



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- 9 / 1 5

3 . 1 5 SUBSTRATE 基板情報アクセスサービス関数

2 0 0 9年7月

株式会社データマップ

文書番号 DSHEng4-09-30329-00



**[ 取り扱い注意 ]**

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

**【改訂履歴】**

番号	改訂日付	項目	概略
1.	2009.6	初版	

## 目 次

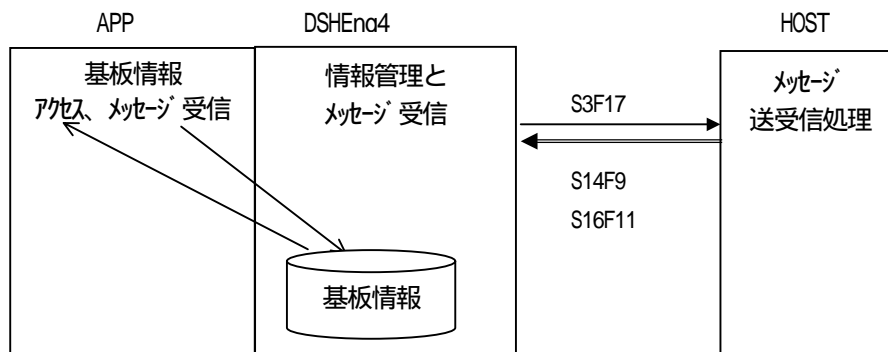
3.15 SUBST 基板情報アクセスサービス関数 .....	1
3.15.1 使用する情報格納構造体 .....	4
3.15.2 SUBST 基板情報アクセス関数 .....	5
3.15.2.1 EngAllocSubstInfo() - 基板の登録 .....	5
3.15.2.2 EngSetSubstInfo() - 基板情報の設定 .....	6
EngSetSubstInfoX() - インデクスでの基板情報の設定 .....	6
3.15.2.3 EngGetSubstInfo() - 基板情報の取得 .....	8
EngGetSubstInfoX() - インデクス指定での基板情報の取得 .....	8
3.15.2.4 EngDelSubstInfo() - 基板の削除 .....	10
EngDelSubstInfoX() - インデクスでの基板の削除 .....	10
3.15.2.5 EngSetSubstInfoState() - 基板情報状態の設定 .....	11
EngSetSubstInfoStateX() - インデクスでの基板情報状態の設定 .....	11
3.15.2.6 EngGetSubstInfoState() - 基板情報状態の取得 .....	13
EngGetSubstInfoStateX() - インデクスでの基板情報状態の取得 .....	13
3.15.2.7 EngSetSubstIdStatus() - 基板ID読取り状態の設定 .....	15
EngSetSubstIdStatusX() - インデクスでの基板ID読取り状態の設定 .....	15
3.15.2.8 EngGetSubstIdStatus() - 基板ID読取り状態の取得 .....	17
EngGetSubstIdStatusX() - インデクスでの基板ID読取り状態の取得 .....	17
3.15.2.9 EngSetSubstAcquiredId() - 読取られたIDの設定 .....	19
EngSetSubstAcquiredIdX() - インデクスでの読取られたIDの設定 .....	19
3.15.2.10 EngGetSubstAcquiredId() - 読取られたIDの取得 .....	20
EngGetSubstAcquiredIdX() - インデクスでの読取られたIDの取得 .....	20
3.15.2.11 EngSetSubstLotId() - LotIdの設定 .....	21
EngSetSubstLotIdX() - インデクスでのLotIdの設定 .....	21
3.15.2.12 EngGetSubstLotId() - LotIdの取得 .....	22
EngGetSubstLotIdX() - インデクスでのLotIdの取得 .....	22
3.15.2.13 EngSetSubstLocId() - LocIdの設定 .....	23
EngSetSubstLocIdX() - インデクスでのLocIdの設定 .....	23
3.15.2.14 EngGetSubstLocId() - LocIdの取得 .....	24
EngGetSubstLocIdX() - インデクスでのLocIdの取得 .....	24
3.15.2.15 EngSetSubstSource() - 移動元ロケーションの設定 .....	25
EngSetSubstSourceX() - インデクスでの移動元ロケーションの設定 .....	25
3.15.2.16 EngGetSubstSource() - 移動元ロケーションの取得 .....	26
EngGetSubstSourceX() - インデクスでの移動元ロケーションの取得 .....	26
3.15.2.17 EngSetSubstDestination() - 移動先ロケーションの設定 .....	27
EngSetSubstDestinationX() - インデクスでの移動先ロケーションの設定 .....	27
3.15.2.18 EngGetSubstDestination() - 移動先ロケーションの取得 .....	28
EngGetSubstDestinationX() - インデクスでの移動先ロケーションの取得 .....	28
3.15.2.19 EngSetSubstBatchLocId() - バッチロケーションの設定 .....	29
EngSetSubstBatchLocIdX() - インデクスでのバッチロケーションの設定 .....	29
3.15.2.20 EngGetSubstBatchLocId() - バッチロケーションの取得 .....	30
EngGetSubstBatchLocIdX() - インデクスでのバッチロケーションの取得 .....	30
3.15.2.21 EngSetSubstPosInBatch() - バッチ内位置の設定 .....	31
EngSetSubstPosInBatchX() - インデクスでのバッチ内位置の設定 .....	31
3.15.2.22 EngGetSubstPosInBatch() - バッチ内位置の取得 .....	32
EngGetSubstPosInBatchX() - インデクスでのバッチ内位置の取得 .....	32

3 . 15 . 2 . 23	EngSetSubsState() – 基板状態の設定.....	33
.	EngSetSubstStateX() – インデクスでの基板状態の設定 .....	33
3 . 15 . 2 . 24	EngGetSubstState() – 基板情報状態の取得.....	35
.	EngGetSubstStateX() – インデクスでの基板情報状態の取得.....	35
3 . 15 . 2 . 25	EngSetSubsMaterialStatus () – 基板処理品質状態の設定.....	37
.	EngSetSubstMaterialStatusX() – インデクスでの基板処理品質状態の設定 .....	37
3 . 15 . 2 . 26	EngGetSubstMaterialStatus() – 基板処理品質状態の取得.....	39
.	EngGetSubstMaterialStatusX() – インデクスでの基板処理品質状態の取得.....	39
3 . 15 . 2 . 27	EngSetSubsProcState () – 基板処理状態の設定.....	41
.	EngSetSubstProcStateX() – インデクスでの基板処理状態の設定.....	41
3 . 15 . 2 . 28	EngGetSubstProcState() – 基板処理状態の取得.....	43
.	EngGetSubstProcStateX() – インデクスでの基板処理状態の取得.....	43
3 . 15 . 2 . 29	EngSetSubsLocState () – 基板ロケーション状態の設定.....	45
.	EngSetSubstLocStateX() – インデクスでの基板ロケーション状態の設定 .....	45
3 . 15 . 2 . 30	EngGetSubstLocState() – 基板ロケーション状態の取得 .....	47
.	EngGetSubstLocStateX() – インデクスでの基板ロケーション状態の取得.....	47
3 . 15 . 2 . 31	EngSetSubsUsage () – 基板Usageの設定.....	49
.	EngSetSubstUsageX() – インデクスでの基板Usageの設定 .....	49
3 . 15 . 2 . 32	EngGetSubstUsage() – 基板Usageの取得 .....	51
.	EngGetSubstUsageX() – インデクスでの基板Usageの取得.....	51
3 . 15 . 2 . 33	EngSetSubsType () – 基板Typeの設定.....	53
.	EngSetSubstTypeX() – インデクスでの基板Typeの設定 .....	53
3 . 15 . 2 . 34	EngGetSubstType() – 基板Typeの取得.....	55
.	EngGetSubstTypeX() – インデクスでの基板Typeの取得.....	55
3 . 15 . 2 . 35	EngSetSubsLocHistory () – 基板通過ロケーション履歴の設定.....	57
.	EngSetSubstLocHistoryX() – インデクスでの基板通過ロケーション履歴の設定.....	57
3 . 15 . 2 . 36	EngGetSubstLocHistory() – 基板通過ロケーション履歴の取得.....	59
.	EngGetSubstLocHistoryX() – インデクスでの基板通過ロケーション履歴の取得.....	59
3 . 15 . 2 . 37	EngAddSubsLocHistory () – 基板通過ロケーション履歴の追加.....	60
.	EngAddSubstLocHistoryX() – インデクスでの基板通過ロケーション履歴の追加 .....	60
3 . 15 . 2 . 38	EngGetSubstList() – 全登録基板ID取得関数.....	62
3 . 15 . 2 . 39	EngGetSubstId() – インデクスから基板IDの取得 .....	63
3 . 15 . 2 . 40	EngGetSubstIdIndex() – 基板IDからインデクスの取得 .....	64
3 . 15 . 3	SUBST基板関連ライブラリ関数.....	65
3 . 15 . 3 . 1	DshFreeTSUBST_INFO() – 基板情報構造体メモリの開放 .....	65
3 . 15 . 3 . 2	DshCopyTSUBST_INFO() – 基板情報構造体メモリのコピー.....	66
3 . 15 . 3 . 3	DshFreeTSUBST_LOC_HIST() – 通過ロケーション履歴構造体メモリの開放.....	67
3 . 15 . 3 . 4	DshCopyTSUBST_LOC_HIST() – 通過ロケーション履歴構造体メモリのコピー .....	68
3 . 15 . 3 . 5	DshInitSubstInfo – TSUBST_INFOの初期設定.....	69
3 . 15 . 3 . 6	DshPutSubstLocHist() – 基板移動履歴情報の追加 .....	72
3 . 15 . 3 . 7	DshInitSubstLocHist() – 基板位置履歴情報の初期設定.....	73
3 . 15 . 3 . 8	DshAddSubstLocHist() – 基板位置履歴情報の追加設定.....	74

(VOL - 10に続く)

### 3.15 SUBST 基板情報アクセスサービス関数

ここで述べる基板情報は、DSHeng4 が管理します。従って、APP はこれらの情報をアクセスと関連メッセージを送信するために以下の DSHeng4 API 関数を使用します。



#### (1) 情報アクセスと送信 API 関数

基板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。

	API 関数名	機能
1	EngAllocSubstInfo()	基板情報領域を割当て登録します。
2	EngSetSubstInfo()	基板情報を設定・変更します。
3	EngSetSubstInfoX()	基板情報インデックス指定で基板情報を設定します。
4	EngGetSubstInfo()	SUBSTID 指定で基板情報を取得します。
5	EngGetSubstInfoX()	基板情報インデックス指定で基板情報を取得します。
6	EngDelSubstInfo()	SUBSTID 指定で基板情報を削除します。
7	EngDelSubstInfoX()	基板情報インデックス指定で基板情報を削除します。
8	EngSetSubstInfoState()	SUBSTID 指定で SUBST 情報の状態を設定します。
9	EngSetSubstInfoStateX()	基板情報インデックス指定で SUBST 情報の状態を設定します。
10	EngGetSubstInfoState()	SUBSTID 指定で SUBST 情報の状態を取得します。
11	EngGetSubstInfoStateX()	基板情報インデックス指定で SUBST 情報の状態を取得します。
12	EngSetSubstIdState()	SUBSTID 指定で SUBST ID の状態を設定します。
13	EngSetSubstIdStateX()	基板情報インデックス指定で SUBST ID の状態を設定します。
14	EngGetSubstIdState()	SUBSTID 指定で SUBSTID の状態を取得します。
15	EngGetSubstIdStateX()	基板情報インデックス指定で SUBST ID の状態を取得します。
16	EngSetSubstAcquiredID()	SUBSTID 指定で基板読取 ID を設定します。
17	EngSetSubstAcquiredIDX()	基板情報インデックス指定で基板読取 ID を設定します。
18	EngGetSubstAcquiredID()	SUBSTID 指定で基板読取 ID を取得します。
19	EngGetSubstAcquiredIDX()	基板情報インデックス指定で基板読取 ID を取得します。
20	EngSetSubstLotId()	SUBSTID 指定で基板の LotId を設定します。
21	EngSetSubstLotIdx()	基板情報インデックス指定で基板の LotId を設定します。
22	EngGetSubstLotId()	SUBSTID 指定で基板の LotId を取得します。
23	EngGetSubstLotIdx()	基板情報インデックス指定で基板の LotId を取得します。
24	EngSetSubstLocId()	SUBSTID 指定で基板の AcquiredIdId を設定します。
25	EngSetSubstLocIdx()	基板情報インデックス指定で基板の AcquiredIdId を設定します。
26	EngGetSubstLocId()	SUBSTID 指定で基板の AcquiredIdId を取得します。

27	EngGetSubstLocIdX()	基板情報インデックス指定で基板の AcquiredId を取得します。
28	EngSetSubstSource()	SUBSTID 指定で基板の移動元アクションを設定します。
29	EngSetSubstSourceX()	基板情報インデックス指定で基板の移動元アクションを設定します。
30	EngGetSubstSource()	SUBSTID 指定で基板の移動元アクションを取得します。
31	EngGetSubstSourceX()	基板情報インデックス指定で基板の移動元アクションを取得します。
32	EngSetSubstDestination()	SUBSTID 指定で基板の移動先アクションを設定します。
33	EngSetSubstDestinationX()	基板情報インデックス指定で基板の移動先アクションを設定します。
34	EngGetSubstDestination()	SUBSTID 指定で基板の移動先アクションを取得します。
35	EngGetSubstDestinationX()	基板情報インデックス指定で基板の移動先アクションを取得します。
36	EngSetSubstBatchLocId()	SUBSTID 指定で基板のバッチアクションを設定します。
37	EngSetSubstBatchLocIdX()	基板情報インデックス指定で基板のバッチアクションを設定します。
38	EngGetSubstBatchLocId()	SUBSTID 指定で基板のバッチアクションを取得します。
39	EngGetSubstBatchLocIdX()	基板情報インデックス指定で基板のバッチアクションを取得します。
40	EngSetSubstPosInBatch()	SUBSTID 指定で基板のバッチ内位置を設定します。
41	EngSetSubstPosInBatchX()	基板情報インデックス指定で基板のバッチ内位置を設定します。
42	EngGetSubstPosInBatch()	SUBSTID 指定で基板のバッチ内位置を取得します。
43	EngGetSubstPosInBatchX()	基板情報インデックス指定で基板のバッチ内位置を取得します。
44	EngSetSubstState()	SUBSTID 指定で SUBST の状態を設定します。
45	EngSetSubstStateX()	基板情報インデックス指定で SUBST の状態を設定します。
46	EngGetSubstState()	SUBSTID 指定で SUBST の状態を取得します。
47	EngGetSubstStateX()	基板情報インデックス指定で SUBST の状態を取得します。
48	EngSetSubstProcState()	SUBSTID 指定で SUBST の処理状態を設定します。
49	EngSetSubstProcStateX()	基板情報インデックス指定で SUBST の処理状態を設定します。
50	EngGetSubstProcState()	SUBSTID 指定で SUBST の処理状態を取得します。
51	EngGetSubstProcStateX()	基板情報インデックス指定で SUBST の処理状態を取得します。
52	EngGetSubstStateX()	基板情報インデックス指定で SUBST の状態を取得します。
53	EngSetSubstLocState()	SUBSTID 指定で SUBST のアクション状態を設定します。
54	EngSetSubstLocStateX()	基板情報インデックス指定で SUBST のアクション状態を設定します。
55	EngGetSubstLocState()	SUBSTID 指定で SUBST のアクション状態を取得します。
56	EngGetSubstLocStateX()	基板情報インデックス指定で SUBST のアクション状態を取得します。
57	EngSetSubstIdStatus()	SUBSTID 指定で SUBSTID の状態を設定します。
58	EngSetSubstIdStatusX()	基板情報インデックス指定で SUBSTID の状態を設定します。
59	EngGetSubstIdStatus()	SUBSTID 指定で SUBSTID の状態を取得します。
60	EngGetSubstIdStatusX()	基板情報インデックス指定で SUBSTID の状態を取得します。
61	EngSetSubstUsage()	SUBSTID 指定で SUBSTID の使用方法を設定します。
62	EngSetSubstUsageX()	基板情報インデックス指定で SUBSTID の使用方法を設定します。
63	EngGetSubstUsage()	SUBSTID 指定で SUBSTID の使用方法を取得します。
64	EngGetSubstUsageX()	基板情報インデックス指定で SUBSTID の使用方法を取得します。
65	EngSetSubstType()	SUBSTID 指定で SUBSTID のタイプを設定します。
66	EngSetSubstTypeX()	基板情報インデックス指定で SUBSTID のタイプを設定します。
67	EngGetSubstType()	SUBSTID 指定で SUBSTID のタイプを取得します。
68	EngGetSubstTypeX()	基板情報インデックス指定で SUBSTID のタイプを取得します。
69	EngSetSubstLocHistory()	SUBSTID 指定で SUBSTID のアクション履歴を設定します。
70	EngSetSubstLocHistoryX()	基板情報インデックス指定で SUBSTID のアクション履歴を設定します。
71	EngGetSubstLocHistory()	SUBSTID 指定で SUBSTID のアクション履歴を取得します。
72	EngGetSubstTLocHistoryX()	基板情報インデックス指定で SUBSTID のアクション履歴を取得します。

73	EngGetSubstList()	基板 ID の一覧リストを取得します。
74	EngGetSubstId()	指定した基板情報インデックスの SUBSTID を取得します。
75	EngGetSubstIdIndex()	指定した SUBSTID の情報インデックスを取得します。

SUBSTID インデクスは、DSHEng4 が管理する各 SUBSTID 領域の番号です。このインデクスの値は、EngAllocSubstInfo()関数実行時に DSHEng4 によって割当てられ、APP に渡されます。また、基板の取得時に、情報格納構造体のメンバー、index に設定されます。

## (2) ライブラリ関数

他に APP が使用できる基板情報処理用 API 関数として、以下の関数があります。

	API 関数名	機能
1	DshFreeTSUBST_INFO()	基板情報が格納されている TSUBST_INFO 構造体の内部で使用されているメモリを開放するための関数です。
2	DshCopyTSUBST_INFO()	TSUBST_INFO の基板情報を別の別の構造体にコピーします。
3	DshFreeTSUBST_LOC_HIST()	TSUBST_LOC_HIST 構造体内のメモリを開放します。
4	DshCopyTSUBST_LOC_HIST()	TSUBST_LOC_HIST 構造体の内容を別の構造体にコピーします。
5	DshInitSubstInfo()	TSUBST_INFO 構造体を初期設定します。
6	DshPutSubstLocHist()	TSUBST_INFO 構造体内に基板ロケーション移動情報を 1 個追加設定します。
7	DshInitSubstLocHist()	TSUBST_INFO 内に設定するための履歴情報構造体 TSUBST_LOC_HIST を初期設定します。
8	DshAddSubstLocHist()	TSUBST_LOC_HIST 構造体内に履歴情報を 1 個追加します。

### 3.15.1 使用する情報格納構造体

基板情報を操作する関数は、共通の情報格納のための TSUBST\_INFO 構造体を使用します。  
基板情報に関連する構造体は下記のとおりです。

#### (1) TSUBST\_INFO Substrate Information

```
typedef struct{
    int      index;
    int      state;                // 情報の状態です。
    char     *substid;            // ID
    char     *acquiredID;
    char     *lotID;
    char     *subst_locID;
    char     *subst_source;
    char     *subst_destination;
    char     *batch_locID;
    char     *subst_pos_in_batch;
    int      substIDstatus;
    int      material_status;
    int      subst_proc_state;
    int      subst_state;
    int      subst_loc_state;
    int      subst_usage;
    int      subst_type;
    TSUBST_LOC_HIST *hist;
} TSUBST_INFO;
```

#### (2) TSLOT\_LOC\_HIST Substrate Location History

```
typedef struct{
    int      hist_count;
    char     **subst_locID;
    char     **TimeIn;
    char     **TimeOut;
} TSUBST_LOC_HIST;
```



### 3.15.2 SUBST 基板情報アクセス関数

#### 3.15.2.1 EngAllocSubstInfo() - 基板の登録

##### (1) 呼出書式

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

```
API int APIX EngAllocSubstInfo(
    char *substid,          // 基板 ID 格納領域のポインタ
    int *index              // 得られた情報領域のインデックス格納用ポインタ
);
```

###### [.NET VB]

```
Function EngAllocSubstInfo (
    ByVal substid As String,
    ByRef index As Int32) As Int32
```

###### [.NET C#]

```
int EngAllocSubstInfo(
    byte[] substid,
    ref int index );
```

##### (2) 引数

substid

登録したい基板 ID が格納されているポインタです。

index

登録された情報領域のインデックス値が格納される領域のポインタです。

##### (3) 戻り値

戻り値	意味
0	正常に登録できた。
1	指定された SUBSTID は既に登録されていた。
(-1)	登録できなかった。

##### (4) 説明

基板を新規にシステムに登録するための関数です。

登録は、引数 substid で与えられる基板 ID をシステムに登録します。

正常に登録できた場合は、index で指定される領域に登録された情報領域のインデックスが設定返却されます。もし、substid に指定された基板が既に登録済みであった場合には関数の戻り値 1 を返却します。index には既に登録されている情報領域のインデックスが設定されます。

得られたインデックスを使って、情報の設定、取得、削除などのアクセスを行うことができます。

### 3.15.2.2 EngSetSubstInfo() - 基板情報の設定

#### EngSetSubstInfoX() インデクスでの基板情報の設定

#### (1) 呼出書式

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

```
API int APIX EngSetSubstInfo(
    TSUBST_INFO *pinfo // 基板情報格納構造体のポインタ
);

API int APIX EngSetSubstInfoX(
    int index, // EngAllocSubstInfo()で得られた基板情報のインデクス
    TSUBST_INFO *pinfo // 基板情報格納構造体のポインタ
);
```

##### [.NET VB]

```
Function EngSetSubstInfo (
    ByRef pinfo As dsh_info.TSUBST_INFO) As Int32

Function EngSetSubstInfoX (
    ByVal index As Int32,
    ByRef pinfo As dsh_info.TSUBST_INFO) As Int32
```

##### [.NET C#]

```
int EngSetSubstInfo(
    ref TSUBST_INFO pinfo );

int EngSetSubstInfoX(
    int index,
    ref TSUBST_INFO pinfo );
```

#### (2) 引数

pinfo

設定したい基板情報が格納されている格納構造体領域のポインタです。

index

基板 ID 情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	設定できなかった。

#### (4) 説明

本関数は、pinfo に格納されている基板情報の設定・変更に使います。

引数 pinfo 内のメンバー subst id に指定される基板 ID の情報として設定されます。

pinfo 内には SUBSTID の他、スロット情報などの情報が含まれます。

指定した SUBSTID が既に登録済みである場合には、pinfo 内の情報にすべて書き換えられます。

pinfo 内の SUBSTID が未登録であった場合は、登録手続きをしてから SUBST 情報を設定します。  
(EngAllocSubstInfo()関数で行われる登録と同じ登録が行われます。)

### 3.15.2.3 EngGetSubstInfo() - 基板情報の取得 EngGetSubstInfoX() - インデクス指定での基板情報の取得

#### (1) 呼出書式

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

```
API int APIX EngGetSubstInfo(
    char *substid, // SUBSTID が格納されている領域のポインタ
    TSUBST_INFO *pinfo // 基板情報を格納する構造体ポインタを格納するポインタ
);
```

```
API int APIX EngGetSubstInfoX(
    int index, // 基板情報のインデクス
    TSUBST_INFO *pinfo // 基板情報を格納するポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstInfo (
    ByVal substid As String,
    ByRef pinfo As dsh_info.TSUBST_INFO) As Int32
```

```
Function EngGetSubstInfoX (
    ByVal index As Int32,
    ByRef pinfo As dsh_info.TSUBST_INFO) As Int32
```

##### [.NET C#]

```
int EngGetSubstInfo(
    byte[] substid,
    ref TSUBST_INFO pinfo );
```

```
int EngGetSubstInfoX(
    int index,
    ref TSUBST_INFO pinfo );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

pinfo

取得した基板情報を格納する構造体領域ポインタです。

index

基板 ID 情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (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。(SUBSTID が未登録であった。)

#### (4) 説明

substid または index に指定されている基板の情報を pinfo 構造体領域に格納します。

TSUBST\_INFO 構造体の中に情報を格納するために必要なメモリは、DSHEng4 が準備確保します。即ち、構造体のメンバーの中でポインタになっている情報の実体即ち、substid、スロット情報などのためのメモリは DSHEng4 が準備します。

これらのメモリは、使用后、ユーザが DSHEng4 の API 関数を使って次のように開放してください。

```
TSUBST_INFO      *pinfo;

if ( EngGetSubstInfo( substid, pinfo ) == 0 ){
    pinfo の処理
    処理終了後
    DshFreeTSUBST_INFO( pinfo );          // pinfo 内に使用されているメモリの開放
}
```

### 3.15.2.4 EngDelSubstInfo() - 基板の削除 EngDelSubstInfoX() インデクスでの基板の削除

#### (1) 呼出書式

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

```
API int APIX EngDelSubstInfo(
    char *substid; // SUBSTID が格納されている領域のポインタ
);
```

```
API int APIX EngDelSubstInfoX(
    int index, // インデクス(0,1,2,...)
);
```

##### [.NET VB]

```
Function EngDelSubstInfo (
    ByVal substid As String) As Int32
```

```
Function EngDelSubstInfoX (
    ByVal index As Int32) As Int32
```

##### [.NET C#]

```
int EngDelSubstInfo(
    byte[] substid );
```

```
int EngDelSubstInfoX(
    int index );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

index

削除したい基板情報のインデクスです。

#### (3) 戻り値

戻り値	意味
0	正常に削除できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

substid または index に指定されている基板 ID の情報を登録から削除します。

### 3.15.2.5 EngSetSubstInfoState() 基板情報状態の設定 EngSetSubstInfoStateX() インデクスでの基板情報状態の設定

#### (1) 呼出書式

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

```
API int APIX EngSetSubstInfoState(
    char    *substid,          // SUBSTID が格納されている領域のポインタ
    int     state              // 設定したい状態値
);
```

```
API int APIX EngSetSubstInfosStateX(
    int     index,            // 基板情報のインデクス
    int     state              // 設定したい状態値
);
```

##### [.NET VB]

```
Function EngSetSubstInfoState (
    ByVal substid As String,
    ByVal state As Int32) As Int32
```

```
Function EngSetSubstInfoStateX (
    ByVal index As Int32,
    ByVal state As Int32) As Int32
```

##### [.NET C#]

```
int EngSetSubstInfoState(
    byte[] substid,
    int state );
```

```
int EngSetSubstInfoStateX(
    int index,
    int state );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

state

設定したい SUBSTID 情報の状態値です。

値=(-1)は使用不可を意味する値になります。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。

インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板情報の状態値を設定します。

状態値の意味合いは値=(-1)以外についてはユーザが定義し、使用します。



### 3.15.2.6 EngGetSubstInfoState() 基板情報状態の取得 EngGetSubstInfoStateX() インデクスでの基板情報状態の取得

#### (1) 呼出書式

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

```
API int APIX EngGetSubstInfoState(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    int     *state             // 取得した状態値格納用領域ポインタ
);
```

```
API int APIX EngGetSubstInfoStateX(
    int     index,             // 基板情報のインデクス
    int     *state             // 取得した状態値格納用領域ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstInfoState (
    ByVal substid As String,
    ByRef state As Int32) As Int32
```

```
Function EngGetSubstInfoStateX (
    ByVal index As Int32,
    ByRef state As Int32) As Int32
```

##### [.NET C#]

```
int EngGetSubstInfoState(
    byte[] substid,
    ref int state );
```

```
int EngGetSubstInfoStateX(
    int index,
    ref int state );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

state

取得した SUBSTID 情報の状態値を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板情報の状態値を取得します。  
状態値の意味合いは値=(-1)以外についてはユーザが定義し、使用します。

### 3.15.2.7 EngSetSubstIdStatus() 基板 ID 読取り状態の設定 EngSetSubstIdStatusX() インデクスでの基板 ID 読取り状態の設定

#### (1) 呼出書式

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

```
API int APIX EngSetSubstIdStatus(
    char    *substid,          // SUBSTID が格納されている領域のポインタ
    int     status            // 設定したい状態値
);
```

```
API int APIX EngSetSubstIdStatusX(
    int     index,            // 基板情報のインデクス
    int     status            // 設定したい状態値
);
```

##### [.NET VB]

```
Function EngSetSubstIdStatus (
    ByVal substid As String,
    ByVal status As Int32) As Int32
```

```
Function EngSetSubstIdStatusX (
    ByVal index As Int32,
    ByVal status As Int32) As Int32
```

##### [.NET C#]

```
int EngSetSubstIdStatus(
    byte[] substid,
    int status );
```

```
int EngSetSubstIdStatusX(
    int index,
    int status );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

status

設定したい SUBSTID の ID 読取り状態値です。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板 ID 読取り状態値を設定します。  
デフォルトの値は次のとおりです。

基板ID状態のデフォルト値

状態値記号	値
ST_SubstIdNotRead	0
ST_SubstIdWaitingForHost	1
ST_SubstIdVerificationOK	2
ST_SubstIdVerificationFail	3
ST_SubstIdNotExist	4

### 3.15.2.8 EngGetSubstIdStatus() 基板 ID 読取り状態の取得 EngGetSubstIdStatusX() インデクスでの基板 ID 読取り状態の取得

#### (1) 呼出書式

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

```
API int APIX EngGetSubstIdStatus(
    char    *substid,          // SUBSTID が格納されている領域のポインタ
    int     *status           // 取得した状態値格納用領域ポインタ
);
```

```
API int APIX EngGetSubstIdStatusX(
    int     index,            // 基板情報のインデクス
    int     *status           // 取得した状態値格納用領域ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstIdStatus (
    ByVal substid As String,
    ByRef status As Int32) As Int32
```

```
Function EngGetSubstIdStatusX (
    ByVal index As Int32,
    ByRef status As Int32) As Int32
```

##### [.NET C#]

```
int EngGetSubstIdStatus(
    byte[] substid,
    ref int status );
```

```
int EngGetSubstIdStatusX(
    int index,
    ref int status );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

status

取得した SUBSTID の ID 読取り状態値を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板 ID 読取りの状態値を取得します。  
デフォルトの値は次のとおりです。

基板ID状態のデフォルト値

状態値記号	値
ST_SubstIdNotRead	0
ST_SubstIdWaitingForHost	1
ST_SubstIdVerificationOK	2
ST_SubstIdVerificationFail	3
ST_SubstIdNotExist	4

### 3.15.2.9 EngSetSubstAcquiredId() 読取られた ID の設定 EngSetSubstAcquiredIdx() インデクスでの読取られた ID の設定

#### (1) 呼出書式

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

```

API int  APIX EngSetSubstAcquiredId(
    char  *substid,           // SUBSTID が格納されている領域のポインタ
    char  *acquired_id       // 設定したい読取られた SUBSTID 格納ポインタ
);
API int  APIX EngSetSubstAcquiredIdx(
    int    index,           // 基板情報のインデクス
    char  *acquired_id       // 設定したい読取られた SUBSTID 格納ポインタ
);

```

##### [.NET VB]

```

Function EngSetSubstAcquiredId (
    ByVal substid As String,
    ByVal acqid As String) As Int32
Function EngSetSubstAcquiredIdx (
    ByVal index As Int32,
    ByVal acqid As String) As Int32

```

##### [.NET C#]

```

int EngSetSubstAcquiredId(
    byte[] substid,
    byte[] acqid );
int EngSetSubstAcquiredIdx(
    int index,
    byte[] acqid );

```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

acquired\_id

設定したい読取られた SUBSTID 値 (文字列) です。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

読取られた SUBSTID(文字列)を設定します。

### 3.15.2.10 EngGetSubstAcquiredId() 読取られた ID の取得 EngGetSubstAcquiredIdx() インデクスでの読取られた ID の取得

#### (1) 呼出書式

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

```
API int  APIX EngGetSubstAcquiredId(
    char  *substid,           // SUBSTID が格納されている領域のポインタ
    char  *acquired_id       // 取得した読取られた SUBSTID 格納ポインタ
);
API int  APIX EngGetSubstAcquiredIdx(
    int    index,           // 基板情報のインデクス
    char  *acquired_id       // 取得した読取られた SUBSTID 格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstAcquiredId (
    ByVal substid As String,
    ByVal acqid As String) As Int32
Function EngGetSubstAcquiredIdx (
    ByVal index As Int32,
    ByVal acqid As String) As Int32
```

##### [.NET C#]

```
int EngGetSubstAcquiredId(
    byte[] substid,
    byte[] acqid );
int EngGetSubstAcquiredIdx(
    int index,
    byte[] acqid );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

acquired\_id

取得した “読取られた SUBSTID 値” を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdx()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板の読取られた SUBSTID(文字列)を取得します。



### 3.15.2.11 EngSetSubstLotId() Lot Id の設定 EngSetSubstLotIdX() インデクスでの Lot Id の設定

#### (1) 呼出書式

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

```
API int  APIX EngSetSubstLotId(
    char  *substid,           // SUBSTID が格納されている領域のポインタ
    char  *lotid             // 設定したい lotid 格納ポインタ
);
API int  APIX EngSetSubstLotIdX(
    int    index,           // 基板情報のインデクス
    char  *lotid           // 設定したい lotid 格納ポインタ
);
```

##### [.NET VB]

```
Function EngSetSubstLotId (
    ByVal substid As String,
    ByVal lotid As String) As Int32
Function EngSetSubstLotIdX (
    ByVal index As Int32,
    ByVal lotid As String) As Int32
```

##### [.NET C#]

```
int EngSetSubstLotId(
    byte[] substid,
    byte[] lotid );
int EngSetSubstLotIdX(
    int index,
    byte[] lotid );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

lotid

設定したい SUBSTID の lotid 値 (文字列) です。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 ID の lotid(文字列)を設定します。

### 3.15.2.12 EngGetSubstLotId() Lot Id の取得 EngGetSubstLotIdX() インデクスでの Lot Id の取得

#### (1) 呼出書式

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

```
API int  APIX EngGetSubstLotId(
    char  *substid,           // SUBSTID が格納されている領域のポインタ
    char  *lotid             // 取得した lotid 格納ポインタ
);
API int  APIX EngGetSubstLotIdX(
    int    index,           // 基板情報のインデクス
    char  *lotid           // 取得した lotid 格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstLotId (
    ByVal substid As String,
    ByVal lotid As String) As Int32
Function EngGetSubstLotIdX (
    ByVal index As Int32,
    ByVal lotid As String) As Int32
```

##### [.NET C#]

```
int EngGetSubstLotId(
    byte[] substid,
    byte[] lotid );
int EngGetSubstLotIdX(
    int index,
    byte[] lotid );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

lotid

取得した SUBSTID の lotid 値を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 Id の lotid(文字列)を取得します。

### 3.15.2.13 EngSetSubstLocId() LocId の設定 EngSetSubstLocIdX() インデクスでの LocId の設定

#### (1) 呼出書式

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

```
API int  APIX EngSetSubstLocId(
    char  *substid,           // SUBSTID が格納されている領域のポインタ
    char  *location          // 設定したい location 格納ポインタ
);
API int  APIX EngSetSubstLocIdX(
    int    index,           // 基板情報のインデクス
    char  *location        // 設定したい location 格納ポインタ
);
```

##### [.NET VB]

```
Function EngSetSubstLocId (
    ByVal substid As String,
    ByVal location As String) As Int32
Function EngSetSubstLocIdX (
    ByVal index As Int32,
    ByVal location As String) As Int32
```

##### [.NET C#]

```
int EngSetSubstLocId(
    byte[] substid,
    byte[] location );
int EngSetSubstLocIdX(
    int index,
    byte[] location );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

location

設定したい SUBSTID の location (文字列) です。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 ID の location(文字列)を設定します。

### 3.15.2.14 EngGetSubstLocId() LocId の取得 EngGetSubstLocIdX() インデクスでの LocId の取得

#### (1) 呼出書式

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

```
API int  APIX EngGetSubstLocId(
    char  *substid,           // SUBSTID が格納されている領域のポインタ
    char  *location          // 取得した location 格納ポインタ
);
API int  APIX EngGetSubstLocIdX(
    int    index,           // 基板情報のインデクス
    char  *location        // 取得した location 格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstLocId (
    ByVal substid As String,
    ByVal location As String) As Int32
Function EngGetSubstLocIdX (
    ByVal index As Int32,
    ByVal location As String) As Int32
```

##### [.NET C#]

```
int EngGetSubstLocId(
    byte[] substid,
    byte[] location );
int EngGetSubstLocIdX(
    int index,
    byte[] location );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

location

取得した SUBSTID の location を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 Id の location(文字列)を取得します。

3.15.2.15 EngSetSubstSource() 移動元ロケーションの設定  
 EngSetSubstSourceX() インデクスでの移動元ロケーションの設定

(1) 呼出書式

[C, C++]

```
API int  APIX EngSetSubstSource(
    char  *substid,           // SUBSTID が格納されている領域のポインタ
    char  *source            // 設定したい移動元ロケーション格納ポインタ
);
API int  APIX EngSetSubstSourceX(
    int   index,             // 基板情報のインデクス
    char  *source            // 設定したい移動元ロケーション格納ポインタ
);
```

[.NET VB]

```
Function EngSetSubstSource (
    ByVal substid As String,
    ByVal source As String) As Int32
Function EngSetSubstSourceX (
    ByVal index As Int32,
    ByVal source As String) As Int32
```

[.NET C#]

```
int EngSetSubstSource(
    byte[] substid,
    byte[] source );
int EngSetSubstSourceX(
    int index,
    byte[] source );
```

(2) 引数

- substid  
基板 ID が格納されている領域のポインタです。
- source  
設定したい SUBSTID の移動元ロケーション (文字列) です。
- index  
基板情報のインデクスです。登録時に EngAllocSubstInfo() 関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex() 関数で取得することができます。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

指定された基板 ID の移動元ロケーション(文字列)を設定します。

### 3.15.2.16 EngGetSubstSource() 移動元ロケーションの取得 EngGetSubstSourceX() インデクスでの移動元ロケーションの取得

#### (1) 呼出書式

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

```
API int  APIX EngGetSubstSource(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    char    *source            // 取得した移動元ロケーション格納ポインタ
);
API int  APIX EngGetSubstSourceX(
    int     index,             // 基板情報のインデクス
    char    *source            // 取得した移動元ロケーション格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstSource (
    ByVal substid As String,
    ByVal source As String) As Int32
Function EngGetSubstSourceX (
    ByVal index As Int32,
    ByVal source As String) As Int32
```

##### [.NET C#]

```
int EngGetSubstSource(
    byte[] substid,
    byte[] source );
int EngGetSubstSourceX(
    int index,
    byte[] source );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

source

取得した SUBSTID の移動元ロケーションを格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 Id の移動元ロケーション(文字列)を取得します。

### 3.15.2.17 EngSetSubstDestination() 移動先ロケーションの設定 EngSetSubstDestinationX() インデクスでの移動先ロケーションの設定

#### (1) 呼出書式

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

```
API int  APIX EngSetSubstDestination(
    char    *substid,          // SUBSTID が格納されている領域のポインタ
    char    *dest              // 設定したい移動先ロケーション格納ポインタ
);
API int  APIX EngSetSubstDestinationX(
    int     index,            // 基板情報のインデクス
    char    *dest              // 設定したい移動先ロケーション格納ポインタ
);
```

##### [.NET VB]

```
Function EngSetSubstDestination (
    ByVal substid As String,
    ByVal dest As String) As Int32
Function EngSetSubstDestinationX (
    ByVal index As Int32,
    ByVal dest As String) As Int32
```

##### [.NET C#]

```
int EngSetSubstDestination(
    byte[] substid,
    byte[] dest );
int EngSetSubstDestinationX(
    int index,
    byte[] dest );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

dest

設定したい SUBSTID の移動先ロケーション (文字列) です。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 ID の移動先ロケーション(文字列)を設定します。

### 3.15.2.18 EngGetSubstDestination() 移動先ロケーションの取得 EngGetSubstDestinationX() インデクスでの移動先ロケーションの取得

#### (1) 呼出書式

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

```
API int  APIX EngGetSubstDestination(
    char  *substid,           // SUBSTID が格納されている領域のポインタ
    char  *dest               // 取得した移動先ロケーション格納ポインタ
);
API int  APIX EngGetSubstDestinationX(
    int   index,             // 基板情報のインデクス
    char  *dest               // 取得した移動先ロケーション格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstDestination (
    ByVal substid As String,
    ByVal dest As String) As Int32
Function EngGetSubstDestinationX (
    ByVal index As Int32,
    ByVal dest As String) As Int32
```

##### [.NET C#]

```
int EngGetSubstDestination(
    byte[] substid,
    byte[] dest );
int EngGetSubstDestinationX(
    int index,
    byte[] dest );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

dest

取得した SUBSTID の移動先ロケーションを格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 Id の移動先ロケーション(文字列)を取得します。



### 3.15.2.19 EngSetSubstBatchLocId() バッチロケーションの設定 EngSetSubstBatchLocIdX() インデクスでのバッチロケーションの設定

#### (1) 呼出書式

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

```
API int APIX EngSetSubstBatchLocId(
    char *substid, // SUBSTID が格納されている領域のポインタ
    char *blocid // 設定したいバッチロケーション格納ポインタ
);
API int APIX EngSetSubstBatchLocIdX(
    int index, // 基板情報のインデクス
    char *blocid // 設定したいバッチロケーション格納ポインタ
);
```

##### [.NET VB]

```
Function EngSetSubstBatchLocId (
    ByVal substid As String,
    ByVal blocid As String) As Int32
Function EngSetSubstBatchLocIdX (
    ByVal index As Int32,
    ByVal blocid As String) As Int32
```

##### [.NET C#]

```
int EngSetSubstBatchLocId(
    byte[] substid,
    byte[] blocid );
int EngSetSubstBatchLocIdX(
    int index,
    byte[] blocid );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

blocid

設定したい SUBSTID のバッチロケーション (文字列) です。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo() 関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex() 関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 ID のバッチロケーション(文字列)を設定します。

3.15.2.20 EngGetSubstBatchLocId() バッチロケーションの取得  
 EngGetSubstBatchLocIdX() インデクスでのバッチロケーションの取得

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSubstBatchLocId(
    char *substid, // SUBSTID が格納されている領域のポインタ
    char *blocid // 取得したバッチロケーション格納ポインタ
);
API int APIX EngGetSubstBatchLocIdX(
    int index, // 基板情報のインデクス
    char *blocid // 取得したバッチロケーション格納ポインタ
);
```

[.NET VB]

```
Function EngGetSubstBatchLocId (
    ByVal substid As String,
    ByVal blocid As String) As Int32
Function EngGetSubstBatchLocIdX (
    ByVal index As Int32,
    ByVal blocid As String) As Int32
```

[.NET C#]

```
int EngGetSubstBatchLocId(
    byte[] substid,
    byte[] blocid );
int EngGetSubstBatchLocIdX(
    int index,
    byte[] blocid );
```

(2) 引数

substid

基板 ID が格納されている領域のポインタです。

blocid

取得した SUBSTID のバッチロケーションを格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllcSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

(3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

指定された基板 Id のバッチロケーション(文字列)を取得します。

### 3.15.2.21 EngSetSubstPosInBatch() バッチ内位置の設定 EngSetSubstPosInBatchX() インデクスでのバッチ内位置の設定

#### (1) 呼出書式

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

```
API int APIX EngSetSubstPosInBatch(
    char *substid, // SUBSTID が格納されている領域のポインタ
    char *posid // 設定したいバッチ内位置格納ポインタ
);
API int APIX EngSetSubstPosInBatchX(
    int index, // 基板情報のインデクス
    char *posid // 設定したいバッチ内位置格納ポインタ
);
```

##### [.NET VB]

```
Function EngSetSubstPosInBatch (
    ByVal substid As String,
    ByVal posid As String) As Int32
Function EngSetSubstPosInBatchX (
    ByVal index As Int32,
    ByVal posid As String) As Int32
```

##### [.NET C#]

```
int EngSetSubstPosInBatch(
    byte[] substid,
    byte[] posid );
int EngSetSubstPosInBatchX(
    int index,
    byte[] posid );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

posid

設定したい SUBSTID のバッチ内位置 (文字列) です。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo() 関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex() 関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 ID のバッチ内位置(文字列)を設定します。

### 3.15.2.22 EngGetSubstPosInBatch() バッチ内位置の取得 EngGetSubstPosInBatchX() インデクスでのバッチ内位置の取得

#### (1) 呼出書式

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

```
API int  APIX EngGetSubstPosInBatch(
    char  *substid,          // SUBSTID が格納されている領域のポインタ
    char  *posid            // 取得したバッチ内位置格納ポインタ
);
API int  APIX EngGetSubstPosInBatchX(
    int    index,           // 基板情報のインデクス
    char  *posid            // 取得したバッチ内位置格納ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstPosInBatch (
    ByVal substid As String,
    ByVal posid As String) As Int32
Function EngGetSubstPosInBatchX (
    ByVal index As Int32,
    ByVal posid As String) As Int32
```

##### [.NET C#]

```
int EngGetSubstPosInBatch(
    byte[] substid,
    byte[] posid );
int EngGetSubstPosInBatchX(
    int index,
    byte[] posid );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

posid

取得した SUBSTID のバッチ内位置を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

指定された基板 Id のバッチ内位置(文字列)を取得します。

### 3.15.2.23 EngSetSubsState() 基板状態の設定 EngSetSubstStateX() インデクスでの基板状態の設定

#### (1) 呼出書式

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

```
API int APIX EngSetSubstState(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    int     state              // 設定したい状態値
);
API int APIX EngSetSubstInfosStateX(
    int     index,            // 基板情報のインデクス
    int     state              // 設定したい状態値
);
```

##### [.NET VB]

```
Function EngSetSubstState (
    ByVal substid As String,
    ByVal state As Int32) As Int32
Function EngSetSubstStateX (
    ByVal index As Int32,
    ByVal state As Int32) As Int32
```

##### [.NET C#]

```
int EngSetSubstState(
    byte[] substid,
    int state );
int EngSetSubstStateX(
    int index,
    int state );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

state

設定したい基板の状態値です。

index

基板情報のインデクスです。登録時に EngAllocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板の状態値を設定します。

デフォルトの値は次のとおりです。

基板状態のデフォルト値

状態値記号	値
ST_StSource	0
ST_AtWork	1
ST_AtDestination	2

### 3.15.2.24 EngGetSubstState() 基板情報状態の取得 EngGetSubstStateX() インデクスでの基板情報状態の取得

#### (1) 呼出書式

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

```
API int APIX EngGetSubstState(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    int     *state             // 取得した状態値格納用領域ポインタ
);
```

```
API int APIX EngGetSubstStateX(
    int     index,             // 基板情報のインデクス
    int     *state             // 取得した状態値格納用領域ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstState (
    ByVal substid As String,
    ByRef state As Int32) As Int32
```

```
Function EngGetSubstStateX (
    ByVal index As Int32,
    ByRef state As Int32) As Int32
```

##### [.NET C#]

```
int EngGetSubstState(
    byte[] substid,
    ref int state );
```

```
int EngGetSubstStateX(
    int index,
    ref int state );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

state

取得した SUBSTID 情報の状態値を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdx()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板の状態値を取得します。  
デフォルトの値は次のとおりです。

基板状態のデフォルト値

状態値記号	値
ST_StSource	0
ST_AtWork	1
ST_AtDestination	2



3.15.2.25 EngSetSubstMaterialStatus ( )      基板処理品質状態の設定  
 EngSetSubstMaterialStatusX( )      インデクスでの基板処理品質状態の設定

(1) 呼出書式

[C,C++]

```
API int APIX EngSetSubstMaterialStatus(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    int     status             // 設定したい状態値
);
```

```
API int APIX EngSetSubstInfosStateX(
    int     index,             // 基板情報のインデクス
    int     status             // 設定したい状態値
);
```

[.NET VB]

```
Function EngSetSubstMaterialStatus (
    ByVal substid As String,
    ByVal state As Int32) As Int32
```

```
Function EngSetSubstMaterialStatusX (
    ByVal index As Int32,
    ByVal state As Int32) As Int32
```

[.NET C#]

```
int EngSetSubstMaterialStatus(
    byte[] substid,
    int state );
```

```
int EngSetSubstMaterialStatusX(
    int index,
    int state );
```

(2) 引数

substid

基板 ID が格納されている領域のポインタです。

status

設定したい基板の処理品質状態値です。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板の処理品質状態値を設定します。

状態値の定義はユーザが行うことになります。

3 . 15 . 2 . 26 EngGetSubstMaterialStatus() 基板処理品質状態の取得  
 EngGetSubstMaterialStatusX() インデクスでの基板処理品質状態の取得

( 1 ) 呼出書式

[c,C++]

```
API int APIX EngGetSubstMaterialStatus(
    char *substid, // SUBSTID が格納されている領域のポインタ
    int *status // 取得した状態値格納用領域ポインタ
);
```

```
API int APIX EngGetSubstMaterialStatusX(
    int index, // 基板情報のインデクス
    int *status // 取得した状態値格納用領域ポインタ
);
```

[.NET VB]

```
Function EngGetSubstMaterialStatus (
    ByVal substid As String,
    ByRef state As Int32) As Int32
```

```
Function EngGetSubstMaterialStatusX (
    ByVal index As Int32,
    ByRef state As Int32) As Int32
```

[.NET C#]

```
int EngGetSubstMaterialStatus(
    byte[] substid,
    ref int state );
```

```
int EngGetSubstMaterialStatusX(
    int index,
    ref int state );
```

( 2 ) 引数

substid  
 基板 ID が格納されている領域のポインタです。

status  
 取得した SUBSTID 情報の処理品質状態値を格納する領域のポインタです。

index  
 基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

( 3 ) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板の処理品質状態値を取得します。

状態値の定義はユーザが行うことになります。

3.15.2.27 EngSetSubstProcState ( ) 基板処理状態の設定  
 EngSetSubstProcStateX( ) インデクスでの基板処理状態の設定

(1) 呼出書式

[C, C++]

```
API int APIX EngSetSubstProcState(
    char *substid, // SUBSTID が格納されている領域のポインタ
    int state // 設定したい状態値
);
```

```
API int APIX EngSetSubstInfosStateX(
    int index, // 基板情報のインデクス
    int state // 設定したい状態値
);
```

[.NET VB]

```
Function EngSetSubstProcState (
    ByVal substid As String,
    ByVal state As Int32) As Int32
```

```
Function EngSetSubstProcStateX (
    ByVal index As Int32,
    ByVal state As Int32) As Int32
```

[.NET C#]

```
int EngSetSubstProcState(
    byte[] substid,
    int state );
```

```
int EngSetSubstProcStateX(
    int index,
    int state );
```

(2) 引数

substid

基板 ID が格納されている領域のポインタです。

state

設定したい基板の処理状態値です。

値=(-1)は使用不可を意味する値になります。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板の処理状態値を設定します。  
デフォルトの値は次のとおりです。

基板処理状態のデフォルト値

状態値記号	値
ST_SUBST_NeedsProcessing	0
ST_SUBST_InProcessing	1
ST_SUBST_processed	2
ST_SUBST_Aborted	3
ST_SUBST_Stopped	4
ST_SUBST_Rejected	5
ST_SUBST_Lost	6

3.15.2.28 EngGetSubstProcState() 基板処理状態の取得  
 EngGetSubstProcStateX() インデクスでの基板処理状態の取得

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSubstProcState(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    int     *state             // 取得した状態値格納用領域ポインタ
);
```

```
API int APIX EngGetSubstProcStateX(
    int     index,             // 基板情報のインデクス
    int     *state             // 取得した状態値格納用領域ポインタ
);
```

[.NET VB]

```
Function EngGetSubstProcState (
    ByVal substid As String,
    ByRef state As Int32) As Int32
```

```
Function EngGetSubstProcStateX (
    ByVal index As Int32,
    ByRef state As Int32) As Int32
```

[.NET C#]

```
int EngGetSubstProcState(
    byte[] substid,
    ref int state );
```

```
int EngGetSubstProcStateX(
    int index,
    ref int state );
```

(2) 引数

substid

基板 ID が格納されている領域のポインタです。

state

取得した SUBSTID 情報の処理状態値を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdx()関数で取得することができます。

(3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板の処理状態値を取得します。  
デフォルトの値は次のとおりです。

基板処理状態のデフォルト値

状態値記号	値
ST_SUBST_NeedsProcessing	0
ST_SUBST_InProcessing	1
ST_SUBST_processed	2
ST_SUBST_Aborted	3
ST_SUBST_Stopped	4
ST_SUBST_Rejected	5
ST_SUBST_Lost	6



3.15.2.29 EngSetSubstLocState () 基板ロケーション状態の設定  
 EngSetSubstLocStateX() インデクスでの基板ロケーション状態の設定

(1) 呼出書式

[C, C++]

```
API int APIX EngSetSubstLocState(
    char    *substid,          // SUBSTID が格納されている領域のポインタ
    int     state              // 設定したい状態値
);
```

```
API int APIX EngSetSubstInfosStateX(
    int     index,            // 基板情報のインデクス
    int     state              // 設定したい状態値
);
```

[.NET VB]

```
Function EngSetSubstLocState (
    ByVal substid As String,
    ByVal state As Int32) As Int32
```

```
Function EngSetSubstLocStateX (
    ByVal index As Int32,
    ByVal state As Int32) As Int32
```

[.NET C#]

```
int EngSetSubstLocState(
    byte[] substid,
    int state );
```

```
int EngSetSubstLocStateX(
    int index,
    int state );
```

(2) 引数

substid

基板 ID が格納されている領域のポインタです。

state

設定したい基板のロケーション状態値です。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

(3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板のロケーション状態値を設定します。  
デフォルトの値は次のとおりです。

基板ロケーション状態のデフォルト値

状態値記号	値
ST_SUBST_Unoccupied	0
ST_SUBST_Occupied	1

### 3.15.2.30 EngGetSubstLocState() 基板ロケーション状態の取得 EngGetSubstLocStateX() インデクスでの基板ロケーション状態の取得

#### (1) 呼出書式

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

```
API int APIX EngGetSubstLocState(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    int     *state             // 取得した状態値格納用領域ポインタ
);
```

```
API int APIX EngGetSubstLocStateX(
    int     index,             // 基板情報のインデクス
    int     *state             // 取得した状態値格納用領域ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstLocState (
    ByVal substid As String,
    ByRef state As Int32) As Int32
```

```
Function EngGetSubstLocStateX (
    ByVal index As Int32,
    ByRef state As Int32) As Int32
```

##### [.NET C#]

```
int EngGetSubstLocState(
    byte[] substid,
    ref int state );
```

```
int EngGetSubstLocStateX(
    int index,
    ref int state );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

state

取得した SUBSTID 情報のロケーション状態値を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdxIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板のロケーション状態値を取得します。  
デフォルトの値は次のとおりです。

基板ロケーション状態のデフォルト値

状態値記号	値
ST_SUBST_Unoccupied	0
ST_SUBST_Occupied	1

3 . 15 . 2 . 31 EngSetSubstUsage ( ) 基板 Usage の設定  
 EngSetSubstUsageX( ) インデクスでの基板 Usage の設定

( 1 ) 呼出書式

[C, C++]

```
API int APIX EngSetSubstUsage(
    char    *substid,          // SUBSTID が格納されている領域のポインタ
    int     usage              // 設定したい Usage 値
);
```

```
API int APIX EngSetSubstInfosStateX(
    int     index,            // 基板情報のインデクス
    int     usage              // 設定したい Usage 値
);
```

[.NET VB]

```
Function EngSetSubstUsage (
    ByVal substid As String,
    ByVal usage As Int32) As Int32
```

```
Function EngSetSubstUsageX (
    ByVal index As Int32,
    ByVal usage As Int32) As Int32
```

[.NET C#]

```
int EngSetSubstUsage(
    byte[] substid,
    int usage );
```

```
int EngSetSubstUsageX(
    int index,
    int usage );
```

( 2 ) 引数

substid

基板 ID が格納されている領域のポインタです。

usage

設定したい基板の Usage 値です。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

( 3 ) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

( 4 ) 説明

基板の Usage 値を設定します。  
デフォルトの値は次のとおりです。

基板Usageのデフォルト値

状態値記号	値
SUBST_USAGE_Product	0
SUBST_USAGE_Test	1
SUBST_USAGE_Filler	2

3.15.2.32 EngGetSubstUsage() 基板 Usage の取得  
 EngGetSubstUsageX() インデクスでの基板 Usage の取得

(1) 呼出書式

[C, C++]

```
API int APIX EngGetSubstUsage(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    int     *usage             // 取得した Usage 値格納用領域ポインタ
);
```

```
API int APIX EngGetSubstUsageX(
    int     index,             // 基板情報のインデクス
    int     *usage             // 取得した Usage 値格納用領域ポインタ
);
```

[.NET VB]

```
Function EngGetSubstUsage (
    ByVal substid As String,
    ByRef usage As Int32) As Int32
```

```
Function EngGetSubstUsageX (
    ByVal index As Int32,
    ByRef usage As Int32) As Int32
```

[.NET C#]

```
int EngGetSubstUsage(
    byte[] substid,
    ref int usage );
```

```
int EngGetSubstUsageX(
    int index,
    ref int usage );
```

(2) 引数

substid

基板 ID が格納されている領域のポインタです。

usage

取得した SUBSTID 情報の Usage 値を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdx()関数で取得することができます。

(3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板の Usage 値を取得します。  
デフォルトの値は次のとおりです。

基板Usageのデフォルト値

状態値記号	値
SUBST_USAGE_Product	0
SUBST_USAGE_Test	1
SUBST_USAGE_Filler	2



### 3.15.2.33 EngSetSubstType () 基板 Type の設定 EngSetSubstTypeX() インデクスでの基板 Type の設定

#### (1) 呼出書式

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

```
API int APIX EngSetSubstType(
    char    *substid,          // SUBSTID が格納されている領域のポインタ
    int     type              // 設定したいType 値
);
```

```
API int APIX EngSetSubstInfosStateX(
    int     index,           // 基板情報のインデクス
    int     type            // 設定したいType 値
);
```

##### [.NET VB]

```
Function EngSetSubstType (
    ByVal substid As String,
    ByVal stype As Int32) As Int32
```

```
Function EngSetSubstTypeX (
    ByVal index As Int32,
    ByVal stype As Int32) As Int32
```

##### [.NET C#]

```
int EngSetSubstType(
    byte[] substid,
    int stype );
```

```
int EngSetSubstTypeX(
    int index,
    int stype );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

type

設定したい基板の Type 値です。

値=(-1)は使用不可を意味する値になります。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。

インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板の Type 値を設定します。

デフォルトの値は次のとおりです。

基板Typeのデフォルト値

状態値記号	値
ST_SUBST_TYPE_Wafer	0
ST_SUBST_TYPE_FlatPanel	1
ST_SUBST_TYPE_CD	2
ST_SUBST_TYPE_Mask	3

### 3.15.2.34 EngGetSubstType() 基板 Type の取得 EngGetSubstTypeX() インデクスでの基板 Type の取得

#### (1) 呼出書式

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

```
API int APIX EngGetSubstType(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    int     *stype             // 取得した Type 値格納用領域ポインタ
);
```

```
API int APIX EngGetSubstTypeX(
    int     index,             // 基板情報のインデクス
    int     *stype             // 取得した Type 値格納用領域ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstType (
    ByVal substid As String,
    ByRef stype As Int32) As Int32
```

```
Function EngGetSubstTypeX (
    ByVal index As Int32,
    ByRef stype As Int32) As Int32
```

##### [.NET C#]

```
int EngGetSubstType(
    byte[] substid,
    ref int stype );
```

```
int EngGetSubstTypeX(
    int index,
    ref int stype );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

stype

取得した SUBSTID 情報の Type 値を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdx()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板の Type 値を取得します。  
デフォルトの値は次のとおりです。

基板Typeのデフォルト値

状態値記号	値
ST_SUBST_TYPE_Wafer	0
ST_SUBST_TYPE_FlatPanel	1
ST_SUBST_TYPE_CD	2
ST_SUBST_TYPE_Mask	3

3 . 15 . 2 . 35 EngSetSubstLocHistory ( )      基板通過ロケーション履歴の設定  
 EngSetSubstLocHistoryX( )      インデクスでの基板通過ロケーション履歴の設定

( 1 ) 呼出書式

[C, C++]

```
API int APIX EngSetSubstLocHistory(
    char      *substid,           // SUBSTID が格納されている領域のポインタ
    TSUBST_LOC_HIST *pinfo       // 設定したい通過ロケーション履歴情報のポインタ
);
```

```
API int APIX EngSetSubstInfosStateX(
    int      index,             // 基板情報のインデクス
    TSUBST_LOC_HIST *pinfo     // 設定したい通過ロケーション履歴情報のポインタ
);
```

[.NET VB]

```
Function EngSetSubstLocHistory (
    ByVal substid As String,
    ByRef hinfo As dsh_info.TSUBST_LOC_HIST) As Int32
```

```
Function EngSetSubstLocHistoryX (
    ByVal index As Int32,
    ByRef hinfo As dsh_info.TSUBST_LOC_HIST) As Int32
```

[.NET C#]

```
int EngSetSubstLocHistory(
    byte[] substid,
    ref TSUBST_LOC_HIST hinfo );
```

```
int EngSetSubstLocHistoryX(
    int index,
    ref TSUBST_LOC_HIST hinfo );
```

( 2 ) 引数

substid

基板 ID が格納されている領域のポインタです。

pinfo

設定したい基板の通過ロケーション履歴情報が格納されている領域のポインタです。  
 値=(-1)は使用不可を意味する値になります。

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
 インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

( 3 ) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

(4) 説明

基板の基板通過ロケーション履歴情報を設定します。

### 3.15.2.36 EngGetSubstLocHistory() 基板通過ロケーション履歴の取得 EngGetSubstLocHistoryX() インデクスでの基板通過ロケーション履歴の取得

#### (1) 呼出書式

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

```
API int APIX EngGetSubstLocHistory(
    char *substid, // SUBSTID が格納されている領域のポインタ
    TSUBST_LOC_HIST *pinfo // 取得した通過ロケーション履歴情報格納用領域ポインタ
);
API int APIX EngGetSubstLocHistoryX(
    int index, // 基板情報のインデクス
    TSUBST_LOC_HIST *pinfo // 取得した通過ロケーション履歴情報格納用領域ポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstLocHistory (
    ByVal substid As String,
    ByRef hinfo As dsh_info.TSUBST_LOC_HIST) As Int32
Function EngGetSubstLocHistoryX (
    ByVal index As Int32,
    ByRef hinfo As dsh_info.TSUBST_LOC_HIST) As Int32
```

##### [.NET C#]

```
int EngGetSubstLocHistory(
    byte[] substid,
    ref TSUBST_LOC_HIST hinfo );
int EngGetSubstLocHistoryX(
    int index,
    ref TSUBST_LOC_HIST hinfo );
```

#### (2) 引数

substid

基板 ID が格納されている領域のポインタです。

pinfo

取得した SUBSTID 情報の通過ロケーション履歴情報を格納する領域のポインタです。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

#### (3) 戻り値

戻り値	意味
0	正常に取得できた。
(-1)	SUBSTID が未登録であった。

#### (4) 説明

基板の基板通過ロケーション履歴情報を取得します。

3.15.2.37 EngAddSubstLocHistory ()      基板通過ロケーション履歴の追加  
 EngAddSubstLocHistoryX()      インデクスでの基板通過ロケーション履歴の追加

(1) 呼出書式

[C, C++]

```
API int APIX EngAddSubstLocHistory(
    char    *substid,           // SUBSTID が格納されている領域のポインタ
    char    *loc,              // ロケーション名
    char    *time_in,         // 入り時刻
    char    *time_out         // 出時刻
);
```

```
API int APIX EngAddSubstInfosStateX(
    int     index,            // 基板情報のインデクス
    char    *loc,            // ロケーション名
    char    *time_in,       // 入り時刻
    char    *time_out      // 出時刻
);
```

[.NET VB]

```
Function EngAddSubstLocHistory (
    ByVal substid As String,
    ByVal loc As String,
    ByVal time_in As String,
    ByVal time_out As String) As Int32
```

```
Function EngAddSubstLocHistoryX (
    ByVal index As Int32,
    ByVal loc As String,
    ByVal time_in As String,
    ByVal time_out As String) As Int32
```

[.NET C#]

```
int EngAddSubstLocHistory(
    byte[] substid,
    byte[] loc,
    byte[] time_in,
    byte[] time_out );
```

```
int EngAddSubstLocHistoryX(
    int index,
    byte[] loc,
    byte[] time_in,
    byte[] time_out );
```

(2) 引数

substid

基板 ID が格納されている領域のポインタです。



loc

通過ロケーション名です。

time\_in

入り時刻です。

time\_out

出時刻です。

index

基板情報のインデクスです。登録時に EngAllLocSubstInfo()関数によって与えられます。  
インデクスは、SUBSTID から EngGetSubstIdIndex()関数で取得することができます。

### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	SUBSTID が未登録であった。

### (4) 説明

基板の基板通過ロケーション履歴情報を追加します。

### 3.15.2.38 EngGetSubstList() 全登録基板 ID 取得関数

#### (1) 呼出書式

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

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

##### [.NET VB]

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

##### [.NET C#]

```
int EngGetSubstList(
    IntPtr list );
```

#### (2) 引数

list

取得できた SUBST が格納されている TTEXT\_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 SUBSTID(基板 ID)を TTEXT\_DLIST 構造体に出すための関数です。

info->name\_list には NULL が設定されます。(定義名がありません。)

取得した情報の処理が終了した後、DshFreeTText\_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。

TTEXT\_DLIST 構造体は次のとおりです。

```
typedef struct{
    int         count;           // 取得できた ID 数
    char       **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
}TTEXT_DLIST;
```

### 3.15.2.39 EngGetSubstId() インデクスから基板 ID の取得

#### (1) 呼出書式

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

```
API int APIX EgnGetSubstId(
    int index,           // 基板情報のインデクス
    char *substid       // 取得した基板 ID を格納する領域のポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstId (
    ByVal index As Int32,
    ByVal substid As String) As Int32
```

##### [.NET C#]

```
int EngGetSubstId(
    int index,
    byte[] substid );
```

#### (2) 引数

index

基板情報のインデクスです。登録時に EngAllSubstInfo()関数によって与えられます。  
インデクスは、基板 ID から EngGetSubstIdIndex()関数で取得することができます。

substid

基板 ID 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	正常に取得できた。
(-1)	取得できなかった。(未登録)

#### (4) 説明

基板情報のインデクスから基板 ID を取得し、substid に格納します。  
正常に取得できた場合は関数戻り値として 0 が返却されます。

### 3.15.2.40 EngGetSubstIdIndex() 基板 ID からインデクスの取得

#### (1) 呼出書式

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

```
API int APIX EgnGetSubstIdIndex(
    char *substid,          // 基板 ID が格納されている領域のポインタ
    int *index              // 取得したインデクスを格納するための領域のポインタ
);
```

##### [.NET VB]

```
Function EngGetSubstIdIndex (
    ByVal substid As String,
    ByRef index As Int32) As Int32
```

##### [.NET C#]

```
int EngGetSubstIdIndex(
    byte[] substid,
    ref int index );
```

#### (2) 引数

substid

インデクスを取得したい対象の基板 ID が格納されている領域のポインタです。

index

取得したインデクスの値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	正常に取得できた。
(-1)	取得できなかった。(未登録)

#### (4) 説明

substid に指定される基板 ID から基板情報インデクスを取得するための関数です。

取得されたインデクスは index で指定された領域に格納されます。

正常に取得できた場合は関数戻り値として 0 が返却されます。

### 3.15.3 SUBST 基板関連ライブラリ関数

#### 3.15.3.1 DshFreeTSUBST\_INFO() - 基板情報構造体メモリの開放

(1) 呼出書式

[C, C++]

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

[.NET VB]

```
Sub DshFreeTSUBST_INFO (  
    ByRef info As dsh_info.TSUBST_INFO)
```

[.NET C#]

```
void DshFreeTSUBST_INFO(  
    ref TSUBST_INFO info );
```

(2) 引数

pinfo

メモリを解放したい基板情報構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

TSUBST\_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、TSUBST\_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。

### 3.15.3.2 DshCopyTSUBST\_INFO() - 基板情報構造体メモリのコピー

#### (1) 呼出書式

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

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

##### [.NET VB]

```
Function DshCopyTSUBST_INFO (
    ByRef info As dsh_info.TSUBST_INFO,
    ByRef sinfo As dsh_info.TSUBST_INFO) As Int32
```

##### [.NET C#]

```
int DshCopyTSUBST_INFO(
    ref TSUBST_INFO info,
    ref TSUBST_INFO sinfo);
```

#### (2) 引数

dinfo

基板情報のコピー先構造体メモリのポインタです。

sinfo

コピー元の基板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 TSUBST\_INFO 構造体内に格納されている基板情報を dinfo が指定する TSUBST\_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、DshFreeTSUBST\_INFO()関数を使って開放してください。

### 3.15.3.3 DshFreeTSUBST\_LOC\_HIST() - 通過ロケーション履歴構造体メモリの開放

(1) 呼出書式

**[C, C++]**

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

**[.NET VB]**

```
Sub DshFreeTSUBST_LOC_HIST (  
    ByRef pinfo As dsh_info.TSUBST_LOC_HIST)
```

**[.NET C#]**

```
void DshFreeTSUBST_LOC_HIST(  
    ref TSUBST_LOC_HIST pinfo );
```

(2) 引数

pinfo

メモリを解放したい通過ロケーション履歴構造体のポインタです。

(3) 戻り値

なし。

(4) 説明

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

### 3.15.3.4 DshCopyTSUBST\_LOC\_HIST() - 通過ロケーション履歴構造体メモリのコピー

#### (1) 呼出書式

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

```
API int APIX DshCopyTSUBST_INFO(
    TSUBST_LOC_HIST **dinfo,           // 北°-先のポインタ格納ポインタ
    TSUBST_LOC_HIST *sinfo            // 北°-元のポインタ
);
```

##### [.NET VB]

```
Function DshCopyTSUBST_LOC_HIST (
    ByRef dinfo As IntPtr,
    ByRef sinfo As dsh_info.TSUBST_LOC_HIST) As Int32
```

##### [.NET C#]

```
int DshCopyTSUBST_LOC_HIST(
    ref TSUBST_LOC_HIST dinfo,
    ref TSUBST_LOC_HIST sinfo );
```

#### (2) 引数

dinfo

通過ロケーション履歴のコピー先構造体メモリのポインタを格納するポインタです。

sinfo

コピー元の通過ロケーション履歴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 TSUBST\_LOC\_HIST 構造体内に格納されている通過ロケーション履歴を新しく確保したメモリの TSUBST\_LOC\_HIST 構造体に構造体ごとコピーします。

dinfo には、新しく確保したメモリのポインタを格納し返却します。



### 3.15.3.5 DshInitSubstInfo TSUBST\_INFO の初期設定

(1) 呼出書式

**[C, C++]**

```
API int APIX DshInitSubstInfo(
    TSUBST_INFO *info,           // TSUBST_INFO 構造体のポインタ
    char *substid,              // 基板 ID
    char *acquiredID,           // 読み込まれた ID
    char *lotID,                 // 基板の lot id
    char *subst_locID,          // 基板の現在 location
    char *source,                // 移動元位置
    char *destination,          // 移動先位置
    char *batch_locID,          // バッチの現在位置
    char *subst_pos_in_batch,    // バッチ内の位置
    int substIDstatus,           // 基板 ID の読み込み状態
    int material_status,         // 処理品質基板状態
    int subst_pro_state,         // 処理状態
    int subst_state,             // TRANSPORT 状態
    int subst_loc_state,         // 位置の状態
    int subst_usage,             // 基板の種類コード
    int subst_type               // 基板のタイプコード
);
```

**[.NET VB]**

```
Sub DshInitSubstInfo (
    ByRef info As dsh_info.TSUBST_INFO,
    ByVal substid As String,
    ByVal acquiredID As String,
    ByVal lotID As String,
    ByVal subst_locID As String,
    ByVal subst_source As String,
    ByVal subst_destination As String,
    ByVal batch_locID As String,
    ByVal subst_pos_in_batch As String,
    ByVal substIDstatus As Int32,
    ByVal material_status As Int32,
    ByVal subst_proc_state As Int32,
    ByVal subst_state As Int32,
    ByVal subst_loc_state As Int32,
    ByVal subst_usage As Int32,
    ByVal subst_type As Int32)

```

**[.NET C#]**

```
void DshInitSubstInfo(
    ref TSUBST_INFO info,
    byte[] substid,
    byte[] acquiredID,
    byte[] lotID,
```

```

byte[] subst_locID,
byte[] subst_source,
byte[] subst_destination,
byte[] batch_locID,
byte[] subst_pos_in_batch,
int substIDstatus,
int material_status,
int subst_proc_state,
int subst_state,
int subst_loc_state,
int subst_usage,
int subst_type );

```

## ( 2 ) 引数

info

TSUBST\_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。

substid

設定する基板 ID (文字列) です。

acquierdID

基板リーダーで読み込まれた基板 ID です。

lotID

基板のロット ID です。

subst\_locID

基板の現在 location (バッチプロセス装置以外)

source

基板の移動元 location

destination

基板の移動先 location

batch\_locID

バッチの現在 location (バッチプロセス装置)

subst\_pos\_in\_batch

バッチ内の現在の基板 location (バッチプロセス装置)

subst\_IDstatus

基板の ID 読み込み状態です。

material\_status

基板の処理品質基板状態です。

subst\_proc\_status

基板の処理状態です。

subst\_state

基板のトランスポート状態です。

subst\_loc\_state

基板の location 状態です。

subst\_usage

基板の種類コードです。

subst\_type

基板のタイプコードです。

## ( 3 ) 戻り値

なし。

#### (4) 説明

本関数は APP が OFFLINE で基板情報を生成する際に使用することができます。  
最初に info 内をクリアします。そして、引数で指定された情報を info 内に設定します。  
メモリが必要なメンバーについてはメモリを確保し情報をコピーします。  
なお、バッチ処理装置かどうかによって引数の値を NULL にする引数もあります。  
batch 名がついている引数はバッチプロセス装置用の引数です。

ユーザのシステムで不要な引数については、NULL または 0 を指定してください。

また、基板の移動履歴情報(TSUBST\_LOC\_HIST)の設定については次の DshPutSubstLocHist()関数を使ってください。この移動履歴情報のための領域を最大 64 個分予約してあります。

TSUBST\_INFO 構造体の情報を使用終了後は DshFreeTSUBST\_INFO()関数を使ってこの構造体内部で使用されたメモリを開放してください。

### 3.15.3.6 DshPutSubstLocHist() 基板移動履歴情報の追加

#### (1) 呼出書式

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

```
API int APIX DshPutSubstLocHist(
    TSUBST_INFO *info,           // 基板情報構造体のポインタ
    char *loc,                   // 移動ロケーション ID
    char *time_in,              // 入時刻
    char *time_out              // 出時刻
);
```

##### [.NET VB]

```
Function DshPutSubstLocHist (
    ByRef info As dsh_info.TSUBST_INFO,
    ByVal loc As String,
    ByVal time_in As String,
    ByVal time_out As String) As Int32
```

##### [.NET C#]

```
int DshPutSubstLocHist(
    ref TSUBST_INFO info,
    byte[] loc,
    byte[] time_in,
    byte[] time_out );
```

#### (2) 引数

info

移動履歴を加える基板情報構造体のポインタです。

loc

基板移動ロケーション ID です。

time\_in

指定されたロケーション ID 位置に入った時刻です。(YYYYMMDDhhmmsscc)

time\_out

指定されたロケーション ID 位置から出た時刻です。(YYYYMMDDhhmmsscc)

#### (3) 戻り値

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

#### (4) 説明

先に DshInitSubstInfo() で初期設定された基板情報構造体の hist が示す TSUBST\_LOC\_HIST に 1 個のロケーション移動情報を加えます。

予め最大 64 個予約された数の履歴リスト上には本関数が実行される順にスロット情報を追加していきます。60 個以内の追加であれば設定後 0 を返却します。

もし、64 個を超える場合は、設定しないで(-1)を返却します。

実際に設定された合計の履歴情報の数は TSUBST\_LOC\_HIST 内の hist\_count に保存されています。

### 3.15.3.7 DshInitSubstLocHist() 基板位置履歴情報の初期設定

#### (1) 呼出書式

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

```
API void APIX DshInitSubstLocHist(
    TSUBST_LOC_HIST *info,          // 基板履歴情報構造体のポインタ
    int count                       // 基板位置履歴数
);
```

##### [.NET VB]

```
Sub DshInitSubstLocHist (
    ByRef info As dsh_info.TSUBST_LOC_HIST,
    ByVal count As Int32)
```

##### [.NET C#]

```
void DshInitSubstLocHist(
    ref TSUBST_LOC_HIST info,
    int count);
```

#### (2) 引数

info

基板位置履歴情報を格納する構造体のポインタです。

count

格納する履歴数です。

#### (3) 戻り値

なし。

#### (4) 説明

info で指定される TSUBST\_LOC\_HIST 構造体を初期設定します。

info 内に格納する履歴数を count で指定します。本関数は、履歴メンバーに count 数分のポインタ配列領域を設けます。

TSUBST\_INFO 構造体内に対する履歴情報の設定は DshAddSubstLocHist() 関数を使用してください。

### 3.15.3.8 DshAddSubstLocHist() 基板位置履歴情報の追加設定

#### (1) 呼出書式

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

```
API int APIX DshAddSubstLocHist(
    TSUBST_LOC_HIST *info,           // 基板履歴情報構造体のポインタ
    char *loc,                       // 位置(名)
    char *time_inf,                 // 入時刻
    char *time_out                  // 出時刻
);
```

##### [.NET VB]

```
Function DshAddSubstLocHist (
    ByRef info As dsh_info.TSUBST_LOC_HIST,
    ByVal loc As String,
    ByVal time_in As String,
    ByVal time_out As String) As Int32
```

##### [.NET C#]

```
int DshAddSubstLocHist(
    ref TSUBST_LOC_HIST info,
    byte[] loc,
    byte[] time_in,
    byte[] time_out);
```

#### (2) 引数

info

基板位置履歴情報を格納する構造体のポインタです。

loc

ロケーション名です。

time\_in

入り時刻です。

timeout

出時刻です。

#### (3) 戻り値

戻り値	意味
0	正常に設定できた。
(-1)	info にはもう追加できなかった

#### (4) 説明

info で指定される TSUBST\_LOC\_HIST 構造体に履歴情報を追加します。

info 内に格納できる履歴数が既に満杯であった場合にはエラーを返します。

正常に追加できた場合は、0 を返します。