

付録A ClientManager SDK

本章では、ClientManager が提供するソフトウェア開発キット (SDK) について説明します。

ClientManager の CM マネージャをセットアップすると、(ESMPRO/CM マネージャホームディレクトリ)¥SDK ディレクトリの下に必要なファイルがコピーされます。各 SDK で提供する実行ファイル (ダイナミックリンクライブラリ : DLL) は、CM クライアントにセットアップされています。

SDK	内容
アラート SDK	アプリケーションプログラムから、アラート通報を CM マネージャに通知する。
ユーザ定義 SDK	ユーザ定義データの入力を、API を通して行う。

A.1 アラート SDK

提供する SDK を利用して開発したユーザアプリケーションプログラムはクライアント PC 上からアラートを CM マネージャに通報することができます。

通知するアラートタイプとしては、以下の 3 通りがあります。それぞれのアラートタイプによりオペレーションウィンドウ上のアイコン(クライアント PC に対応したアイコン)の色が変化します。

状態	アラートタイプ	オペレーションウィンドウ上のアイコンの色
エラー	ALERT_ERROR	赤色に変化
注意	ALERT_WARNING	黄色に変化
情報	ALERT_INFORMATION	緑色に変化

オペレーションウィンドウの設定により色は変更可能です。

■ ファイル

アラート API SDK が提供するファイルは、(ESMPRO/CM マネージャホームディレクトリ)¥SDK にあります。

¥SDK¥Include	ヘッダファイル(CMAPI.H)
¥BIN	DLL ファイル(CMAPI.DLL)
¥Lib	ライブラリファイル(CMAPI.LIB)

■ 注意事項

- (1) ClientManager 2.1 以降で提供する CMAPI.DLL は、ClientManager 2.0 以前のクライアント上では動作しません。
- (2) CMAPI.DLL を利用する場合には、CM クライアントのインストール時にアラートマネージャのインストールを選択する必要があります。

■ API リファレンス

関数一覧

名前	機能
SendAlert	アラートを送る。

SendAlert 関数

```
int WINAPI SendAlert (  
    int AlertType,  
    char *MessageString);
```

関数の説明

アラート通報を行う。

パラメータ

int AlertType

アラートタイプを指定する。アラートタイプは、以下のアラートタイプから選択する。

ALERT_ERROR (2)エラー

ALERT_WARNING (1)注意

ALERT_INFORMATION (0)情報

char *MessageString

アラートで通知するメッセージの文字列を指定する。転送する文字列長は、**MAX_MESSAGE_SIZE (256)**バイトまでである。

戻り値

正常にアラート通報が行えた場合は **FALSE** を返し、失敗した場合は、**TRUE** を返します。

解説

アラートメッセージ

SendAlert 関数によって通知されるアラートメッセージは、以下のような形式となります。

「CMクライアント管理ID :%1 において以下の事象が発生しました。¥n %2 ¥n」

%1 はクライアント管理 ID が自動的に格納され、**AlertSend** 関数のパラメータ (**char *MessageString**)で指定したメッセージ文字列が%2 の部分に格納されます。

A.2 ユーザ定義 SDK

提供する SDK を利用して、GUI ではなくアプリケーションアプリケーションプログラムよりユーザ定義情報を入力することができます。

■ ファイル

ユーザ定義 SDK が提供するファイルは、(CM マネージャホームディレクトリ)¥SDK にあります。

¥SDK¥Include	ヘッダファイル(CMUSRAPI.H)
¥BIN	DLL ファイル(CMUSRAPI.DLL)
¥Lib	ライブラリファイル(CMUSRAPI.LIB)

■ 注意事項

- (1) ClientManager 2.1 以降で提供する CMUSTAPI.DLL は、ClientManager 2.0 以降の CM クライアント上で動作します。
- (2) CM GUI を用いて、ユーザ定義データを収集する設定にしておく必要があります。

■ API リファレンス

関数一覧

名前	機能
CmuGetItemCount	アイテム数を取得する。
CmuGetItemType	指定アイテムのタイプを取得する。
CmuGetItemLength	指定アイテムのデータサイズを取得する。
CmuSetItemData	指定アイテムに 32 ビット符号付き整数値を設定する。
CmuSetItemString	指定アイテムに ASNI または DBCS コードで文字列を設定する。

CmuGetItemCount 関数

```
int WINAPI  
CmuGetItemCount();
```

関数の説明
アイテム数を得る。

パラメータ
なし

戻り値	
0 および負の数	エラー
正の数	アイテムの個数

解説
正の数が返却された場合、アイテム番号 1 から返却された値までが有効なアイテム番号となる。

CmuGetItemType 関数

```
int WINAPI  
CmuGetItemType(  
    int Index);
```

関数の説明

指定アイテムのデータのタイプを取得する。

パラメータ

int Index アイテムのデータのタイプ

戻り値

- 1 符号付き 32 ビット整数
- 2 文字列

解説

アイテムのタイプ番号が 1 (32 ビット整数) の場合には、**CmuSetItemData** で値を設定する。タイプ番号が 2 (文字列) の場合には、**CmuSetItemString** で値を設定する。

CmuGetItemLength 関数

```
int WINAPI  
CmuGetItemLength(  
    int Index);
```

関数の説明

指定アイテムのデータサイズを取得する。

パラメータ

int Index データ長さを求めるアイテム番号

戻り値

0 および負の値 エラー

正の値 長さ

CmuSetItemData 関数

BOOL WINAPI

```
CmuSetItemData(  
    int Index,  
    int Value);
```

関数の説明

指定アイテムに **32** ビット符号付き整数を設定する。

パラメータ

int Index	設定を行うアイテム番号
int Value	設定を行う値

戻り値

TRUE: 設定に成功した。

FALSE: 設定に失敗した。

CmuSetItemString 関数

BOOL WINAPI

```
CmuSetItemString(  
    int Index,  
    char *Value);
```

関数の説明

指定アイテムに、文字列を設定する。

パラメータ

int Index 設定を行うアイテム番号
char *Value 設定を行う文字列

戻り値

TRUE: 設定に成功した。

FALSE: 設定に失敗した。

解説

設定可能な最大文字数は、**CmuGetItemLength()**を利用して取得する。