配列サイズを取得します。
3	SV(装置状態変数)関連		対象は SV 変数だけです。
	(1)	EngSetSvVal()	SVID で指定された状態変数の値を設定変更します。
	(2)	EngGetSvVal()	SVID で指定された状態変数の値を取得します。
	(3)	EngGetSvName()	SVID で指定された状態変数の SV 名を取得します。
	(4)	EngGetSvUnits()	SVID で指定された状態変数の単位名を取得します。
	(5)	EngGetSvFormat()	SVID で指定された状態変数の値の Format と単位 [°] 、 [′] 、 [″] 長を取得します。
	(6)	EngGetSvArraySize()	SVID で指定された状態変数の値の [°] 、 [′] 、 [″] 配列サイズを取得します。
	(7)	EngGetSvMin()	SVID で指定された状態変数の値の最小値を取得します。
	(8)	EngGetSvMax()	SVID で指定された状態変数の値の最大値を取得します。
	(9)	EngGetSvNominal()	SVID で指定された状態変数の値の初期設定値を取得します。
	(10)	EngCheckEcVal()	SVID で指定された状態変数の値をチェックします。
	(11)	EngGetSvList()	全 SV 変数の一覧リストを取得します。
	(12)	EngGetSvSizeMinMax()	状態変数値の最小、最大配列サイズを取得します。
4	DVVAL(装置[°]、[′]、[″]変数)関連		対象は DVVAL 変数だけです。
	(1)	EngSetDvVal()	DVID で指定された [°] 、 [′] 、 [″] 変数の値を設定変更します。
	(2)	EngGetDvVal()	DVID で指定された [°] 、 [′] 、 [″] 変数の値を取得します。
	(3)	EngGetDvName()	DVID で指定された [°] 、 [′] 、 [″] 変数の [°] 、 [′] 、 [″] 名を取得します。
	(4)	EngGetDvUnits()	DVID で指定された [°] 、 [′] 、 [″] 変数の単位名を取得します。
	(5)	EngGetDvFormat()	DVID で指定された [°] 、 [′] 、 [″] 変数の値の Format と単位 [°] 、 [′] 、 [″] 長を取得します。
	(6)	EngGetDvArraySize()	DVID で指定された [°] 、 [′] 、 [″] 変数の値の [°] 、 [′] 、 [″] 配列サイズを取得します。
	(7)	EngGetDvMin()	DVID で指定された [°] 、 [′] 、 [″] 変数の値の最小値を取得します。
	(8)	EngGetDvMax()	DVID で指定された [°] 、 [′] 、 [″] 変数の値の最大値を取得します。
	(9)	EngGetDvNominal()	DVID で指定された [°] 、 [′] 、 [″] 変数の値の初期設定値を取得します。
	(10)	EngCheckDvVal()	DVID で指定された [°] 、 [′] 、 [″] 変数の値をチェックします。
	(11)	EngGetDvList()	全 DVVAL 変数の一覧リストを取得します。
	(12)	EngGetDvSizeMinMax()	[°] 、 [′] 、 [″] 変数値の最小、最大配列サイズを取得します。

3.3.1 使用する情報構造体

- (1) S1F3 ならびに S2F13 要求に対して応答された変数値を格納するために以下の構造体を使用します。
また、S2F29 定数値変更時にも設定、送信したい値を与えるために使用します。

```

typedef struct{
    TVID      vid;          // 変数 ID
    int       format;      // 変数値のフォーマット( DSHDR2 で定める ITEM のフォーマット)
    int       asize;       // 変数値の配列サイズ
    void      *value;      // 変数値が格納されているメモリ 領域
} TV_VALUE;

typedef struct{
    int       count;       // 変数 ID の数
    TV_VALUE **vw_list;    // 変数 ID と変数値格納構造体ポインタへのポインタ
                        (count 分)
} TV_VALUE_LIST;

```

- (2) S2F29 装置定数名一覧要求の応答情報を格納するために以下の構造体を使用します。

```

typedef struct{
    TECID     ecid;        // 装置定数 ID
    char      *name;       // 名前
    int       format;      // 値のフォーマット
    int       asize;       // 値の配列サイズ
    void      *ecmin;      // 値の最小値
    void      *ecmax;      // 値の最大値
    void      *ecdef;      // 値のデフォルト値(=Nominal, 初期設定値)
    char      *units;      // 値の単位名
} TEC_NAME;

typedef struct{
    int       count;       // 装置定数 ID の数
    TEC_NAME **name_list;  // 定数名情報が格納されている構造体ポインタへのポインタ
} TEC_NAME_LIST;

```

- (3) S1F11 装置状態変数名一覧要求の応答情報を格納するために以下の構造体を使用します。

```

typedef struct{
    TSVID     svid;        // 装置状態変数 ID
    char      *name;       // 名前
    char      *units;      // 値の単位名
} TSV_NAME;

typedef struct{
    int       count;       // 装置状態変数 ID の数
    TSV_NAME **name_list;  // 状態変数名情報が格納されている構造体ポインタへのポインタ
} TSV_NAME_LIST;

```

3.3.2 装置変数情報アクセス関数

以下説明する装置変数情報アクセス関数は、EC(装置定数)、SV(装置状態変数)ならびにDWAL(装置データ変数)の3つの変数が対象になります。

3.3.3以降にEC, SV, DWALそれぞれ個別対象のAPI関数の説明を行います。

3.3.2.1 EngSetWal 装置変数() - 装置変数値設定関数

(1) 呼出書式

[C, C++]

```
API int APIX EngSetWal(
    TVID vid,                // 変数 ID
    void *val,              // 設定値ポインタ
    int asize                // データの配列サイズ
);
```

[.NET VB]

```
Function EngSetWal (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByVal arraysize As Int32) As Int32
```

[.NET C#]

```
int EngSetWal(
    uint vid,
    byte[] val,
    int arraysize );
```

(2) 引数

vid

装置変数 ID です。
装置管理情報定義ファイルに登録されている装置変数の ID でなければなりません。

val

設定したい値の格納ポインタです。vidのフォーマットに合わせた値を設定してください。

asize

設定値領域の配列サイズです。
フォーマット A, J の場合は、文字列長になります。

(3) 戻り値

戻り値	意味
0	正常に設定された。
(-1)	vidの値が正しくなかった。(登録されていなかった)

(4) 説明

vidで指定されたV(装置変数)に新しいV値を設定します。戻り値が設定結果になります。
正常に設定できた場合は、関数の戻り値は0になります。

3.3.2.2 EngGetWal() - 装置変数値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetWal(
    TVID vid,                // 変数 ID
    void *val,               // 取得値格納ポインタ
    int *asize               // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetWal (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetWal(
    uint vid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

vid

装置変数 ID です。

装置管理情報定義ファイルに登録されている装置変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。vid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。

フォーマット A, J の場合は、文字列長になり、文字列の最後に NULL 文字 (=0) が付加されます。

(3) 戻り値

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

(4) 説明

vid で指定された V (装置変数) の現在値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。

3.3.2.3 EngGetVName() - 装置変数名取得関数 EngGetVNameA()

(1) 呼出書式

[C, C++]

```
API int APIX EngGetVName(
    TVID vid, // 変数 ID
    char *name, // 取得した名前の格納領域のポインタ
    int *asize // 名前の文字列長格納用
);
```

```
API int APIX EngGetVNameA(
    TVID vid, // 変数 VID
    char *name // 取得した名前の格納領域のポインタ
);
```

[.NET VB]

```
Function EngGetVName (
    ByVal vid As Int32,
    ByVal val As String,
    ByRef arraysize As Int32) As Int32
```

```
Function EngGetVName (
    ByVal eqid As Int32,
    ByVal vid As Int32,
    ByVal val As String) As Int32
```

[.NET C#]

```
int EngGetVName(
    uint vid,
    byte[] val,
    ref int arraysize );
```

```
int EngGetVName(
    uint vid,
    byte[] val);
```

(2) 引数

vid

装置変数 ID です。

装置管理情報定義ファイルに登録されている装置変数の 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。

asize

取得された名前の文字列長 (バイト) です。

(3) 戻り値

戻り値	意味
0	正常に取得できた。 - EngGetVName()の場合
> 0	正常に取得できた。 - EngGetVNameA()の場合は名前のバイト長が返る
(-1)	vidの値が正しくなかった。

(4) 説明

vidで指定されたV(装置変数 EC, SV or DV)の名前を取得します。

正常に取得できた場合は、EngGetVName()の戻り値は0になり、取得された名前の配列サイズはasizeで指定された領域に返却されます。

EngGetVNameA()は取得できた名前の長さが返却されます。

引数asizeが無い場合は、名前のバイト長が返却されます。

3.3.2.4 EngGetVUnits() - 装置変数単位名取得関数 EngGetVUnitsA()

(1) 呼出書式

[C, C++]

```
API int APIX EngGetVUnits(
    TVID vid,                // VID
    char *units,            // 取得した単位名の格納領域のポインタ
    int *asize              // 単位名の文字列長格納用
);
```

```
API int APIX EngGetVUnitsA(
    TVID vid,                // VID
    char *units              // 取得した単位名の格納領域のポインタ
);
```

[.NET VB]

```
Function EngGetVUnits (
    ByVal vid As Int32,
    ByVal val As String,
    ByRef arraysize As Int32) As Int32
```

```
Function EngGetVUnits (
    ByVal vid As Int32,
    ByVal val As String) As Int32
```

[.NET C#]

```
int EngGetVUnits(
    uint vid,
    byte[] val,
    ref int arraysize );
```

```
int EngGetVUnits(
    uint vid,
    byte[] val);
```

(2) 引数

vid

装置変数 ID(EC, SV, DWAL)です。

装置管理情報定義ファイルに登録されている変数の ID でなければなりません。

units

取得した単位名(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。

asize

取得された単位名の文字列長です。もし、単位名が無い場合には =0 が格納されます。

(3) 戻り値

戻り値	意味
-----	----

0	正常に取得できた。 - EngGetVUnits()の場合
>0	正常に取得できた。 - EngGetVUnitsA()の場合は名前のバイト長が返る
(-1)	vidの値が正しくなかった。

(4) 説明

vidで指定されたV(装置変数)の値の単位名を取得します。

正常に取得できた場合は、EngGetVUnits()の戻り値は0になり、取得された単位名の配列サイズはasizeで指定された領域に返却されます。

EngGetVUnitsA()は取得できた単位名の長さが返却されます。
引数 asize が無い場合は、単位名のバイト長が返却されます。

3.3.2.5 EngGetVFormat() - 装置変数フォーマットコード取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetVFormat(
    TVID vid,           // 変数 ID
    int *fmt            // 取得フォーマット値格納ポインタ
);
```

[.NET VB]

```
Function EngGetVFormat (
    ByVal vid As Int32,
    ByRef fmt As Int32) As Int32
```

[.NET C#]

```
int EngGetVFormat(
    uint vid,
    ref int fmt );
```

(2) 引数

vid

装置変数 ID です。
装置管理情報定義ファイルに登録されている装置変数の ID でなければなりません。

fmt

取得したフォーマットの値を格納する領域のポインタです。
返却されるフォーマット値は、DSHDR2 ドライバーが定義するアイテムフォーマットです。

(3) 戻り値

戻り値	意味
> 0	正常に取得できた。戻り値はフォーマットの単位バイト長です。
(-1)	vid の値が正しくなかった。

(4) 説明

vid で指定された V(装置変数)のフォーマット値を取得します。
フォーマット値は、DSHDR2 ドライバーで定義するアイテムフォーマットです。
(dsh.h ファイルでマクロ定義されています。)
正常に取得できた場合は、関数の戻り値はアイテムフォーマットの単位配列サイズ になります。

3.3.2.6 EngGetVArraySize() - 装置変数値の配列サイズ取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetVArraySize(
    TVID vid,           // 変数 ID
    int *asize         // 取得変数配列サイズ 格納ポインタ
);
```

[.NET VB]

```
Function EngGetVArraySize (
    ByVal vid As Int32,
    ByRef val As Int32) As Int32
```

[.NET C#]

```
int EngGetVArraySize(
    uint vid,
    ref int val );
```

(2) 引数

vid

装置変数 ID です。

装置管理情報定義ファイルに登録されている装置変数の ID でなければなりません。

asize

取得した配列サイズ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)	vid の値が正しくなかった。

(4) 説明

vid で指定された V(装置変数)の配列サイズを取得します。

例えば、指定された変数のフォーマットが、FORMAT A[6] の場合は、配列サイズとして=6 が返却されます。

ただし、可変配列文字列変数については最小配列サイズが返却されます。

FORMAT : A[4..16] の場合は =4 が返却されます。

3.3.2.7 EngGetVMin() - 装置変数最小(Min)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetVMin(
    TVID vid,                // 変数 ID
    void *val,               // 取得値格納ポインタ
    int *asize               // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetVMin (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetVSizeMinMax(
    uint vid,
    ref int min,
    ref int max );
```

(2) 引数

vid

装置変数 ID です。

装置管理情報定義ファイルに登録されている装置変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。vid のフォーマットと配列サイズ分の値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズを格納します。フォーマット A, J の場合は、文字列長(byte)になります。最小値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

vid で指定された V (装置変数) の最小(Min)値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された V が最小値を持たない場合は、asize に = 0 が返却されます。

3.3.2.8 EngGetVMax() - 装置変数最大(Max)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetVMax(
    TVID vid,                // 変数 ID
    void *val,               // 取得値格納ポインタ
    int *asize               // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetVMax (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetVMax(
    uint vid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

vid

装置変数 ID です。

装置管理情報定義ファイルに登録されている装置変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。vidのフォーマットと配列サイズ分の値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。最大値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

vid で指定された V(装置変数)の最大(Max)値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された V が最大値を持たない場合は、asize に = 0 が返却されます。

3.3.2.9 EngGetVNominal() - 装置変数初期(Nominal)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetVNominal(
    TVID vid,                // 変数 ID
    void *val,               // 取得値格納ポインタ
    int *asize               // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetVNominal (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetVNominal(
    uint vid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

vid

装置変数 ID です。

装置管理情報定義ファイルに登録されている装置変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。vid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。初期値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

vid で指定された V (装置変数) の初期値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された V が初期値を持たない場合は、asize に = 0 が返却されます。

3.3.2.10 EngCheckVal() - 装置変数値 Min,Max チェック関数

(1) 呼出書式

[C, C++]

```
API int APIX EngCheckVal (
    TVID vid,                // 変数 ID
    void *val,              // チェック値格納ポインタ
    int asize               // データの配列サイズ
);
```

[.NET VB]

```
Function EngCheckVal (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByVal arraysize As Int32) As Int32
```

[.NET C#]

```
int EngCheckVal (
    uint vid,
    byte[] val,
    int arraysize );
```

(2) 引数

vid

変数 ID です。装置管理情報定義ファイルに登録されている装置変数の ID でなければなりません。

val

チェックする V の値が格納されている領域のポインタです。

asize

チェック値領域の配列サイズです。

(3) 戻り値

戻り値	意味
0	正常な値である。
(-1)	vid の値が正しくなかった。
(-2)	値が最小値より小さかった。
(-3)	値が最大値より大きかった。

(4) 説明

vid で指定された V (装置変数) としての val 領域に格納されているデータ値をチェックします。チェック方法は、val に格納されている値が、当該 vid の V のために指定された最小値、最大値の範囲内であるかどうかの判断です。最小値、最大値がもし指定されていない場合は、指定されていないものとの比較判断は行いません。(正常であるとみなします。)

3.3.2.11 EngGetVList() 全登録装置変数 ID 取得関数

(1) 呼出書式

[C, C++]

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

[.NET VB]

```
Function EngGetVList (
    ByRef plist As IntPtr) As Int32
```

[.NET C#]

```
int EngGetVList(
    IntPtr plist );
```

(2) 引数

list

取得できた VID 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 VID とその名前を TBIN_DLIST 構造体に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装置変数 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;
```

3.3.2.12 EngGetVSizeMinMax() 装置変数値の最小、最大配列サイズ取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetVSizeMinMax(
    TVID vid,           // 変数 ID
    int *min,          // 最小配列サイズ 格納ポインタ
    int *max           // 最大配列サイズ 格納ポインタ
);
```

[.NET VB]

```
Function EngGetVSizeMinMax (
    ByVal vid As Int32,
    ByRef min As Int32,
    ByRef max As Int32) As Int32
```

[.NET C#]

```
int EngGetVSizeMinMax(
    uint vid,
    ref int min,
    ref int max );
```

(2) 引数

vid

装置変数 ID です。

装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

min

取得した最小配列サイズを格納する領域のポインタです。

max

取得した最大配列サイズ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)	vid の値が正しくなかった。

(4) 説明

vid で指定された WAL (装置変数) の最小と最大配列サイズを取得します。

変数の配列サイズが固定の場合は、min, max に同じ値が返却されます。

可変配列文字列変数については最小、最大配列サイズがそれぞれ min, max に返却されます。

FORMAT : A[4..16] の場合は min = 4 , max = 16 となります。

3.3.3 EC 装置定数情報アクセスとメッセージ送信関数

3.3.3.1 EngSetEcVal() - EC 値設定関数

(1) 呼出書式

[C, C++]

```
API int APIX EngSetEcVal(
    TECID ecid,           // ECID
    void *val,           // 設定値ポインタ
    int asize            // データの配列サイズ
);
```

[.NET VB]

```
Function EngSetEcVal (
    ByVal ecid As Int32,
    ByVal val As IntPtr,
    ByVal arraysize As Int32) As Int32
```

```
Function EngSetEcVal (
    ByVal ecid As Int32,
    ByRef val As Int32,
    ByVal arraysize As Int32) As Int32
```

```
Function EngSetEcVal (
    ByVal ecid As Int32,
    ByVal val As String,
    ByVal arraysize As Int32) As Int32
```

```
Function EngGetEcVal (
    ByVal ecid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngSetEcVal(
    uint ecid,
    byte[] val,
    int arraysize );
```

(2) 引数

ecid

装置定数 ID です。
装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

val

設定したい値が格納されている領域のポインタです。ecid のフォーマットに合わせた値を設定してください。

asize

設定値領域の配列サイズです。
フォーマット A, J の場合は、文字列長になり、文字列の最後に NULL 文字(=0)が付加されます。

(3) 戻り値

戻り値	意味
0	正常に設定された。
(-1)	ecidの値が正しくなかった。(登録されていなかった)

(4) 説明

ecidで指定されたEC(装置定数)に新しい値を設定します。戻り値が設定結果になります。
正常に設定できた場合は、関数の戻り値は0になります。

3.3.3.2 EngGetEcVal() - EC 値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetEcVal(
    TECID ecid,           // ECID
    void *val,           // 取得値格納ポインタ
    int *asize           // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetEcVal (
    ByVal ecid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetEcVal(
    uint ecid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

ecid

装置定数 ID です。

装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。ecid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。

(3) 戻り値

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

(4) 説明

ecid で指定された EC (装置定数) の現在値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。

3.3.3.3 EngGetEcName() - EC 名取得関数 EngGetEcNameA()

(1) 呼出書式

[C, C++]

```
API int APIX EngGetEcName(
    TECID ecid,                // ECID
    char *name,                // 取得した名前の格納領域ポインタ
    int *asize                 // 名前の文字列長格納用
);
```

```
API int APIX EngGetEcNameA(
    TECID ecid,                // ECID
    char *name                 // 取得した名前の格納領域ポインタ
);
```

[.NET VB]

```
Function EngGetEcName (
    ByVal ecid As Int32,
    ByVal val As String,
    ByRef arraysize As Int32) As Int32
```

```
Function EngGetEcName (
    ByVal ecid As Int32,
    ByVal val As String) As Int32
```

[.NET C#]

```
int EngGetEcName(
    uint ecid,
    byte[] val,
    ref int arraysize );
```

```
int EngGetEcName(
    int eqid,
    uint ecid,
    byte[] val);
```

(2) 引数

eqid

GEM 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。

ecid

装置定数 ID です。

装置管理情報定義ファイルに登録されている定数の 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。

asize

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。 - EngGetEcName()の場合
> 0	正常に取得できた。 - EngGetEcNameA()の場合は名前のバイト長が返る
(-1)	ecidの値が正しくなかった。

(4) 説明

ecidで指定されたEC(装置定数)の名前を取得します。

正常に取得できた場合は、EngGetEcName()の戻り値は0になり、取得された名前の配列サイズはasizeで指定された領域に返却されます。

EngGetEcNameA()は取得できた名前の長さが返却されます。
引数 asize が無い場合は、名前のバイト長が返却されます。

3.3.3.4 EngGetEcUnits() - EC 単位名取得関数 EngGetEcUnitsA()

(1) 呼出書式

[C, C++]

```
API int APIX EngGetEcUnits(
    TECID ecid,                // ECID
    char *units,               // 取得した単位名の格納領域のポインタ
    int *asize                 // 単位名の文字列長格納用
);
```

```
API int APIX EngGetEcUnitsA(
    TECID ecid,                // ECID
    char *units                // 取得した単位名の格納領域のポインタ
);
```

[.NET VB]

```
Function EngGetEcUnits (
    ByVal ecid As Int32,
    ByVal val As String,
    ByRef arraysize As Int32) As Int32
```

```
Function EngGetEcUnits (
    ByVal ecid As Int32,
    ByVal val As String) As Int32
```

[.NET C#]

```
int EngGetEcUnits(
    uint ecid,
    byte[] val,
    ref int arraysize );
```

```
int EngGetEcUnits(
    uint ecid,
    byte[] val);
```

(2) 引数

ecid

装置定数 ID です。

装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

units

取得した単位名 (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。

asize

取得された単位名の文字列長です。もし、単位名が無い場合には =0 が格納されます。

(3) 戻り値

戻り値	意味
-----	----

0	正常に取得できた。 - EngGetEcUnits()の場合
>0	正常に取得できた。 - EngGetEcUnitsA()の場合は名前のバイト長が返る
(-1)	vidの値が正しくなかった。

(4) 説明

ecid で指定された EC(装置定数)の値の単位名を取得します。

正常に取得できた場合は、EngGetEcUnits()の戻り値は 0 になり、取得された単位名の配列サイズは asize で指定された領域に返却されます。

EngGetEcUnitsA()は取得できた単位名の長さが返却されます。
引数 asize が無い場合は、単位名のバイト長が返却されます。

3.3.3.5 EngGetEcFormat() - EC フォーマットコード取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetEcFormat(
    TECID ecid,           // ECID
    int *fmt             // 取得フォーマット値格納ポインタ
);
```

[.NET VB]

```
Function EngGetEcFormat (
    ByVal ecid As Int32,
    ByRef fmt As Int32) As Int32
```

[.NET C#]

```
int eqid,
uint ecid,
ref int fmt );
```

(2) 引数

ecid

装置定数 ID です。

装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

fmt

取得したフォーマットの値を格納する領域のポインタです。

返却されるフォーマット値は、DSHDR2 ドライバーが定義するアイテムフォーマットです。

(3) 戻り値

戻り値	意味
> 0	正常に取得できた。戻り値はフォーマットの単位バイト長です。
(-1)	ecid の値が正しくなかった。

(4) 説明

ecid で指定された EC (装置定数) のフォーマット値を取得します。

フォーマット値は、DSHDR2 ドライバーで定義するアイテムフォーマットです。

(dsh.h ファイルでマクロ定義されています。)

正常に取得できた場合は、関数の戻り値は アイテムフォーマットの単位配列サイズ になります。

3.3.3.6 EngGetEcArraySize() - EC 値の配列サイズ取得関数

(1) 呼出書式

[C, C++]

```
API int WINAPI EngGetEcArraySize(
    TECID ecid,           // ECID
    int *asize           // 取得配列サイズ 格納ポインタ
);
```

[.NET VB]

```
Function EngGetEcArraySize (
    ByVal ecid As Int32,
    ByRef val As Int32) As Int32
```

[.NET C#]

```
int EngGetEcArraySize(
    uint ecid,
    ref int val );
```

(2) 引数

ecid

装置定数 ID です。
装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

asize

取得した配列サイズ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)	ecid の値が正しくなかった。

(4) 説明

ecid で指定された EC (装置定数) の配列サイズを取得します。

例えば、指定された EC が、EC_mdIn A が FORMAT: A[6] の場合は、配列サイズとして=6 が返却されます。

ただし、可変配列文字列変数については最小配列サイズが返却されます。

FORMAT: A[4..16] の場合は =4 が返却されます。

3.3.3.7 EngGetEcMin() - EC 最小(Min)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetEcMin(
    TECID ecid,           // ECID
    void *val,           // 取得値格納ポインタ
    int *asize           // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetEcMin (
    ByVal ecid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetEcMin(
    uint ecid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

ecid

装置定数 ID です。

装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。ecid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。最小値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

ecid で指定された EC (装置定数) の最小(Min)値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された EC が最小値を持たない場合は、asize に = 0 が返却されます。

3.3.3.8 EngGetEcMax() - EC 最大(Max)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetEcMax(
    TECID ecid,           // ECID
    void *val,           // 取得値格納ポインタ
    int *asize           // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetEcMax (
    ByVal ecid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetEcMax(
    uint ecid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

ecid

装置定数 ID です。

装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。ecid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。最大値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

ecid で指定された EC (装置定数) の最大(Max)値を取得します。正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された EC が最大値を持たない場合は、asize に = 0 が返却されます。

3.3.3.9 EngGetEcNominal() - EC 初期(Nominal)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetEcNominal(
    TECID ecid,           // ECID
    void *val,           // 取得値格納ポインタ
    int *asize           // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetEcNominal (
    ByVal ecid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetEcNominal(
    uint ecid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

ecid

装置定数 ID です。装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。ecid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。初期値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

ecid で指定された EC (装置定数) の初期値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された EC が初期値を持たない場合は、asize に = 0 が返却されます。

3.3.3.10 EngCheckEcVal() - EC 値 MinMax チェック関数

(1) 呼出書式

[C, C++]

```
API int APIX EngCheckEcVal(
    TECID ecid,           // ECID
    void *val,           // チェック値格納ポインタ
    int asize            // データの配列サイズ
);
```

[.NET VB]

```
Function EngCheckEcVal (
    ByVal ecid As Int32,
    ByVal val As IntPtr,
    ByVal arraysize As Int32) As Int32
```

[.NET C#]

```
int EngCheckEcVal(
    uint ecid,
    byte[] val,
    int arraysize );
```

(2) 引数

ecid

装置定数 ID です。装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

val

チェックする EC の値が格納されている領域のポインタです。

asize

チェック値領域の配列サイズです。

(3) 戻り値

戻り値	意味
0	正常な値である。
(-1)	ecid の値が正しくなかった。
(-2)	値が最小値より小さかった。
(-3)	値が最大値より大きかった。

(4) 説明

ecid で指定された EC (装置定数) としての val 領域に格納されているデータ値をチェックします。

チェック方法は、val に格納されている値が、当該 ecid の EC のために指定された最小値、最大値の範囲内であるかどうかの判断です。最小値、最大値がもし指定されていない場合は、指定されていないものとの比較判断は行いません。(正常であるとみなします。)

3.3.3.11 EngGetEcList() 全登録 ECID 取得関数

(1) 呼出書式

[C, C++]

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

[.NET VB]

```
Function EngGetEcList (
    ByRef plist As IntPtr) As Int32
```

[.NET C#]

```
int EngGetEcList(
    IntPtr plist );
```

(2) 引数

eqid

GEM 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。

list

取得できた ECID 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 ECID 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 EC 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;
```

3.3.3.12 EngGetEcSizeMinMax() - EC 値の最小、最大配列サイズ取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetEcArraySize(
    TECID ecid,           // ECID
    int *min,             // 最小配列サイズ 格納ポインタ
    int *max              // 最大配列サイズ 格納ポインタ
);
```

[.NET VB]

```
Function EngGetEcSizeMinMax (
    ByVal ecid As Int32,
    ByRef min As Int32,
    ByRef max As Int32) As Int32
```

[.NET C#]

```
int EngGetSvSizeMinMax(
    uint ecid,
    ref int min,
    ref int max );
```

(2) 引数

ecid

装置定数 ID です。装置管理情報定義ファイルに登録されている定数の ID でなければなりません。

min

取得した最小配列サイズを格納する領域のポインタです。

max

取得した最大配列サイズ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)	ecid の値が正しくなかった。

(4) 説明

ecid で指定された EC (装置定数) の最小と最大配列サイズを取得します。
変数の配列サイズが固定の場合は、min, max に同じ値が返却されます。

可変配列文字列変数については最小、最大配列サイズがそれぞれ min, max に返却されます。

FORMAT : A[4..16] の場合は min = 4 , max = 16 となります。

3.3.4 SV 装置状態変数情報アクセスとメッセージ送信関数

3.3.4.1 EngSetSvVal() - SV 値設定関数

(1) 呼出書式

[C, C++]

```
API int APIX EngSetSvVal(
    TSVID svid,                // SVID
    void *val,                 // 設定値ポインタ
    int asize                  // データの配列サイズ
);
```

[.NET VB]

```
Function EngSetSvVal (
    ByVal svid As Int32,
    ByVal val As IntPtr,
    ByVal arraysize As Int32) As Int32
```

```
Function EngSetSvVal (
    ByVal svid As Int32,
    ByVal val As String,
    ByVal arraysize As Int32) As Int32
```

```
Function EngSetSvVal (
    ByVal svid As Int32,
    ByRef val As Integer,
    ByVal arraysize As Int32) As Int32
```

```
Function EngGetSvVal (
    ByVal svid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngSetSvVal(
    uint svid,
    byte[] val,
    int arraysize );
```

(2) 引数

svid

装置状態変数 ID です。
装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

val

設定したい値の格納ポインタです。svid のフォーマットに合わせた値を設定してください。

asize

設定値領域の配列サイズです。
フォーマット A, J の場合は、文字列長になります。

(3) 戻り値

戻り値	意味
0	正常に設定された。
(-1)	svidの値が正しくなかった。(登録されていなかった)

(4) 説明

svidで指定されたSV(装置状態変数)に新しい値を設定します。戻り値が設定結果になります。正常に設定できた場合は、関数の戻り値は0になります。

3.3.4.2 EngGetSvVal() - SV 値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvVal(
    TSVID svid,           // SVID
    void *val,           // 取得値格納ポインタ
    int *asize           // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetSvVal (
    ByVal svid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetSvVal(
    uint svid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

svid

装置状態変数 ID です。
装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。svid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。
フォーマット A, J の場合は、文字列長になり、最後に NULL 文字(=0) が付加されます。

(3) 戻り値

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

(4) 説明

svid で指定された SV(装置状態変数)の現在値を取得します。
正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。

3.3.4.3 EngGetSvName() - SV 名取得関数 EngGetSvNameA()

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvName(
    TSVID svid,                // SVID
    char *name,                // 取得した名前の格納領域ポインタ
    int *asize                 // 名前の文字列長格納用
);
```

```
API int APIX EngGetSvNameA(
    TECID svid,               // SVID
    char *name                // 取得した名前の格納領域ポインタ
);
```

[.NET VB]

```
Function EngGetSvName (
    ByVal svid As Int32,
    ByVal val As String,
    ByRef arraysize As Int32) As Int32
```

```
Function EngGetSvName (
    ByVal svid As Int32,
    ByVal val As String) As Int32
```

[.NET C#]

```
int EngGetSvName(
    uint svid,
    byte[] val,
    ref int arraysize );
```

```
int EngGetSvName(
    uint svid,
    byte[] val );
```

(2) 引数

svid

装置状態変数 ID です。

装置管理情報定義ファイルに登録されている状態変数の 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。

asize

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。 - EngGetSvName()の場合
> 0	正常に取得できた。 - EngGetSvnameA()の場合は名前のバイト長が返る
(-1)	ecid の値が正しくなかった。

(4) 説明

svid で指定された SV(装置状態変数)の名前を取得します。

正常に取得できた場合は、EngGetSvName()の戻り値は 0 になり、取得された名前の配列サイズは asize で指定された領域に返却されます。

EngGetSvnameA()は取得できた名前の長さが返却されます。
引数 asize が無い場合は、名前のバイト長が返却されます。

3.3.4.4 EngGetSvUnits() - SV 単位名取得関数 EngGetSvUnitsA()

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvUnits(
    TSVID svid,                // SVID
    char *units,               // 取得した単位名の格納領域ポインタ
    int *asize                 // 単位名の文字列長格納用
);
```

```
API int APIX EngGetSvUnitsA(
    TSVID svid,                // SVID
    char *units                 // 取得した単位名の格納領域ポインタ
);
```

[.NET VB]

```
Function EngGetSvUnits (
    ByVal svid As Int32,
    ByVal val As String,
    ByRef arraysize As Int32) As Int32
```

```
Function EngGetSvUnits (
    ByVal svid As Int32,
    ByVal val As String) As Int32
```

[.NET C#]

```
int EngGetSvUnits(
    uint svid,
    byte[] val,
    ref int arraysize );
```

```
int EngGetSvUnits(
    uint svid,
    byte[] val);
```

(2) 引数

svid

装置状態変数 ID です。

装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

units

取得した単位名 (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。

asize

取得された単位名の文字列長です。もし、単位名が無い場合には =0 が格納されます。

(3) 戻り値

戻り値	意味
-----	----

0	正常に取得できた。 - EngGetSvUnits()の場合
>0	正常に取得できた。 - EngGetSvUnitsA()の場合は名前のバイト長が返る
(-1)	vidの値が正しくなかった。

(4) 説明

svid で指定された SV(装置状態変数)の値の単位名を取得します。

正常に取得できた場合は、EngGetSvUnits()の戻り値は 0 になり、取得された単位名の配列サイズは asize で指定された領域に返却されます。

EngGetSvUnitsA()は取得できた単位名の長さが返却されます。
引数 asize が無い場合は、単位名のバイト長が返却されます。

3.3.4.5 EngGetSvFormat() - SV フォーマットコード取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvFormat(
    TSVID svid,                // SVID
    int *fmt                   // 取得フォーマット値格納ポインタ
);
```

[.NET VB]

```
Function EngGetSvFormat (
    ByVal svid As Int32,
    ByRef fmt As Int32) As Int32
```

[.NET C#]

```
int EngGetSvFormat(
    uint svid,
    ref int fmt );
```

(2) 引数

svid

装置状態変数 ID です。
装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

fmt

取得したフォーマットの値を格納する領域のポインタです。
返却されるフォーマット値は、DSHDR2 ドライバーが定義するアイテムフォーマットです。

(3) 戻り値

戻り値	意味
> 0	正常に取得できた。戻り値はフォーマットの単位バイト長です。
(-1)	svid の値が正しくなかった。

(4) 説明

svid で指定された SV(装置状態変数)のフォーマット値を取得します。
フォーマット値は、DSHDR2 ドライバーで定義するアイテムフォーマットです。
(dsh.h ファイルでマクロ定義されています。)
正常に取得できた場合は、関数の戻り値はアイテムフォーマットの単位配列サイズ になります。

3.3.4.6 EngGetSvArraySize() - SV 値の配列サイズ取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvArraySize(
    TSVID svid,           // SVID
    int *asize           // 取得配列サイズ 格納ポインタ
);
```

[.NET VB]

```
Function EngGetSvArraySize (
    ByVal svid As Int32,
    ByRef val As Int32) As Int32
```

[.NET C#]

```
int EngGetSvArraySize(
    uint svid,
    ref int val );
```

(2) 引数

svid

装置状態変数 ID です。
装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

asize

取得した配列サイズ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)	svid の値が正しくなかった。

(4) 説明

svid で指定された SV (装置状態変数) の配列サイズを取得します。

例えば、指定された SV が、SV_mdIn が FORMAT A[6] の場合は、配列サイズとして=6 が返却されます。

ただし、可変配列文字列変数については最小配列サイズが返却されます。

FORMAT : A[4..16] の場合は =4 が返却されます。

3.3.4.7 EngGetSvMin() - SV 最小(Min)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvMin(
    TSVID svid,                // SVID
    void *val,                 // 取得値格納ポインタ
    int *asize                 // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetSvMin (
    ByVal svid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetSvMin(
    uint svid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

svid

装置状態変数 ID です。装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。svid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。最小値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

svid で指定された SV (装置状態変数) の最小(Min)値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された SV が最小値を持たない場合は、asize に = 0 が返却されます。

3.3.4.8 EngGetSvMax() - SV 最大(Max)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvMax(
    TSVID svid,           // SVID
    void *val,           // 取得値格納ポインタ
    int *asize           // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetSvMax (
    ByVal svid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetSvMax(
    uint svid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

svid

装置状態変数 ID です。装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。svid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。最大値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

svid で指定された SV (装置状態変数) の最大(Max)値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された SV が最大値を持たない場合は、asize に = 0 が返却されます。

3.3.4.9 EngGetSvNominal() - SV 初期(Nominal)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvNominal(
    TSVID svid,           // SVID
    void *val,           // 取得値格納ポインタ
    int *asize           // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetSvNominal (
    ByVal svid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetSvNominal(
    uint svid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

svid

装置状態変数 ID です。装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。svid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。初期値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

svid で指定された SV (装置状態変数) の初期値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された SV が初期値を持たない場合は、asize に = 0 が返却されます。

3.3.4.10 EngCheckSvVal() - SV 値 MinMax チェック関数

(1) 呼出書式

[C, C++]

```
API int APIX EngCheckSvVal(
    TSVID svid,           // SVID
    void *val,           // チェック値格納ポインタ
    int asize            // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngCheckSvVal (
    ByVal svid As Int32,
    ByVal val As IntPtr,
    ByVal arraysize As Int32) As Int32
```

[.NET C#]

```
int EngCheckSvVal(
    uint svid,
    byte[] val,
    int arraysize );
```

(2) 引数

svid

装置状態変数 ID です。
装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

val

チェックする SV の値が格納されている領域のポインタです。

asize

チェック値領域の配列サイズです。

(3) 戻り値

戻り値	意味
0	正常な値である。
(-1)	svid の値が正しくなかった。
(-2)	値が最小値より小さかった。
(-3)	値が最大値より大きかった。

(4) 説明

svid で指定された SV (装置状態変数) としての val 領域に格納されているデータ値をチェックします。チェック方法は、val に格納されている値が、当該 svid の SV のために指定された最小値、最大値の範囲内であるかどうかの判断です。最小値、最大値がもし指定されていない場合は、指定されていないものとの比較判断は行いません。(正常であるとみなします。)

3.3.4.11 EngGetSvList() 全登録 SVID 取得関数

(1) 呼出書式

[C, C++]

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

[.NET VB]

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

[.NET C#]

```
int EngGetSvList(
    IntPtr list );
```

(2) 引数

list

取得できた SVID 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 SVID 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 SV 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;
```

3.3.4.12 EngGetSvSizeMinMax() - SV 値の最小、最大配列サイズ取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSvSizeMinMax(
    TSVID svid,           // SVID
    int *min,            // 最小配列サイズ 格納ポインタ
    int *max             // 最大配列サイズ 格納ポインタ
);
```

[.NET VB]

```
Function EngGetSvSizeMinMax (
    ByVal svid As Int32,
    ByRef min As Int32,
    ByRef max As Int32) As Int32
```

[.NET C#]

```
int EngGetSvSizeMinMax(
    uint svid,
    ref int min,
    ref int max );
```

(2) 引数

svid

装置状態変数 ID です。
装置管理情報定義ファイルに登録されている状態変数の ID でなければなりません。

min

取得した最小配列サイズを格納する領域のポインタです。

max

取得した最大配列サイズ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)	svid の値が正しくなかった。

(4) 説明

svid で指定された SV(装置状態変数)の最小と最大配列サイズを取得します。
変数の配列サイズが固定の場合は、min, max に同じ値が返却されます。

可変配列文字列変数については最小、最大配列サイズがそれぞれ min, max に返却されます。

FORMAT : A[4..16] の場合は min = 4 , max = 16 となります。

3.3.5 DWAL データ変数情報アクセスとメッセージ送信関数

3.3.5.1 EngSetDvVal() - DWAL 値設定関数

(1) 呼出書式

[C, C++]

```
API int APIX EngSetDvVal(
    TVID vid,                // VID
    void *val,              // 設定値ポインタ
    int asize               // データの配列サイズ
);
```

[.NET VB]

```
Function EngSetDvVal (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByVal arraysize As Int32) As Int32
```

```
Function EngSetDvVal (
    ByVal vid As Int32,
    ByRef val As Int32,
    ByVal arraysize As Int32) As Int32
```

```
Function EngSetDvVal (
    ByVal vid As Int32,
    ByVal val As String,
    ByVal arraysize As Int32) As Int32
```

[.NET C#]

```
int EngSetDvVal(
    uint vid,
    byte[] val,
    int arraysize );
```

(2) 引数

vid

装置データ変数 ID です。
装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

val

設定したい値の格納ポインタです。vid のフォーマットに合わせた値を設定してください。

asize

設定値の配列サイズ格納です。
フォーマット A, J の場合は、文字列長になります。

(3) 戻り値

戻り値	意味
0	正常に設定された。
(-1)	vid の値が正しくなかった。(登録されていなかった)

(4) 説明

vid で指定された V(装置データ変数)に新しい値を設定します。戻り値が設定結果になります。
正常に設定できた場合は、関数の戻り値は 0 になります。

3.3.5.2 EngGetDvVal() - DWAL 値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvVal(
    TVID vid,                // VID
    void *val,              // 取得値格納ポインタ
    int *asize              // データのバイト長格納用
);
```

[.NET VB]

```
Function EngGetDvVal (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetDvVal(
    uint vid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

vid

装置データ変数 ID です。
装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。vid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。
フォーマット A, J の場合は、文字列長になり、文字列の最後に NULL 文字 (=0) が付加されます。

(3) 戻り値

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

(4) 説明

vid で指定された V (装置データ変数) の現在値を取得します。
正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。

3.3.5.3 EngGetDvName() - DWAL 名取得関数 EngGetDvNameA()

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvName(
    TVID vid,                // VID
    char *name,              // 取得した名前の格納領域ポインタ
    int *asize               // データの配列サイズ 格納用
);
```

```
API int APIX EngGetDvNameA(
    TVID vid,                // VID
    char *name               // 取得した名前の格納領域ポインタ
);
```

[.NET VB]

```
Function EngGetDvName (
    ByVal vid As Int32,
    ByVal val As String,
    ByRef arraysize As Int32) As Int32
```

```
Function EngGetDvName (
    ByVal vid As Int32,
    ByVal val As String) As Int32
```

[.NET C#]

```
int EngGetDvName(
    uint vid,
    byte[] val,
    ref int arraysize );
```

```
int EngGetDvName(
    uint vid,
    byte[] val);
```

(2) 引数

vid

装置データ変数 ID です。

装置管理情報定義ファイルに登録されているデータ変数の 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。

asize

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。 - EngGetDvName()の場合
> 0	正常に取得できた。 - EngGetDvNameA()の場合は名前のバイト長が返る
(-1)	vidの値が正しくなかった。

(4) 説明

vidで指定されたDV(装置データ変数)の名前を取得します。

正常に取得できた場合は、EngGetDvName()の戻り値は0になり、取得された名前の配列サイズはasizeで指定された領域に返却されます。

EngGetDvNameA()は取得できた名前の長さが返却されます。
引数 asize が無い場合は、名前のバイト長が返却されます。

3.3.5.4 EngGetDvUnits() - DWAL 単位名取得関数 EngGetDvUnitsA()

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvUnits(
    TDVID dvid,                // DVID
    char *units,               // 取得した単位名の格納領域のポインタ
    int *asize                 // 単位名の文字列長格納用
);
```

```
API int APIX EngGetDvUnitsA(
    TDVID dvid,                // DVID
    char *units                 // 取得した単位名の格納領域のポインタ
);
```

[.NET VB]

```
Function EngGetDvUnits (
    ByVal vid As Int32,
    ByVal val As String,
    ByRef arraysize As Int32) As Int32
```

```
Function EngGetDvUnits (
    ByVal vid As Int32,
    ByVal val As String) As Int32
```

[.NET C#]

```
int EngGetDvUnits(
    uint vid,
    byte[] val,
    ref int arraysize );
```

```
int EngGetDvUnits(
    uint vid,
    byte[] val );
```

(2) 引数

dvid

装置データ変数 ID です。
装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

units

取得した単位名 (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。

asize

取得された単位名の文字列長です。もし、単位名が無い場合には =0 が格納されます。

(3) 戻り値

戻り値	意味
-----	----

0	正常に取得できた。 - EngGetDvUnits()の場合
>0	正常に取得できた。 - EngGetDvUnitsA()の場合は名前のバイト長が返る
(-1)	vidの値が正しくなかった。

(4) 説明

dvid で指定された DV(装置データ変数)の値の単位名を取得します。

正常に取得できた場合は、EngGetDvUnits()の戻り値は 0 になり、取得された単位名の配列サイズは asize で指定された領域に返却されます。

EngGetDvUnitsA()は取得できた単位名の長さが返却されます。
引数 asize が無い場合は、単位名のバイト長が返却されます。

3.3.5.5 EngGetDvFormat() - DWVAL フォーマットコード取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvFormat(
    TVID vid,                // VID
    int *fmt                 // 取得フォーマット値格納ポインタ
);
```

[.NET VB]

```
Function EngGetDvFormat (
    ByVal vid As Int32,
    ByRef fmt As Int32) As Int32
```

[.NET C#]

```
int EngGetDvFormat(
    uint vid,
    ref int fmt );
```

(2) 引数

vid

装置データ変数 ID です。
装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

fmt

取得したフォーマットの値を格納する領域のポインタです。
返却されるフォーマット値は、DSHDR2 ドライバーが定義するアイテムフォーマットです。

(3) 戻り値

戻り値	意味
> 0	正常に取得できた。戻り値はフォーマットの単位バイト長です。
(-1)	vid の値が正しくなかった。

(4) 説明

vid で指定された V(装置データ変数)のフォーマット値を取得します。
フォーマット値は、DSHDR2 ドライバーで定義するアイテムフォーマットです。
(dsh.h ファイルでマクロ定義されています。)
正常に取得できた場合は、関数の戻り値は アイテムフォーマットの単位配列サイズ になります。

3.3.5.6 EngGetDvArraySize() - DWAL 値の配列サイズ取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvArraySize(
    TVID vid,                // VID
    int *asize                // 取得配列サイズ 格納ポインタ
);
```

[.NET VB]

```
Function EngGetDvArraySize (
    ByVal vid As Int32,
    ByRef val As Int32) As Int32
```

[.NET C#]

```
int EngGetDvArraySize(
    uint vid,
    ref int val );
```

(2) 引数

vid

装置データ変数 ID です。

装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

asize

取得した配列サイズ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)	vid の値が正しくなかった。

(4) 説明

vid で指定された V (装置データ変数) の配列サイズを取得します。

例えば、指定された V が、DV_temp_FORMAT: A[6] の場合は、配列サイズとして=6 が返却されます。

ただし、可変配列文字列変数については最小配列サイズが返却されます。

FORMAT: A[4..16] の場合は =4 が返却されます。

3.3.5.7 EngGetDvMin() - DWAL 最小(Min)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvMin(
    TVID vid,                // VID
    void *val,               // 取得値格納ポインタ
    int *asize               // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetDvMin (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetDvMin(
    uint vid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

vid

装置データ変数 ID です。

装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。vid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。最小値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

vid で指定された V (装置データ変数) の最小(Min)値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された V が最小値を持たない場合は、asize に = 0 が返却されます。

3.3.5.8 EngGetDvMax() - DWAL 最大(Max)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvMax(
    TVID vid,                // VID
    void *val,              // 取得値格納ポインタ
    int *asize              // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetDvMax (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetDvMax(
    uint vid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

vid

装置データ変数 ID です。

装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。vid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。最大値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

vid で指定された V (装置データ変数) の最大(Max)値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された V が最大値を持たない場合は、asize に = 0 が返却されます。

3.3.5.9 EngGetDvNominal() - DWVAL 初期(Nominal)値取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvNominal(
    TVID vid,                // VID
    void *val,               // 取得値格納ポインタ
    int *asize               // データの配列サイズ 格納用
);
```

[.NET VB]

```
Function EngGetDvNominal (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByRef arraysize As Int32) As Int32
```

[.NET C#]

```
int EngGetDvNominal(
    uint vid,
    byte[] val,
    ref int arraysize );
```

(2) 引数

vid

装置データ変数 ID です。

装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

val

取得した値を格納する領域のポインタです。vid のフォーマットと配列サイズの値を格納するために十分な領域を準備してください。実際に取得した値の配列サイズは asize に格納されます。

asize

取得された値の配列サイズ格納用です。フォーマット A, J の場合は、文字列長になります。初期値が定義されていない場合は、=0 が格納されます。

(3) 戻り値

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

(4) 説明

vid で指定された V (装置データ変数) の初期値を取得します。

正常に取得できた場合は、関数の戻り値は 0 になります。取得データの配列サイズは asize で指定された領域に返却されます。ただし、指定された V が初期値を持たない場合は、asize に = 0 が返却されます。

3.3.5.10 EngCheckDvVal() - DWAL 値 Min,Max チェック関数

(1) 呼出書式

[C, C++]

```
API int APIX EngCheckDvVal(
    TVID vid,                // VID
    void *val,              // チェック値格納ポインタ
    int asize               // データの配列サイズ
);
```

[.NET VB]

```
Function EngCheckDvVal (
    ByVal vid As Int32,
    ByVal val As IntPtr,
    ByVal arraysize As Int32) As Int32
```

[.NET C#]

```
int EngCheckDvVal(
    uint vid,
    byte[] val,
    int arraysize );
```

(2) 引数

vid

装置データ変数 ID です。

装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

val

チェックする V の値が格納されている領域のポインタです。

asize

チェック値領域の配列サイズです。

(3) 戻り値

戻り値	意味
0	正常な値である。
(-1)	vid の値が正しくなかった。
(-2)	値が最小値より小さかった。
(-3)	値が最大値より大きかった。

(4) 説明

vid で指定された V (装置データ変数) としての val 領域に格納されているデータ値をチェックします。

チェック方法は、val に格納されている値が、当該 vid の V のために指定された最小値、最大値の範囲内であるかどうかの判断です。最小値、最大値がもし指定されていない場合は、指定されていないものとの比較判断は行いません。(正常であるとみなします。)

3.3.5.11 EngGetDvList() 全登録 DWALID 取得関数

(1) 呼出書式

[C, C++]

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

[.NET VB]

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

[.NET C#]

```
int EngGetDvList(
    IntPtr list );
```

(2) 引数

list

取得できた DVID 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 DVID 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 DWAL 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;
```

3.3.5.12 EngGetDvSizeMinMax() DWAL 値の最小、最大配列サイズ取得関数

(1) 呼出書式

[C, C++]

```
API int APIX EngGetDvSizeMinMax(
    TDVID dvid,           // DVID
    int *min,            // 最小配列サイズ 格納ポインタ
    int *max             // 最大配列サイズ 格納ポインタ
);
```

[.NET VB]

```
Function EngGetDvSizeMinMax (
    ByVal vid As Int32,
    ByRef min As Int32,
    ByRef max As Int32) As Int32
```

[.NET C#]

```
int EngGetDvSizeMinMax(
    uint vid,
    ref int min,
    ref int max );
```

(2) 引数

dvid

装置データ変数 ID です。
装置管理情報定義ファイルに登録されているデータ変数の ID でなければなりません。

min

取得した最小配列サイズを格納する領域のポインタです。

max

取得した最大配列サイズ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)	dvid の値が正しくなかった。

(4) 説明

dvid で指定された DWAL (装置データ変数) の最小と最大配列サイズを取得します。
変数の配列サイズが固定の場合は、min, max に同じ値が返却されます。

可変配列文字列変数については最小、最大配列サイズがそれぞれ min, max に返却されます。

FORMAT : A[4..16] の場合は min = 4 , max = 16 となります。