

b-CAP INtime プロバイダ b-CAP 通信

Version 1.0.0

ユーザーズ ガイド

May 20, 2015

【備考】

目次

1. はじめに	4
2. プロバイダの概要	5
2.1. 概要	5
2.1.1. b-CAP INtime プロバイダのセットアップ	6
2.2. メソッド・プロパティ	7
2.2.1. CaoWorkspace::AddController メソッド	7
2.2.2. AddController 以外のメソッド・プロパティ	8
3. INtime b-CAP リスナ	9
3.1. 概要	9
3.2. メソッド・プロパティ	9
3.2.1. ListenerInitialize	9
3.2.2. ListenerStart	9
3.2.3. ListenerStop	10
3.2.4. ListenerSetCallbackFunc	10
3.2.5. ListenerSetProcessName	11
3.2.6. ListenerSetMailboxName	11
3.2.7. ListenerSetInterval	11
3.2.8. ListenerGetStatus	12

1. はじめに

本書は b-CAP を使用して、INtime カーネル上のリアルタイムアプリケーションと通信を行うプロバイダである b-CAP INtime プロバイダのユーザーズガイドです。

b-CAP は CAP の概念を踏襲しつつ、通信速度の向上を狙ったプロトコルです。b-CAP の詳細は「b-CAP 通信仕様書」(b-CAP_Spec_ja.pdf)をご参照ください。

本書は、この b-CAP INtime プロバイダの機能と実装されているメソッドについて説明します。

2. プロバイダの概要

2.1. 概要

b-CAP INtime プロバイダは、INtime のメールボックス機能を利用して、Windows 上のアプリケーションと INtime カーネル上のリアルタイムアプリケーションの間で b-CAP メッセージを通信します。

以下に b-CAP INtime プロバイダを用いたシステムの構成図を示します。

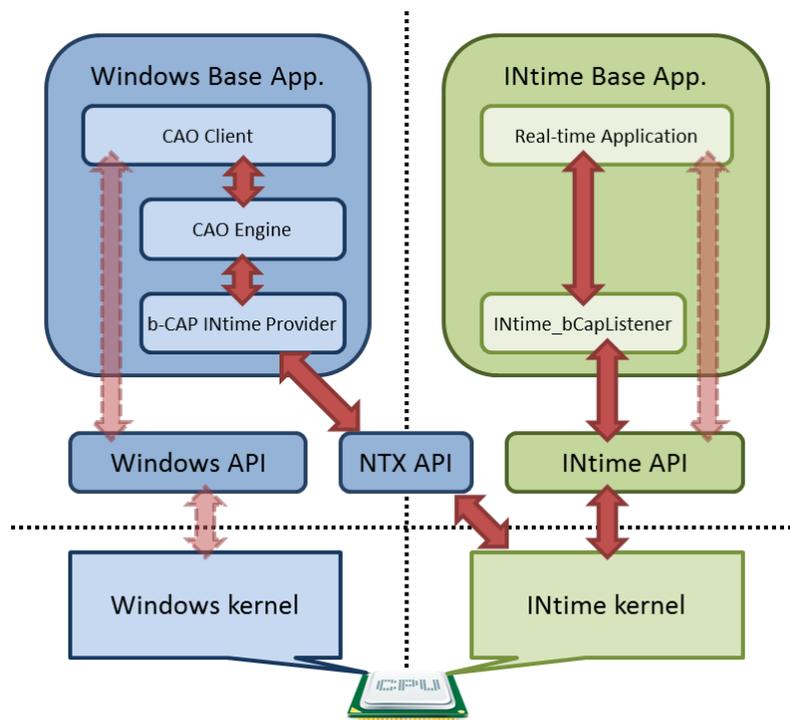


図 2-1 b-CAP INtime システム構成図

b-CAP INtime プロバイダが送信したメッセージを INtime カーネル上で処理するためのライブラリとして INtime b-CAP リスナがあります。リアルタイムアプリケーションに INtime b-CAP リスナを組み込むと、メッセージ受信スレッドが起動されます。INtime b-CAP リスナはメッセージを受信すると対応する関数をコールバックすることで Windows アプリケーションからの要求をリアルタイムアプリケーションに通知します。

b-CAP INtime プロバイダでは、パケットの最大サイズは 504 バイトです。504 バイトを超えるメッセージを送信しようとした場合、E_TOO_MUCH_DATA(0x80000909L)が返ります。

INtime b-CAP リスナを組み込んだリアルタイムアプリケーションを開発する場合は、「b-CAP サーバ&クライアント ユーザーズガイド」(b-CAP_Server_Client_Guide_ja.pdf) および付属のサンプルを参照ください。

2.1.1. b-CAP INtime プロバイダのセットアップ

b-CAP INtime プロバイダのファイル形式は DLL(Dynamic Link Library)であり, CAO エンジンから使用時に動的にロードされます. b-CAP INtime プロバイダを使用するにあたっては下表を参照して手作業でレジストリ登録を行う必要があります. このとき INtime がインストールされている必要があります.

表 2-1 b-CAP INtime プロバイダ

ファイル名	b-CAP_INtime.dll
ProgID	CaoProv.b-CAP.INtime
レジストリ登録 ¹	regsvr32 b-CAP_INtime.dll
レジストリ登録の抹消	regsvr32 /u b-CAP_INtime.dll

¹ ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません.

【参考】

AddControllerメソッドを実行すると、b-CAPの(1)Service_Start, (2)Controller_Connectの2つの関数がこの順で呼び出されます。また、AddControllerにより作成されたCaoControllerオブジェクトが消滅する際には、(3) Controller_Disconnect, (4) Service_Stopの2つの関数がこの順で呼び出されます。

2.2.2. AddController 以外のメソッド・プロパティ

b-CAP INtime プロバイダは、コントローラ、ロボット、ファイル、タスク、変数、拡張ボードクラスのすべてのメソッド、プロパティが実装されています。前述の AddController(2.2.1)以外のメソッド、プロパティは INtime b-CAP リスナのコールバックによって実行されます。

3. INtime b-CAP リスナ

3.1. 概要

INtime b-CAP リスナは、b-CAP メッセージを受け取り、対応する関数をコールバックします。b-CAP による通信を行う時にはあらかじめ INtime カーネル上で INtime b-CAP リスナを組み込んだリアルタイムアプリケーションを起動しておく必要があります。

3.2. メソッド・プロパティ

3.2.1. ListenerInitialize

書式

```
HRESULT ListenerInitialize(BOOL bForceCatalog);
```

説明

INtime b-CAP リスナをカタログに登録する。

引数

[in] bForceCatalog 既にカタログに登録されている場合、削除して再登録するかどうか

返回值

成功した場合は S_OK(0)を、失敗した場合は負値を返す。

備考

プロセスハンドルおよびメールボックスハンドルがカタログに登録されます。

登録される際の名前は、初期設定ではプロセスハンドルが“IntimeBCap”，メールボックスハンドルが“BCapInBox”です。

変更する場合は、ListenerSetProcessName や ListenerSetMailboxName を実行します。

3.2.2. ListenerStart

書式

```
HRESULT ListenerStart(void);
```

説明

メッセージ受信スレッドを起動する。

引数

なし

返回值

成功した場合は S_OK(0)を、失敗した場合は負値を返す。

備考

ListenerInitialize 実行後に呼ばれる必要があります。未実行状態で ListenerStart を実行すると E_FAIL が返ります。

3.2.3. ListenerStop

書式

```
HRESULT ListenerStop(void);
```

説明

メッセージ受信スレッドを停止する。

引数

なし

返回值

成功した場合は S_OK(0)を, 失敗した場合は負値を返す。

備考

なし

3.2.4. ListenerSetCallbackFunc

書式

```
HRESULT ListenerSetCallbackFunc(DWORD dwId, CALLBACK func, void *obj);
```

説明

メッセージ受信時にコールバックされる関数を指定する。

引数

[in]	id	b-CAP 関数 ID
[in]	func	コールバック関数
[in]	obj	汎用ポインタ

コールバック関数

```
typedef struct {
    VARIANT          *vntInPara;      // 受信パラメータ配列
    int              nInParaNum;      // 受信パラメータ数
    VARIANT          *pvntOutPara;    // 応答パラメータポインタ
    void             *obj;            // 汎用ポインタ
    // (コールバック関数設定時に渡したポインタが返される)
} CALLBACK_T;
```

```
typedef HRESULT (*CALLBACK)(CALLBACK_T strCallback);
```

[in]	strCallback	コールバック引数
------	-------------	----------

返回值

成功した場合は S_OK(0)を, 失敗した場合は負値を返す。

備考

コールバック関数の引数の詳細は、「b-CAP サーバ&クライアント ユーザーズガイド」を参照。

3.2.5. ListenerSetProcessName

書式

```
void ListenerSetProcessName(LPSTR name);
```

説明

カタログに登録するプロセスハンドル名を指定する。

引数

[in] name プロセスハンドル名

返回值

なし

備考

プロセスハンドル名は 11 文字以下である必要があります。

3.2.6. ListenerSetMailboxName

書式

```
void ListenerSetMailboxName(LPSTR name);
```

説明

カタログに登録するメールボックスハンドル名を指定する。

引数

[in] name メールボックスハンドル名

返回值

なし

備考

メールボックスハンドル名は 11 文字以下である必要があります。

3.2.7. ListenerSetInterval

書式

```
void ListenerSetInterval(DWORD dwInterval);
```

説明

受信スレッドのメッセージ確認周期を指定する。

引数

[in] dwInterval メッセージ確認周期

返回值

なし

備考

なし

3.2.8. ListenerGetStatus

書式

```
int ListenerGetStatus(void);
```

説明

INtime b-CAP リスナの状態を取得する。

引数

なし

返り値

INtime b-CAP リスナの状態

- 0: 未初期化 (ListenerInitialize 未実行)
- 1: カタログ登録中 (ListenerInitialize 実行中)
- 2: カタログ登録完了 (ListenerInitialize 完了)
- 3: 受信スレッド起動 (ListenerStart 完了)
- 4: 受信スレッド停止中 (ListenerStop 実行中)
- 5: 終了処理実行中

備考

なし