

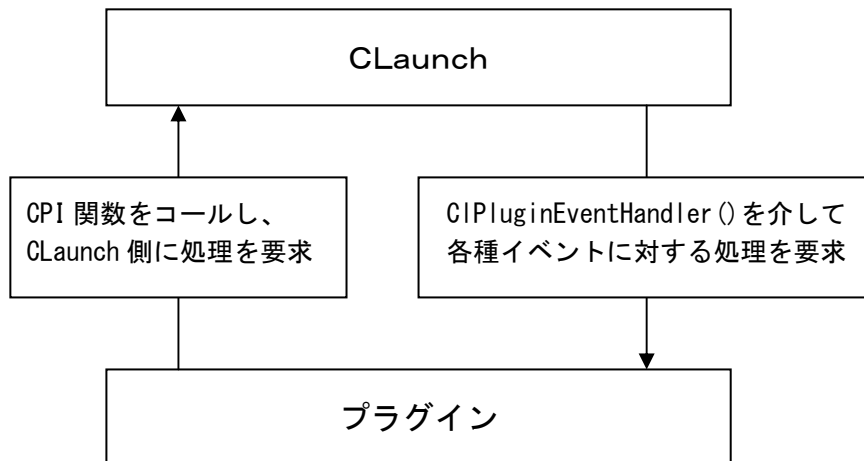
# CLaunch プラグイン仕様

改版履歴	
2005/04/16	第 1 版
2007/05/15	第 2 版
2009/04/09	第 3 版
2009/06/14	第 4 版
2010/11/11	第 5 版

## 1. プラグインの概要

はじめに、CLaunchプラグインの概要を説明します。

プラグインの実態は、通常のWin32 DLLファイルです。プラグインとCLaunch本体とのインターフェースは下図のようになっています。



CLaunch側からの処理要求及びイベントは、イベントハンドラ関数 `CIPluginEventHandler()` を通じて送られてきます。また、プラグイン側からCLaunchへの処理要求および情報の取得は、関数アドレステーブル `pCpiFunc` で渡されるCPI (CLaunch Plugin Interface) 関数によって行います。

以下に、CLaunchのオプションでプラグインがリストに追加される処理の流れを説明します。

- (1) CLaunchオプションのプラグインタブ初期化処理では、“Plugins”フォルダ内の\*.dllファイルを検索し、見つかったDLLファイルをロードします。
- (2) 正常にロードされた場合は、`CIPluginEventHandler()` の関数アドレスを取得し、プラグイン名取得要求 (`PRQ_GET_TITLE`) のための呼び出しを行います。ここでプラグインタイトルの文字列が取得できなかった場合は、プラグインリストに追加されません。

次に、選択されたプラグインが登録され、プラグイン側の機能が開始される時の流れを説明します。

- (1) CLaunchのオプションダイアログBOXでプラグインが選択されると、プラグインDLLがロードされ、`CIPluginEventHandler()` の関数アドレスを取得します。
- (2) プラグイン初期化要求 (`PRQ_INIT`) のために、`CIPluginEventHandler()` をコールします。ここでFALSEが返されると、プラグインのロードを中止します。
- (3) 正常にロードされた場合は、次のCLaunch起動時にも自動でロードされますが、このときにも上記の処理が実行されます。

プラグイン開発キットのソースには上記の流れに必要なプラグイン側の処理が含まれているため、そのままコンパイルすれば、プラグインとして登録できるようになっています。

## 2. イベント処理

イベントハンドラ関数 `CIPluginEventHandler()` は、下のように宣言されています。

```
BOOL CIPluginEventHandler(UINT uEvent, LONG_PTR lpParam1, LONG_PTR lpParam2)
```

第1引数の `uEvent` にはイベントの種類が設定され、第2、第3引数の `lpParam1`、`lpParam2` には、イベントの種類に応じて異なる値が渡されます。(Windowsのメッセージのようなイメージです。)

プラグイン開発キットのソースでは、全てのイベントに対して個別のハンドラ関数が呼び出される構造になっており、追加機能の実装は、これらの関数内に追加コーディングをするだけで済むようになっています。

以下では、各イベントの詳細について解説します。

### 2-1. プラグイン処理要求(イベント定数名 `PRQ_xxx`)

個々のプラグインに対して、`CLaunch`側から処理を要求します。

#### ◆ `PRQ_GET_TITLE`

イベント内容	プラグイン名取得要求
<code>lpParam1</code>	(LPSTR) プラグイン名文字列取得バッファ
<code>lpParam2</code>	NULL
プラグイン開発キットのハンドラ関数	
<code>void CIGetPluginTitle(LPSTR pszTitle)</code>	
説 明	
CLaunch オプションダイアログ BOX のプラグインタブが初期化されるときに呼び出されます。 <code>pszTitle</code> に長さ <code>MAX_PLUGIN_TITLE</code> 以内のタイトル文字列をコピーしてください。このイベントに 応答しない場合は、プラグインリストに追加されません。	

## ◆ PRQ\_INIT

イベント内容	プラグイン初期化要求
lpParam1	(int) プラグイン ID (－1 : 設定ダイアログ BOX 用の一時登録)
lpParam2	(LPVOID *)CPI 関数アドレステーブルのポインタ
プラグイン開発キットのハンドラ関数	
BOOL CPluginInit(int id, LPVOID *pvFunc)	
説 明	
<p>CLaunch オプションのプラグインリストで選択され、OK ボタンがクリックされた時、及び CLaunch の起動時に前回登録されていたプラグインがロードされるタイミングで呼び出されます。また CLaunch オプションでプラグインの設定ボタンがクリックされた時にも呼び出されます。</p> <p>プラグイン設定時の呼び出しでは id に－1 が渡され、その他の場合にはプラグイン ID (<math>\geq 0</math>) が渡されます。プラグイン ID は、CLaunch 側で各プラグインを識別するための値であり、CPI_REG_BUTTONITEM 等、特定の CPI 関数呼び出し時のパラメータとして使用するため、グローバル変数 iID に保存しておく必要があります。</p> <p>id が－1 の場合と、0 以上の場合とでは、以下のように処理を分けてください。</p> <ul style="list-style-type: none"> <li>・ id が－1 の時は、設定ダイアログ BOX の表示前に、固有のデータを初期化しておく必要のあるプラグインのための呼び出しです。登録が確定し、実行が開始されているわけではないので、プラグインの機能をスタートさせる必要はありません。</li> <li>・ id が 0 以上の場合は、プラグインの登録が確定しているため、実行開始要求となります。この時点でプラグインの機能をスタートさせてください。</li> </ul> <p>pvFunc には CPI 関数のポインタが格納された配列のアドレスが渡されます。この関数アドレスは以降の処理で、CLaunch 側機能を呼び出すために必要となるため、グローバル変数 pCpiFunc に保存しておく必要があります。</p>	

## ◆ PRQ\_EXIT

イベント内容	プラグイン終了要求
lpParam1	(int) プラグイン ID (－1 : 設定ダイアログ BOX 用の一時登録)
lpParam2	NULL
プラグイン開発キットのハンドラ関数	
BOOL CPluginExit(int id)	
説 明	
<p>CLaunch 終了時、および CLaunch オプションでプラグインの選択が解除されたときに呼び出されます。このイベント呼び出し後にプラグイン DLL がアンロードされるため、使用したメモリの解放等を行ってください。</p> <p>また、プラグインの設定ダイアログ BOX を終了した後にも呼び出されますが、この時の id は－1 となります。</p>	

#### ◆ PRQ\_ID\_CHANGE

イベント内容	プラグイン ID 変更要求
lpParam1	(int) プラグイン ID
lpParam2	NULL
プラグイン開発キットのハンドラ関数	
なし (CPluginEventHandler () 内で直接処理)	
説 明	
<p>プラグイン初期化要求 (PRQ_INIT) で保存したプラグイン ID は、他のプラグインの登録・解除によって変更される場合があります。ID 変更時にはこのイベントが発生します。このイベントの処理要求を受けた場合には、グローバル変数に保存されているプラグイン ID を更新し、以降の CPI 関数呼び出しには、新しい ID を使用しなければなりません。</p>	

#### ◆ PRQ\_INIT\_ITEM

イベント内容	プラグインアイテムデータ初期化要求
lpParam1	(LPBTNINFO) 初期化対象ボタン情報のポインタ
lpParam2	NULL
プラグイン開発キットのハンドラ関数	
BOOL CPluginInitItem(LPBTNINFO pBI)	
説 明	
<p>ボタン登録タイプのプラグインが、ボタンへ登録される時に呼び出されます。pBI でポインタが渡されるボタン情報 (BTNINFO 構造体) は、ボタンに登録されている全てのアイテムが個々に保有するデータであり、プラグインアイテムの登録ボタンへも割り当てられます。BTNINFO 構造体のメンバー変数 szParam、szDir、iSW、uFlags の 4 つは、プラグイン固有の設定記憶用に使用できます。ただし、これらの変数はボタン登録時に 0 クリアされているため、特に初期値を設定する必要がない場合、このイベントを処理する必要はありません。</p> <p>BTNINFO 構造体の内容は、プラグイン開発キットに添付されているヘッダファイル "CPlugin.h" を参照してください。</p>	

#### ◆ PRQ\_EXECUTE

イベント内容	プラグインアイテム実行要求
lpParam1	(LPBTNINFO) 実行するアイテムのボタン情報のポインタ
lpParam2	(UINT) 起動オプションフラグ (0x00000001 = ボタン長押しで起動)
プラグイン開発キットのハンドラ関数	
BOOL CIPuginExecute(LPBTNINFO pBI, UINT uOption)	
説 明	
<p>プラグインアイテムを登録したボタンがクリックされた時に呼び出されます。必要に応じて pBI のボタン情報を参照し、プラグインアイテムの処理を実行してください。uOption の最下位ビットは、ボタンの長押しで起動された場合にセットされるため、通常押しと長押しで処理を分けたい場合に使用してください。</p> <p>BTNINFO 構造体の内容は、プラグイン開発キットに添付されているヘッダファイル "CIPugin.h" を参照してください。</p>	

#### ◆ PRQ\_ITEM\_PROP

イベント内容	プラグインアイテムのプロパティダイアログ表示要求
lpParam1	(HWND) ボタンプロパティダイアログ BOX のハンドル
lpParam2	(LPBTNINFO) 設定対象ボタン情報のポインタ
プラグイン開発キットのダイアログ関数	
LRESULT CALLBACK CIPuginItemPropDlg(HWND hDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)	
説 明	
<p>CLaunch 側のボタンのプロパティダイアログ BOX で、プラグインアイテムのプロパティボタンがクリックされたときに呼び出されます。lpParam1 で渡されるウィンドウハンドルを親ウィンドウにして、プラグインアイテムのプロパティダイアログ BOX を表示してください。</p> <p>このダイアログ BOX では、ボタンに登録された個々のプラグインアイテムに対する設定を行います。WM_INITDIALOG の lParam にはプロパティ設定の対象ボタンに対応する BTNINFO 構造体のポインタが渡されるので、必要に応じて個別設定データを元にダイアログ BOX の初期化を行ってください。また、ダイアログ BOX で設定が変更された場合には、BTNINFO 構造体のメンバー変数 szParam、szDir、iSW、uFlags に変更内容を反映してください。</p> <p>BTNINFO 構造体の内容は、プラグイン開発キットに添付されているヘッダファイル "CIPugin.h" を参照してください。</p>	

#### ◆ PRQ\_SETTING

イベント内容	プラグイン設定ダイアログ表示要求
lpParam1	(HWND) ボタンプロパティダイアログ BOX のハンドル
lpParam2	NULL
プラグイン開発キットのダイアログ関数	
LRESULT CALLBACK CIPluginSettingDlg(HWND hDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)	
説 明	
<p>CLaunch オプションのプラグインタブで、プラグインの設定ボタンがクリックされた時に呼び出されます。lpParam1 で渡されるウィンドウハンドルを親ウィンドウにして、プラグインの設定ダイアログ BOX を表示してください。</p> <p>このダイアログ BOX では、プラグイン全体のオプション設定を行います。</p>	

#### ◆ PRQ\_ABOUT

イベント内容	プラグインバージョン情報ダイアログ表示要求
lpParam1	(HWND) ボタンプロパティダイアログ BOX のハンドル
lpParam2	NULL
プラグイン開発キットのダイアログ関数	
LRESULT CALLBACK CIAboutPluginDlg(HWND hDlg, UINT uMsg, WPARAM wParam, LPARAM lParam)	
説 明	
<p>CLaunch オプションのプラグインタブで、バージョン情報ボタンがクリックされた時に呼び出されます。lpParam1 で渡されるウィンドウハンドルを親ウィンドウにして、プラグインのバージョン情報ダイアログ BOX を表示してください。</p>	

## 2-2. CLaunchイベント(イベント定数名 PEV\_xxx)

CLaunchの実行中に発生する様々なイベントに対応して呼び出されます。

lpParam1には常にCLaunchステータスフラグが渡されます。

CLaunchの各状態とフラグビットとの対応は下表の通りです。

### CLaunchステータスフラグ

定数名	フラグビット	フラグの内容
CS_DISPMODE	0x00000001	表示モード (0:Mode1 1:Mode2)
CS_LOCK	0x00000002	ウィンドウロック中
CS_RESTRAINT	0x00000004	抑止モード中

### ◆ PEV\_SHOW

イベント内容	CLaunch 表示イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	NULL
プラグイン開発キットのハンドラ関数	
void C1OnShowWindow(DWORD dwStatus)	
説 明	
CLaunch が表示された時に呼び出されます。	

### ◆ PEV\_HIDE

イベント内容	CLaunch 非表示イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	NULL
プラグイン開発キットのハンドラ関数	
void C1OnHideWindow(DWORD dwStatus)	
説 明	
CLaunch が非表示化された時に呼び出されます。	



#### ◆ PEV\_PRE\_EXECUTE

イベント内容	アイテム起動前イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	上位ワード：起動アイテムを含むページの番号 下位ワード：起動アイテムのボタン番号
プラグイン開発キットのハンドラ関数	
void C1OnPreExecute(DWORD dwStatus, int iPage, int iButton)	
説 明	
<p>ボタンに登録されているアイテムが起動される前に呼び出されます。iButton、iPage には、起動されようとするボタンの番号と、そのボタンを含むページの番号が渡されます。これらの値をボタン情報取得関数「CpiGetButtonInfo()」へ渡すことで、ボタン情報を取得できます。</p> <p>具体的には下のような記述になります。</p> <pre>LPBTNINFO pbi = CpiGetButtonInfo(iPage, iButton); MessageBox(NULL, pbi-&gt;szName, NULL, MB_OK);</pre> <p>iPage はページの表示順ですが、iButton はボタンの登録順となっているため注意してください。</p>	

#### ◆ PEV\_EXECUTE

イベント内容	アイテム起動後イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	上位ワード：起動アイテムを含むページの番号 下位ワード：下位ワード：起動アイテムのボタン番号
プラグイン開発キットのハンドラ関数	
void C1OnExecute(DWORD dwStatus, int iPage, int iButton)	
説 明	
<p>ボタンに登録されているアイテムが起動された後に呼び出されます。iButton、iPage には、起動されたボタンの番号と、そのボタンを含むページの番号が渡されます。起動されたアイテムのボタン情報の取得方法は、アイテム起動前イベント(PEV_PRE_EXECUTE)の説明を参照してください。</p>	

◆ PEV\_PAGE\_CHANGE

イベント内容	ページ切替えイベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	(int) 新しく表示されたページの番号
プラグイン開発キットのハンドラ関数	
void C1OnPageChange (DWORD dwStatus, int iPage)	
説 明	
CLaunch の表示ページが切り替えられた時に呼び出されます。iPage には新しく表示されたページの番号が渡されます。	

◆ PEV\_TAB\_HOVER

イベント内容	タブのホバー状態変化イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	(int) ホバー状態になったページの番号 (－1 : ホバー状態 OFF)
プラグイン開発キットのハンドラ関数	
void C1OnTabHover (DWORD dwStatus, int iPage)	
説 明	
ページタブ上でマウスが移動し、タブのホバー状態が変化した時に呼び出されます。iPage にはホバー状態になったタブに対応するページ番号が渡されます。iPage が－1 のときは、ホバー状態が解除され、現在ホバー状態のタブが無いことを示します。	

#### ◆ PEV\_BUTTON\_HOVER

イベント内容	ボタンのホバー状態変化イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	(int) ホバー状態になったボタンの位置番号 (－ 1 : ホバー状態 OFF)
プラグイン開発キットのハンドラ関数	
void C1OnButtonHover (DWORD dwStatus, int iBtnPos)	
説 明	
<p>ボタン上でのマウス移動、又はカーソルキーの操作によって、ボタンのホバー状態が変化した時に呼び出されます。iBtnPos にはホバー状態になったボタンの位置番号が渡されます。ボタンの位置番号は左上隅のボタンを 0 とした値で、表示ボタン情報取得関数「CpiGetDispBtnInfo()」へ渡すことにより、ボタン情報を取得できます。ただし、空白ボタンに対応する位置番号を指定した場合、NULL ポインタが返されるので、注意してください。</p> <p>具体的には下のような記述になります。</p> <pre> LPBTNINFO pbiDisp = CpiGetDispBtnInfo(iBtnPos); if (pbi)           // 登録ボタン? {     MessageBox(NULL, pbi-&gt;szName, NULL, MB_OK); } </pre> <p>iBtnPos が－ 1 のときは、ホバー状態が解除され、現在ホバー状態のボタンが無いことを示します。</p>	

#### ◆ PEV\_ADD\_PAGE

イベント内容	ページ追加イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	(int) 追加されたページの番号
プラグイン開発キットのハンドラ関数	
void C1OnAddPage (DWORD dwStatus, int iPage)	
説 明	
ページが追加された後に呼び出されます。iPage には追加されたページの番号が渡されます。	

#### ◆ PEV\_DEL\_PAGE

イベント内容	ページ削除イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	(int) 削除されるページの番号
プラグイン開発キットのハンドラ関数	
void ClOnDelPage(DWORD dwStatus, int iPage)	
説 明	
ページが削除される前に呼び出されます。iPage には削除されるページの番号が渡されます。	

#### ◆ PEV\_ADD\_BUTTON

イベント内容	ボタン追加イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	上位ワード：ボタンが追加されたページの番号 下位ワード：追加されたボタンの番号
プラグイン開発キットのハンドラ関数	
void ClOnAddButton(DWORD dwStatus, int iPage, int iButton)	
説 明	
ボタンが追加された後に呼び出されます。iPage にはボタンが追加されたページ、iButton には追加されたボタンの番号が渡されます。	

#### ◆ PEV\_DEL\_BUTTON

イベント内容	ボタン削除イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	上位ワード：削除されるボタンを含むページの番号 下位ワード：削除されるボタンの番号
プラグイン開発キットのハンドラ関数	
void ClOnDelButton(DWORD dwStatus, int iPage, int iButton)	
説 明	
ボタンが削除される前に呼び出されます。iPage には削除されるボタンを含むページ、iButton には削除されるボタンの番号が渡されます。	

#### ◆ PEV\_MOVE\_PAGE

イベント内容	ページ移動イベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	上位ワード：移動元ページ位置 下位ワード：移動先ページ位置
プラグイン開発キットのハンドラ関数	
void C1OnMovePage(DWORD dwStatus, int iSrc, int iDst)	
説 明	
<p>ページが移動される前に呼び出されます。iSrc には移動元のページ位置、iDst には移動先となるページ位置の番号が渡されます。移動先のページ位置は、移動元のページ位置が抜ける前の値であることに注意してください。「移動元 &lt; 移動先」である場合、移動した後のページ位置は iDst-1 となります。</p>	

#### ◆ PEV\_MOVE\_BUTTON

イベント内容	ボタン移動イベント(ページ外へのボタン移動のみ)
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	上位ワード：移動元ページの番号 下位ワード：移動元ボタンの番号
プラグイン開発キットのハンドラ関数	
void C1OnMoveButton(DWORD dwStatus, int iPage, int iButton)	
説 明	
<p>ボタンが他のページへ移動される前に呼び出されます。iPage には移動元のページ、iButton には移動されるボタンの移動元ページ内での番号が渡されます。</p> <p>移動先は現在表示されているページで、ページ番号は、表示ページ取得関数「CpiGetCurrentPage()」から取得できます。</p>	

#### ◆ PEV\_WIN\_LOCK

イベント内容	ウィンドウロック切替えイベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	NULL
プラグイン開発キットのハンドラ関数	
void ClOnWindowLock (DWORD dwStatus)	
説 明	
<p>ウィンドウロックモードのON/OFF時に呼び出されます。ロックされたか、解除されたかは、CLaunch ステータスフラグで判断できます。</p> <pre> if (dwStatus &amp; CS_LOCK)      // ロックされた？     .... </pre>	

#### ◆ PEV\_RESTRAINT\_MODE

イベント内容	抑止モード切替えイベント
lpParam1	(DWORD) CLaunch ステータスフラグ
lpParam2	NULL
プラグイン開発キットのハンドラ関数	
void ClOnRestraintMode (DWORD dwStatus)	
説 明	
<p>抑止モードに切り替わった時、または解除された時に呼び出されます。抑止モードのON/OFF状態は、CLaunch ステータスフラグで判断できます。</p> <pre> if (dwStatus &amp; CS_RESTRAINT) // 抑止モード中？     .... </pre>	

### 3. CPI (CLaunch Plugin Interface) 関数

プラグイン側からCLaunchへの処理要求はCPI関数を通じて行います。CPI関数のアドレスは、プラグイン初期化要求 (PRQ\_INIT) のハンドラ関数において、グローバル変数pCpiFuncに保存されており、この配列要素をインデックス指定して関数を呼び出します。ただし、プラグイン開発キットのヘッダファイルには関数名が定義されているため、コーディングは関数名での記述となります。

以下では、各CPI関数の詳細について解説します。

#### ◆ CpiGetVersion

関数の機能	CLaunch のバージョンを取得する	
関数書式	DWORD CpiGetVersion(void)	
	引 数	なし
	リターン値	CLaunch のバージョン
説 明		
CLaunch のバージョンを DWORD 値で返します。プラグインが対応可能なバージョンであるか否かのチェックに使用してください。返される値は、Version 2.78 であれば 0x02070800 となります。		
使用例		
DWORD dwVersion = CpiGetVersion();		

### ◆ CpiShowWindow

関数の機能	CLaunch を表示する
関数書式	void CpiShowWindow(UINT uPos, int x, int y, int iPage)
uPos	CLaunch の表示位置指定フラグ
x	表示座標指定時の X 座標
y	表示座標指定時の Y 座標
iPage	初期表示ページ指定 (PN_MEMORY : 前回のページ)
リターン値	なし
説 明	
<p>CLaunch を表示します。API 関数の ShowWindow() で直接表示させることは避け、必ずこの関数を使用するようにしてください。</p> <p>uPos には初期表示位置として、WPOS_MOUSE_CENTER(マウスポインタの周囲)、WPOS_MOUSE_LEFTTOP(マウスポインタの右下)、WPOS_NOMOVE(現在位置)、WPOS_CENTER(画面中央)、WPOS_SPECIFY(座標指定)の何れか1つを選択してセットします。また、WPOS_AUTO_HIDE(自動非表示フラグ)を OR で付加すると、表示後にマウスポインタがウィンドウ外へ出たとき、自動的に非表示となります。</p> <p>x、y は uPos で WPOS_SPECIFY を選択した場合のみ、表示位置をスクリーン座標で指定してください。それ以外の場合は無視されます。</p>	
使用例	
<pre>POINT pt; GetCursorPos(&amp;pt); CpiShowWindow(WPOS_SPECIFY, pt.x, pt.y, PN_MEMORY);</pre>	

### ◆ CpiHideWindow

関数の機能	CLaunch を非表示にする
関数書式	void CpiHideWindow(void)
引 数	なし
リターン値	なし
説 明	
<p>CLaunch を非表示にします。API 関数の ShowWindow() で直接非表示化することは避け、必ずこの関数を使用するようにしてください。</p>	
使用例	
<pre>CpiHideWindow();</pre>	



### ◆ CpiGetHandle

関数の機能	CLaunch のインスタンスハンドル、ウィンドウハンドルを取得する
関数書式	void CpiGetHandle(HINSTANCE *phInst, HWND *phWnd)
*phInst	インスタンスハンドル取得ポインタ
*phWnd	ウィンドウハンドル取得ポインタ
リターン値	なし
説 明	
CLaunch のインスタンスハンドル、及びウィンドウハンドルを取得します。CLaunch に直接メッセージを送信する必要がある場合等に利用してください。 一方のハンドルのみ必要な場合、他方のパラメータは NULL でも構いません。	
使用例	
<pre>HINSTANCE    hInstance; HWND         hWnd; CpiGetHandle(&amp;hInstance, &amp;hWnd);</pre>	

### ◆ CpiGetPageNum

関数の機能	ページ数を取得する
関数書式	int CpiGetPageNum(void)
引 数	なし
リターン値	登録ページ数
説 明	
CLaunch に現在登録されているページの数进行返します。	
使用例	
<pre>int nPage = CpiGetPageNum();</pre>	

### ◆ CpiGetCurrentPage

関数の機能	現在のページ番号を取得する
関数書式	int CpiGetCurrentPage(void)
引 数	なし
リターン値	現在のページ番号 (1 ページ目が 0)
説 明	
現在表示しているページの番号を返します。	
使用例	
<pre>int iCurPage = CpiGetCurrentPage();</pre>	

### ◆ CpiGetPageInfo

関数の機能	ページ情報を取得する
関数書式	LPPAGEINFO CpiGetPageInfo(int iPage)
iPage	ページ番号 (1 ページ目が 0)
リターン値	ページ情報構造体のポインタ
説 明	
指定ページの情報を、ページ情報構造体 (PAGEINFO) のポインタで返します。	
使用例	
<pre>LPPAGEINFO  pinf = CpiGetPageInfo(0);           // 1 ページ目のページ情報を取得 MessageBox(NULL, pinf-&gt;szPage, NULL, MB_OK);    // ページ名を表示</pre>	

### ◆ CpiGetButtonInfo

関数の機能	指定ページ、指定ボタンのボタン情報を取得する	
関数書式	LPBTNINFO CpiGetButtonInfo(int iPage, int iBtn)	
	iPage	ページ番号（1 ページ目が0）
	iBtn	ボタン番号（1 番目の登録ボタンが0）
	リターン値	ボタン情報のポインタ
説 明		
<p>指定ページ、指定ボタンの情報を、ボタン情報構造体 (BTNINFO) のポインタで返します。パラメータのページ番号はページの表示順と一致していますが、ボタン番号はボタンが登録された順番 (BTNINFO 構造体の iNumber メンバー変数の値) となっているため、表示順とは一致しません。</p> <p>BTNINFO 構造体の内容は、プラグイン開発キットに添付されているヘッダファイル "CIPlugin.h" を参照してください。</p>		
使用例		
<pre>LPBTNINFO pbi = CpiGetButtonInfo(2, 4);           // 3 ページ目、ボタン5の情報取得 MessageBox(NULL, pbi-&gt;szName, NULL, MB_OK);      // ボタン名を表示</pre>		

### ◆ CpiGetDispBtnInfo

関数の機能	表示ページ内、指定位置のボタン情報を取得する	
関数書式	LPBTNINFO CpiGetDispBtnInfo(int iPos)	
	iPos	ボタン位置番号（左上のボタンが0）
	リターン値	ボタン情報のポインタ
説 明		
<p>現在表示されているページ内の、指定位置にあるボタンの情報を、ボタン情報構造体 (BTNINFO) のポインタで返します。空白ボタンに対応する位置番号を指定した場合、NULL ポインタが返されるので、注意してください。</p> <p>BTNINFO 構造体の内容は、プラグイン開発キットに添付されているヘッダファイル "CIPlugin.h" を参照してください。</p>		
使用例		
<pre>LPBTNINFO pbi = CpiGetDispBtnInfo(4);             // 5 番目のボタンの情報取得 if (pbi)   // 登録ボタン? {     MessageBox(NULL, pbi-&gt;szName, NULL, MB_OK);    // ボタン名を表示 }</pre>		

## ◆ CpiSetCaption

関数の機能	キャプションを変更する
関数書式	void CpiSetCaption(int id, LPCSTR pszCaption, BOOL bSave)
id	プラグイン ID
pszCaption	表示文字列のポインタ
bSave	保存指定フラグ (TRUE:CLaunch の設定に保存 FALSE:一時的に変更)
リターン値	なし
説 明	
<p>キャプションの表示文字列を変更します。最大文字数は MAX_CAPTION で定義されています。</p> <p>bSave が TRUE の場合、pszCaption の内容は CLaunch 側のバッファにコピーされるため、呼び出し後にバッファを解放しても問題ありません。また、次回起動時にも復元されます。</p> <p>bSave が FALSE の場合、pszCaption の内容はコピーされません。従って文字列用のバッファは、プラグイン側で静的配列を確保し、呼び出し後も解放されないようにして下さい。</p>	
使用例	
<pre>static char szTitle[MAX_CAPTION+1]; lstrcpy(szTitle, "キャプション"); CpiSetCaption(iID, szTitle, FALSE);</pre>	

## ◆ CpiRegisterButtonItem

関数の機能	プラグインをボタンアイテムとして登録する
関数書式	void CpiRegisterButtonItem(int id, LPCSTR pszRegName)
id	プラグイン ID
pszRegName	プラグインアイテム名のポインタ
リターン値	なし
説 明	
<p>ボタン登録タイプのプラグインを、プラグインアイテムとして CLaunch へ登録します。</p> <p>id には、プラグイン初期化要求 (PRQ_INIT) で受け取ったプラグイン ID を指定します。</p> <p>アイテム登録後は、CLaunch 側のボタン登録ダイアログ BOX で、特殊アイテムのドロップダウンリストに、pszRegName で渡したプラグインアイテム名が表示されるようになります。</p>	
使用例	
<pre>CpiRegisterButtonItem(iID, "プラグインアイテム");</pre>	

### ◆ CpiChangePage

関数の機能	指定ページに切替える
関数書式	void CpiChangePage(int iPage)
iPage	表示させるページ番号（1 ページ目が 0）
リターン値	なし
説 明	
CLaunch の表示ページを指定したページに切り替えます。	
使用例	
<pre>CpiChangePage(3);           // 4 ページ目に切り替え</pre>	

### ◆ CpiSelectButton

関数の機能	指定位置のボタンを選択する
関数書式	void CpiSelectButton(int iBtnPos)
iBtnPos	ボタン位置番号（左上のボタンが 0）
リターン値	なし
説 明	
<p>現在表示されているページ内の、指定位置にあるボタンを選択状態（ホバー状態）にします。</p> <p>ボタンの表示位置番号は、BTNINFO 構造体の iPos メンバー変数から取得することができます。</p>	
使用例	
<pre>CpiSelectButton(5);         // 6 個目のボタンを選択</pre>	

### ◆ CpiExecuteItem

関数の機能	指定アイテムを起動する
関数書式	void CpiExecuteItem(int iPage, int iButton)
iPage	起動させるアイテムのページ番号（1 ページ目が0）
iButton	起動させるアイテムのボタン番号（1 番目の登録ボタンが0）
リターン値	なし
説 明	
<p>指定ページの、指定ボタンに登録されているアイテムを起動します。</p> <p>パラメータのボタン番号は、表示位置番号ではないので注意してください。（CpiGetButtonInfo() に渡すパラメータと同様です）</p>	
使用例	
<pre>CpiExecuteItem(2, 3);           // 3 ページ目、ボタン4 のアイテムを起動</pre>	

### ◆ CpiCheckStatus

関数の機能	CLaunch のステータを取得する
関数書式	BOOL CpiCheckStatus(UINT uBit)
uBit	フラグビット
リターン値	TRUE:対象ステータスは ON FALSE: 対象ステータスは OFF
説 明	
<p>指定項目について、CLaunch の現在の動作状態をチェックし、BOOL 値で返します。</p> <p>パラメータのフラグビットは、プラグイン開発キットに添付されているヘッダファイル "CIPlugin.h" で定義されています。</p>	
使用	
<pre>if (CpiCheckStatus(CS_LOCK))      // ウィンドウロック中？ {     .... }</pre>	

### ◆ CpiCheckSetting

関数の機能	CLaunch の動作オプション設定を取得する
関数書式	BOOL CpiCheckSetting(UINT uBit)
uBit	オプションビット
リターン値	TRUE: 指定オプションは有効設定 FALSE: 無効設定
説 明	
CLaunch オプションのうち、動作全般に関する項目の設定状態をチェックし、BOOL 値で返します。パラメータのオプションビットは、プラグイン開発キットに添付されているヘッダファイル "CIPlugin.h" で定義されています。	
使用例	
<pre> if (CpiCheckSetting(OPT_INV_PAGE_DIR))          // ページ切替え動作反転設定？ {     ..... } </pre>	

### ◆ CpiCheckDispSetting

関数の機能	CLaunch の表示関連オプション設定を取得する
関数書式	BOOL CpiCheckDispSetting(UINT uBit)
uBit	フラグビット
リターン値	TRUE: 指定オプションは有効設定 FALSE: 無効設定
説 明	
CLaunch オプションのうち、画面表示に関する項目の設定状態をチェックし、BOOL 値で返します。パラメータのオプションビットは、プラグイン開発キットに添付されているヘッダファイル "CIPlugin.h" で定義されています。	
使用例	
<pre> if (CpiCheckDispSetting(DSP_TOPMOST))          // 最前面表示設定？ {     ..... } </pre>	

## 4. プラグイン作成時のポイント

### 4-1. 設定の保存

ボタン登録タイプのプラグインでは、ボタン登録時に割り当てられたBTNINFO構造体のメンバー変数 szParam、szDir、iSW、uFlagsに、ボタンごとの固有設定を保持させることができます。これらのデータはCLaunch側でセーブ、ロードが行われるため、プラグイン側での保存処理は不要です。また、設定バックアップの対象となります。

一方、プラグインの全体設定については、各プラグインが独自処理で保存する必要があり、設定バックアップの対象にはなりません。

### 4-2. ボタンに対応したイベントを処理するプラグイン

例えば、あるボタンに登録されたアイテムが起動されたとき、他のボタンのアイテムを同時起動させるようなプラグインを作ろうとした場合、トリガとなるボタンと、同時起動させるボタンのページ番号、ボタン番号をプラグイン側で記憶しておく必要があります。しかしCLaunch側でボタンやページが移動又は削除されると、記憶していたページ番号、ボタン番号が目的のアイテムと一致しくなくなります。そのためこのようなプラグインでは、PEV\_DEL\_PAGE、PEV\_DEL\_BUTTON、PEV\_MOVE\_PAGE、PEV\_MOVE\_BUTTONの4イベントが発生したときに、プラグイン側で管理しているページ番号やボタン番号を更新しなければなりません。