

# e-CAP 通信仕様書

Version 1.0

October 15, 2008

【備考】

**【改版履歴】**

日付	版数	内容
2008-10-15	1.0.0	初版

## 目次

1. はじめに.....	4
2. e-CAP の構造.....	5
3. e-CAP メッセージ.....	7
3.1. メッセージ構造.....	7
3.2. メッセージ規則.....	7
3.3. 関数 ID.....	8
3.4. リターンコード.....	12
3.5. VARIANT データ書式.....	13

## 1. はじめに

本仕様書は、e-CAP(Embedded CAP)の通信プロトコルを規定するものです。

このe-CAPと同様にリモートにコマンドの送受信を行う手段として、CAP(Controller Access Protocol)があります。しかし、CAPではSOAPを採用しているため、サーバ側がWindowsマシンでない場合は、実装に手間がかかります。

e-CAPは、CAPよりも簡素になるように設計されています。これによりe-CAPは以下のような特徴を持っています。

- XMLを使用しないでHTTPのみによる単純化。
- 複雑なデータ表現(多次元構造体配列等)はサポートされていない。
- その他の機能はCAPと同等。

## 2. e-CAP の構造

e-CAP は、CAO プロバイダのオブジェクトモデルと同様なサービス構造を持ち、1つの e-CAP メッセージが1つのサービス(関数)に対応する。この構造を以下に図で示します。

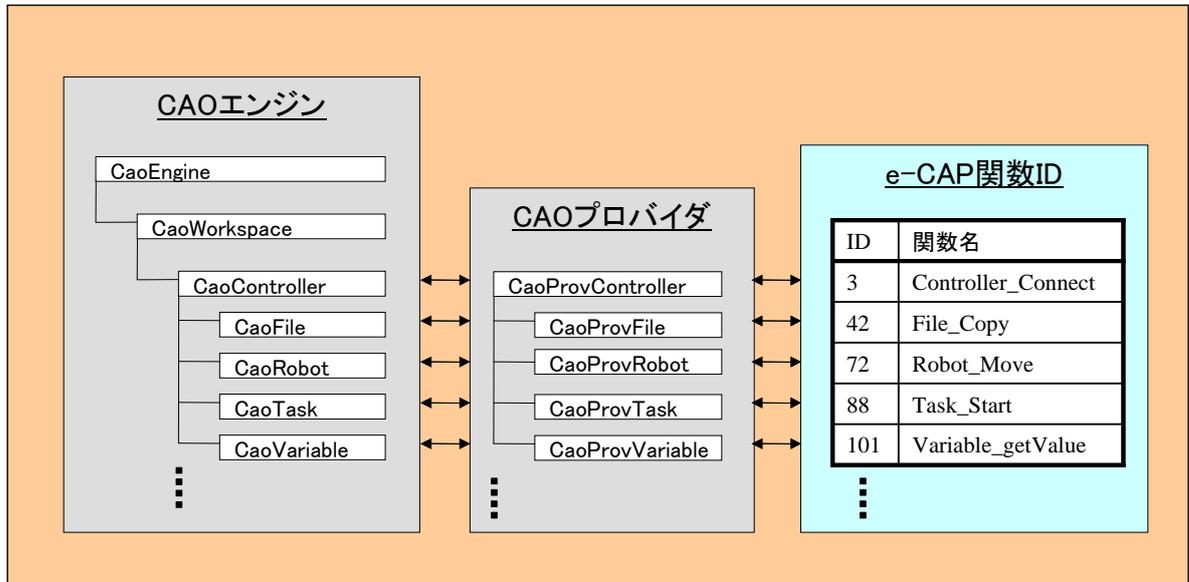


図 2-1 e-CAP の構造

e-CAP は、HTTP を用いて CGI のように利用したい機能を表現することにより、サーバ側のコマンド解析の負荷を軽減し、CAP と比べて実装を簡易化するように設計されています。

e-CAP は、サービスを要求する e-CAP クライアントとサービスを実行し結果を返す e-CAP サーバの 2 つのプログラムで構成される。

e-CAP クライアントは要求するサービスに必要な情報を格納した HTTP リクエストメッセージを作成し、サーバ側に送信し、HTTP レスポンスメッセージを受信して実行結果を確認する。

e-CAP サーバでは、クライアントからの HTTP リクエストメッセージを受信し、関数 ID に対応するサービスを実行する。サービス実行後は、結果及び値を HTTP レスポンスメッセージに格納し、クライアント側に送信する。

以下に b-CAP による接続例を示す。

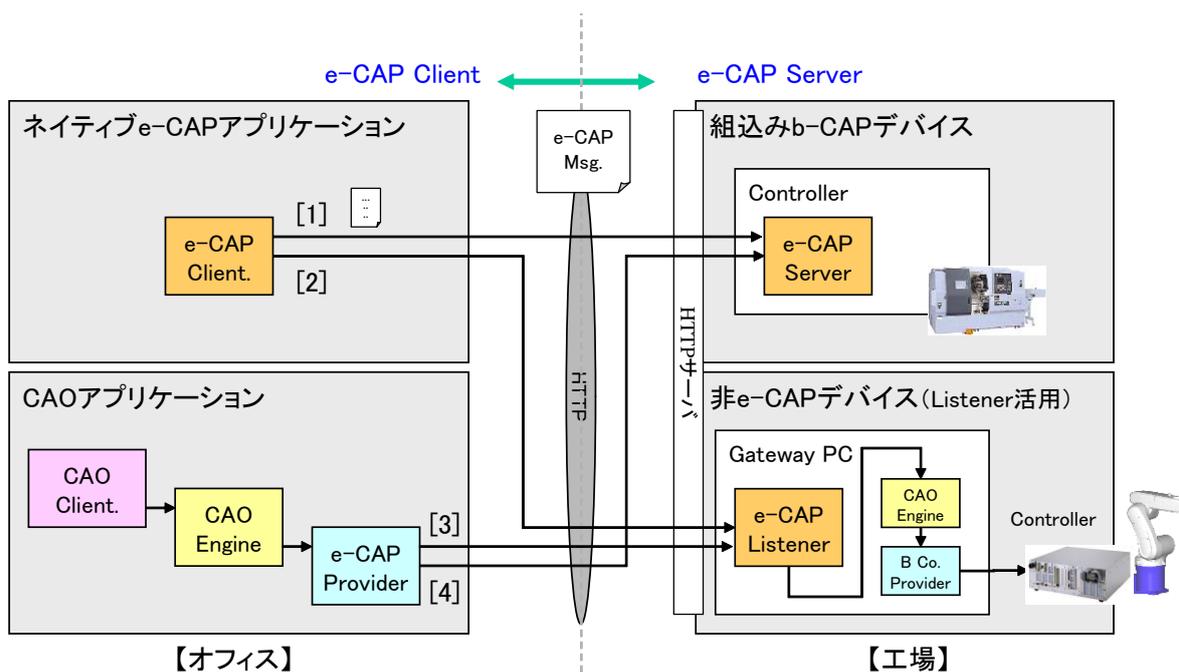


図 2-2 e-CAP 接続例

## 3. e-CAP メッセージ

### 3.1. メッセージ構造

e-CAP は HTTP を用いて CGI のようにメッセージの送受信を行います。このとき e-CAP メッセージの表記規則は以下のものに従います。

### 3.2. メッセージ規則

e-CAP は HTTP を用いて CGI のようにメッセージの送受信を行います。このとき e-CAP メッセージの表記規則は以下のものに従います。

- HTTP リクエストメッセージは，“POST”コマンドを送信してください。
- HTTP リクエストメッセージの HTTP ヘッダに，以下のメタデータを追加してください。  
“Content-Type:application/x-www-form-urlencoded”
- HTTP リクエストメッセージの HTTP ボディは以下の構造になります。  
Func=<関数 ID>,Arg1=<第 1 引数>,Arg2=<第 2 引数>,Arg3=<第 3 引数>,Arg4=<第 4 引数>,  
Arg5=<第 5 引数>  
関数 ID            関数 ID. 詳細については, 3.3 を参照してください。  
第1～5 引数        実行関数のパラメータ。  
                    引数の型が VARIANT 型の場合は, 3.5 の書式に従って記述しなければなりません。VARIANT 型以外のときは, 入力値を文字列で設定します。  
                    実行関数を使用しない引数は, から文字列を設定してください。
- HTTP レスポンスメッセージの HTTP ボディは以下の構造になります。  
<リターンコード>[<結果データ>]  
リターンコード    実行関数のリターンコード. 詳細については, 3.4 を参照してください。  
結果データ        実行関数の結果データ  
                    実行関数がデータを返さない場合は, 結果データは省略されます。  
                    結果データは, VARIANT 型で取得します。VARIANT 型の書式については, 3.5 を参照してください。

以下に，通信の一例を示します。

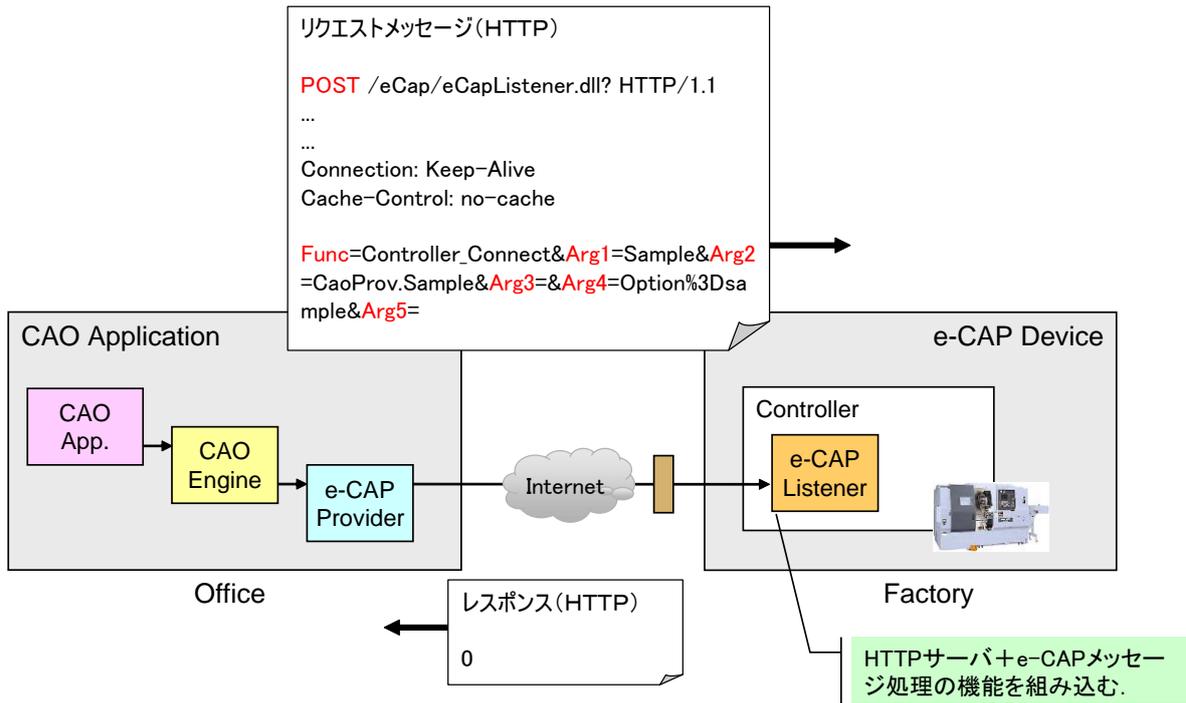


図 3-1 e-CAP メッセージ例

### 3.3. 関数 ID

b-CAP では関数 ID は以下のように割り振られている。

表 1 関数 ID の割り当て

関数 ID	説明
1～137	既定の関数
138～255	予約領域
255～	ユーザ関数

既定の関数以外の関数を使用したいときは、ユーザ関数に任意の関数を割り当てて使用することができる。

b-CAP の既定の関数の一覧を以下に示す。

表 2 既定の関数一覧

関数 ID	関数名	説明
1	Service_Start	サーバサービスの開始
2	Service_Stop	サーバサービスの停止

3	Controller_Connect	コントローラとの接続
4	Controller_Disconnect	コントローラとの切断
5	Controller_GetExtension	コントローラの拡張ボード取得
6	Controller_GetFile	コントローラのファイル取得
7	Controller_GetRobot	コントローラのロボット取得
8	Controller_GetTask	コントローラのタスク取得
9	Controller_GetVariable	コントローラの変数取得
10	Controller_GetCommand	コントローラのコマンド取得
11	Controller_GetExtensionNames	コントローラの拡張ボード名一覧取得
12	Controller_GetFileNames	コントローラのファイル名一覧取得
13	Controller_GetRobotNames	コントローラのロボット名一覧取得
14	Controller_GetTaskNames	コントローラのタスク名一覧取得
15	Controller_GetVariableNames	コントローラの変数名一覧取得
16	Controller_GetCommandNames	コントローラのコマンド名一覧取得
17	Controller_Execute	コントローラの拡張関数の実行
18	Controller_GetMessage	コントローラのイベントメッセージ取得
19	Controller_GetAttribute	コントローラの属性値取得
20	Controller_GetHelp	コントローラのヘルプ文字列取得
21	Controller_GetName	コントローラの名前取得
22	Controller_GetTag	コントローラのタグ情報取得
23	Controller_PutTag	コントローラのタグ情報設定
24	Controller_GetID	コントローラの ID 取得
25	Controller_PutID	コントローラの ID 設定
26	Extension_GetVariable	拡張ボードの変数取得
27	Extension_GetVariableNames	拡張ボードの変数名一覧取得
28	Extension_Execute	拡張ボードの拡張関数実行
29	Extension_GetAttribute	拡張ボードの属性値取得
30	Extension_GetHelp	拡張ボードのヘルプ文字列取得
31	Extension_GetName	拡張ボードの名前取得
32	Extension_GetTag	拡張ボードのタグ情報取得
33	Extension_PutTag	拡張ボードのタグ情報設定
34	Extension_GetID	拡張ボードの ID 取得
35	Extension_PutID	拡張ボードの ID 設定
36	Extension_Release	拡張ボードの解放
37	File_GetFile	ファイルの別ファイル取得

38	File_GetVariable	ファイルの変数取得
39	File_GetFileNames	ファイルの別ファイル名一覧取得
40	File_GetVariableNames	ファイルの変数名一覧取得
41	File_Execute	ファイルの拡張関数実行
42	File_Copy	ファイルのコピー
43	File_Delete	ファイルの削除
44	File_Move	ファイルの移動
45	File_Run	ファイルの実行
46	File_GetDateCreated	ファイルの作成日時取得
47	File_GetDateLastAccessed	ファイルの最終アクセス日時取得
48	File_GetDateLastModified	ファイルの最終更新日時取得
49	File_GetPath	ファイルのパス取得
50	File_GetSize	ファイルのサイズ取得
51	File_GetType	ファイルのファイルタイプ取得
52	File_GetValue	ファイルの内容取得
53	File_PutValue	ファイルの内容設定
54	File_GetAttribute	ファイルの属性取得
55	File_GetHelp	ファイルのヘルプ文字列取得
56	File_GetName	ファイルの名前取得
57	File_GetTag	ファイルのタグ情報取得
58	File_PutTag	ファイルのタグ情報設定
59	File_GetID	ファイルの ID 取得
60	File_PutID	ファイルの ID 設定
61	File_Release	ファイルの解放
62	Robot_GetVariable	ロボットの変数取得
63	Robot_GetVariableNames	ロボットの変数名一覧取得
64	Robot_Execute	ロボットの拡張関数実行
65	Robot_Accelerate	ロボットの ACCEL 文実行
66	Robot_Change	ロボットの CHANGE 文実行
67	Robot_Chuck	ロボットの GRASP 文実行
68	Robot_Drive	ロボットの DRIVE 文実行
69	Robot_GoHome	ロボットの GOHOME 文実行
70	Robot_Halt	ロボットの HALT 文実行
71	Robot_Hold	ロボットの HOLD 文実行
72	Robot_Move	ロボットの MOVE 文実行

73	Robot_Rotate	ロボットの ROTATE 文実行
74	Robot_Speed	ロボットの SPEED/JSPEED 文実行
75	Robot_Unchuck	ロボットの REELASE 文実行
76	Robot_Unhold	ロボットの HOLD 文解除
77	Robot_GetAttribute	ロボットの属性値取得
78	Robot_GetHelp	ロボットのヘルプ文字列取得
79	Robot_GetName	ロボットの名前取得
80	Robot_GetTag	ロボットのタグ情報取得
81	Robot_PutTag	ロボットのタグ情報設定
82	Robot_GetID	ロボットの ID 取得
83	Robot_PutID	ロボットの ID 設定
84	Robot_Release	ロボットの解放
85	Task_GetVariable	タスクの変数取得
86	Task_GetVariableNames	タスクの変数名一覧取得
87	Task_Execute	タスクの拡張関数実行
88	Task_Start	タスクの開始
89	Task_Stop	タスクの停止
90	Task_Delete	タスクの削除
91	Task_GetFileName	タスクの元ファイル名
92	Task_GetAttribute	タスクの属性取得
93	Task_GetHelp	タスクのヘルプ文字列取得
94	Task_GetName	タスクの名前取得
95	Task_GetTag	タスクのタグ情報取得
96	Task_PutTag	タスクのタグ情報設定
97	Task_GetID	タスクの ID 取得
98	Task_PutID	タスクの ID 設定
99	Task_Release	タスクの解放
100	Variable_GetDateTime	変数のタイムスタンプ取得
101	Variable_GetValue	変数の値取得
102	Variable_PutValue	変数の値設定
103	Variable_GetAttribute	変数の属性値取得
104	Variable_GetHelp	変数のヘルプ文字列取得
105	Variable_GetName	変数の名前取得
106	Variable_GetTag	変数のタグ情報取得
107	Variable_PutTag	変数のタグ情報設定

108	Variable_GetID	変数の ID 取得
109	Variable_PutID	変数の ID 設定
110	Variable_GetMicrosecond	変数のタイムスタンプ(ミリ秒)取得
111	Variable_Release	変数の解放
112	Command_Execute	コマンドの実行
113	Command_Cancel	コマンドのキャンセル
114	Command_GetTimeout	コマンドのタイムアウト時間取得
115	Command_PutTimeout	コマンドのタイムアウト時間設定
116	Command_GetState	コマンドの状態取得
117	Command_GetParameters	コマンドのパラメータ取得
118	Command_PutParameters	コマンドのパラメータ設定
119	Command_GetResult	コマンドの実行結果取得
120	Command_GetAttribute	コマンドの属性値取得
121	Command_GetHelp	コマンドのヘルプ文字列取得
122	Command_GetName	コマンドの名前取得
123	Command_GetTag	コマンドのタグ情報取得
124	Command_PutTag	コマンドのタグ情報設定
125	Command_GetID	コマンドの ID 取得
126	Command_PutID	コマンドの ID 設定
127	Command_Release	コマンドの解放
128	Message_Reply	イベントメッセージの応答
129	Message_Clear	イベントメッセージのクリア
130	Message_GetDateTime	イベントメッセージのタイムスタンプ取得
131	Message_GetDescription	イベントメッセージの説明文取得
132	Message_GetDestination	イベントメッセージの送信先取得
133	Message_GetNumber	イベントメッセージのメッセージ番号取得
134	Message_GetSerialNumber	イベントメッセージのシリアル番号取得
135	Message_GetSource	イベントメッセージの送信元取得
136	Message_GetValue	イベントメッセージの値取得
137	Message_Release	イベントメッセージの解放

### 3.4. リターンコード

b-CAP ではリターンコードは以下のように割り振られている。

**表 3 リターンコードの割り当て**

リターンコード	説明
0x00000000～0x8000FFFF	既定リターンコード, 予約領域
0x80010000～0x8001FFFF	ユーザ定義エラー

以下に示す“既定エラーコード”以外のエラーコードを作成するときは，“ユーザ定義エラー”の値の範囲内で任意のエラーコードを割り当てることができる。

**表 4 既定のリターンコード一覧**

リターンコード	エラー	説明
0x00000000	S_OK	正常終了.
0x80004001	E_NOTIMPL	未実装.
0x80004004	E_ABORT	関数が中断されました.
0x80004005	E_FAIL	関数が失敗しました.
0x80070005	E_ACCESSDENIED	アクセスできません.
0x80070006	E_HANDLE	ハンドルが不正です.
0x8007000E	E_OUTOFMEMORY	メモリが不足しています.
0x80070057	E_INVALIDARG	引数が不正です.
0x8000FFFF	E_UNEXPECTED	致命的エラーが発生しました.

### 3.5. VARIANT データ書式

e-CAP における VARIANT 型のデータ書式は、ORiN2 の VARIANT 型を文字列で表現する方法に従っています。その書式は、以下に示すようにデータ型とデータ列をカンマ区切りで表現します。

<データ型>, <データ列>

ここで、<データ型>には VARTYPE 型で表記される整数値を示します。表 3-5 に使用できるデータ型とその値を示します。

**表 3-5 使用できるデータ型**

データ型	値	意味
VT_I2	2	2 バイト整数型
VT_I4	3	4 バイト整数型
VT_R4	4	単精度浮動小数点型
VT_R8	5	倍精度浮動小数点型
VT_CY	6	通貨型
VT_DATE	7	日付型
VT_BSTR	8	文字列型

VT_BOOL	11	ブール型
VT_VARIANT	12	VARIANT 型
VT_UI1	17	バイト型
VT_ARRAY	8192	配列型

データ型が配列のときは、VT\_ARRAY とデータ型の論理和で表記します。

データ列にはデータを文字列で表記します。配列データの表記は“,”(カンマ)で区切って表記します。

例 1)	2,100	型:VT_I2	値:100
例 2)	8,Sample	型:VT_BSTR	値:Sample
例 3)	12,(8,Sample)	型:VT_VARIANT	値:“Sample”(文字列型)
例 4)	8194,100,200,300	型:VT_I2   VTARRAY	値:100, 200, 300
例 5)	8200,Sample,Test	型:VT_BSTR   VTARRAY	値:“Sample”, “Test”
例 6)	8,Sample,Test	型:VT_BSTR	値:“Sample,Test”
例 7)	8204,(8,Sample),(2,100)	型:VT_VARIANT   VTARRAY	値:“Sample”, 100