

# Memory Access Controller INpMac

## -取扱説明書-

Rev.14  
2020/01/23

### 目次

<b>1. 概要</b>	<b>2</b>	<b>4. OCXについて</b>	<b>29</b>
1-1. Windowsアプリケーション間通信機能(INpMacWinAPI)	2	4-1. INpMacWinコントロール	29
1-2. INtimeアプリケーション間通信機能(INpMacRtAPI)	2	(1) プロパティ	29
1-3. INplc間通信機能(INpMacAPI)	3	(2) メソッド	29
(1) I/Oイメージへのアクセス	3	(3) イベント	30
(2) 共有メモリ領域へのアクセス	3	4-2. INpMacコントロール	30
<b>2. インストール</b>	<b>4</b>	(1) プロパティ	30
2-1. システム要件	4	(2) メソッド	31
(1) Windowsアプリケーション間通信機能(INpMacWinAPI)	4	(3) イベント	31
(2) INtimeアプリケーション間通信機能(INpMacRtAPI)	4	<b>5. エラーコード一覧</b>	<b>32</b>
(3) INplc間通信機能(INpMacAPI)	4	5-1. INpMacWinAPI	32
(4) 開発環境	4	5-2. INpMacRtAPI	32
2-2. インストールについて	4	5-3. INpMacAPI	32
(1) はじめに	4	<b>6. サンプルプログラム</b>	<b>33</b>
(2) インストール	4	6-1. フォルダ構成	33
(3) 旧INpMacインストール環境へのインストール	6	6-2. プログラムの内容	33
2-3. アンインストールについて	6	(1) DLLサンプルの概要	34
2-4. ACTIVE TOUCH環境の場合	7	(2) OCXサンプルの概要	34
(1) はじめに	7	<b>7. 使用例</b>	<b>35</b>
(2) インストール	7	7-1. Windows用共有メモリのダンプ	35
2-5. アンインストールについて	7	7-2. INplcシステムを介したエコー動作テスト	36
<b>3. APIについて</b>	<b>8</b>	(1) PLCプログラムの設定	36
3-1. INpMacWinAPI	8	(2) IEリアヘデータを書込む	38
(1) INpMacWin_CreateMemory()	8	(3) Qエリアのデータを取得	38
(2) INpMacWin_DeleteMemory()	9	7-3. HMIソフト『ACTIVE TOUCH』での使用方法	40
(3) INpMacWin_ReadMemory()	9	(1) コントロール(部品)の登録	40
(4) INpMacWin_WriteMemory()	10	(2) データの関連付け	41
(5) INpMacWin_MapMemory()	11	<b>8. 更新履歴</b>	<b>43</b>
(6) INpMacWin_UnMapMemory()	12		
(7) INpMacWin_GetSize()	12		
(8) INpMacWin_CreateMemoryEx()	13		
(9) INpMacWin_DeleteMemoryEx()	13		
(10) INpMacWin_ReadMemoryEx()	14		
(11) INpMacWin_WriteMemoryEx()	15		
(12) INpMacWin_MapMemoryEx()	16		
(13) INpMacWin_UnMapMemoryEx()	17		
(14) INpMacWin_GetSizeEx()	17		
3-2. INpMacRtAPI	18		
(1) INpMacRt_ReadMemory()	18		
(2) INpMacRt_WriteMemory()	19		
(3) INpMacRt_GetSize()	20		
(4) INpMacRt_ReadMemoryEx()	21		
(5) INpMacRt_WriteMemoryEx()	22		
(6) INpMacRt_GetSizeEx()	23		
3-3. INpMacAPI	24		
(1) INpMac_Get_Qarea_Size()	24		
(2) INpMac_Get_Qarea()	25		
(3) INpMac_Get_Iarea_Size()	26		
(4) INpMac_Set_Iarea()	26		
(5) INpMac_Get_Marea_Size()	27		
(6) INpMac_Get_Marea()	27		
(7) INpMac_Set_Marea()	28		

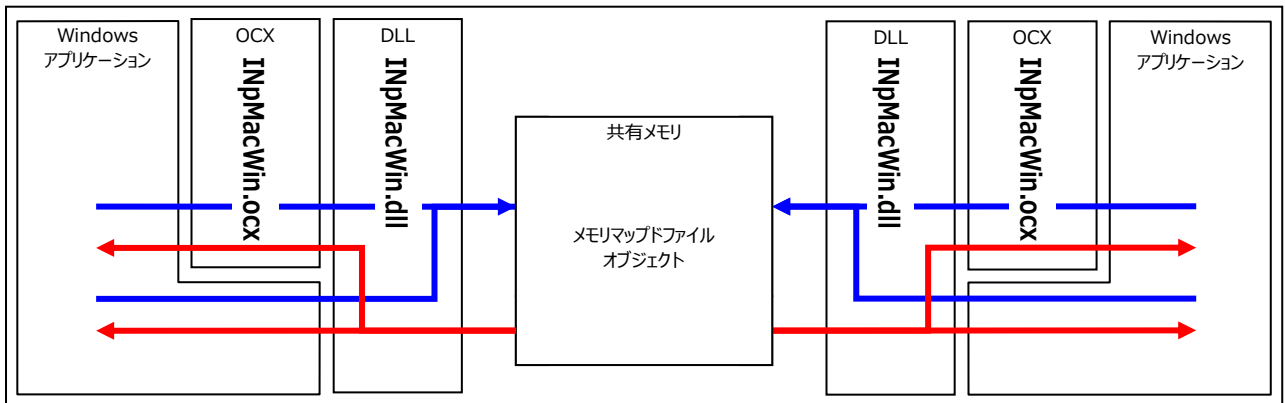
# 1. 概要

本製品(INpMac)は、Windowsアプリケーション向け共有メモリ/Fコンポーネントです。  
Windowsアプリケーションで使用することができ、他のアプリケーションなどと共有メモリを介したデータ交換(通信)を実現します。  
INpMacは、DLLファイル型式とActiveXコントロール(OCX)の形式で提供します。  
DLLファイルを利用したコンソールアプリケーションやWin32等のアプリケーションと、ActiveXコントロールを利用したC#やダイアログベースのアプリケーションで使用するコンポーネントを使い分けることが可能です。

INpMacの各機能について以下にご説明します。

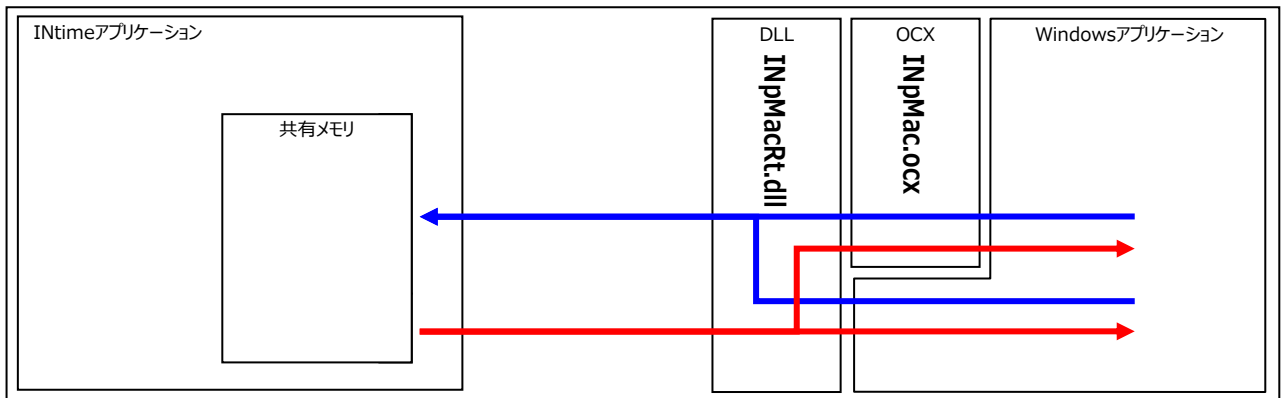
## 1-1. Windowsアプリケーション間通信機能(INpMacWinAPI)

- 1) WindowsアプリケーションとWindowsアプリケーション間のデータ交換をする場合に使用します。
- 2) Windows共有メモリ(名前付メモリマップドファイルオブジェクト)に対して、双方のWindowsアプリケーションからデータの読み込み(赤線)と書き込み(青線)が可能です。
- 3) 指定した名前の共有メモリを新たに確保することができます。(4KByte単位)  
指定した名前の共有メモリが既に存在する場合は、その共有メモリにアクセスします。



## 1-2. INtimeアプリケーション間通信機能(INpMacRtAPI)

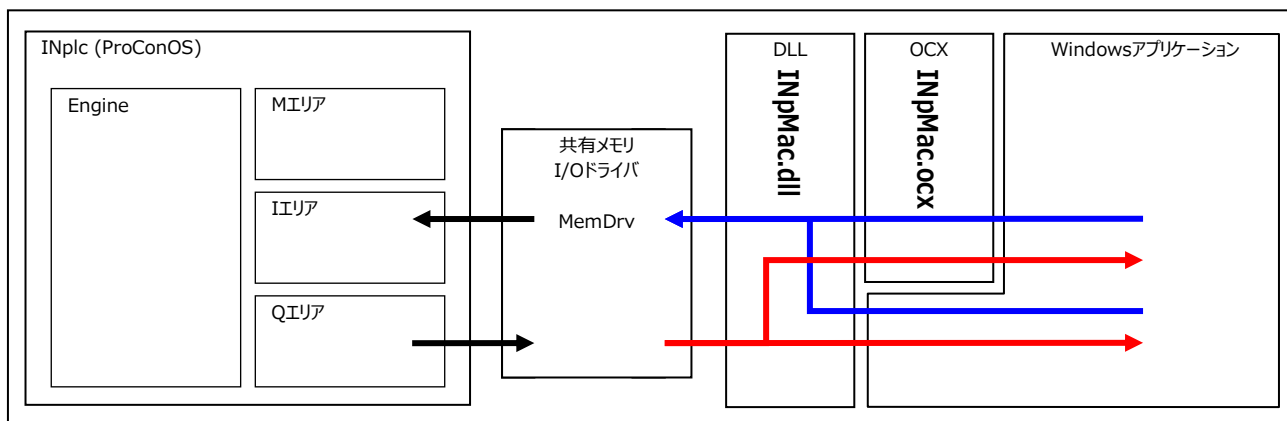
- 1) WindowsアプリケーションとINtimeアプリケーション間のデータ交換をする場合に使用します。
- 2) カタログされたINtime共有メモリに対して、Windowsアプリケーションからデータの読み込み(赤線)と書き込み(青線)が可能です。
- 3) 指定したカタログ名の共有メモリを検索して読み書きを行います。
- 4) ROOTプロセス以外のプロセスにカタログされた共有メモリを参照する場合は、そのプロセスをROOTプロセスにカタログすることで参照可能です。



## 1-3. INplc間通信機能(INpMacAPI)

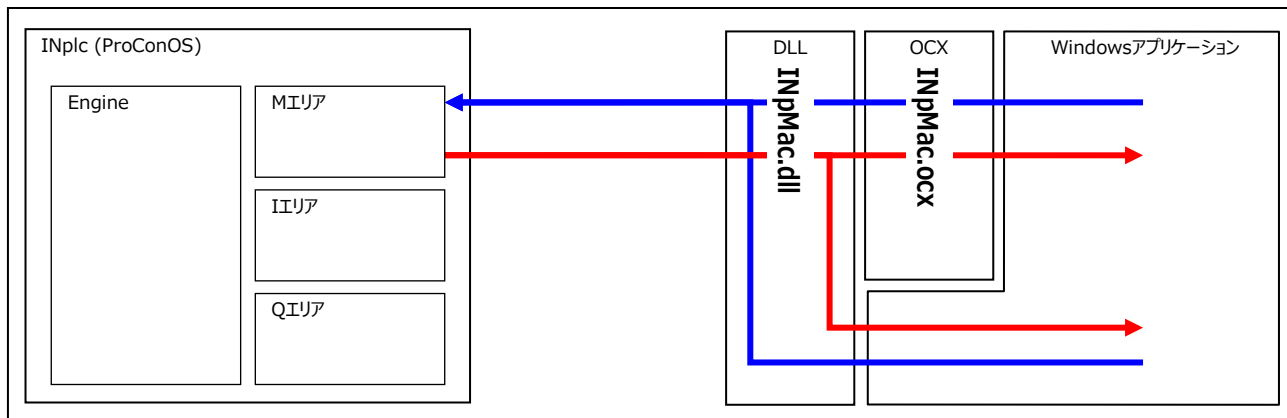
### (1) I/Oイメージへのアクセス

- 1) INplcのIエリア(入力読み用エリア)とQエリア(出力書き出し用エリア)を、Windowsアプリケーションと連動して利用する場合に使用します。(Iエリア、Qエリアについては以下I/Oイメージと記述)
- 2) I/Oイメージに対しての読書きは、INplcで標準提供されている共有メモリI/Oドライバ(MemDrv)を利用します。
- 3) 共有メモリI/Oドライバに対して、Windowsアプリケーションからデータの読み込み(赤線)と書き込み(青線)が可能です。
- 4) 共有メモリI/Oドライバを利用することで、実際にI/Oデバイスが接続されている状態を構成することができます。  
これにより、外部アプリケーションとProConOSアプリケーション間において、入出力を行うことができます。
- 5) I/Oイメージに対しては、共有メモリI/Oドライバで指定した範囲に対しアクセスします。そのため、他のI/Oドライバで割付けられたエリアに対してはアクセスすることはできません。



### (2) 共有メモリ領域へのアクセス

- 1) WindowsアプリケーションとINplcの共有メモリ領域『Mエリア』(%Mのセクション3)間のデータ交換をする場合に使用します。
- 2) INplcの共有メモリ領域『Mエリア』に対しての、読み込み(赤線)と書き込み(青線)が可能です。
- 3) Mエリア全体16,777,216Byte(16MByte)にアクセス出来ます。



## 2. インストール

### 2-1. システム要件

以下の条件に当てはまるコンピュータ上にて、INpMacを使用することができます。

#### (1) Windowsアプリケーション間通信機能(INpMacWinAPI)

- ・ Windows 7以降が動作すること。

#### (2) INtimeアプリケーション間通信機能(INpMacRtAPI)

- ・ INtime5.2以降が動作すること。

#### (3) INplc間通信機能(INpMacAPI)

- ・ 開発環境で、INplc-SDK v2.10以降が動作すること。
- ・ 実行環境で、INplc-RT v2.10以降が動作すること。
- ・ 共有メモリ/ODドライバ[MemDrv.iopack]がインストールされていること。

#### (4) 開発環境

- ・ Microsoft Visual Studio 2008以降。

### 2-2. インストールについて

#### (1) はじめに

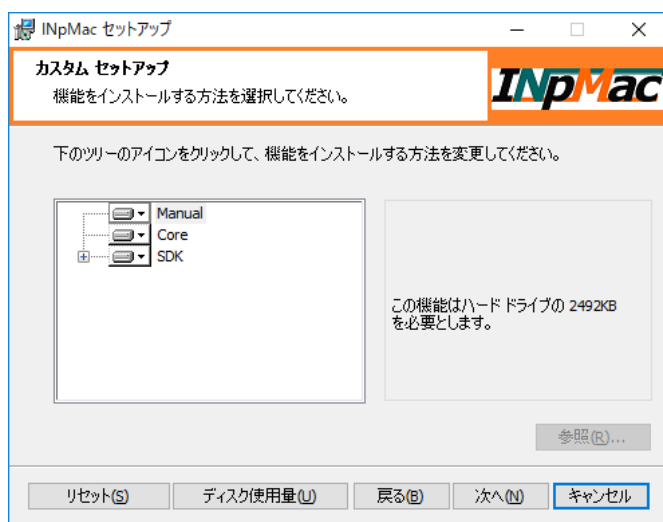
すでにINpMacがインストールされている場合はINpMacをアンインストールしてください。(2-3.アンインストールについてを参照)

#### (2) インストール

専用のインストーラ【INpMac.msi】を実行することにより、インストールが行われます。



以下の画面でインストールする機能を選択できます。



### 1) マニュアル (Manual)

「Manual」アイコンの機能を有効にすると、以下の場所にINpMacのマニュアルがインストールされます。

No.	フォルダ	ファイル名	内容
1	Program Files¥Micronet¥ INpMac¥	INpMac_UsersManual.pdf	INpMacユーザーズマニュアル(※本書)

※ 64bit版OSの場合、フォルダは[Program Files(x86)¥Micronet¥INpMac¥]になります。

### 2) 実行環境(Core)

「Core」アイコンの機能を有効にすると、以下の場所にINpMacの実行環境がインストールされます。

No.	フォルダ	ファイル名	内容
1	Windows¥System32¥	INpMacRt.dll	INtime用共有メモリ/ライブラリ(INpMacRtAPI)
2	"	INpMac.dll	INplc用共有メモリ/ライブラリ(INpMacAPI)
3	"	INpMac.ocx	INtime/INplc用ActiveXコントロール(OCX)
4	"	INpMacWin.dll	Windows用共有メモリ/ライブラリ(INpMacWinAPI)
5	"	INpMacWin.ocx	Windows用ActiveXコントロール(OCX)
6	Program Files¥Micronet¥ INpMac¥	INpMacRt.dll	INtime用共有メモリ/ライブラリ(INpMacRtAPI)
7	"	INpMac.dll	INplc用共有メモリ/ライブラリ(INpMacAPI)
8	"	INpMac.ocx	INtime/INplc用ActiveXコントロール(OCX)
9	"	INpMacWin.dll	Windows用共有メモリ/ライブラリ(INpMacWinAPI)
10	"	INpMacWin.ocx	Windows用ActiveXコントロール(OCX)

※ 64bit版OSの場合、フォルダは[Windows¥SysWOW64¥],[Program Files (x86)¥Micronet¥INpMac¥]になります。

### 3) 開発環境 (SDK)

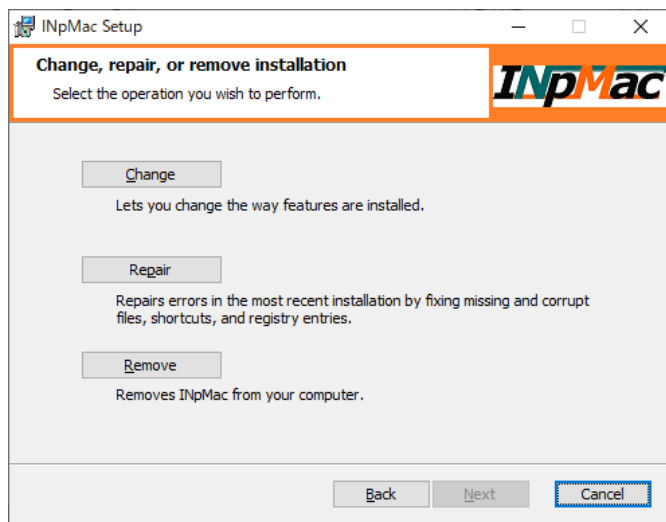
「SDK」アイコンの機能を有効にすると、以下の場所にINpMacの開発環境がインストールされます。

No.	フォルダ	ファイル名	内容
1	Program Files¥Micronet¥ INpMac¥	INpMac.lib	INpMacAPI定義ライブラリ(C/C++用)
2	"	INpMac.h	INpMacAPI定義ヘッダー(C/C++用)
3	"	INpMacRt.lib	INpMacRtAPI定義ライブラリ(C/C++用)
4	"	INpMacRt.h	INpMacRtAPI定義ヘッダー(C/C++用)
5	"	INpMacWin.lib	INpMacWinAPI定義ライブラリ(C/C++用)
6	"	INpMacWin.h	INpMacWinAPI定義ヘッダー(C/C++用)
7	Program Files¥Micronet¥ INpMac¥Sample	¥DLL¥SamplePLC	INpMac.dll を使用したサンプルプロジェクト
8	"	¥DLL¥SampleRT	INpMacRt.dll を使用したサンプルプロジェクト
9	"	¥DLL¥SampleWIN	INpMacWin.dll を使用したサンプルプロジェクト
10	"	¥OCX¥SampleRT	INpMac.ocx を使用したサンプルプロジェクト
11	"	¥OCX¥SampleWIN	INpMacWin.ocx を使用したサンプルプロジェクト

※ 64bit版OSの場合、フォルダは[Program Files (x86)¥Micronet¥INpMac¥]になります。

### (3) 旧INpMacインストール環境へのインストール

既にINpMacをインストールしていた環境に対して新しいINpMacをインストールする場合は、本インストーラを再び実行して「Repair」を実行してください。



### 2-3. アンインストールについて

DLLおよびOCXがアプリケーションから使用されていない状態で、[コントロールパネル]>[プログラムと機能] の一覧の中から "INpMac" を選択することで削除できます。

## 2-4. ACTIVE TOUCH環境の場合

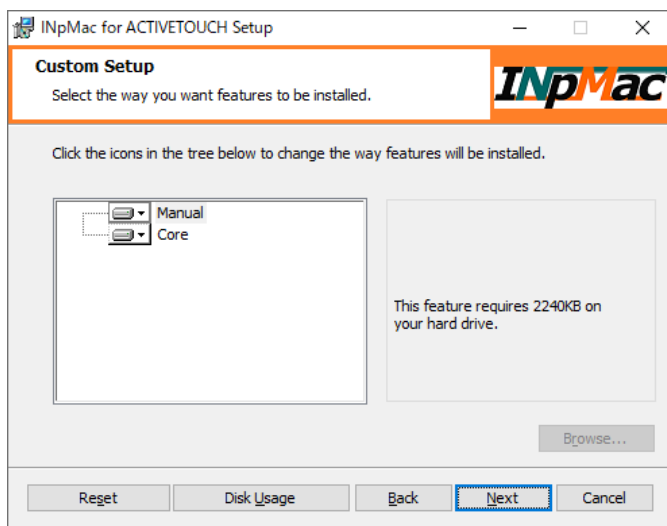
### (1) はじめに

すでにINpMacがインストールされている場合はアンインストールしてください。(2-5.アンインストールについてを参照)

### (2) インストール

専用のインストーラ【INpMac\_for\_ACTIVETOUCH.msi】を実行することにより、インストールが行われます。

以下の画面でインストールする機能を選択できます。



#### 1) マニュアル (Manual)

「Manual」アイコンの機能を有効にすると、以下の場所にINpMacのマニュアルがインストールされます。

No.	フォルダ	ファイル名	内容
1	Program Files¥Micronet¥ INpMac¥	INpMac_UsersManual.pdf	INpMacユーザーズマニュアル(※本書)

※ 64bit版OSの場合、フォルダは[Program Files(x86)¥Micronet¥INpMac¥]になります。

#### 2) 実行環境(Core)

「Core」アイコンの機能を有効にすると、以下の場所にINpMacの実行環境がインストールされます。

No.	フォルダ	ファイル名	内容
1	[32bit版OS] Program Files¥Micronet¥ INpMac¥ [64bit版OS] Program Files¥Micronet¥ INpMac¥x64	INpMacRt.dll	INtime用共有メモリ/ライブラリ(INpMacRtAPI)
2	"	INpMac.dll	INplc用共有メモリ/ライブラリ(INpMacAPI)
3	"	INpMac.ocx	INtime/INplc用ActiveXコントロール(OCX)
4	"	INpMacWin.dll	Windows用共有メモリ/ライブラリ(INpMacWinAPI)
5	"	INpMacWin.ocx	Windows用ActiveXコントロール(OCX)

※ 64bit版OSの場合、フォルダは[Program Files (x86)¥Micronet¥INpMac¥]になります。

## 2-5. アンインストールについて

DLLおよびOCXがアプリケーションから使用されていない状態で、[コントロールパネル]>[プログラムと機能]の一覧の中から "INpMac for ACTIVETOUCH" を選択することで削除できます。

## 3. APIについて

### 3-1. INpMacWinAPI

INpMacWin.dllに実装するAPIについて解説します。

#### ■ 使用方法

定義ファイルを各種アプリケーションのプロジェクトに組み込んで使用します。

- ・ Visual C/C++ プロジェクト ……【INpMacWin.h】及び【INpMacWin.lib】

#### (1) INpMacWin\_CreateMemory()

Windowsシステム上に共有メモリを確保します。

```
int INpMacWin_CreateMemory(  
    HANDLE *   phMem,           // メモリのハンドル  
    LPTSTR     pName,           // メモリの名称  
    int        Mem_Size         // メモリサイズ  
);
```

#### パラメータ

<i>phMem</i>	HANDLE 型変数のポインタを指定します。 メモリマップドファイルオブジェクトのHANDLE 値が格納されます。
<i>pName</i>	確保する共有メモリの名前を半角英数字文字列で指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。※NULL文字含まず)
<i>Mem_Size</i>	確保する共有メモリのサイズ(Byte)を65,536Byte(64KByte)以内で指定します。

#### 戻り値

関数が成功すると、0以上の値が返ります。

新たに共有メモリ領域を確保した場合は、戻り値として0が返ります。

共有メモリが既に存在する場合は、戻り値としてアクセス可能な範囲(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_CREATE	0x80010000	共有メモリが確保できません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>phMem</i> ]がNULLです。 パラメータ[ <i>pName</i> ]がNULLです。 パラメータ[ <i>Mem_Size</i> ]が範囲外です。

#### 内部動作

共有メモリは、4,096Byte(4KByte)単位で確保されます。

指定した名前の共有メモリが存在する場合は、既存の共有メモリをオープンします。GetLastError関数は[ERROR\_ALREADY\_EXISTS(183)]を返します。この時、共有メモリサイズは既存の共有メモリのサイズとなるため、関数の戻り値としてアクセス可能な範囲(最大65,536Byte)を返します。



## (2) INpMacWin\_DeleteMemory()

Windowsシステム上の共有メモリを解放します。

```
int INpMacWin_DeleteMemory(  
    HANDLE    hMem          // メモリのハンドル  
);
```

### パラメータ

*hMem* `INpMacWin_CreateMemory()`で取得した共有メモリのHANDLE値を設定します。

### 戻り値

関数が成功すると、0以上の値が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_DELETE	0x80010001	共有メモリが解放できません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>hMem</i> ]が不正です。

## (3) INpMacWin\_ReadMemory()

Windowsシステム上の共有メモリから値を取得します。

```
int INpMacWin_ReadMemory(  
    HANDLE    hMem,          // メモリのハンドル  
    int       Offset,        // オフセット  
    BYTE *    lpData,        // データを格納するバッファ  
    int       Buff_Size      // 取得するサイズ  
);
```

### パラメータ

*hMem* `INpMacWin_CreateMemory()`で取得した共有メモリのHANDLE値を設定します。

*Offset* 取得を開始するメモリ領域のオフセットバイト値を指定します。

*lpData* データを格納するバッファへのポインタを指定します。

*Buff\_Size* 共有メモリから取得するデータサイズ(Byte)を指定します。

### 戻り値

関数が成功すると、共有メモリから取得したデータ数(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_INVALID_OFFSET	0x80010002	指定したオフセット値は無効です。
E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>hMem</i> ]が不正です。 パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>Buff_Size</i> ]が範囲外です。

### 内部動作

取得するデータサイズは、以下の方式で決定されます。

(共有メモリサイズ - Offset) > Buff\_Size ... Buff\_Size 分のデータを取得する。  
(共有メモリサイズ - Offset) < Buff\_Size ... (共有メモリサイズ - Offset)分のデータを取得する。

### 制限

- 本関数で取り扱える共有メモリの領域は最大65,536Byte(64KByte)の範囲内です。この範囲を超えた領域や、メモリサイズを超えた領域へのアクセスはできません。
- [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (4) INpMacWin\_WriteMemory()

Windowsシステム上の共有メモリへ値を書込みます。

```
int INpMacWin_WriteMemory(  
    HANDLE    hMem,                // メモリのハンドル  
    int        Offset,              // オフセット  
    BYTE *     lpData,              // データが格納されたバッファ  
    int        Buff_Size            // 書込むサイズ  
);
```

### パラメータ

<i>hMem</i>	INpMacWin_CreateMemory()で取得した共有メモリのHANDLE値を設定します。
<i>Offset</i>	書き込みを開始するメモリ領域のオフセットバイト値を指定します。
<i>lpData</i>	データが格納されたバッファへのポインタを指定します。
<i>Buff_Size</i>	共有メモリへ書込むデータサイズ(Byte)を指定します。

### 戻り値

関数が成功すると、共有メモリへ書込んだデータ数(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_INVALID_OFFSET	0x80010002	指定したオフセット値は無効です。
E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>hMem</i> ]が不正です。 パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>Buff_Size</i> ]が範囲外です。

### 内部動作

書込むデータサイズは、以下の方式で決定されます。

(共有メモリサイズ - Offset) > Buff\_Size ... Buff\_Size 分のデータを書込む。  
(共有メモリサイズ - Offset) < Buff\_Size ... (共有メモリサイズ - Offset)分のデータを書込む。

### 制限

- 本関数で取り扱える共有メモリの領域は最大65,536Byte(64KByte)の範囲内です。この範囲を超えた領域や、メモリサイズを超えた領域へのアクセスはできません。
- [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (5) INpMacWin\_MapMemory()

Windowsシステム上の共有メモリのマップしたポインタを取得します。

```
int INpMacWin_MapMemory(
    HANDLE      hMem,           // メモリのハンドル
    LPVOID *    lpMap           // メモリのポインタ
);
```

### パラメータ

*hMem* INpMacWin\_CreateMemory()で取得した共有メモリのHANDLE値を設定します。

*lpMap* 取得したポインタを格納する変数へのポインタを指定します。

### 戻り値

関数が成功すると、0以上の値が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>hMem</i> ]が不正です。 パラメータ[ <i>lpMap</i> ]がNULLです。

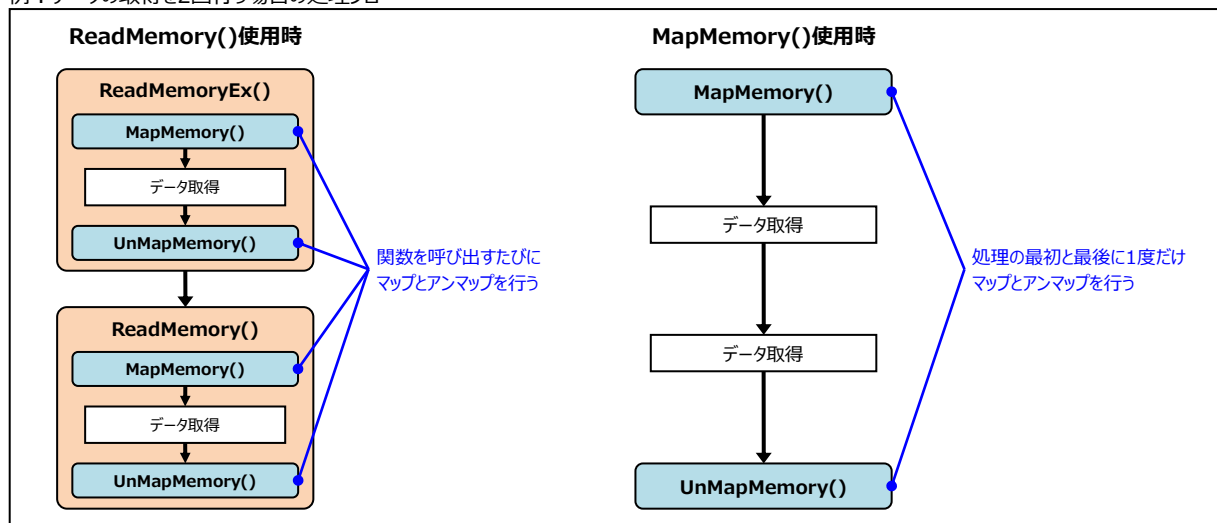
### 制限

- ・ [*lpMap*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

### 補足

連続してデータの取得や書き込みを行う場合には処理の最初に本関数でマップを行い、取得したメモリポインタを直接利用することにより、INpMacWin\_ReadMemory()やINpMacWin\_WriteMemory()を使用したときと比較して高速にデータの取得や書き込みを行えます。

例：データの取得を2回行う場合の処理フロー



## (6) INpMacWin\_UnMapMemory()

Windowsシステム上の共有メモリのマップした領域を解放します。

```
int INpMacWin_UnMapMemory(  
    LPVOID    pMap                // メモリのポインタ  
);
```

### パラメータ

*pMap* `INpMacWin_MapMemory()`で取得したポインタを指定します。

### 戻り値

関数が成功すると、0以上の値が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。 パラメータ[ <i>pMap</i> ]が不正です。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>pMap</i> ]がNULLです。

### 制限

- ・ [*pMap*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (7) INpMacWin\_GetSize()

Windowsシステム上の共有メモリのマップした領域のサイズを取得します。

```
int INpMacWin_GetSize(  
    LPVOID    pMap                // メモリのポインタ  
);
```

### パラメータ

*pMap* `INpMacWin_MapMemory()`で取得したポインタを指定します。

### 戻り値

関数が成功すると、マップした共有メモリのサイズ(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。 パラメータ[ <i>pMap</i> ]が不正です。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>pMap</i> ]がNULLです。

### 制限

- ・ [*pMap*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (8) INpMacWin\_CreateMemoryEx()

Windowsシステム上に共有メモリを確保します。 ※INpMacWin\_CreateMemory()のサイズ制限無し版です。

```
int INpMacWin_CreateMemoryEx(  
    HANDLE *   phMem,           // メモリのハンドル  
    LPTSTR     pName,           // メモリの名称  
    DWORD      dwMaxSize        // メモリサイズ  
);
```

### パラメータ

<i>phMem</i>	HANDLE 型変数のポインタを指定します。 メモリマップドファイルオブジェクトのHANDLE 値が格納されます。
<i>pName</i>	確保する共有メモリの名前を半角英数字文字列で指定します。 (大文字小文字の区別はされず。また最大文字数は12文字までです。※NULL文字含まず)
<i>dwMaxSize</i>	確保する共有メモリのサイズ(Byte)を指定します。確保できる最大サイズはシステムに依存します。

### 戻り値

関数が成功すると、以下の値が返ります。

E_OK	0x00000000	共有メモリを作成しました。
S_INPMACWIN_ALREADY_EXISTS	0x000000B7	既存の共有メモリをオープンしました。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_CREATE	0x80010000	共有メモリが確保できません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>phMem</i> ]がNULLです。 パラメータ[ <i>pName</i> ]がNULLです。
E_INPMACWIN_NOT_ENOUGH_MEMORY	0x80010005	システムのメモリが不足しています。

### 内部動作

共有メモリは、4,096Byte(4KByte)単位で確保されます。

指定した名前の共有メモリが存在する場合は、既存の共有メモリをオープンします。この時、共有メモリサイズは指定の[*dwMaxSize*]ではなく既存の共有メモリのサイズとなるため、INpMacWin\_GetSizeEx()で確認することを推奨します。

## (9) INpMacWin\_DeleteMemoryEx()

Windowsシステム上の共有メモリを解放します。 ※INpMacWin\_DeleteMemory()のサイズ制限無し版です。

```
int INpMacWin_DeleteMemoryEx(  
    HANDLE      hMem            // メモリのハンドル  
);
```

### パラメータ

<i>hMem</i>	INpMacWin_CreateMemoryEx()で取得した共有メモリのHANDLE値を設定します。
-------------	---

### 戻り値

関数が成功すると、0が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_DELETE	0x80010001	共有メモリが解放できません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>hMem</i> ]が不正です。

## (10) INpMacWin\_ReadMemoryEx()

Windowsシステム上の共有メモリから値を取得します。※INpMacWin\_ReadMemory()のサイズ制限無し版です。

```
int INpMacWin_ReadMemoryEx(  
    HANDLE    hMem,                // メモリのハンドル  
    DWORD     dwOffset,            // オフセット  
    void *    lpData,              // データを格納するバッファ  
    DWORD     dwMaxSize,           // 取得するサイズ  
    DWORD *   pdwReadSize          // 取得したサイズ  
);
```

### パラメータ

<i>hMem</i>	INpMacWin_CreateMemoryEx()で取得した共有メモリのHANDLE値を設定します。
<i>dwOffset</i>	取得を開始するメモリ領域のオフセットバイト値を指定します。
<i>lpData</i>	データを格納するバッファへのポインタを指定します。
<i>dwMaxSize</i>	共有メモリから取得するデータサイズ(Byte)を指定します。
<i>pdwReadSize</i>	共有メモリから取得したデータ数(Byte)を格納する変数へのポインタを指定します。

### 戻り値

関数が成功すると、0が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_INVALID_OFFSET	0x80010002	指定したオフセット値は無効です。
E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>hMem</i> ]が不正です。 パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>dwMaxSize</i> ]が範囲外です。

### 内部動作

取得するデータサイズは、以下の方式で決定されます。

(共有メモリサイズ - dwOffset) > dwMaxSize ... dwMaxSize 分のデータを取得する。  
(共有メモリサイズ - dwOffset) < dwMaxSize ... (共有メモリサイズ - dwOffset)分のデータを取得する。

### 制限

- 本関数で取り扱える共有メモリの領域は、共有メモリサイズの範囲内です。メモリサイズを超えた領域へのアクセスはできません。
- [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (11) INpMacWin\_WriteMemoryEx()

Windowsシステム上の共有メモリへ値を書込みます。 ※INpMacWin\_WriteMemory()のサイズ制限無し版です。

```
int INpMacWin_WriteMemoryEx(  
    HANDLE    hMem,                // メモリのハンドル  
    DWORD     dwOffset,            // オフセット  
    void *    lpData,              // データが格納されたバッファ  
    DWORD     dwMaxSize,           // 書込むサイズ  
    DWORD *   pdwWriteSize         // 書込んだサイズ  
);
```

### パラメータ

<i>hMem</i>	INpMacWin_CreateMemoryEx()で取得した共有メモリのHANDLE 値を設定します。
<i>Offset</i>	書込みを開始するメモリ領域のオフセットバイト値を指定します。
<i>lpData</i>	データが格納されたバッファへのポインタを指定します。
<i>dwMaxSize</i>	共有メモリへ書込むデータサイズ(Byte)を指定します。
<i>pdwWriteSize</i>	共有メモリへ書込んだデータ数(Byte)を格納する変数へのポインタを指定します。

### 戻り値

関数が成功すると、0が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_INVALID_OFFSET	0x80010002	指定したオフセット値は無効です。
E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>hMem</i> ]が不正です。 パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>dwMaxSize</i> ]が範囲外です。

### 内部動作

書込むデータサイズは、以下の方式で決定されます。

(共有メモリサイズ - dwOffset) > dwMaxSize ... dwMaxSize 分のデータを書込む。  
(共有メモリサイズ - dwOffset) < dwMaxSize ... (共有メモリサイズ - dwOffset)分のデータを書込む。

### 制限

- 本関数で取り扱える共有メモリの領域は、共有メモリサイズの範囲内です。メモリサイズを超えた領域へのアクセスはできません。
- [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (12) INpMacWin\_MapMemoryEx()

Windowsシステム上の共有メモリのマップしたポインタを取得します。 ※INpMacWin\_MapMemoryEx()のサイズ制限無し版です。

```
int INpMacWin_MapMemoryEx(  
    HANDLE    hMem,                // メモリのハンドル  
    DWORD     dwOffset,            // オフセット  
    LPVOID *   lpMap,              // メモリのポインタ  
    DWORD     dwMaxSize            // マップするサイズ  
);
```

### パラメータ

<i>hMem</i>	INpMacWin_CreateMemoryEx()で取得した共有メモリのHANDLE 値を設定します。
<i>dwOffset</i>	マップする領域のオフセットバイト値を指定します。
<i>lpMap</i>	取得したポインタを格納する変数へのポインタを指定します。
<i>dwMaxSize</i>	マップするメモリサイズ(Byte)を指定します。

### 戻り値

関数が成功すると、オフセット値の端数(0以上の値)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[ <i>hMem</i> ]が不正です。 パラメータ[ <i>lpMap</i> ]がNULLです。
E_INPMACWIN_NOT_ENOUGH_MEMORY	0x80010005	システムのメモリが不足しています。

### 内部動作

共有メモリをマップする際、オフセットは64KB(65,536Byte)単位で指定します。dwOffset に指定した値が64KBの倍数ではない場合、戻り値としてオフセット値の端数を返します。

例: dwOffset=66000の場合、lpMapには65,536Byte目のメモリポインタが格納され、戻り値には端数の464が返ります。

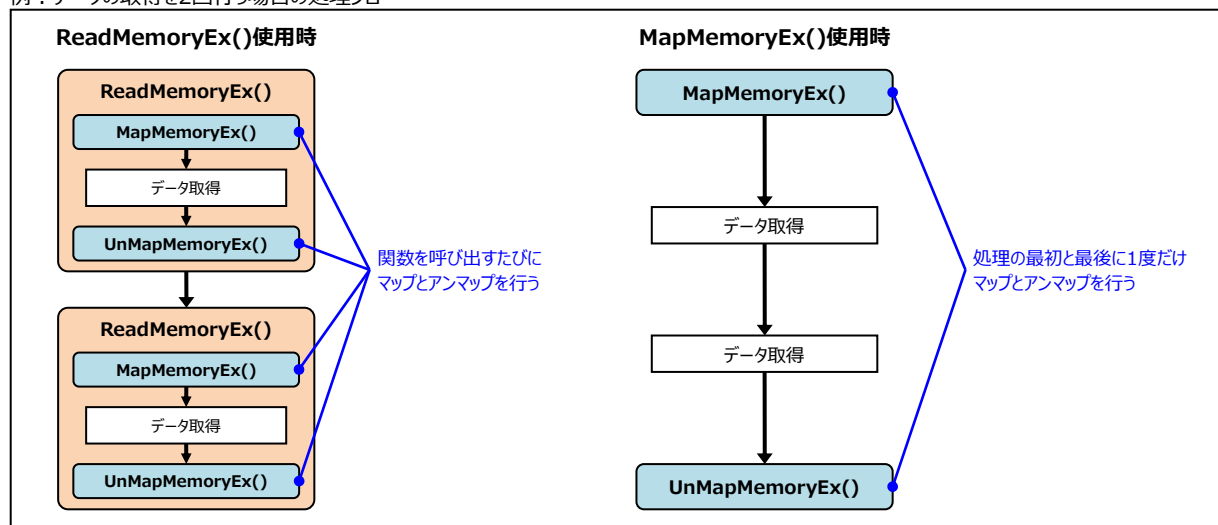
### 制限

- ・ [*lpMap*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。
- ・ 共有メモリをマップする際に連続したメモリ領域が確保できない場合、関数は失敗します。
- ・ 共有メモリのマップを解除する場合は、INpMacWin\_UnMapMemoryEx()を使用してください。

### 補足

連続してデータの取得や書き込みを行う場合には処理の最初に本関数でマップを行い、取得したメモリポインタを直接利用することにより、INpMacWin\_ReadMemoryEx()やINpMacWin\_WriteMemoryEx()を使用したときと比較して高速にデータの取得や書き込みを行えます。

例：データの取得を2回行う場合の処理フロー





### (13) INpMacWin\_UnMapMemoryEx()

Windowsシステム上の共有メモリのマップした領域を解放します。 ※INpMacWin\_UnMapMemory()のサイズ制限無し版です。

```
int INpMacWin_UnMapMemoryEx (
    LPVOID    pMap                // メモリのポインタ
);
```

#### パラメータ

*pMap*                      INpMacWin\_MapMemoryEx()で取得したポインタを指定します。

#### 戻り値

関数が成功すると、0以上の値が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。 UnmapViewOfFileに失敗しました。 パラメータ[pMap]が不正です。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[pMap]がNULLです。

#### 制限

- ・ [pMap]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

### (14) INpMacWin\_GetSizeEx()

Windowsシステム上の共有メモリのマップした領域のサイズを取得します。 ※INpMacWin\_GetSize()のサイズ制限無し版です。

```
int INpMacWin_GetSizeEx (
    LPVOID    pMap,                // メモリのポインタ
    DWORD *   pdwMemSize           // マップしたメモリサイズ
);
```

#### パラメータ

*pMap*                      INpMacWin\_MapMemoryEx()で取得したポインタを指定します。

*pdwMemSize*              メモリサイズを格納する変数へのポインタを指定します。

#### 戻り値

関数が成功すると、0が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACWIN_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。 パラメータ[pMap]が不正です。
E_INPMACWIN_INVALID_PARAM	0x80010004	パラメータ[pMap]がNULLです。 パラメータ[pdwMemSize]がNULLです。

#### 制限

- ・ [pMap]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## 3-2. INpMacRtAPI

INpMacRt.dllに実装するAPIについて解説します。

### ■ 使用方法

定義ファイルを各種アプリケーションのプロジェクトに組み込んで使用します。

- ・ Visual C/C++ プロジェクト ……【INpMacRt.h】及び【INpMacRt.lib】

### (1) INpMacRt\_ReadMemory()

INtime上の共有メモリから値を取得します。

```
int INpMacRt_ReadMemory(
    LPCTSTR pProcName, // プロセスの名前
    LPCTSTR pName,      // メモリ の名前
    int Offset,          // オフセット
    BYTE * lpData,       // データを格納するバッファ
    int Buff_Size        // 取得するサイズ
);
```

#### パラメータ

<i>pProcName</i>	共有メモリがカタログされているプロセスの名前を半角英数字文字列で指定します。 ROOTプロセスの場合は、空文字("")を指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>pName</i>	共有メモリの名前を半角英数字文字列で指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>Offset</i>	取得を開始するメモリ領域のオフセットバイト値を指定します。
<i>lpData</i>	データを格納するバッファへのポインタを指定します。
<i>Buff_Size</i>	共有メモリから取得するデータサイズ(Byte)を指定します。

#### 戻り値

関数が成功すると、共有メモリから取得したデータ数(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACRT_SYSTEM_DOWN	0x80020000	INtimeが起動していません。 指定したプロセスがありません。
E_INPMACRT_NOT_FOUND	0x80020001	指定した共有メモリがありません。
E_INPMACRT_INVALID_OFFSET	0x80020002	指定したオフセット値は無効です。
E_INPMACRT_FAILED_ACCESS	0x80020003	共有メモリにアクセスできません。
E_INPMACRT_INVALID_PARAM	0x80020004	パラメータ[lpData]がNULLです。 パラメータ[Buff_Size]が範囲外です。
E_INPMACRT_NOT_EXECUTION	0x80020005	INtimeがインストールされていません。

#### 内部動作

取得するデータサイズは、以下の方式で決定されます。

- (共有メモリサイズ - Offset) > Buff\_Size …… Buff\_Size 分のデータを取得する。
- (共有メモリサイズ - Offset) < Buff\_Size …… (共有メモリサイズ - Offset)分のデータを取得する。

#### 制限

- ・ 本関数で取り扱える共有メモリの領域は最大65,536Byte(64KByte)の範囲内です。この範囲を超えた領域や、メモリサイズを超えた領域へのアクセスはできません。
- ・ [lpData]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (2) INpMacRt\_WriteMemory()

INtime上の共有メモリへ値を書込みます。

```
int INpMacRt_WriteMemory(  
    LPCTSTR    pProcName,          // プロセスの名前  
    LPCTSTR    pName,              // メモリの名前  
    int        Offset,             // オフセット  
    BYTE *     lpData,              // データが格納されたバッファ  
    int        Buff_Size            // 書き込むサイズ  
);
```

### パラメータ

<i>pProcName</i>	共有メモリがカタログされているプロセスの名前を半角英数字文字列で指定します。 ROOTプロセスの場合は、空文字("")を指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>pName</i>	共有メモリの名前を半角英数字文字列で指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>Offset</i>	書き込みを開始するメモリ領域のオフセットバイト値を指定します。
<i>lpData</i>	データが格納されたバッファへのポインタを指定します。
<i>Buff_Size</i>	共有メモリへ書き込むデータサイズ(Byte)を指定します。

### 戻り値

関数が成功すると、共有メモリへ書き込んだデータ数(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACRT_SYSTEM_DOWN	0x80020000	INtimeが起動していません。 指定したプロセスがありません。
E_INPMACRT_NOT_FOUND	0x80020001	指定した共有メモリがありません。
E_INPMACRT_INVALID_OFFSET	0x80020002	指定したオフセット値は無効です。
E_INPMACRT_FAILED_ACCESS	0x80020003	共有メモリにアクセスできません。
E_INPMACRT_INVALID_PARAM	0x80020004	パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>Buff_Size</i> ]が範囲外です。
E_INPMACRT_NOT_EXECUTION	0x80020005	INtimeがインストールされていません。

### 内部動作

取得するデータサイズは、以下の方式で決定されます。

(共有メモリサイズ - Offset) > Buff\_Size ... Buff\_Size 分のデータを書込む。  
(共有メモリサイズ - Offset) < Buff\_Size ... (共有メモリサイズ - Offset)分のデータを書込む。

### 制限

- 本関数で取り扱える共有メモリの領域は最大65,536Byte(64KByte)の範囲内です。この範囲を超えた領域や、メモリサイズを超えた領域へのアクセスはできません。
- [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

### (3) INpMacRt\_GetSize()

INtimeシステム上の共有メモリのサイズを取得します。

```
int INpMacRt_GetSize(  
    LPCTSTR pProcName, // プロセスの名前  
    LPCTSTR pName       // メモリの名前  
);
```

#### パラメータ

<i>pProcName</i>	共有メモリがカタログされているプロセスの名前を半角英数字文字列で指定します。 ROOTプロセスの場合は、空文字("")を指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>pName</i>	共有メモリの名前を半角英数字文字列で指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)

#### 戻り値

関数が成功すると、共有メモリのサイズ(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACRT_SYSTEM_DOWN	0x80020000	INtimeが起動していません。 指定したプロセスがありません。
E_INPMACRT_NOT_FOUND	0x80020001	指定した共有メモリがありません。
E_INPMACRT_FAILED_ACCESS	0x80010003	共有メモリにアクセスできません。
E_INPMACRT_INVALID_PARAM	0x80010004	パラメータ[pProcName]がNULLです。 パラメータ[pName]がNULLです。

## (4) INpMacRt\_ReadMemoryEx()

INtime上の共有メモリから値を取得します。 ※INpMacRt\_ReadMemory()のサイズ制限無し版です。

```
int INpMacRt_ReadMemoryEx(  
    LPCTSTR    pProcName,          // プロセスの名前  
    LPCTSTR    pName,              // メモリの名前  
    DWORD      dwOffset,           // オフセット  
    void *     lpData,              // データを格納するバッファ  
    DWORD      dwMaxSize,          // 取得するサイズ  
    DWORD *     pdwReadSize         // 取得したサイズ  
);
```

### パラメータ

<i>pProcName</i>	共有メモリがカタログされているプロセスの名前を半角英数字文字列で指定します。 ROOTプロセスの場合は、空文字("")を指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>pName</i>	共有メモリの名前を半角英数字文字列で指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>dwOffset</i>	取得を開始するメモリ領域のオフセットバイト値を指定します。
<i>lpData</i>	データを格納するバッファへのポインタを指定します。
<i>dwMaxSize</i>	共有メモリから取得するデータサイズ(Byte)を指定します。
<i>pdwReadSize</i>	共有メモリから取得したデータ数(Byte)を格納する変数のポインタを指定します。

### 戻り値

関数が成功すると、0が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACRT_SYSTEM_DOWN	0x80020000	INtimeが起動していません。 指定したプロセスがありません。
E_INPMACRT_NOT_FOUND	0x80020001	指定した共有メモリがありません。
E_INPMACRT_INVALID_OFFSET	0x80020002	指定したオフセット値は無効です。
E_INPMACRT_FAILED_ACCESS	0x80020003	共有メモリにアクセスできません。
E_INPMACRT_INVALID_PARAM	0x80020004	パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>dwMaxSize</i> ]が範囲外です。
E_INPMACRT_NOT_EXECUTION	0x80020005	INtimeがインストールされていません。

### 内部動作

取得するデータサイズは、以下の方式で決定されます。

(共有メモリサイズ - dwOffset) > dwMaxSize ... dwMaxSize分のデータを取得する。  
(共有メモリサイズ - dwOffset) < dwMaxSize ... (共有メモリサイズ - dwOffset)分のデータを取得する。

### 制限

- 本関数で取り扱える共有メモリの領域は、共有メモリサイズの範囲内です。メモリサイズを超えた領域へのアクセスはできません。
- [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (5) INpMacRt\_WriteMemoryEx()

INtime上の共有メモリへ値を書込みます。 ※INpMacRt\_WriteMemory()のサイズ制限無し版です。

```
int INpMacRt_WriteMemoryEx(
    LPCTSTR    pProcName,          // プロセスの名前
    LPCTSTR    pName,              // メモリの名前
    DWORD      dwOffset,           // オフセット
    void *     lpData,              // データが格納されたバッファ
    DWORD      dwMaxSize,           // 書込むサイズ
    DWORD *    pdwReadSize         // 書込んだサイズ
);
```

### パラメータ

<i>pProcName</i>	共有メモリがカタログされているプロセスの名前を半角英数字文字列で指定します。 ROOTプロセスの場合は、空文字("")を指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>pName</i>	共有メモリの名前を半角英数字文字列で指定します。 (大文字小文字の区別はされます。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>dwOffset</i>	書き込みを開始するメモリ領域のオフセットバイト値を指定します。
<i>lpData</i>	データが格納されたバッファへのポインタを指定します。
<i>dwMaxSize</i>	共有メモリへ書込むデータサイズ(Byte)を指定します。
<i>pdwWriteSize</i>	共有メモリへ書込んだデータ数(Byte)を格納する変数のポインタを指定します。

### 戻り値

関数が成功すると、0が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMACRT_SYSTEM_DOWN	0x80020000	INtimeが起動していません。 指定したプロセスがありません。
E_INPMACRT_NOT_FOUND	0x80020001	指定した共有メモリがありません。
E_INPMACRT_INVALID_OFFSET	0x80020002	指定したオフセット値は無効です。
E_INPMACRT_FAILED_ACCESS	0x80020003	共有メモリにアクセスできません。
E_INPMACRT_INVALID_PARAM	0x80020004	パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>dwMaxSize</i> ]が範囲外です。
E_INPMACRT_NOT_EXECUTION	0x80020005	INtimeがインストールされていません。

### 内部動作

取得するデータサイズは、以下の方式で決定されます。

(共有メモリサイズ - dwOffset) > dwMaxSize ... dwMaxSize分のデータを書込む。  
(共有メモリサイズ - dwOffset) < dwMaxSize ... (共有メモリサイズ - dwOffset)分のデータを書込む。

### 制限

- 本関数で取り扱える共有メモリの領域は、共有メモリサイズの範囲内です。メモリサイズを超えた領域へのアクセスはできません。
- [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (6) INpMacRt\_GetSizeEx()

INtimeシステム上の共有メモリのサイズを取得します。 ※INpMacRt\_GetSize()のサイズ制限無し版です。

```
int INpMacRt_GetSizeEx (
    LPCTSTR    pProcName,          // プロセスの名前
    LPCTSTR    pName,              // メモリの名前
    DWORD *    pdwMemSize          // メモリサイズ
);
```

### パラメータ

<i>pProcName</i>	共有メモリがカタログされているプロセスの名前を半角英数字文字列で指定します。 ROOTプロセスの場合は、空文字("")を指定します。 (大文字小文字の区別はされず。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>pName</i>	共有メモリの名前を半角英数字文字列で指定します。 (大文字小文字の区別はされず。また最大文字数は12文字までです。 ※NULL文字含まず)
<i>pdwMemSize</i>	メモリサイズを格納する変数へのポインタを指定します。

### 戻り値

関数が成功すると、0に戻ります。

関数が失敗すると、以下の値が返ります。

E_INPMACRT_SYSTEM_DOWN	0x80020000	INtimeが起動していません。 指定したプロセスがありません。
E_INPMACRT_NOT_FOUND	0x80020001	指定した共有メモリがありません。
E_INPMACRT_FAILED_ACCESS	0x80020003	共有メモリにアクセスできません。
E_INPMACRT_INVALID_PARAM	0x80020004	パラメータ[ <i>pProcName</i> ]がNULLです。 パラメータ[ <i>pName</i> ]がNULLです。

### 3-3. INpMacAPI

---

INpMac.dllに実装するAPIについて解説します。

#### ■ 使用方法

定義ファイルを各種アプリケーションのプロジェクトに組み込んで使用します。

- ・ Visual C/C++ プロジェクト … 【INpMac.h】及び【INpMac.lib】

#### (1) INpMac\_Get\_Qarea\_Size()

Qエリアに割り当てた共有メモリ/Oドライブのメモリサイズを取得します。

```
int INpMac_Get_Qarea_Size(  
    int      GroupNo           // グループ番号  
);
```

#### パラメータ

*GroupNo*                      対象となる1～16のI/Oグループ番号を指定します。

#### 戻り値

関数が成功すると、指定したグループ番号の共有メモリのサイズ(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMAC_SYSTEM_DOWN	0x80000000	INplcが起動していません。
E_INPMAC_GROUP_NO_UNKNOWN	0x80000001	指定したグループ番号が不正です。 共有メモリにアクセスできません。
E_INPMAC_NOT_EXECUTION	0x80000005	INplc環境ではありません。



## (2) INpMac\_Get\_Qarea()

Qエリアに割り当てた共有メモリ/Oドライブを経由して、出力値を取得します。

```
int INpMac_Get_Qarea(  
    int      GroupNo,          // グループ番号  
    int      Offset,          // オフセット  
    BYTE *   lpData,          // 情報を格納するバッファ  
    int      Buff_Size        // 情報バッファのサイズ  
);
```

### パラメータ

<i>GroupNo</i>	対象となる1～16のI/Oグループ番号を指定します。
<i>Offset</i>	取得を開始するメモリ領域のオフセットバイト値を指定します。
<i>lpData</i>	データを格納するバッファへのポインタを指定します。
<i>Buff_Size</i>	共有メモリから取得するデータサイズ(Byte)を指定します。

### 戻り値

関数が成功すると、指定したグループ番号の共有メモリから取得したデータ数(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMAC_SYSTEM_DOWN	0x80000000	INplcが起動していません。
E_INPMAC_GROUP_NO_UNKNOWN	0x80000001	指定したグループ番号が不正です。 共有メモリにアクセスできません。
E_INPMAC_INVALID_OFFSET	0x80000002	指定したオフセット値は無効です。
E_INPMAC_INVALID_PARAM	0x80000004	パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>Buff_Size</i> ]が範囲外です。
E_INPMAC_NOT_EXECUTION	0x80000005	INplc環境ではありません。

### 内部動作

取得するデータサイズは、以下の方式で決定されます。

- (共有メモリサイズ - Offset) > Buff\_Size ... Buff\_Size 分のデータを取得する。
- (共有メモリサイズ - Offset) < Buff\_Size ... (共有メモリサイズ - Offset)分のデータを取得する。

### 制限

- [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

### (3) INpMac\_Get\_Iarea\_Size()

IEリアに割り当てた共有メモリ/Oドライブのメモリサイズを取得します。

```
int INpMac_Get_Iarea_Size(  
    int      GroupNo          // グループ番号  
);
```

#### パラメータ

*GroupNo*                      対象となる1～16のI/Oグループ番号を指定します。

#### 戻り値

関数が成功すると、指定したグループ番号の共有メモリのサイズ(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMAC_SYSTEM_DOWN	0x80000000	INplcが起動していません。
E_INPMAC_GROUP_NO_UNKNOWN	0x80000001	指定したグループ番号が不正です。 共有メモリにアクセスできません。
E_INPMAC_NOT_EXECUTION	0x80000005	INplc環境ではありません。

### (4) INpMac\_Set\_Iarea()

IEリアに割り当てた共有メモリ/Oドライブを経由して、入力値を書込みます。

```
int INpMac_Set_Iarea(  
    int      GroupNo,          // グループ番号  
    int      Offset,          // オフセット  
    BYTE *   lpData,          // データが格納されたバッファ  
    int      Buff_Size        // 書込むサイズ  
);
```

#### パラメータ

*GroupNo*                      対象となる1～16のI/Oグループ番号を指定します。

*Offset*                        取得を開始するメモリ領域のオフセットバイト値を指定します。

*lpData*                        データを格納するバッファへのポインタを指定します。

*Buff\_Size*                    共有メモリから取得するデータサイズ(Byte)を指定します。

#### 戻り値

関数が成功すると、指定したグループ番号の共有メモリへ書込んだデータ数(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMAC_SYSTEM_DOWN	0x80000000	INplcが起動していません。
E_INPMAC_GROUP_NO_UNKNOWN	0x80000001	指定したグループ番号が不正です。 共有メモリにアクセスできません。
E_INPMAC_INVALID_OFFSET	0x80000002	指定したオフセットは無効です。
E_INPMAC_INVALID_PARAM	0x80000004	パラメータ[ <i>lpData</i> ]がNULLです。 パラメータ[ <i>Buff_Size</i> ]が範囲外です。
E_INPMAC_NOT_EXECUTION	0x80000005	INplc環境ではありません。

#### 内部動作

取得するデータサイズは、以下の方式で決定されます。

(共有メモリサイズ - Offset) > Buff\_Size ... Buff\_Size 分のデータを書込む。  
(共有メモリサイズ - Offset) < Buff\_Size ... (共有メモリサイズ - Offset)分のデータを書込む。

#### 制限

- ・ [*lpData*]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (5) INpMac\_Get\_Marea\_Size()

INplcシステムの共有メモリ領域『Mエリア』のメモリサイズを取得します。

```
int INpMac_Get_Marea_Size( void );
```

### パラメータ

なし

### 戻り値

関数が成功すると、Mエリアのサイズ(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMAC_SYSTEM_DOWN	0x80000000	INplcが起動していません。
E_INPMAC_NOT_EXECUTION	0x80000005	INplc環境ではありません。

## (6) INpMac\_Get\_Marea()

INplcシステムの共有メモリ領域『Mエリア』から値を取得します。

```
int INpMac_Get_Marea(  
    int      Offset,           // オフセット  
    BYTE *   lpData,           // 情報を格納するバッファ  
    int      Buff_Size         // 情報バッファのサイズ  
);
```

### パラメータ

Offset	取得を開始するMエリアのオフセットバイト値を指定します。
lpData	データを格納するバッファへのポインタを指定します。
Buff_Size	Mエリアから取得するデータサイズ(Byte)を指定します。

### 戻り値

関数が成功すると、Mエリアから取得したデータ数(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMAC_SYSTEM_DOWN	0x80000000	INplcが起動していません。
E_INPMAC_INVALID_OFFSET	0x80000002	指定したオフセット値は無効です。
E_INPMAC_INVALID_PARAM	0x80000004	パラメータ[lpData]がNULLです。 パラメータ[Buff_Size]が範囲外です。
E_INPMAC_NOT_EXECUTION	0x80000005	INplc環境ではありません。

### 内部動作

取得するデータサイズは、以下の方式で決定されます。

- (Mエリアサイズ - Offset) > Buff\_Size ... Buff\_Size 分のデータを取得する。
- (Mエリアサイズ - Offset) < Buff\_Size ... (Mエリアサイズ - Offset)分のデータを取得する。

### 制限

- [lpData]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## (7) INpMac\_Set\_Marea()

INplcシステムの共有メモリ領域『Mエリア』へ値を書込みます。

```
int INpMac_Set_Marea(  
    int      Offset,           // オフセット  
    BYTE *    lpData,          // データが格納されたバッファ  
    int      Buff_Size         // 書込むサイズ  
);
```

### パラメータ

<i>Offset</i>	取得を開始するMエリアのオフセットバイト値を指定します。
<i>lpData</i>	データを格納するバッファへのポインタを指定します。
<i>Buff_Size</i>	Mエリアから取得するデータサイズ(Byte)を指定します。

### 戻り値

関数が成功すると、Mエリアへ書込んだデータ数(Byte)が返ります。

関数が失敗すると、以下の値が返ります。

E_INPMAC_SYSTEM_DOWN	0x80000000	INplcが起動していません。
E_INPMAC_INVALID_OFFSET	0x80000002	指定したオフセット値は無効です。
E_INPMAC_INVALID_PARAM	0x80000004	パラメータ[lpData]がNULLです。 パラメータ[Buff_Size]が範囲外です。
E_INPMAC_NOT_EXECUTION	0x80000005	INplc環境ではありません。

### 内部動作

取得するデータサイズは、以下の方式で決定されます。

- (Mエリアサイズ - Offset) > Buff\_Size ... Buff\_Size 分のデータを書込む。
- (Mエリアサイズ - Offset) < Buff\_Size ... (Mエリアサイズ - Offset)分のデータを書込む。

### 制限

- [lpData]に不正なポインタを指定しないでください。予期しない結果や保護違反が発生する恐れがあります。

## 4. OCXについて

### 4-1. INpMacWinコントロール



INpMacWin.ocxは、Windows上で動作するウインドウアプリケーション(表示系アプリケーション)に配置して使用します。プロパティ設定に基づいてWindows共有メモリに接続し、情報の設定/取得を行います。メソッドは.NET A/P以外で使用可能で、コントロールのプロパティ設定に関わらず使用できます。.NET A/Pではプロパティのみの使用、またはDLLを使用してください。

#### ■ 使用方法

開発環境にて、[Program Files¥INpMac]にインストールおよび登録されているActiveXコントロールをアプリケーションのプロジェクトに組み込んで使用します。

・ ActiveXコントロール・・・名称【 INpMacWin Control 】ファイル名【 INpMacWin.ocx 】

#### (1) プロパティ

No.	名称	データ型	初期値	機能
1	Mode	BYTE	0	各エリアへのアクセスモードを指定します。 0 本機能を無効化 1 Windows共有メモリへアクセス 2 Windows共有メモリのサイズを取得
2	MemoryName	String	"INpMacWin"	参照する共有メモリの名前を指定します。 ※メモリマップドファイルオブジェクトの名前を指定
3	CreateMemory	LONG	0	参照する共有メモリが見つからない場合、メモリを生成するかを設定します。 0 生成しない 1 生成する
4	Offset	LONG	0	各エリアへのアクセスするオフセットバイト値を指定します。
5	Type	BYTE	0	メモリ領域にアクセスするデータ型を指定します。 0 符号無整数(1Byte) unsigned char 0 ~ 255 1 符号無整数(2Bytes) unsigned short 0 ~ 65535 2 符号無整数(4Bytes) unsigned long 0 ~ 4294967295 3 整数(1Byte) char -128 ~ +127 4 整数(2Bytes) short -32768 ~ +32767 5 整数(4Bytes) long -2147483648 ~ 2147483647 6 実数(4Bytes) float ±10-37 ~ ±1038 7 実数(8Bytes) double ±10-307 ~ ±10308 8 文字列(ワイド) string 最大256文字 + NULL文字 9 文字列(マルチバイト) string 最大256文字 + NULL文字 10 ビット bool 0 or 1
6	Value	VARIANT	0	値の設定によってメモリライト、取得によってメモリリードが行われます。 取り扱うデータ型はTypeプロパティの設定に基づき、メモリへのアクセスバイト数が増減します。データ型の範囲外の値を設定した場合、メモリライトは行われず、Valueプロパティは{0}が設定され、Statusプロパティには【パラメータエラー(E_INPMACWIN_INVALID_PARAM)】が格納されます。Type プロパティで文字列(8,9)を指定した場合、メモリライト時はValue プロパティに設定されたNULL 終端文字列(最大256文字)を書込みます。メモリリード時は、オフセットバイト~NULL 文字、または最大256文字までを取得します。
7	Status	LONG	0	実行結果が返ります。 失敗すると、APIのエラー定義(E_INPMACWIN_~)が格納されます。なお、本プロパティには、最後に実行した結果が格納されます。

#### (2) メソッド

No.	名称	機能
1	void AboutBox(void)	コントロールのバージョン情報ダイアログボックスを表示します。
2	LONG WinCreateMemory(LONG*,BSTR,LONG)	INpMacWin_CreateMemory() に準拠した機能を提供します。
3	LONG WinDeleteMemory(LONG)	INpMacWin_DeleteMemory() に準拠した機能を提供します。
4	LONG WinReadMemory(LONG, LONG, VARIANT*, LONG)	INpMacWin_ReadMemory() に準拠した機能を提供します。
5	LONG WinWriteMemory(LONG, LONG, VARIANT*, LONG)	INpMacWin_WriteMemory() に準拠した機能を提供します。
6	LONG WinMapMemory(LONG, LONG*)	INpMacWin_MapMemory() に準拠した機能を提供します。
7	LONG WinUnMapMemory(LONG)	INpMacWin_UnMapMemory() に準拠した機能を提供します。
8	LONG WinGetSize(LONG)	INpMacWin_GetSize() に準拠した機能を提供します。

### (3) イベント

No.	名称	機能
1	void OnError( int ErrCode )	エラーの発生を示します。パラメータにはAPIのエラーコードが設定されます。

## 4-2. INpMacコントロール

INpMac.ocxは、Windows上で動作するウィンドウアプリケーション(表示系アプリケーション)に配置して使用します。



プロパティ設定に基づいてINplc及びINtimeの共有メモリに接続し、情報の設定/取得を行います。

メソッドは.NET A/P以外で使用可能で、コントロールのプロパティ設定に関わらず使用できます。

.NET A/PではRtReadMemoryCs(), RtWriteMemoryCs(), プロパティのみの使用、またはDLLを使用してください。

### ■ 使用方法

開発環境にて、[Program Files¥INpMac]にインストールおよび登録されているActiveXコントロールをアプリケーションのプロジェクトに組み込んで使用します。

・ ActiveXコントロール・・・名称【 INpMac Control 】ファイル名【 INpMac.ocx 】

### (1) プロパティ

No.	名称	データ型	初期値	機能
1	Mode	BYTE	0	各エリアへのアクセスモードを指定します。 0 本機能を無効化 1 Qエリアのサイズを取得(INplcの出力書き出し用エリアのサイズを取得) 2 Iエリアのサイズを取得(INplcの入力読み込み用エリアのサイズを取得) 3 Qエリアの情報を読み込む(INplcの出力書き出し用エリアの情報を読み込む) 4 Iエリアに対し情報を書き込む(INplcの入力読み込み用エリアへ情報を書き込む) 5 INtime共有メモリへアクセス 6 INtime共有メモリのサイズを取得 7 Mエリアのサイズを取得(INplcの共有メモリ領域のサイズを取得) 8 Mエリアの情報を読み込む(INplcの共有メモリ領域の情報を読み込む) 9 Mエリアに対し情報を書き込む(INplcの共有メモリ領域へ情報を書き込む)
2	GroupNo	BYTE	0	IエリアまたはQエリアのI/Oグループ番号(1～16)を指定します。 ※ Mode プロパティで1～4を指定したときに有効
3	MemoryName	String	"INpMac"	参照する共有メモリの名前を指定します。 ※ Modeプロパティで5～6を指定したときに、[共有メモリのカatalog名]を指定。
4	ProcessName	String	""	共有メモリの名前を定義しているプロセスの名 ※ Modeプロパティで5～6を指定したときに、[共有メモリの名前をCatalogしているプロセスのカatalog名]を指定。(空白の場合、ROOTプロセスと認識します)
5	Offset	LONG	0	各エリアへのアクセスするオフセットバイト値を指定します。
6	Type	BYTE	0	メモリ領域にアクセスするデータ型を指定します。 0 符号無整数(1Byte) unsigned char 0 ～ 255 1 符号無整数(2Bytes) unsigned short 0 ～ 65535 2 符号無整数(4Bytes) unsigned long 0 ～ 4294967295 3 整数(1Byte) char -128 ～ +127 4 整数(2Bytes) short -32768 ～ +32767 5 整数(4Bytes) long -2147483648 ～ 2147483647 6 実数(4Bytes) float ±10-37 ～ ±1038 7 実数(8Bytes) double ±10-307 ～ ±10308 8 文字列(ワイド) string 最大256文字 + NULL文字 9 文字列(マルチバイト) string 最大256文字 + NULL文字 10 ビット bool 0 or 1
7	Value	VARIANT	0	値の設定によってメモライト、取得によってメモリアードが行われます。 取り扱うデータ型はTypeプロパティの設定に基づき、メモリへのアクセスバイト数が増減します。データ型の範囲外の値を設定した場合、メモライトは行われず、Valueプロパティに【0】が設定され、Statusプロパティにはパラメータエラー (E_INPMAC_INVALID_PARAM, または E_INPMACRT_INVALID_PARAM) が格納されます。 Type プロパティで文字列(8, 9)を指定した場合、メモライト時はValue プロパティに設定されたNULL 終端文字列(最大256文字)を書込みます。メモリアード時は、オフセットバイト～NULL 文字、または最大256文字までを取得します。
8	Status	LONG	0	実行結果が返ります。 失敗すると、APIのエラー定義(E_INPMAC_~, E_INPMACRT_~)が格納されます。なお、本プロパティには、最後に実行した結果が格納されます。

## (2) メソッド

No.	名称	機能
1	void AboutBox(void)	コントロールのバージョン情報ダイアログボックスを表示します。
2	LONG GetQareaSize(LONG)	INpMac_Get_Qarea_Size() に準拠した機能を提供します。
3	LONG GetQarea(LONG,LONG,BYTE*,LONG)	INpMac_Get_Qarea() に準拠した機能を提供します。(※1)
4	LONG GetIareaSize(LONG)	INpMac_Get_Iarea_Size() に準拠した機能を提供します。
5	LONG SetIarea(LONG,LONG,BYTE*,LONG)	INpMac_Set_Iarea() に準拠した機能を提供します。(※1)
6	LONG GetMareaSize(void)	INpMac_Get_Marea_Size() に準拠した機能を提供します。
7	LONG GetMarea(LONG,BYTE*,LONG)	INpMac_Get_Marea() に準拠した機能を提供します。(※1)
8	LONG SetMarea(LONG,BYTE*,LONG)	INpMac_Set_Marea() に準拠した機能を提供します。(※1)
9	LONG GetQareaEx(LONG,LONG,VARIANT*,LONG)	INpMac_Get_Qarea() に準拠した機能を提供します。
10	LONG SetIareaEx(LONG,LONG,VARIANT*,LONG)	INpMac_Set_Iarea() に準拠した機能を提供します。
11	LONG GetMareaEx(LONG,VARIANT*,LONG)	INpMac_Get_Marea() に準拠した機能を提供します。
12	LONG SetMareaEx(LONG,VARIANT*,LONG)	INpMac_Set_Marea() に準拠した機能を提供します。
13	LONG RtReadMemory(BSTR,BSTR,LONG,VARIANT*,LONG)	INpMacRt_ReadMemory() に準拠した機能を提供します。
14	LONG RtWriteMemory(BSTR,BSTR,LONG,VARIANT*,LONG)	INpMacRt_WriteMemory() に準拠した機能を提供します。
15	LONG RtGetSize(BSTR,BSTR)	INpMacRt_GetSize() に準拠した機能を提供します。
16	LONG RtReadMemoryCs(BSTR,BSTR,LONG,VARIANT,LONG)	INpMacRt_ReadMemory() に準拠した機能を提供します。 (※2)
17	LONG RtWriteMemoryCs(BSTR,BSTR,LONG,VARIANT,LONG)	INpMacRt_WriteMemory() に準拠した機能を提供します。 (※2)

(※1)・・・ACTIVE TOUCHからは利用できません。

(※2)・・・.NET A/Pからのみ利用できます。

### [.NET A/PからのRtReadMemoryCs()、RtWriteMemoryCs()の使用方法について]

.NET A/PでOCX参照する場合、関数の引数・戻り値の型は以下のようになります。

```
int RtReadMemoryCs(string procName, string memName, int offset, ref object lpData, int buff_Size);
```

```
int RtWriteMemoryCs(string procName, string memName, int offset, object lpData, int buff_Size);
```

また、lpDataに渡すobject型の変数は、以下のように配列をobjectの変数に代入してください。

```
int aBuffSize = 100;
object aObj;
byte[] aByte;
aByte = new byte[aBuffSize];
aObj = aByte;
axINpMac1.RtReadMemoryCs("PROC", "MEM", 0, ref aObj, aBuffSize);
```

## (3) イベント

No.	名称	機能
1	void OnError( int ErrCode )	エラーの発生を示します。パラメータにはAPIのエラーコードが設定されます。

## 5. エラーコード一覧

### 5-1. INpMacWinAPI

No.	定義	コード	内容
1	E_INPMACWIN_FAILED_CREATE	-2147418112 [0x80010000]	共有メモリが確保できません。
2	E_INPMACWIN_FAILED_DELETE	-2147418111 [0x80010001]	共有メモリが解放できません。
3	E_INPMACWIN_INVALID_OFFSET	-2147418110 [0x80010002]	指定したオフセット値は無効です。
4	E_INPMACWIN_FAILED_ACCESS	-2147418109 [0x80010003]	共有メモリにアクセスできません。
5	E_INPMACWIN_INVALID_PARAM	-2147418108 [0x80010004]	パラメータが不正です。
6	E_INPMACWIN_NOT_ENOUGH_MEMORY	-2147418107 [0x80010005]	メモリが不足しています。
7	S_INPMACWIN_ALREADY_EXISTS	183 [0x000000B7]	既存のメモリを参照します。

### 5-2. INpMacRtAPI

No.	定義	コード	内容
1	E_INPMACRT_SYSTEM_DOWN	-2147352576 [0x80020000]	指定したプロセスが起動していません。
2	E_INPMACRT_NOT_FOUND	-2147352575 [0x80020001]	指定した共有メモリがありません。
3	E_INPMACRT_INVALID_OFFSET	-2147352574 [0x80020002]	指定したオフセット値は無効です。
4	E_INPMACRT_FAILED_ACCESS	-2147352573 [0x80020003]	共有メモリにアクセスできません。
5	E_INPMACRT_INVALID_PARAM	-2147352572 [0x80020004]	パラメータが不正です。
6	E_INPMACRT_NOT_EXECUTION	-2147352571 [0x80020005]	INtime環境がないため実行できません。

### 5-3. INpMacAPI

No.	定義	コード	内容
1	E_INPMAC_SYSTEM_DOWN	-2147418112 [0x80000000]	INplcが起動していません。
2	E_INPMAC_GROUP_NO_UNKNOWN	-2147418111 [0x80000001]	指定したグループ番号が不正です。
3	E_INPMAC_INVALID_OFFSET	-2147418110 [0x80000002]	指定したオフセット値は無効です。
4	E_INPMAC_MEM_ALIGMENT	-2147418109 [0x80000003]	指定した共有メモリが4KB(4,096Byte)単位で作成されていません。
5	E_INPMAC_INVALID_PARAM	-2147418108 [0x80000004]	パラメータが不正です。
6	E_INPMAC_NOT_EXECUTION	-2147418107 [0x80000005]	INplc環境がないため実行できません。

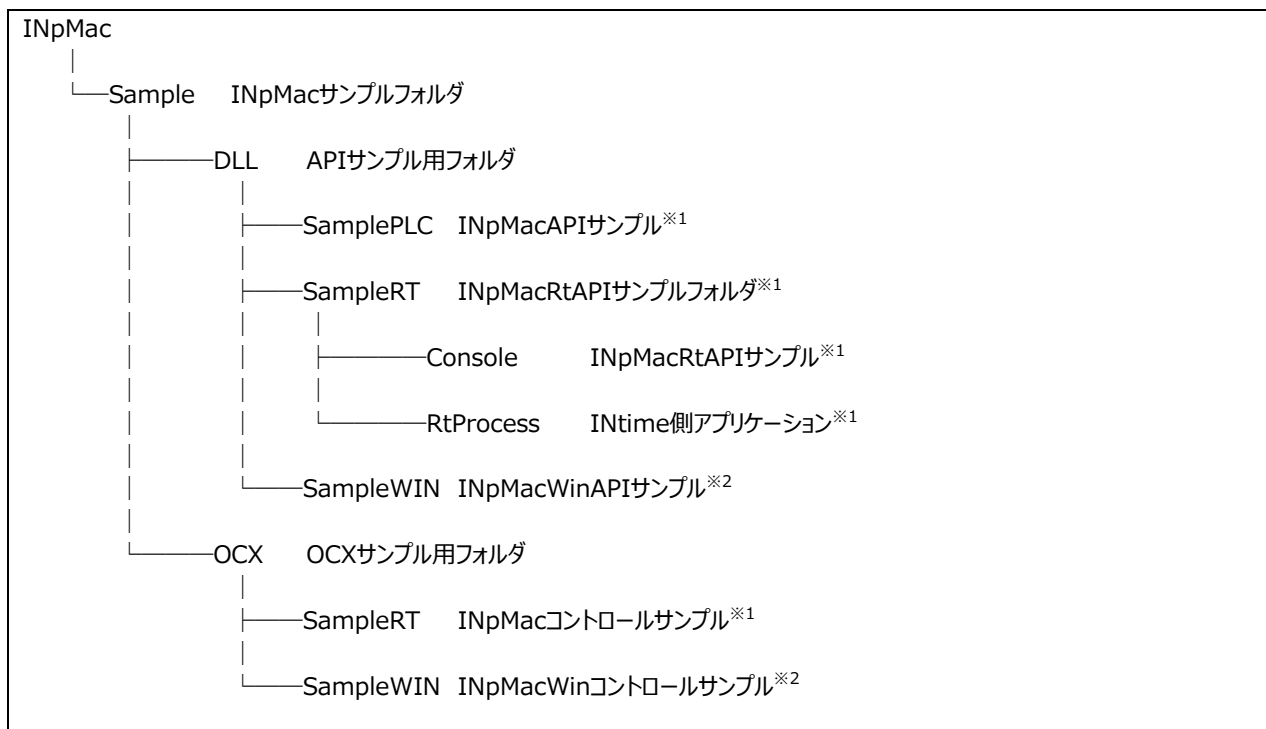


## 6. サンプルプログラム

基本的な使い方を、ソースコード付きのサンプルプログラムとして提供しています。  
サンプルプログラムはできるだけシンプルに作成されていますので、プログラミングの際に活用していただけます。

### 6-1. フォルダ構成

INpMacの開発環境【Develop】をインストールしたフォルダに、以下のように構成されます。



※1 インストール時に【INtime】機能が選択されていた場合に配置されます。  
※2 インストール時に【Windows】機能が選択されていた場合に配置されます。

### 6-2. プログラムの内容

サンプルプログラムは「DLLサンプル」、「OCXサンプル」の2種類に分けられます。  
DLLサンプルは、Visual C++ 2008 の Win32コンソールアプリケーションで作成しています。  
コンソール上にてMenu Listから選択したAPIを、入力された引数で実行し、処理結果を表示します。  
OCXサンプルは、Visual C# 2008 の フォームアプリケーションで作成しています。  
画面上にてOCXのプロパティを変更し、処理結果を表示します。

DLLサンプル	
DLL¥SampleWIN	INpMacWinAPIを使用して、Windows のメモリ領域にアクセスするサンプルです。
DLL¥SampleRT	INpMacRtAPIを使用して、INtimeのメモリ領域にアクセスするサンプルです。 ※ 付属のINtime側アプリケーションを実行してから使用してください。
DLL¥SamplePLC	INpMacAPIを使用して、INplcのメモリ領域にアクセスするサンプルです。 ※ (7. 使用例 - 7-2. INplcシステムを介したエコー動作テスト)を参照してください。
OCXサンプル	
OCX¥SampleWIN	INpMacWin コントロールを介して、Windows のメモリ領域にアクセスするサンプルです。
OCX¥SampleRT	INpMac コントロールを介して、INtime及びINplcのメモリ領域にアクセスするサンプルです。

## (1) DLLサンプルの概要

```
////////////////////////////////////////
// Shared Memory I/F for Windows SAMPLE
////////////////////////////////////////

Set menu (100:Menu list) >> 100
** [ Menu ] *****
** ID NAME          | ID NAME
** 1 API Test       | 2 Memory Dump
** 99 End

Set menu (100:Menu list) >> 1
*****
** API Test
*****

Set API menu (100:Menu list) >> 100
** [ API Menu ] *****
** ID NAME          | ID NAME
** 1 CreateMemory   | 2 DeleteMemory
** 3 ReadMemory     | 4 WriteMemory
** 5 GetSize        | 6 CreateMemory (Unlimited)
** 7 DeleteMemory (Unlimited) | 8 ReadMemory (Unlimited)
** 9 WriteMemory (Unlimited) | 10 GetSize (Unlimited)
** 99 Back

Set API menu (100:Menu list) >> 1
MemoryName -> TEST01
MemorySize -> 4096
iRet = 0(00000000h) | HANDLE = 0x44
```

APIを使用して、Windowsの共有メモリ領域にアクセスするサンプルです。

本サンプルは、メニュー番号と設定値を入力することで各APIの処理を実行します。

### ■ 主なメニュー番号

メニュー番号 "100" :

メニューの一覧が表示されます。

メニュー番号 "1" :

APIメニューが表示され、各APIのテストが行えます。

メニュー番号 "99" :

サンプルを終了します。APIメニューが表示されている場合は、メインメニューに戻ります。

## (2) OCXサンプルの概要

Mode: [Windows Memory] Data Read / Write

MemoryName: INpMacWin CreateMemory: ☒

Offset: 0 Bit Offset: 0

Type: string(multi-byte)

Value (NULL)=Read: TestData!

Status: 0

SET

Mode: [Windows Memory] Data Read / Write

MemoryName: INpMacWin

Offset: 0 Bit Offset: 0

Type: string(multi-byte)

Value: "TestData!"

Status: 0(00000000h)

ActiveXコントロールを介して、Windowsの共有メモリ領域にアクセスするサンプルです。

本サンプルは、ActiveXコントロールのプロパティと一対一になっており、画面中央のSETボタンを押すことで処理を実行します。

### ■ 画面概要

画面上部 :

プロパティに設定する値を指定します。

画面中央 :

SETボタンを押すことで処理を実行します。

画面下部 :

実行結果を表示します。

## 7. 使用例

### 7-1. Windows用共有メモリのダンプ

サンプルを使用して、Windows用共有メモリの内容をダンプすることができます。

※ APIを使用するアプリケーションとして、[Program Files¥INpMac¥Sample¥DLL¥SampleWIN]にインストールされる INpMacWinAPI用サンプル(Win32 コンソールアプリケーション)を使用します。

```
////////////////////////////////////
// INpMacWin SAMPLE
////////////////////////////////////

Set menu (100:Menu list) >> 100
** [ Menu ] *****
** ID NAME          | ID NAME
** 1 API Test       | 2 Memory Dump
** 99 End

Set menu (100:Menu list) >> 2
*****
** Memory Dump
*****

Set Dump menu (100:Menu list) >> 100
** [ Menu ] *****
** ID NAME
** 1 Dump Start
** 99 Back

Set Dump menu (100:Menu list) >> 1
MemoryName (MAX:12byte) --> MemTest
Offset      (MAX:65535)  --> 0
Dump_Size   (MAX:65536)  --> 5000

MEM_SZ = 8192(00002000h) | HANDLE = 0x3C | ADDR = 0x240000

Dump_Size = 5000(00001388h)
0000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
:
:
1340: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1350: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1360: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1370: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
1380: 00 00 00 00 00 00 00 00 .....

Set Dump menu (100:Menu list) >>
```

メニューリストの表示

ダンプのメニューリストの表示

MemoryName及びダンプ条件を指定した後、INpMacWin\_CreateMemory() INpMacWin\_MapMemory()を行い、ポインタを取得。

メモリの内容を表示

## 7-2. INplcシステムを介したエコー動作テスト

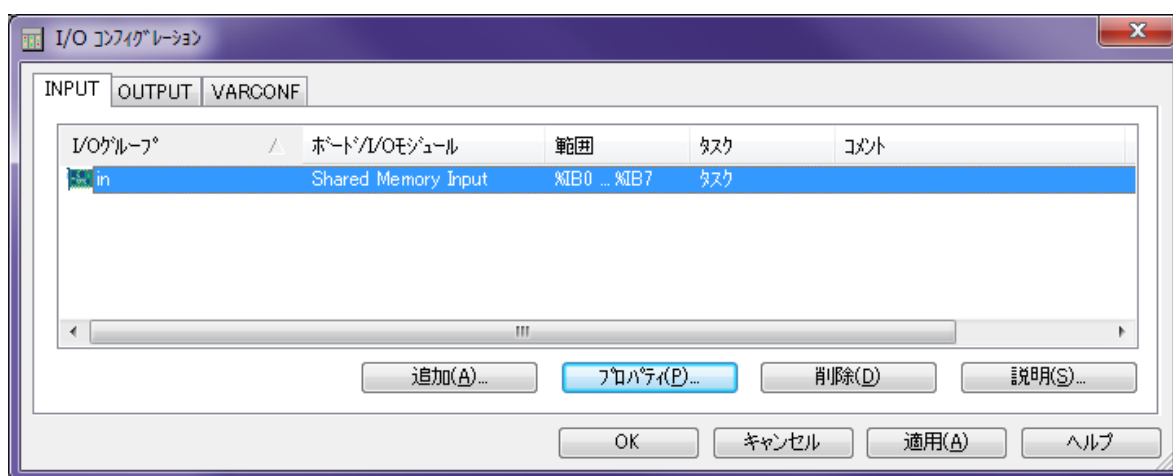
IEリアへ書込んだデータをQエリアから読み込む動作を行う手順を解説します。

- ※ INplcを動作させるために、INplcの実行環境が必要です。
- ※ INplcに共有メモリ/ODライバ[MemDrv.iopack]がインストールされている必要があります。
- ※ APIを使用するアプリケーションとして、[Program Files¥INpMac¥Sample¥DLL¥SamplePLC]にインストールされるサンプルアプリケーション(Win32 コンソールアプリケーション)を使用します。

### (1) PLCプログラムの設定

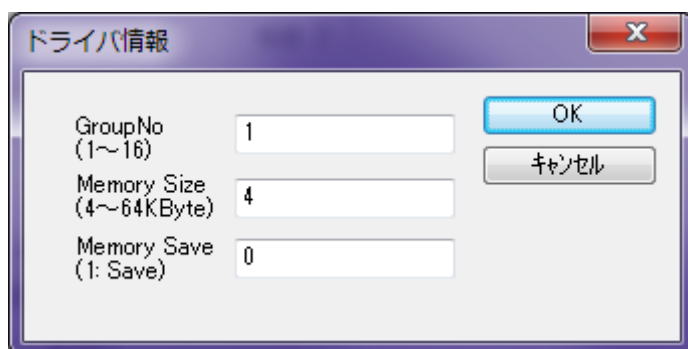
IEリアのデータをQエリアへ出力するPLCプログラムを作成します。

- 1) MULTIPROGを起動し、新しいプロジェクトを作成します。
- 2) IO\_Configurationの[INPUT]から、IEリアの情報を設定します。

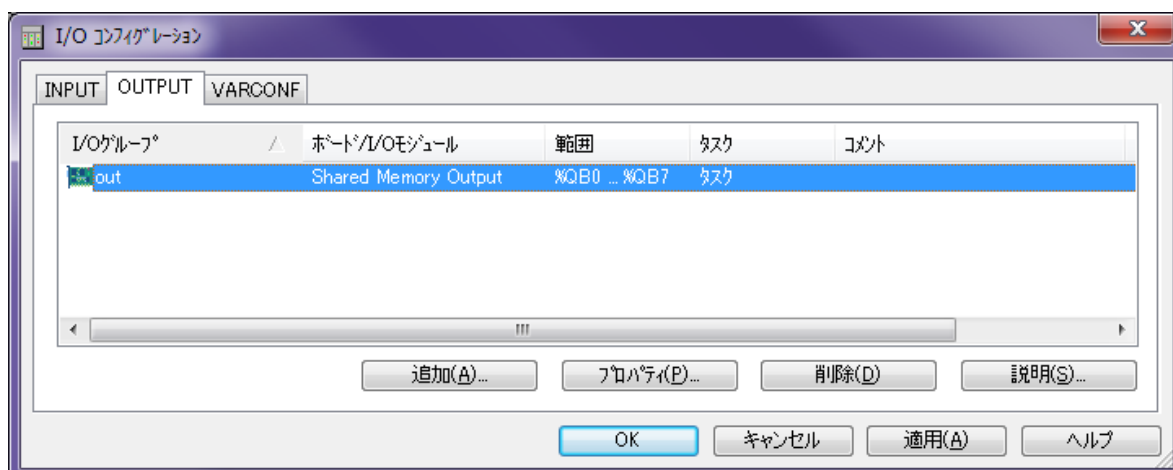


【IEリアの設定】

GroupNo : 1  
Memory Size : 4 KByte  
Memory Save : 0

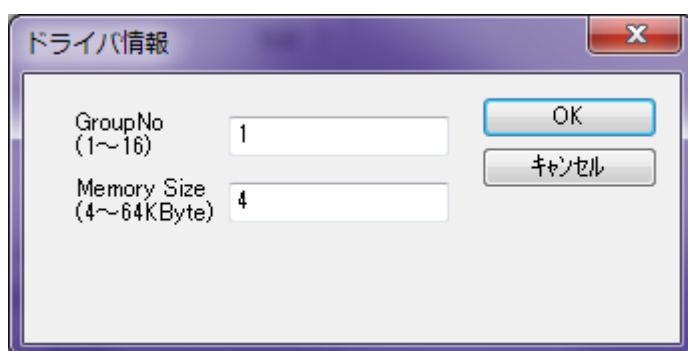


3) 同様に[OUTPUT]から、Qエリアの情報を設定します。

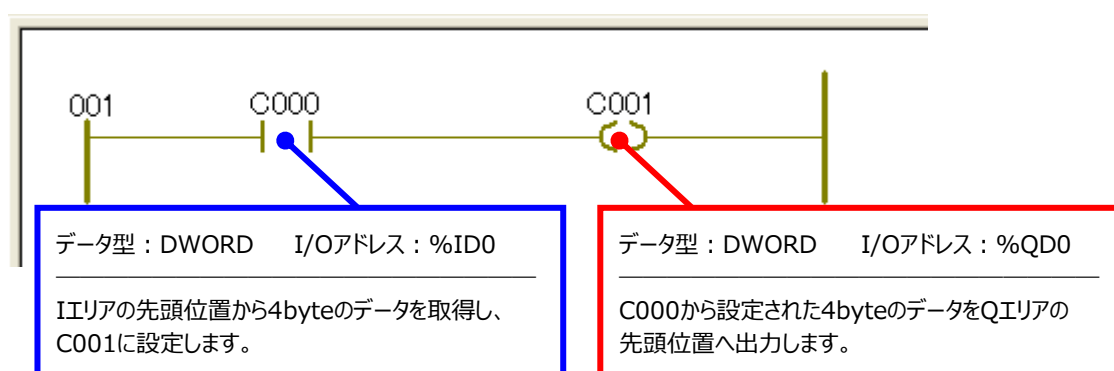


【Qエリアの設定】

GroupNo : **1**  
Memory Size : **4** KByte



4) 以下のようなラダーを作成して、INplc コントローラにダウンロードします。



以上で、PLCプログラムの設定は完了です。

## (2) Iエリアヘデータを書込む

INpMacのAPI【INpMac\_Set\_Iarea()】を使用して、Iエリアにデータを書込みます。

以下は、サンプルアプリケーションを使用した際の画面イメージです。

データ["12345678"]計8Byteを、Iエリア(GroupNo:1)の先頭から書込みます。

```
////////////////////////////////////
// INpMac SAMPLE
////////////////////////////////////

Set menu (100:Menu list) >> 100
** [ Menu ] *****
** ID NAME | ID NAME
** 1 [Q-Area] Get_Size | 2 [Q-Area] Get_Data
** 3 [I-Area] Get_Size | 4 [I-Area] Set_Data
** 5 [M-Area] Get_Size | 6 [M-Area] Get_Data
** 7 [M-Area] Set_Data
** 99 End

Set menu (100:Menu list) >> 4
GroupNo --> 1
Offset --> 0
Set Data (MAX:256Byte) --> 12345678

I-Area Set_Data = 8(0x00000008)
0000: 31 32 33 34 35 36 37 38
Set menu (100:Menu list) >>
```

メニューリストの表示

INpMac\_Set\_Iarea()を実行

```
Data = {"12345678"};
INpMac_Set_Iarea( 1, 0, Data, sizeof(Data) );
```

INpMac\_Set\_Iarea()の戻り値を表示  
(エラーがなければ、書込んだデータをダンプする)

12345678

## (3) Qエリアのデータを取得

INpMacのAPI【INpMac\_Get\_Qarea()】を使用して、Qエリアのデータを取得します。

以下は、サンプルアプリケーションを使用した際の画面イメージです。

Qエリア(GroupNo:1)の先頭から、8Byte分のデータを取得します。

```

////////////////////////////////////
// INpMac SAMPLE
////////////////////////////////////

Set menu (100:Menu list) >> 100
** [ Menu ] *****
** ID NAME | ID NAME
** 1 [Q-AEA] Get_Size | 2 [Q-AEA] Get_Data
** 3 [I-AEA] Get_Size | 4 [I-AEA] Set_Data
** 5 [M-AEA] Get_Size | 6 [M-AEA] Get_Data
** 7 [M-AEA] Set_Data
** 99 End

Set menu (100:Menu list) >> 2
GroupNo --> 1
Offset --> 0
Buff_Size --> 8

Q-AEA Get_Data = 8(0x00000008)
0000: 31 32 33 34 00 00 00 00
Set menu (100:Menu list) >>

```

メニューリストの表示

INpMac\_Get\_Qarea()を実行  
INpMac\_Get\_Qarea( 1, 0, Data, 8 );

INpMac\_Get\_Qarea()の戻り値を表示  
(エラーがなければ、取得したデータをダンプする)

※ 取得した8Byteのデータの内、5～8Byte目が[0]となっている理由は、PLCプログラムでDWORD型(4Byte)のデータのみをQエリア側に出力しているためです。



MULTIPROGをデバッグモードにすると、以下のような入力値と出力値が確認できます。

	名前	オンライン値	型
	<b>Default</b>		
	C000	16#34333231	DWORD
	C001	16#34333231	DWORD

### 7-3. HMIソフト『ACTIVE TOUCH』での使用方法

ACTIVE TOUCH上でINpMac OCX コントロールを使用する際の基本的手順について説明します。  
ACTIVE TOUCH Editorの詳細につきましては、ACTIVE TOUCHのマニュアル等をご参照ください。

#### (1) コントロール(部品)の登録

コントロール	ファイル名	用途
	INpMacRt.ocx	INtime共有メモリ、INplc共有メモリ、I/Oイメージへのアクセス
	INpMacWin.ocx	Windowsメモリへのアクセス

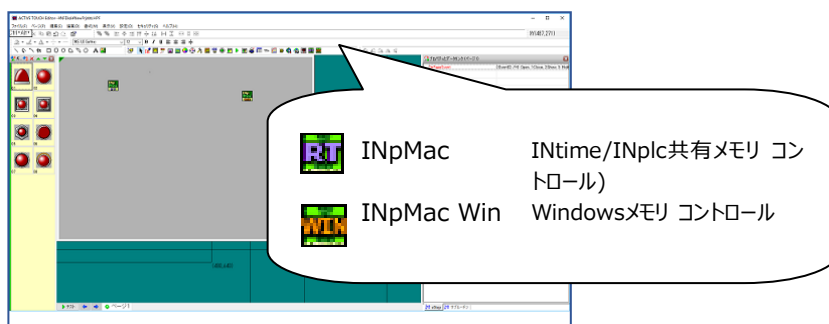
ACTIVE TOUCH Editorに、INpMacコントロールを登録する手順を以下に示します：

- 1) ACTIVE TOUCH Editorを起動します。
- 2) ACTIVE TOUCHプロジェクトを未作成の場合、ACTIVE TOUCH Editorの【ファイル】メニューから“プロジェクト新規作成”を選択し、プロジェクトを作成します。  
表示されるウィザードに従って、プロジェクトの初期設定を行います。
- 3) コントロールの登録を行います。
  - ① ツールバーの“コントロールの登録と解除”アイコンをクリックして、コントロールの登録と削除画面を起動します。
  - ② コントロールの登録と解除画面の【利用可能なコントロール】リストの中から利用するコントロールを選択して【追加】ボタンをクリックします：

コントロール表示名	用途
INpMac Control	INtime共有メモリ、INplc共有メモリ、I/Oイメージへのアクセス
INpMac Win Control	Windowsメモリへのアクセス



- ③ 画面下部の【登録済みコントロール】リストの中に選択したコントロールが表示されていることを確認後、【閉じる】ボタンにより【コントロールの登録と解除】画面を終了します。
- ④ ツールバーに登録したコントロールのアイコンが表示を確認し、コントロールの登録を完了します。





## (2) データの関連付け

コントロールの登録後、データの関連付けを行います：

1) 登録したINpMacアイコンをコントロールバーから選択し、画面ページ上にコントロールを作成します。

2) コントロール上で右クリックにより表示されるポップアップメニュー 上から、[プロパティ]を選択します。



3) プロパティ画面上で各項目を設定します。

※プロパティ設定は、INpMacとINpMac Winにて設定項目数に相違点があります：

### ■ INpMac (INtime/INplc)

INpMac RTのプロパティ画面では以下のようにプロパティの各項目を設定します：

名称	概要
モード	各エリアへのアクセスモードを指定します： 0 本機能を無効化 1 Qエリアのサイズを取得(INplcの出力書出し用エリアのサイズを取得) 2 Iエリアのサイズを取得(INplcの入力読み込み用エリアのサイズを取得) 3 Qエリアの情報を読み込む(INplcの出力書出し用エリアの情報を読み込む) 4 Iエリアに対し情報を書き込む(INplcの入力読み込み用エリアへ情報を書き込む) 5 INtime共有メモリへアクセス 6 INtime共有メモリのサイズを取得 7 Mエリアのサイズを取得(INplcの共有メモリ領域のサイズを取得) 8 Mエリアの情報を読み込む(INplcの共有メモリ領域の情報を読み込む) 9 Mエリアに対し情報を書き込む(INplcの共有メモリ領域へ情報を書き込む)
グループNo.	IエリアまたはQエリアのI/Oグループ番号(1～16)を指定します。 ※ Mode プロパティで1～4を指定したときに有効(INplcアクセス時のみ)
プロセス名	共有メモリの名前を定義しているプロセスの名 ※ Modeプロパティで5～6を指定したときに、[共有メモリの名前をカタログしているプロセスのカタログ名]を指定。 空白の場合、ROOTプロセスと認識します(INtimeアクセス時のみ)
メモリ名	参照する共有メモリの名前を指定します。 ※ Modeプロパティで5～6を指定したときに、[共有メモリのカタログ名]を指定(INtimeアクセス時のみ)
データ型	メモリ領域にアクセスするデータ型を指定します。 0 符号無整数(1Byte) unsigned char 0 ～ 255 1 符号無整数(2Bytes) unsigned short 0 ～ 65535 2 符号無整数(4Bytes) unsigned long 0 ～ 4294967295 3 整数(1Byte) char -128 ～ +127 4 整数(2Bytes) short -32768 ～ +32767 5 整数(4Bytes) long -2147483648 ～ 2147483647 6 実数(4Bytes) float ±10-37 ～ ±1038 7 実数(8Bytes) double ±10-307 ～ ±10308 8 文字列(ワイド) string 最大256文字 + NULL文字 9 文字列(マルチバイト) string 最大256文字 + NULL文字 10 ビット bool 0 or 1
オフセット	各エリアへのアクセスするオフセットバイト値を指定します。

## ■ INpMac Win (Windows)

INpMac Winのプロパティ画面では、以下のようにプロパティの各項目を設定します：

名称	概要
モード	【Windows共有メモリへアクセスする】にすることで、Windows共有メモリにアクセスできるようになります。
メモリ名	参照するWindows共有メモリのメモリ名を指定します。
データ型	メモリ領域にアクセスするデータ型を指定します。 データ型についてはP.29の「4-1. (1) プロパティ」にある"Type"プロパティの説明を参照してください。
オフセット	メモリ領域へアクセスするオフセット値を指定します。(バイト単位)
メモリ領域を生成する	チェックをONにすると、参照するメモリが見つからない場合に新規に共有メモリを生成します。 (生成サイズ：4,096Byte)

4) プロパティ設定後、INpMacの「プロパティとデータリンク」を設定します。

[Value]プロパティを選択し、データを受け渡すための変数を設定します。  
[データリンク]の領域に設定した名称でACTIVE TOUCH上のオブジェクトからデータにアクセスするための変数として生成されます。  
※例では、[BitData]としています。

設定したデータリンク領域の変数入出力設定

データリンクエリアの ボタンにより、データの入出力方向を決定します：

アイコン	方向
	共有メモリ基準入力：メモリに対し、書込専用としてデータリンクを割り当てます。ACTIVE TOUCHアプリケーションは変数データに対し書き込みのみ可能となります。
	共有メモリ基準出力：メモリに対し、読取専用としてデータリンクを割り当てます。ACTIVE TOUCHアプリケーションは変数データに対し読み込みのみ可能となります。
	共有メモリ基準入出力：メモリに対し、読み書き可能としてデータリンクを割り当てます。ACTIVE TOUCHアプリケーションは変数データの読み書きが可能となります。

データ・ツリー表示から設定した変数の登録状況を確認します：

データリンクに設定した文字[BitData]変数が登録されていることを確認します。

ACTIVE TOUCH上のオブジェクトは本変数を割り当てデータ参照・更新を行います。

## 8. 更新履歴

版

日付

更新説明

1 4

2 0 2 0 . 1

● 全体的に記載内容を見直しました。

## INpMac 取扱説明書

株式会社 **マイクロネット**

TEL: +81(0)299-90-1733

FAX: +81(0)299-92-8557

初 版 2010 年 11 月 09 日

第十四版 2020 年 01 月 23 日

▷ Windowsは、米国Microsoft Corporationの米国およびその他の国における商標または登録商標です。

▷ ACTIVE TOUCHは、株式会社コンテックの登録商標です。

▷ その他、本書に記載されている会社名、商品名は、各社の商標または登録商標です。

▷ 本書の内容を無断で転載することは禁止されています。

▷ 本書の内容に関しては、予告なしに変更することがあります。あらかじめご了承ください。

©Micronet Corporation 2010-2020