

MITSUBISHI

三菱Web地理情報システム構築パッケージ



三菱マッピングシステム

PreSerV[®] WebTcl

PreSerV WebTcl 用語集

2008/04/25

マニュアル

はじめに

はじめに

本書では、PreSerV WebTcl で使用する用語を説明します。

目次

1	共通項目	1
1.1	マクロ	1
1.2	名前空間	1
2	ウィンドウ (W) マクロに関する用語	2
2.1	ウィンドウ部品	2
2.2	ウィンドウID	2
2.3	タグ文字列	2
2.4	親ウィンドウ／子ウィンドウ	3
2.5	相対座標	3
2.6	絶対座標	3
2.7	アームボタン	3
3	グラフィックス (G) マクロに関する用語	4
3.1	図形オブジェクト	4
3.2	コンテキストオブジェクト	4
3.3	カレント値	4
3.4	ディスプレイリスト (メモリ)	5
3.5	マップオブジェクト	5
3.6	グラフィックスウィンドウ	6
3.7	ビューポート	6
3.8	DC座標	7
3.9	WC座標	7
3.10	グループオブジェクト	7
3.11	レイヤオブジェクト	7
3.12	ハッシュオブジェクト	8
3.13	色ID	8
3.14	線色	8
3.15	(塗りつぶし) 前景色	9
3.16	(塗りつぶし) 背景色	9
3.17	タグID	9
3.18	スケール	9
3.19	(ベクトル) シンボル	9
3.20	ビットマップシンボル	10
3.21	ベクトルフォント	10
4	標準部品 (ST) に関する用語	11

4.1	標準部品ID	11
5	WebTclに関する用語	12
5.1	WebTclエンジン	12
5.2	FE-COM	12
5.3	PsvWTclClient (WebTclセッション管理クラス)	13
5.4	PsvWTclFilter (WebTclフィルタークラス)	14
5.5	起動HTML	14
5.6	偽接続	14
6	地図に関する用語	15
6.1	測地系	15
6.2	楕円体	15

1 共通項目

1.1 マクロ

本来は命令の集合体を示す意味に使用する用語です。PreSerV では PreSerV 命令の集合体と言う概念で、命令を記述したファイルをマクロと表記します。

また PreSerV 命令自体も、複数のコマンドを組合せて構成しているため、グラフィックスマクロや、ウィンドウマクロと表記します。

これらの表記は過去ユーザに定着しているため、そのまま使用します。WebTcl 版からは、これらを区別して表現したい場合は、命令を記述したファイルを「スクリプト」、PreSerV 命令を「API」と表記します。

1.2 名前空間

名前空間とは、コマンド（プロシジャ）や変数を1つの領域にまとめ、名前で領域を管理する機能です。ある名前空間からは、別の名前空間にあるコマンドや変数にはそのままではアクセスできません。また、名前空間が異なれば、同名のコマンドや変数を別のものとして扱います。

リスト 1-1に名前空間を作成する例を示します。色定義ダイアログサンプルでは、処理全体を"WTCD"という名前空間でカプセル化して、変数名やプロシジャ名の衝突を避けています。

リスト 1-1 名前空間"WTCD"を使用した色定義ダイアログサンプル

```
namespace eval WTCD {
...
### 変数宣言
# 部品を追加して、別のプロシジャーから参照する際はここで宣言してください
array set col {}
array set pb {}
...
# 名称 : col_init
# 概要 : 色定義ツール初期化処理を行います。
# 引数説明 : なし
# 戻り値 : なし
proc col_init { } {
    variable pb;
    variable col;
```

2 ウィンドウ(W)マクロに関する用語

2.1 ウィンドウ部品

Wマクロで生成／表示する、ウィンドウの構成要素をウィンドウ部品と総称します。ウィンドウ部品は表 2-1に示す3種類の部品です。

表 2-1 ウィンドウ部品分類

分類	説明
ウィンドウ部品	他の部品を配置するウィンドウとなる部品です。トップウィンドウとダイアログがあります。
一般部品	トップウィンドウやダイアログ上に配置し、ユーザのデータ入力／選択や、文字、図形の表示などを行うために使用します。ボタン、テキストボックス、キャンバスなどがあります。
メニュー部品	トップウィンドウやダイアログにメニューを付加するための部品です。プルダウンメニューとポップアップメニューの2種類が作成できます。プルダウンメニューは、メニューバー、プルダウンメニュー、カスケードなどを組み合わせて作成します。ポップアップメニューは、ポップアップ、ポップアップメニュー、カスケードなどを組み合わせて作成します。

2.2 ウィンドウ ID

個々のウィンドウ部品を識別するための ID です。ウィンドウ部品の生成時に自動的に採番し、W マクロの戻り値として返します。

2.3 タグ文字列

個々のウィンドウ部品に付加する任意の文字列です。ウィンドウ部品の生成時に W マクロの引数で指定します。W マクロで、生成済のウィンドウ部品を指定する時に、ウィンドウ ID の代わりに「@タグ文字列」を指定することもできます。

(例) :

タグ文字列が「TOP_1」のトップウィンドウ上にボタンを生成するとき
w_button btn_11 @TOP_1 -x 300-y 280...

ただし、ウィンドウ ID の代わりにタグ文字列を使用する場合、各ウィンドウ部品に一意なタグ文字列を設定してください。

参照 : ウィンドウ ID

2.4 親ウィンドウ／子ウィンドウ

ウィンドウ部品はすべて、画面（ルートウィンドウと呼びます）を根（ルート）とする階層構造で管理します（図 2-1参照）。

階層構造中で、あるウィンドウ部品が従属するウィンドウ部品を「親ウィンドウ」と呼び、ウィンドウ部品に従属するウィンドウを「子ウィンドウ」と呼びます。

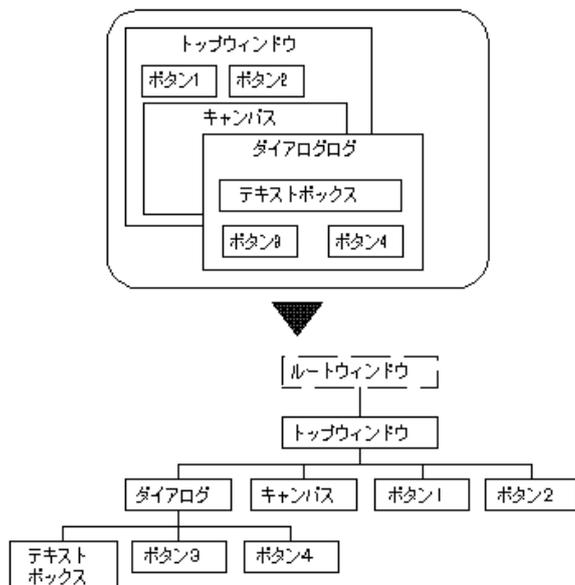


図 2-1 ウィンドウの階層構造

トップウィンドウはルートウィンドウの子ウィンドウになります。また、ボタンやテキストボックスは、トップウィンドウまたはダイアログの子ウィンドウになります。トップウィンドウ以外のウィンドウ部品を生成するには、あらかじめ親ウィンドウを生成しておき、そのウィンドウ ID を引数で指定する必要があります。

2.5 相対座標

親ウィンドウの左上を原点とする座標です。単位はピクセルです。

ウィンドウマクロでは、特に断りのない限り、座標は全て相対座標を使用します。

参照：絶対座標

2.6 絶対座標

ルートウィンドウ（画面）の左上を原点とする座標です。単位はピクセルです。

参照：相対座標

2.7 アームボタン

一般部品の1つで、押下時のイベントをタイマーで連続して発生することができるボタン。

参照：ウィンドウ部品

3 グラフィックス (G) マクロに関する用語

3.1 図形オブジェクト

G マクロで作成する個々の図形です。図形オブジェクトには、生成時に図形オブジェクト ID を割り当て、作成後の処理ではこの ID で図形を識別します。

図形オブジェクトは、図形の種類、座標値、色、属性などの情報を持っており、作成後に移動、拡大／縮小、属性変更、消去などの図形操作が可能です。

図形オブジェクトは、作成しただけでは描画しません。描画するには `g_draw_obj` マクロを実行するか、ディスプレイリストに投入後、`g_redraw` で再描画設定することで、描画します。

なおGマクロでは、図形オブジェクトを作成せずに、`g_draw_XXX`^{※1} マクロを使用して図形を描画することも可能です。

※1 : XXX部分は"polyline"など各図形を示す文字列です。

ただしこの方法では ID や座標値などの情報は保持しません。イベントハンドラなどで、一時的に図形を赤反転するケースなどで使用します。

参照 : カレント値

3.2 コンテキストオブジェクト

グラフィックスマクロにおけるカレント値などの情報を保持するオブジェクトです。グラフィックスウィンドウ ID、ディスプレイオブジェクト ID や、図形オブジェクトの属性値の情報を保持します。

コンテキストオブジェクトは `g_context` マクロで作成します。コンテキストオブジェクトは複数を作成可能です。

`g_context` を除くほとんどのグラフィックスマクロでは、`g_context` の戻り値であるコンテキスト ID を指定します。コンテキストオブジェクトより、各マクロの引数の既定値として、カレント値の情報を適用します。

3.3 カレント値

カレント値は、G マクロの引数として頻繁に指定するオブジェクトや属性を保存する概念です。例えば、頻繁に使用するディスプレイリストメモリの ID は `g_cur_dlist` でコンテキストオブジェクトに保存しておく、図形オブジェクトを格納する時に使用する `g_ins_obj` では"-d"オプションで指定する必要がありません。

カレント値の情報の種類と対応するGマクロを表 3-1に示します。

表 3-1 カレント値の種類と G マクロ一覧

カレント値の情報の種類	対応する G マクロ
グラフィックスウィンドウ	g_cur_gwin
ディスプレイリストメモリ	g_cur_dlist
グループオブジェクト	g_cur_group
レイヤオブジェクト	g_cur_layer
旧版属性値	g_cur_attrib
オブジェクト属性	g_cur_obj_attrib
描画属性	g_cur_draw_attrib
色 ID	g_cur_color_id
文字列属性値	g_cur_tx_attrib

このようにカレント値を効率よく使用すると、スクリプト自体の文字数を減らすことが可能で、コーディングの効率化や、Tcl インタプリタの実行の効率化につながります。

3.4 ディスプレイリスト(メモリ)

作成した図形オブジェクトを格納するためのオブジェクトです。

ディスプレイリストメモリに格納することにより、以下の機能を実現します。

① 図形の一括描画・削除

ディスプレイリストメモリ単位で描画や、ディスプレイリスト破棄による図形オブジェクトの削除が可能です。

② 空間検索および検索と連動した再描画

ディスプレイリストメモリ内部は、空間検索可能な構造になっています。これにより g_search_obj を使用して、検索条件の座標内の図形オブジェクトを高速に検索可能です。また画面の再描画時も、この空間検索を使用し高速に再描画候補図形を検索しています。

ディスプレイリストメモリは g_open_dlist マクロで作成します。複数個作成でき、g_open_dlist の戻り値であるディスプレイリストメモリ ID で識別します。

ディスプレイリストメモリの用途は、図面単位の図形の一括管理や、地図のメッシュ単位の図形管理です。

3.5 マップオブジェクト

ディスプレイリストをまとめて地図という単位で管理するためのオブジェクトです。

この時、ディスプレイリストメモリは地図内の一定の区画単位(以下メッシュと記述)で作成すると、メッシュ単位の図形作成・開放処理が実装しやすくなり、アプリケーションのメモリ使用効率を高めることができます。

またマップオブジェクトには、表示縮尺範囲の設定、オブジェクト属性の設定が可能であるため、複数のディスプレイリストメモリを一括して操作することが可能です。

マップオブジェクトは、処理対象によりベクター図形を管理する“vector”マップと、ラスターイメージを管理する“raster”マップの2種類のマップオブジェクトに分けます。また、PreSerVで提供するマップオブジェクトの機能は基本的機能だけですが、これを拡張して、効率的なメッシュの管理や、特殊な地図フォーマットのリアルタイム変換が可能なカスタマイズしたマップオブジェクトを作成することも可能です。^{※2}

※2 プラグインによるマップオブジェクトのカスタマイズは、別途 VC++言語レベルの技術情報の提供が必要です。

製品に基本サービス範囲ではこれは提供しません。別途開発元にご相談ください。

オプション製品で提供するマップオブジェクトが、これに相当するカスタマイズマップオブジェクトです。

マップオブジェクトは `g_open_map` で作成します。複数個作成でき、`g_open_map` の戻り値であるマップオブジェクト ID で識別します。

3.6 グラフィックスウィンドウ

グラフィックスマクロが図形描画の対象とする仮想的なウィンドウです。

グラフィックスウィンドウはWC座標を管理し、キャンバスの実ウィンドウの座標となるDC座標と関連付けて管理します。グラフィックスウィンドウは図形の表示が行われた場合は、WC座標をDC座標に座標投影(変換)して、ウィンドウの実体であるキャンバスに図形を描画する命令を発行します(図 3-1参照)。

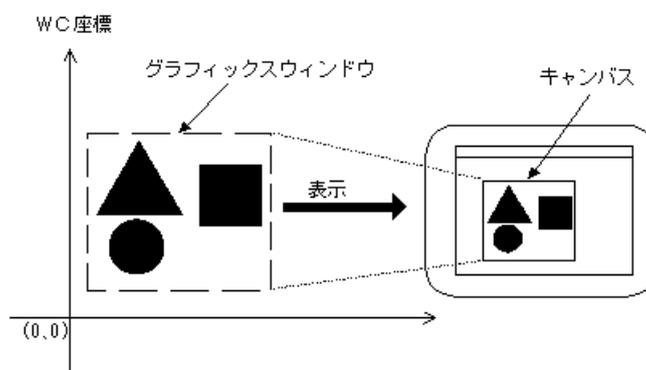


図 3-1 グラフィックスウィンドウの表示概念

グラフィックスウィンドウの範囲は、WC 座標の矩形で指定します。グラフィックスウィンドウのキャンバス上の表示範囲は DC 座標の矩形で指定します。描画時にはこの2つの情報から WC 座標と DC 座標の変換は自動的に行われます。グラフィックスウィンドウの位置やサイズを変更することにより、描画内容のスクロールや、拡大/縮小表示が可能です。

参照：ビューポート、WC座標、DC座標

3.7 ビューポート

キャンバス上のグラフィックスウィンドウと対応付ける領域をビューポートと定義します。グラフィックスウィンドウ生成時には、キャンバスの実領域サイズと一致しています。

ビューポートのサイズ、位置はg_viewportマクロで、グラフィックスウィンドウごとに変更できます。ビューポートにより、キャンバスを分割して複数のグラフィックスウィンドウを使用可能になります（図 3-2参照）。

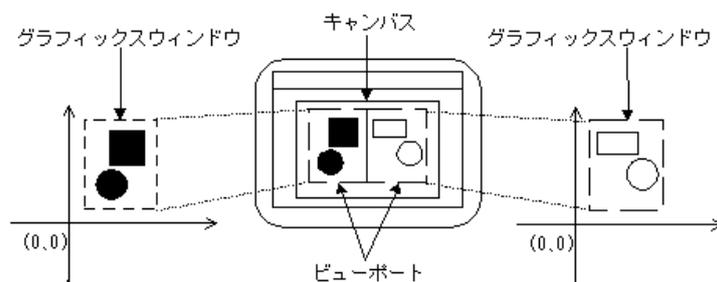


図 3-2 ビューポート使用例

参照：グラフィックスウィンドウ

3.8 DC 座標

キャンバス上の座標系です。キャンバスの左上を原点とし、単位はピクセルです。

参照：グラフィックスウィンドウ、WC座標

3.9 WC 座標

グラフィックスマクロ内で使用する仮想的な座標系です。特に断りのない限り、グラフィックスマクロでは座標はWC座標で指定します。

PreSerVで使用するこのWC座標は、北をY軸正方向、東をX軸正方向としたXY座標^{※3}で、座標値1 = 1mとして扱います。

※3 国土地理院が定義する19座標系は北をX軸正方向、東をY軸正方向とするため、取り込む時はXY座標を入れ替える必要があります。

3.10 グループオブジェクト

図形オブジェクトをグループとして管理するオブジェクトです。^{※4}g_open_groupマクロで作成し、グループオブジェクトIDで識別します。

※4 現在グループ管理は非推奨機能です。図形の包括した管理はレイヤオブジェクトの機能を利用してください。グループオブジェクトは、グループ番号と、各グループ番号に属する図形オブジェクトの情報を保持します。

g_make_groupマクロを使用して、グループオブジェクトIDとグループ番号を指定して図形オブジェクトをグループ化することにより、図形オブジェクトをグループ単位で描画することができます。

3.11 レイヤオブジェクト

図形オブジェクトをレイヤ（層）に分けて管理するオブジェクトです。g_open_layerマクロで作成し、レイヤオブジェクトIDで識別します。

レイヤオブジェクトは、レイヤ番号のセットを保持します（図 3-3参照）。g_layer_noマクロを使用して図形オブジェクトにレイヤID、レイヤ番号を設定して、図形オブジェクトをレイヤ番号単位で管理することができます。

レイヤ番号は1図形オブジェクトに1つだけ設定可能です。また、レイヤ番号で図形オブジェクトを検索し一括処理するには、図形オブジェクトをディスプレイリストメモリに格納しておく必要があります。

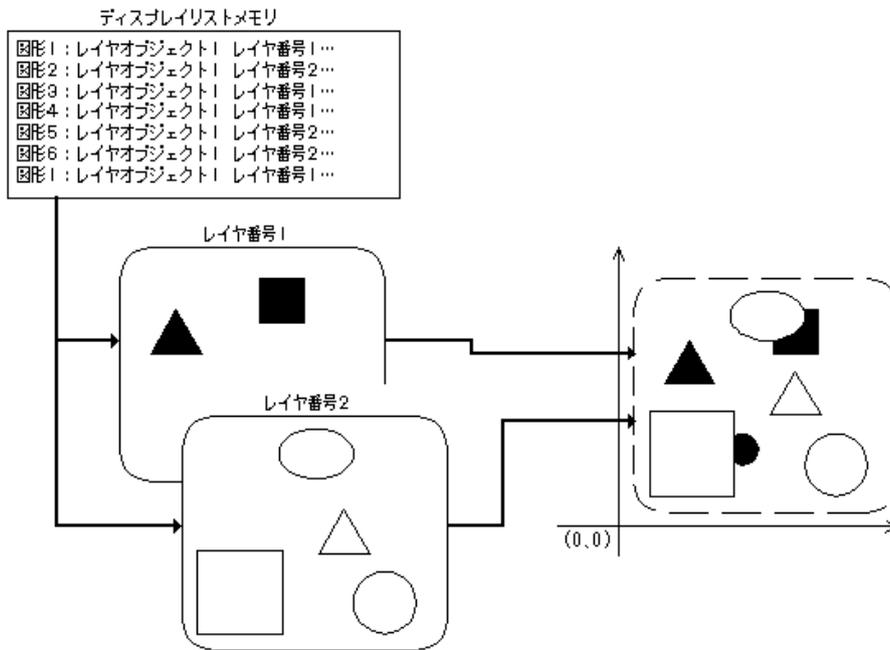


図 3-3 レイヤオブジェクト概念

3.12 ハッシュオブジェクト

播種オブジェクトは、ハッシュでキーと対応するデータを管理するための機能です。

`g_open_hash` マクロで作成し、ハッシュオブジェクト ID で識別します。

データは、一意なキーを指定してハッシュオブジェクトに格納（挿入）します。格納したデータに対しては、キーを指定してデータの検索、消去が可能です。

3.13 色 ID

グラフィックスマクロ中で色を指定するための ID です。色 ID と RGB 値の対応は、`g_alloc_color` マクロで設定します。PreSerV では 1~255 までの 255 パレットを保持しており各パレット番号を色 ID にビットをずらして設定します。^{※5}

※5 色 ID は現在 8 ビットごとに分割し、線色、前景色、背景色の 3 つの情報を圧縮して格納します。

参照：線色、(塗りつぶし)前景色、(塗りつぶし)背景色

3.14 線色

図形オブジェクトや、一時的描画マクロを使用して描画する時の、線の色を色 ID で指定します。

線色は、図形オブジェクト作成マクロや描画関連のマクロの引数に指定する“-c” 引数の数値 bit0~bit7 で指定します。

参照：色 ID、(塗りつぶし)前景色、(塗りつぶし)背景色

3.15 (塗りつぶし)前景色

図形オブジェクトや、一時的描画マクロで塗りつぶしモード2を指定して描画する際の、塗りつぶしパターンの前景色を色IDで指定します。

塗りつぶし前景色は、図形オブジェクト作成マクロや描画関連のマクロの引数に指定する“-c”引数の数値のbit8~bit15で指定します。

参照：色ID、線色、(塗りつぶし)背景色

3.16 (塗りつぶし)背景色

図形オブジェクトや、一時的描画マクロで塗りつぶしモード2を指定して描画する際の、塗りつぶしパターンの背景色を色IDで指定します。

塗りつぶし背景色は、図形オブジェクト作成マクロや描画関連のマクロの引数に指定する“-c”引数の数値のbit16~bit23で指定します。

参照：色ID、線色、(塗りつぶし)前景色

3.17 タグID

図形オブジェクトに設定できる任意の数字です。図形に通し番号を設定するなど任意の用途にご使用ください。

3.18 スケール

現在のグラフィックスウィンドウで表示する地図の表示縮尺の分母値です。スケールの変換は、グラフィックスウィンドウ上で設定するのWC系座標値1を実寸の1mとし、画面の表示や、印刷時に出力する時のサイズとの比率から計算します。^{※6}

※6 印刷上WC座標値100の直線が10cmで出力する時、

$10\text{cm}/100\text{m} = 10\text{cm}/10,000 = 1/1,000$ となりスケール値1,000となります。

スケールは、マップオブジェクトやレイヤオブジェクトの表示範囲にも使用します。WC系座標値1を1mとしてアプリケーションを作成しない場合、PreSerVのスケール関連マクロに設定するスケール値は変換して渡すようにしてください(例WC系座標値1が1cmの場合、スケール関連マクロに渡す値は1/100にするなど)。

3.19 (ベクトル)シンボル

PreSerVでは、地図上に表示する記号などを「シンボル」というオブジェクトを使用して、表示することが可能です。「シンボル」のパターンは、複数の基本的なベクトル図形を、1つのオブジェクトにしたもので、PreSerV基本キットに添付する『シンボル定義ツール』を使用して、シンボル定義ファイルを作成します。1つのシンボル定義ファイルには、複数のシンボルを定義して、シンボルは番号で特定します(図3-4参照)。



図 3-4 『シンボル定義ツール』のシンボルの一覧

3.20 ビットマップシンボル

「ビットマップシンボル」は、Windows ビットマップを PreSerV 図形として表示するオブジェクトです。「ビットマップシンボル」の番号とデータとなる Windows ビットマップは、バイナリ形式定義ファイルで定義します。この定義ファイルは、PreSerV 基本キットに添付するコンソールコマンド『mkbsys.exe』を使用して作成してください。1つの定義ファイルには、複数の Windows ビットマップを定義可能です。

「ビットマップシンボル」の表示例を図 3-5に示します。

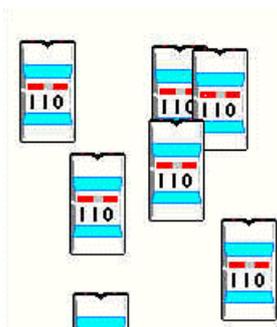


図 3-5 「ビットマップシンボル」オブジェクト表示例

3.21 ベクトルフォント

一般に「ベクトルフォント」とは、「ベクトル形式のフォント」を意味する広義な用語ですが、PreSerVでは独自のストロークで構成されるフォントを、「ベクトルフォント」として記載します。^{※7}

これに対比して使用するフォント名は、「TrueType フォント」です。Windows の MS 明朝体と MS ゴシック体を文字オブジェクトに指定して描画可能です。

※7 一般にストロークで構成されるフォントは「ストロークフォント」と表記します。

4 標準部品 (ST) に関する用語

4.1 標準部品 ID

標準部品の実行環境に関する情報を管理するための ID です。st_open マクロによる標準部品の初期処理で作成します。

st_open 以外の標準部品マクロでは、引数に標準部品 ID を指定する必要があります。

5 WebTcl に関する用語

本章ではWebTclで追加した用語を説明します。本章で説明するWebTclのコンポーネントの概要を図 5-1に示します。

WebTcl ではFE-COM と WebTclJavaAPI により、独自のセッション管理を行い、C/S システムで実現していた双方向の通信処理を HTTP プロトコル上で実現可能です。

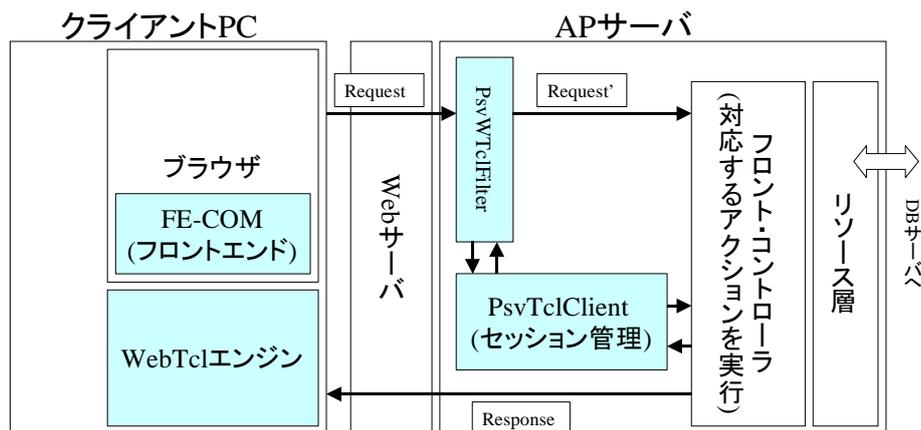


図 5-1 WebTcl 機能コンポーネント (網掛け、括弧数字部分)

5.1 WebTcl エンジン

Tcl コマンドの解析、実行および、各種ウィンドウを制御する、WebTcl の独立したプロセスを示します。WebTcl エンジンは、Web サーバ経由で自動的にダウンロード・インストールが可能です。

※ ダウンロード・インストールは初回と更新時だけ実行します。

5.2 FE-COM

Internet Explorer (以下IE)のフロント・エンドの役割を行う、COMオブジェクトです。FE-COMはIEのアドオンとして認識されるActiveXとして動作します (図 5-2参照)。

※ Front End COM オブジェクトの略称から、WebTcl では"FE-COM"と表記します。

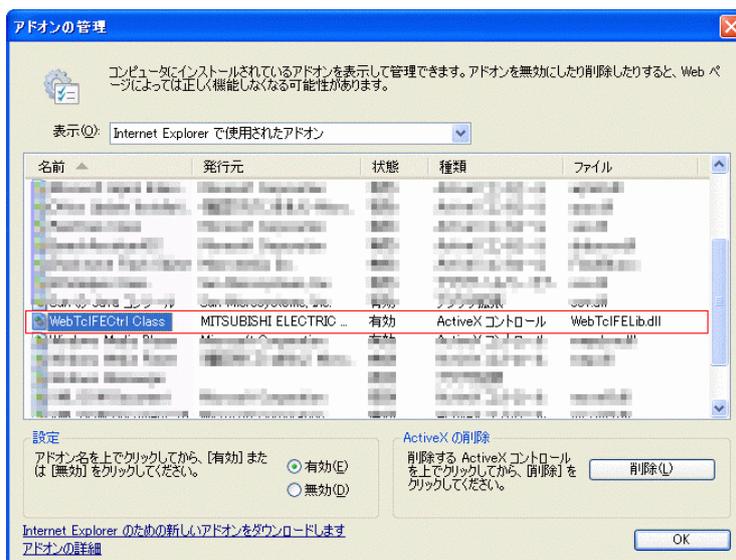


図 5-2 IE7 の「アドオンの管理」での FE-COM

オブジェクト生成時は自動的にダウンロードされたエンジンの起動を行います。

FE-COM では以下の機能を提供します。

- ① 初期ウィンドウ表示までの処理
WebTcl エンジンの起動、*FE-COM*.InitSource プロパティで URL を指定する初期起動マクロのダウンロードを行い、実行します。
- ② HTTP 通信の代行
WebTcl エンジンと係し、AP サーバ上のアプリケーションへのリクエスト送信と、レスポンスの受信を行います。FE-COM が通信を代行するのは、IE と同一プロセスで動作し、HTTP 通信層を共用するためです。
- ③ IE との関係
IE 上の別アプリケーションと、JavaScript 経由の双方向通信が可能です。通信は JavaScript からの FE-COM メソッド実行によるメッセージ文字列の受信と、FE-COM からの JavaScript イベント発生による文字列送信が可能です。

5.3 PsvWTclClient (WebTcl セッション管理クラス)

WebTcl Java API のクラスの 1 つで、Tcl 版と同等の通信管理を実現するため、AP サーバ上に常駐するクラスです。

クライアント ID+ダミーのポート番号+IE のプロセス ID をキーとして、WebTcl セッション ID と関連付け、WebTcl セッション ID に関連付ける各種情報の管理を行います。PSV_Send()、PSV_SendFile() で返信していた通信内容は、PsvWTclClient のメソッドを呼び出し FE-COM に対するレスポンスとして返信します。

5.4 PsvWTclFilter (WebTcl フィルタークラス)

WebTcl Java API のクラスの 1 つで、Servlet2.3 以降のフィルター機能を利用したクラスです。主に通信の前処理・後処理を担当し、一部の処理を自動化します。

アプリケーションを定義する web.xml に定義して使用可能で、既存のフィルターのチェーンに組み込むことも可能です。

5.5 起動 HTML

FE-COM の起動のトリガーをかける HTML ファイルです。起動 HTML には以下の内容を記述します。詳細な記述内容は「WebTcl アプリケーション作成ガイド」の「起動 HTML の準備」を参照してください。

① FE-COM のオブジェクト生成記述

FE-COM のオブジェクト生成記述を通して、WebTcl の自動インストール機能を実行します。

② FE-COM のプロパティ定義

WebTcl の動作を規定する、FE-COM のプロパティを定義します。

③ onLoad の function 定義

IE の強制終了を防止するための JavaScript コードを記述します。

5.6 偽接続

WebTcl Java API の PsvWTclClient#createResponse() メソッドで指定する、PsvWTclClient#CREATE_PSEUDO_CONNECT_RESPONSE モードの動作を示します。

e_connect 時に、アプリケーションの判断で接続を拒否したい場合に使用する動作モードで、PsvWTclClient#createResponse() メソッドの「加工用データ」の引数にエラーダイアログを表示するコマンドなどを指定可能です。

偽接続を受信した FE-COM は WebTcl エンジンに「加工用データ」の値をコマンドとして送信し、直後に AP サーバに接続終了を発行します。

エラー処理のコマンドを実行するため、「偽(にせ)」の接続状態を保つことから「偽接続」と名称を定義しました。

6 地図に関する用語

6.1 測地系

測地系は「地球上の位置をあらわすための基準面・原点・三角網の体系」を示します。一般に旧「日本測地系」(Tokyo97・Tokyo Datum)と呼ぶものは、「測地曲線座標系」と「3次元直角座標系」という座標系と「ベッセル楕円体」を使用した平面直交系座標を意味します。

「日本測地系 2000」(JGD2000)と呼ぶものは、「ITRF94」(国際地球基準フレーム)という座標系と、「GRS-80 楕円体」を使用した平面直交座標を意味します。

「世界測地系」と呼ぶものは、「ITRF」(国際地球基準フレーム)と、「WGS84」という座標系と、「WGS84 楕円体」を意味します。

6.2 楕円体

地球は真球でなく、赤道方向に広がった楕円形をしています。このため地球上の位置を正確に計算するためには、楕円体のモデルを定義して、楕円体のパラメータを計算に使用します。楕円体モデルは測地系ごとに異なり、旧日本測地系では「ベッセル楕円体」を使用して、地球の楕円体を平面の座標に投影計算を行います。

「日本測地系 2000」では「GRS-80 楕円体」を使用して、平面の座標に投影計算を行います。楕円体モデルごとのパラメータを表 6-1に示します。

表 6-1 楕円体モデルのパラメータ

楕円体モデル	赤道直径 (m)	扁平率の逆数 (1/f)
ベッセル楕円体	6,377,397.155	299.152813
GRS-80 楕円体	6,378,137	298.257222101