

MITSUBISHI

三菱Web地理情報システム構築パッケージ



PreSerV WebTcl 概要書

Ver. 6.2

マニュアル

はじめに

はじめに

本書は、PreSerV WebTcl の製品概要を説明します。

目次

1	マニュアルに関して	4
1.1	改訂内容一覧	4
1.2	表記に関して	8
2	PreServ WebTcl 製品概要	9
2.1	製品概要	9
2.1.1	WebTcl 開発の目的	9
2.1.2	WebTcl の製品特性	11
2.1.3	製品構成	12
2.1.4	機能概要	13
2.1.5	機能比較	13
2.2	WebTcl の動作に関して	15
2.2.1	WebTcl の動作環境	15
2.2.2	処理フローの違い	18
2.2.3	API 変更概要	20
2.3	A 版互換モードとは？	22
2.3.1	概要	22
2.3.2	A 版互換モードで起動するには？	22
2.3.3	特殊線を使用したアプリケーションの実行に関して	22
2.3.4	A 版互換モードの動作仕様差異	23
2.3.5	A 版互換モード制限事項	28
2.4	A 版統合モードとは？	29
2.4.1	概要	29
2.4.2	A 版統合モードの起動方法	29
2.4.3	A 版統合モードで使用可能となる機能	29
2.4.4	A 版アプリケーションへの影響	32
2.5	Tcl インタプリタについて	34
2.5.1	Tcl の言語仕様	34
3	データ仕様	37
3.1	ベクトルシンボル定義ファイル	37
3.2	ビットマップシンボル定義ファイル	46

4	エラーメッセージ及びデバッグツール	47
4.1	Tclインタプリタ独自のエラーメッセージ	47
4.2	PreSerVのTcl/IF独自のエラーメッセージ	48
4.3	デバッグツールによるエラー情報の取得	51
5	サンプルに関して	54
5.1	サンプルの動かし方	54
5.2	Wマクロサンプル	54
5.3	Gマクロサンプル	58
5.4	その他サンプル	64

1 マニュアルに関して

本章では、マニュアルの改訂内容を記載します。

1.1 改訂内容一覧

製品のリリース情報及びマニュアルの改訂内容を表 1-1に記載します。

記載情報は新しいものから順に記載しています。

表 1-1 改訂内容一覧

Ver. 6.2.0.13 リリース	
Ver. 6.2.0.13	他アプリケーションと同時実行時に IE 異常終了
他アプリケーションと同時実行時に IE 異常終了	他アプリケーションと同時実行時に通信待機が WebTcl 終了後も継続するケースに対応しました。
大量印刷時に不要なログを出力する	大量の印刷リクエストが発生した場合、極まれに不要なデバッグ・ログが出力するケースに対応しました。
ログに不要な文字列が入るケースの対応	動作環境により、起動時のデバッグ・ログに不要な文字列が入るケースに対応しました。

Ver. 6.2.0.12 リリース	
Ver. 6.2.0.12	e_error の動作調整改修
e_error と一部の通信エラーの動作条件調整	e_send 実行時の一部の致命的な通信エラー時に e_error でユーザマクロの実行が可能になりました。
w_unmap を発行後、画面が消去されない問題の修正	w_dialog に対して高負荷時に w_unmap を実行すると、描画イベントが貯まり、非表示処理が無効化される問題に対応しました。

Ver. 6.2.0.10 リリース	
Ver. 6.2.0.10	w_label のビットマップラベルの表示不具合改修
w_label のビットマップラベル表示不具合	w_label のラベルとして特定のビットマップを指定した場合に発生する表示不具合を修正しました。

Ver. 6.2.0.9 リリース	
Ver. 6.2.0.9	テキストボックスの文字入力イベント処理不具合改修
w_text_box コード判定修正	文字入力時のイベント処理の文字コード判定を修正しました。

Ver. 6.2.0.8 リリース	
Ver. 6.2.0.8	e_recv の通信効率改善、その他 A 版互換モードの調整
e_recv の通信効率改善	e_recv 時に通信処理を最小にする動作モードを追加し、デフォルトの動作としました。 また既存動作との互換性のため、FE-COM プロパティ「StandardRecvFlag」を追加しました。
A 版互換モードの調整	標準部品使用時の図形変形に関する仕様の調整を行いました。
起動 HTML の非表示機能の実装	起動 HTML を非表示にする機能を実装しました。

Ver. 6.1.1.4 リリース	
Ver. 6.1.1.4	重複エラーメッセージ指定機能対応・ドキュメント改善・FE-COM エラー表示・ログ出力・エラー後処置を Ver. 6.1.1.4 としてリリースします。
起動 HTML の非表示機能を追加	起動 HTML を非表示にする機能を追記しました。 注）本機能は、Ver. 6.2.0.2 で実装予定です。
重複起動エラーメッセージの指定機能を追加	重複起動エラー時のメッセージを指定する、FE-COM プロパティを追加しました。
ボタンのラベル変更処理改修	w_button で作成したボタンのラベルを、w_set_value で変更した時の不具合動作を改修しました。
FE-COM プロパティ説明改善	FE-COM プロパティ「CertRetryCount」「TimeOutExecCmd」「TimeOutExecInterval」「TimeOutVal」の説明を分かり易い様に改善しました。

Ver. 6.1.1.0 リリース	
Ver. 6.1.1.0	ビットマップファイルの拡張子の大文字対応・ドキュメント改善を Ver. 6.1.1.0 としてリリースします。
ビットマップファイルの拡張子の 大文字対応を追加	ビットマップファイルの拡張子に大文字を使用できるようにしました。

Ver. 6. 1. 0. 5 リリース	
Ver. 6. 1. 0. 5	各種機能・ドキュメント改善を Ver. 6. 1. 0. 5 としてリリースします。
WebTcl Java API のレスポンスのエラーチェック追加。	PsvWTclClient#createResponse() メソッドにエラーの戻り値-4を追加しました。 WebTcl Java API の JavaDoc を参照してください。
WebTcl Basic Media インストーラの改善	WebTcl の CD からメディアをインストールするインストーラが、Administrators 権限を持つユーザしか実行できないように修正しました。
バイナリアダプタのサンプル追加	バイナリアダプタのサンプルを追加しました。 以下のマニュアルを参照してください。 「PreSerV_WebTcl バイナリアダプタマニュアル」→2.1 節、2.2 節
e_recv のサンプル追加	e_recv のサンプルを追加しました。「e_recv 使用サンプル」を CD から自動再生される画面に追加しています。 ファイルセットを解凍後、スタートメニューの「12 追加ドキュメントフォルダを開く」を選択し、表示されるフォルダの「WebTcl e_recv 利用例サンプルインストール手順書」を参考にサンプルの設定を行ってください。

Ver. 6. 1. 0 リリース	
Ver. 6. 1. 0	インデックス機能追加版を Ver. 6. 1. 0 としてリリースします。
インデックス機能追加	<p>プロシジャの自動ロードが可能になる、インデックス機能を追加しました。</p> <p>以下のマニュアルを参照してください。</p> <p>「PreSerV WebTcl API リファレンス (その他)」→3. 4. 15 項「TclIndexURL」、4 章「その他のコマンド」</p>
印刷余白指定オプション追加	<p>Windows プリンタで余白を指定するオプションを追加しました。</p> <p>以下のマニュアルを参照してください。</p> <p>「PreSerV WebTcl API リファレンス (Gマクロ)」→2. 7. 6 項「g_open_gwin」</p> <p>「PreSerV WebTcl 概要書」→5. 4 節「GUI 構築例サンプル」の印刷ダイアログサンプル (print.psv)</p>
偽接続モード追加	<p>WebTcl Java API に、サーバサイドのアプリケーションで接続を拒否する場合、エラー処理を実装可能な偽接続モードを追加しました。</p> <p>以下のマニュアルを参照してください。</p> <p>「PreSerV WebTcl 用語集」→5. 6 節「偽接続」</p> <p>「WebTcl アプリケーション作成ガイド」→6. 1. 5 項「アプリケーションによる接続拒否の実装」</p>
インストーラ改善	基本モジュールのインストーラ、WebTclInstFull.exe のコントロールパネルからの動作を改善しました。

初版リリース	
初版リリース	初版を Ver. 6.0.0 としてリリースします。
マニュアルの PDF 化	マニュアルを PDF 化して、印刷しやすくしました。
PreSerV WebTcl 変換ユーティリティの追加	文字列⇔PreSerV 地物オブジェクト間の変換をサポートする変換ユーティリティを追加しました。
動作環境の追加	以下の環境をサーバ構成例に追加しました。 Java : J2EE 1.5 AP サーバ : uCosminexus V7
初版以前の改訂履歴	
A 版統合モードの追加	A 版互換 API を V5 版標準モードに追加した A 版統合モードを追加しました。
レイヤ名のサポート	既存のレイヤ番号に名称を関連付ける API、g_layer_name を追加しました。
API の追加	JavaScript のコールバックを実行する、e_external_send API の記述を追加しました。
ホイールイベント検知	w_event_handler にホイールマウスのホイールイベントを検知してマクロを実行する-w オプションを追加しました。
クリック時のビットマップ設定追加	w_button と w_arm_button に、マウスクリック時のビットマップ指定オプション(-g)を追加しました。
デバッグツールのバージョンアップ	デバッグツールにコマンドヒストリー機能を追加しました。

1.2 表記に関して

マニュアル中では、PreSerV で提供する API で、ウィンドウ操作コマンド群を「W マクロ」、図形操作コマンド群を「G マクロ」と表記します。詳細は「PreSerV WebTcl API リファレンス(W マクロ)」、「PreSerV WebTcl API リファレンス(G マクロ)」、「PreSerV WebTcl 用語集」をお読みください。

2 PreSerV WebTcl 製品概要

2.1 製品概要

2.1.1 WebTcl 開発の目的

PreSerV のクライアントサーバ系列製品（以下C/S版、V5 版と表記）は、1990 年代後半から製品提供を行っています（図 2-1参照）。

C/S 版はTcl 言語のインターフェースを使った、スクリプトベースのプログラミングが特徴です。この C/S 版の初版である A 版を使用したシステムは、クライアントサーバシステム（以下 C/S システム）から、現在主流の Web システムへと移行する時期になっています。

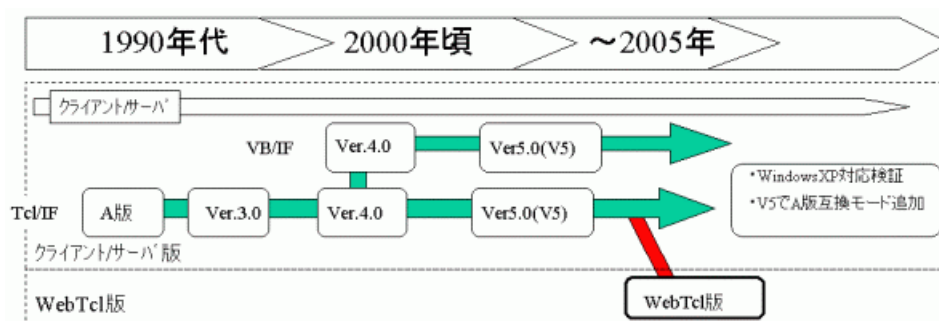


図 2-1 製品ロードマップ C/S 関係抜粋

WebTcl は、C/S 版の Tcl インターフェースを使用して開発した GIS アプリケーションプログラムを保有し、Web アーキテクチャに移行したいユーザを対象として、以下の目的で開発されました。

① Tcl 過去資産の有効活用

WebTcl 版ではアーキテクチャの互換性を保つため、GUI 層の命令の互換性を重視しています。GUI 層の命令の互換性によって、Tcl 言語で作成した過去の S/W 資産※¹を有効に活用し、Web システム移行時のアプリケーションの再生産量を低減します。

※¹ アプリケーションのプレゼンテーション層ロジック部分

② Web システムやネットワーク環境への適合

通信層の命令を HTTP ベースに移行して、PreSerV Tcl エンジン自体を HTTP クライアント・アプリケーションとして動作させることにより、現在主流の Web システムベースのネットワーク・インフラ環境に適合します。

③ PreSerV for Web との関係

Web システム向けの既存の Web-GIS である PreSerV for Web とのデータ関係を可能とし、製品間の相互運用性を高めます。

WebTcl 版の C/S システムから Web システムへの移行の概念を図 2-2 に示します。

図の左側が、一般的な C/S システムから Web システムへの移行の概念図です。

図の右側が、C/S 版から WebTcl 版を適用して Web システムへ移行する時の概念図です。

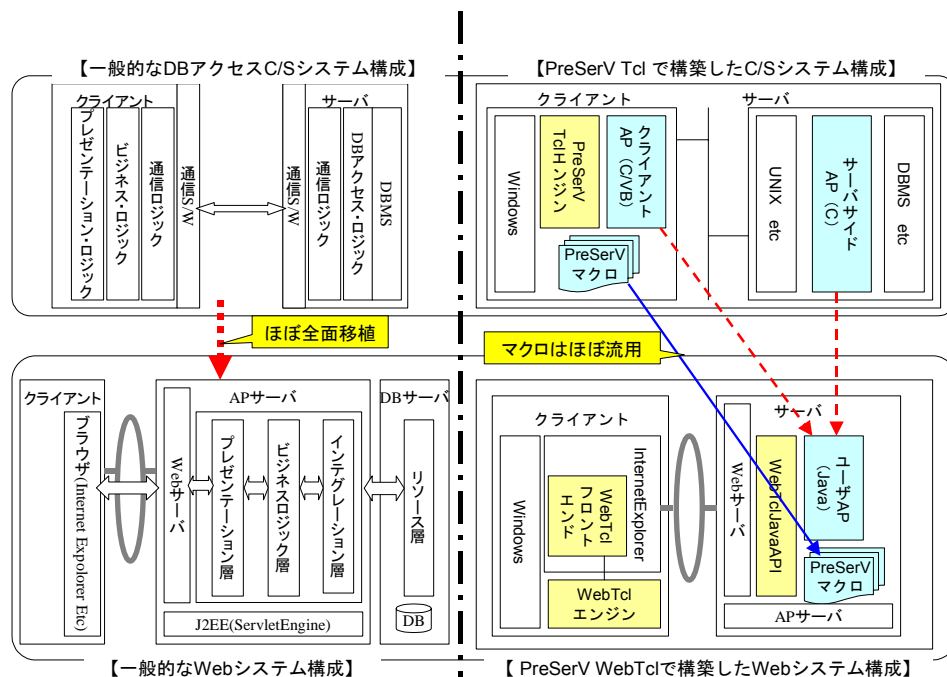


図 2-2 WebTcl への移行の概念

一般的な Web システムへの移行の場合、全て異なるアーキテクチャへの移行となるため、全面的なプログラム移植が必要となります。

C/S 版からの WebTcl 版への移行の場合、GUI を記述する PreSerV マクロファイルをほとんど流用可能であるため、一般的な移行と比較して、過去資産の流用率が高くなります。

2.1.2 WebTcl の製品特性

PreSerVファミリー製品における、WebTcl版の位置付けを図 2-3に示します。

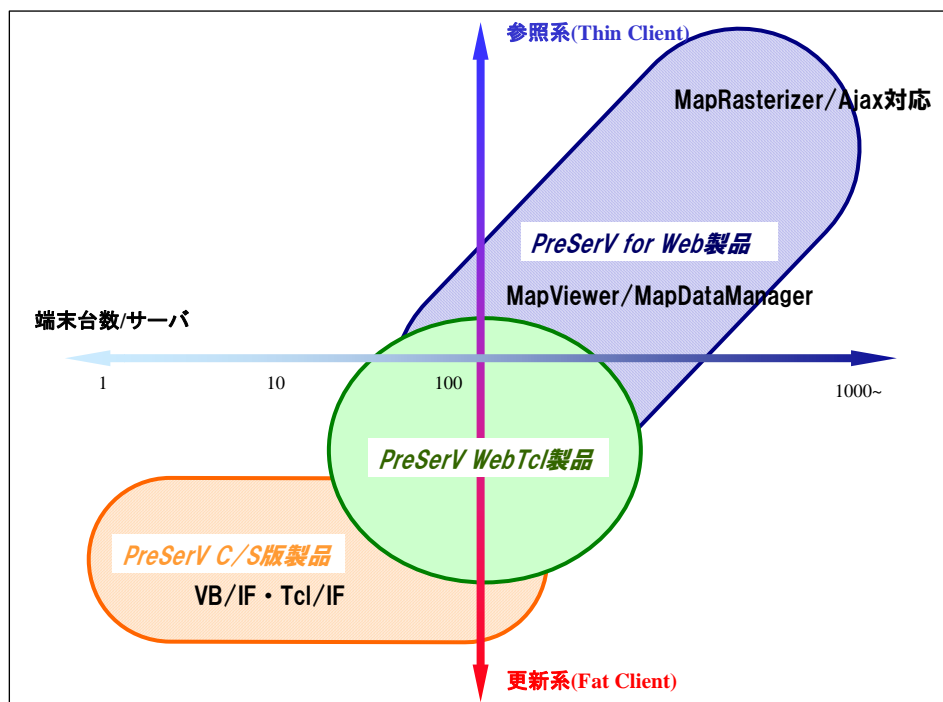


図 2-3 WebTcl の位置付け

図のXY軸は以下のように定義しています。

X軸：対象システムの規模を端末台数で表現しています。

Y軸：更新系か、参照系のアプリケーションかを表現しています。

このように、WebTcl 版は C/S 版から Web 版製品の中間的な特性を持つ製品です。

2.1.3 製品構成

WebTcl版で実行環境に必要なコンポーネントは、図 2-4の赤太線で示します。開発環境では、赤太点線の範囲のC/S版のツール類も必要のため、「PreSerV WebTcl基本パック」ではV5 版のメディアを同封します（図 2-4参照）。

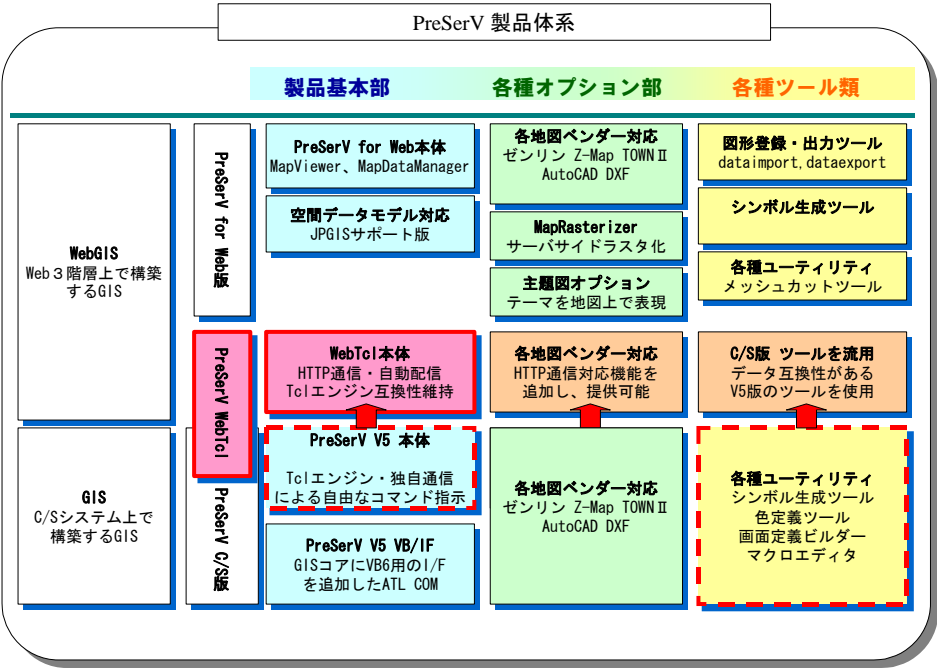


図 2-4 PreSerV 製品体系

2.1.4 機能概要

WebTcl 版で実現する機能を以下に示します。

① 過去の資産流用可能

V5 版のマクロ命令／Tcl 組み込みコマンド命令の互換性維持によりスクリプトの流用を実現します。W・G マクロ命令の互換性維持は、ユーザインターフェース層のマクロ命令や、画面を構成するマクロ命令のそのままの実行を可能として、過去資産の高い流用性を実現します。

② アプリケーションプログラムの配布不要

スクリプトファイルを Web サーバからダウンロード可能にすることで、アプリケーションプログラムを配布する必要がありません。この機能により、クライアント PC の管理負担を軽減します。

③ 高いユーザ操作レスポンス

GUI の制御は、クライアント側で処理するため、高い操作レスポンスが実現可能です。

④ JavaScript⇔Tcl のメッセージ交換

JavaScript からの Tcl コマンド、Tcl からの JavaScript イベントのメッセージ交換を可能とする機能を実装しました。この機能により、GIS 画面と一般の JavaScript ベースのアプリケーションと関係処理を構築可能です。

2.1.5 機能比較

WebTcl 版は V5 版をベースに基本機能を実現しながら、一部の機能の変更と追加を行っています。

表 2-1にV5 版とWebTcl版の機能比較を示します。

表 2-1 WebTcl との機能比較

機能分類	機能名称	PreSerV V5 版	PreSerV WebTcl 版	WebTcl 版での変更点
マクロ命令	W マクロ	○	○	HTML ウィンドウとの統合可能
	G マクロ	○	○	なし
	E マクロ※1	○	○	JavaScript 連係追加 一部通信コマンドの動作仕様変更
通信機能	クライアント関数	Socket ベース	HTTP ベース	WebTcl Java API へ移行
Tcl 組み込み コマンド	source	○	○	source で指定するファイル名は相対パスとして、FE-COM. codeBase と合成して URL を作成し、HTTP 経由でダウンロード
	cd、pwd など	○	○	AP サーバ上の相対パスの概念でエミュレート
外部連係	—	×	○	JavaScript との連係機能や HTML ウィンドウとの統合
A 版互換機能	エンジン部 ※2	○	○	
複数起動	エンジン部 ※2	○	×	V5 版でサポートした 1 クライアント上で複数エンジンを起動する機能は、WebTcl 初版ではサポートしていません。

※1 E マクロは、通信機能の API や、タイマーによるスクリプト実行、スレッドでのイベント待ちなどを行う API です。

※2 エンジン部は 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、プロセスを示します。

2.2 WebTcl の動作に関して

2.2.1 WebTcl の動作環境

WebTcl の動作環境に関して、構成例などの概要を説明します。

(1) WebTcl システム構成概要

WebTcl を適用した場合の、システム構成例を図 2-5 に図示します。※2

※2 網掛け部分が WebTcl の製品コンポーネント

WebTcl では J2EE ベースの最小構成の Web システムと一般的な性能のクライアントを想定して開発しました。

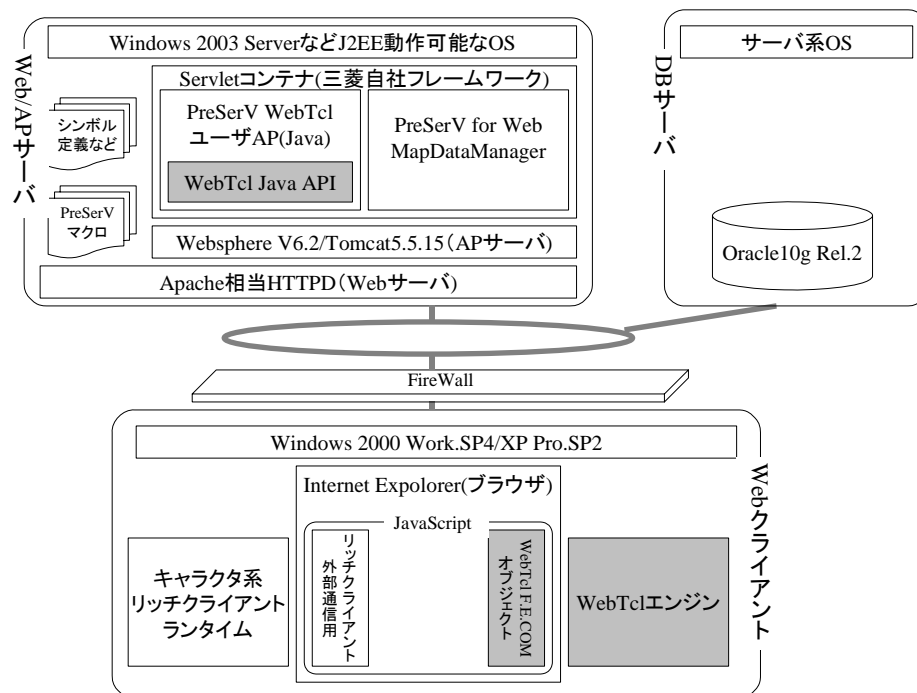


図 2-5 WebTcl システム構成例概要図

(2) Web クライアント PC 構成例

WebクライアントPCの構成例を表 2-2に示します。

WebクライアントPC内部のプロセス間通信で 30001～30003 のポートを使用するため、重複するポートのプログラム使用は避けてください。

表 2-2 Web クライアント PC の構成例

No.	環境カテゴリ	環境定義	備考
1	機種	x86 互換の PC	マルチプラットフォームは想定していません。
2	CPU 性能	1GHz 以上の x86 互換 CPU 推奨	
3	メモリ性能	512MB 以上推奨	
4	HDD	数 MB～数十 MB	クライアントのランタイム部分で容量概算しています。
5	グラフィックス	インテル社チップセット Intel915 以降のオンボード VGA の 2D 性能推奨	最近の PC オンボード VGA の 2D 性能があれば快適に動作可能です。
6	OS	Windows2000 Work. SP4 WindowsXP Pro. SP2 (Vista は追加検証で対応)	Windows2000 は既存資産クライアントの有効利用のため対応します。
7	ブラウザ	Internet Explorer6.0 SP1/SP2 (他 ActiveX が実行可能なブラウザは追加検証で対応予定)	ネイティブ・コードによる Web クライアントであるため、ブラウザは限定します。左記のブラウザであっても、タブブラウジング機能を付加したものは、動作が保障できません。タブブラウジング機能が付加されたブラウザは、使用しないでください。
8	通信機能	一般のイントラネットの通信速度を推奨 (数 Mbps～100Mbps)	

(3) サーバ構成例

Webサーバ/APサーバの構成例を表 2-3に示します。DBサーバに関してはWebTclでは制限がないため、一般的なRDBの例を記載しています。サーバに関してはJ2EE1.4のAPサーバが動作することが前提で、特に機種、OSは制限しません。以下の記述は最小構成の例として構成例を示します。

表 2-3 Web サーバ/AP サーバの構成例

No.	環境カテゴリ	環境定義	備考
1	機種・H/W	PC サーバ/PC ワークステーション	メモリ：1GB 以上 CPU：3GHz 以上
2	OS	Windows 2003 Server/WindowsXP Professional	
3	AP サーバの S/W	Tomcat5.0.23 Tomcat5.5.20 uCosminexus V7 (Websphere V6.2 ^{注1)})	J2EE1.4かJ2EE1.5のサーブレットコンテナとなるAPサーバ。
4	Web サーバの S/W	Apache 相当の HTTPD	AP サーバ付属の HTTPD があるため、明確にバージョンは定義しません。
5	DB サーバ	Oracle10g Release2	
6	Java プラットフォーム	J2EE 1.4 環境	
7	プレゼンテーション層・ビジネスロジック層のプラットフォーム	一般的なサーブレットで作成	PreSerV フレームワーク機能は使用しません。
8	その他	log4jのモジュールをAPサーバ共通ライブラリに配置	Tomcat の場合、log4j-1.2.9.jar を common/lib に配置する。

注 1) Websphere V6.2 はβ版までの動作確認で、参考環境とさせていただきます。

2.2.2 処理フローの違い

C/S 版と WebTcl 版の処理フロー、データフローの違いを説明します。

(1) C/S 版処理フロー

C/S版でアプリケーションを構築した時の、一般的な通信フローを図 2-6に示します。

① パターン 1-1

PreSerV サーバ(Tcl エンジン)から通信を開始した場合の接続処理です。

② パターン 1-2

e_send で処理の依頼・受信をコールバック処理 (PSV クライアント関数の概念) で実現した処理です。

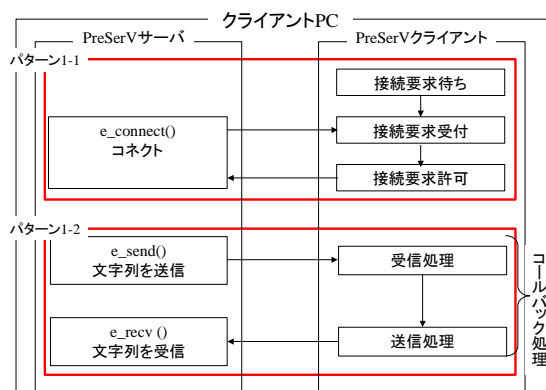


図 2-6 C/S 版処理フロー

(2) WebTcl 版処理フロー

前述のC/S版の通信フローをWebTcl 側に移植した場合を図 2-7に示します。

① パターン 1-1

C/S 版の接続処理は、ほぼ同等の処理が移植可能です。

② パターン 1-2

C/S 版のコールバックを、アクション URI (WebTcl Java API 参照) で登録した、アクションキーワードに対応した AP サーバサイドの処理に変更することで同等の処理フローが実現可能です。

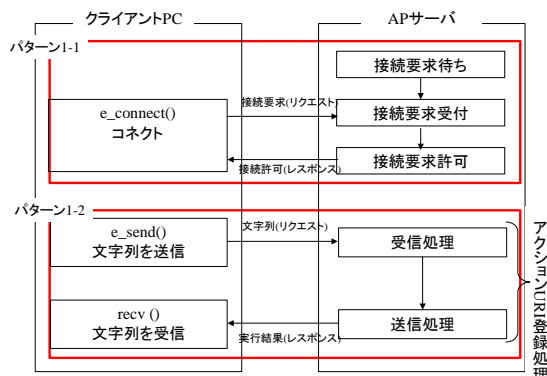


図 2-7 WebTcl 版処理フロー

(3) 実現不可能な C/S 版処理フロー

WebTcl 版では、アプリケーション側からの通信開始パターンの処理フローパターンに関しては、別途処理フローの移植の検討が必要です（図 2-8参照）^{注 2)}。

この処理フローでは、全てアプリケーション側から通信を開始しています。WebTcl ではこの処理は、通信開始の検出をクライアントからポーリングすることで、AP サーバからの通信開始をエミュレートすることになりますが、現在はこの処理は未サポートです。

注 2) 使用ケースが少ないフローであり、ポーリングによる Web システムへの負荷が大きいため、実装を省略しています。

したがって、アプリケーション側からの通信開始パターンが必要な場合は、通信シーケンスを2.2.2-(2)の処理フローへ移植してください。

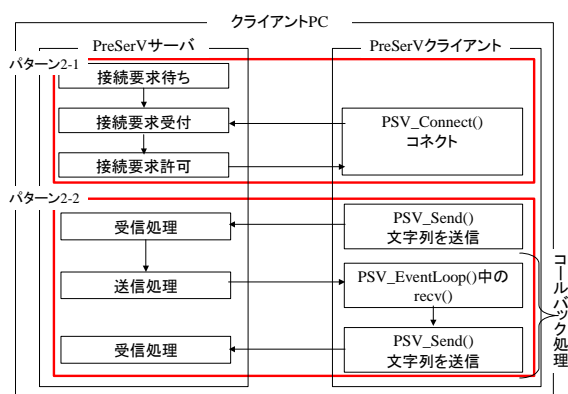


図 2-8 アプリケーション側からの通信開始パターン

2.2.3 API 変更概要

API 種別ごとに、V5 版と WebTcl 版の、API の変更点の概要を説明します。

(1) W マクロ

WebTcl 版では、W マクロは `w_top` マクロに“-a”を追加しました。他の W マクロの仕様は変更ありません。

(2) G マクロ

WebTcl では、v5.7.0.1 で `g_layer_name` を追加しました。

(3) E マクロ

E マクロでは、通信に関係するマクロの内部動作を変更しました。WebTcl 版の通信先は FE-COM が指定する AP サーバです。V5 版までの通信先のホスト名は互換性のためダミーとして残しています。

また、`e_plugin` に関しては Web サーバからプラグインモジュールをダウンロードして実行するように内部動作を変更しています。

(4) Tcl 内部コマンド

`cd`、`pwd`、`source` といった、ローカル・ディレクトリの概念に関連するコマンドと、マクロファイルの読み込みに関するコマンドは、AP サーバ上の URL からの相対パスの概念に置き換えました。

その他のローカル・リソースアクセスのコマンド(ファイルの書き込みコマンドなど)は旧仕様どおり PC 上のファイルにアクセスするため、`cd`、`pwd`、`source` コマンドの対象リソースと同じリソースを想定したプログラムは移植する必要があります。

例) マクロコマンドをローカルファイルに書き出して、後で `source` コマンドを実行する処理は WebTcl 版ではそのままでは動作しません。

(5) クライアント関数

PSVクライアント関数に関しては、WebTcl Java APIの機能に置き換えました。

表 2-4にWebTcl版でのクライアント関数の対応を示します。

表 2-4 PSV クライアント関数の置換え一覧

クライアント関数	対応する WebTcl Java API クラス	対応概要
PSV_Initialize	なし	初回コネクション時の起動アクションで初期処理を代行するためこのAPIは廃止しました (2.2.2(3) 参照)。
PSV_Connect	なし	サーバからの通信開始動作はサポートしていません (2.2.2(3) 参照)。
PSV_Close	PsvWTclClient	PsvWTclClient のメソッドを実行し、WebTcl セッション ID や、関連する情報の破棄を行い、通信の終了処理とします。
PSV_Listen	なし	サーバからの通信開始動作はサポートしていません (2.2.2(3) 参照)。
PSV_AddCallback	PsvWTclClient	アクションキーワードとアクション URI の対応付けを PsvWTclClient のメソッドで登録します。
PSV_ErrorHandler	PsvWTclFilter PsvWTclClient	アクションキーワード”error” とアクション URI の対応付けを PsvWTclClient のメソッドで登録します。
PSV_EventLoop	なし	リクエスト時に指定したアクション URI からフレームワーク側で定義するアクションが起動する形式になるため、不要です。
PSV_Send	PsvWTclClient	送信文字列を、PsvWTclClient のメソッドをコールしてレスポンス形式に加工し、ユーザのアウトプット・アクションから返す処理に置き換えます。非同期の返信時は PsvWTclClient の情報を蓄積するメソッドをコールし、e_recv によるクライアントからの問合せリクエストを待ちます。
PSV_SendFile	PsvWTclClient	PSV_SendFile は”source” マクロの実行処理に置き換えてください。
PSV_AddInput	なし	ユーザ独自の通信は、AP サーバ上でアプリケーションが自由に実装できるため機能提供は行いません。

2.3 A 版互換モードとは？

A 版互換動作モードについて説明します。

2.3.1 概要

A 版互換動作モードとは、V5 版開発時に追加した機能で、A 版とほぼ同一の API 仕様・動作仕様でサポートする互換動作モードを示します（2.1.1 参照）。

この動作モードは A 版ユーザ向けで、A 版の機能をほぼ再現することを目的としており、A 版以降追加した V5 版の API は、API の仕様統一のため動作しません。新規アプリケーション作成時は本モードを使用しないでください。

また A 版互換モードの API を、V5 標準モードの API と排他的に設定するため、A 版互換モード時は A 版以降追加した API は実行できません。

2.3.2 A 版互換モードで起動するには？

- (1) FE-COM.TclVersionOption プロパティに“-A”を指定して、起動 HTML を作成してください。
- (2) A 版のマクロで最初に実行するマクロスクリプトを FE-COM.InitSource プロパティに設定してください。

2.3.3 特殊線を使用したアプリケーションの実行に関して

A 版の標準機能である特殊線機能は、外部アプリケーションという扱いに変更しているため、以下のようにして実行してください。

- (1) WebTcl 版向けの特殊線プラグインのプロジェクトを入手してください。
特殊線プラグインの描画仕様は A 版で作成したシステムの動作仕様がベースとなっており、WebTcl の製品構成には含みません。特殊線プラグインは個別に配布しません。必要な場合は別途連絡してください。
- (2) 特殊線プラグインを WebTcl 版のヘッダ・ライブラリで再コンパイルしてください。
コンパイルした DLL は、FE-COM オブジェクトの CodeBase プロパティで指定する URL 以下に配置してください。
- (3) 特殊線を実行するアプリケーションのマクロスクリプトの先頭に以下の命令を追加してください。e_plugin は、Ver. 3 で追加した API ですが、A 版互換モードでの特殊線の実現や、拡張性の目的のため有効にしています。

リスト 2-1 特殊線プラグイン宣言例

```
# A 版互換用の特殊線プラグイン宣言
set vcheck "0"
catch {set vcheck $psvTclVersionOption}
if { $vcheck == "1" } {e_plugin "applines.dll" g_applines g_draw_applines}
```

- (4) 前述の A 版互換モードの起動を行ってください。

2.3.4 A 版互換モードの動作仕様差異

A版互換動作モードでは、ほぼA版と同じ動作仕様を実現しますが、表 2-5の項目の動作仕様に関しては、動作が異なります。

“No.” の色分けは以下の規則です。

① 水色の項目

A 版正常動作に関する差異です。

② 黄緑の項目

A 版エラー処理に関する差異です。

表 2-5 A 版互換モードの差異

No.	動作概要	A 版動作仕様	A 版互換モード動作仕様
1	w_dialog モーダル時の親ウィンドウの挙動	モーダル時は親ウィンドウをクリックしても、表示は変化ありません。	モーダル時に操作禁止中の親ウィンドウをクリックすると、子ウィンドウが点滅します。
2	フォーカス	w_combo_box など、親ウィンドウがアクティブでなくなった時、フォーカスがなくなります。	w_combo_box など、親ウィンドウがアクティブでなくなった時、フォーカスが残ります。
3	w_combo_box のテキストボックスの表示位置	w_combo_box のテキストボックスの表示位置が 1 ドット程度左上にずれています。	w_combo_box のテキストボックスの表示位置を正しい位置に表示します。
4	w_list_box のスクロールバーの挙動	w_list_box の項目がある場合、表示範囲内でもスクロールバーを表示し、クリックすると消去します。	w_list_box の項目がある場合、表示範囲内の時、スクロールバーは表示しません。
5	w_slider の背景色	w_slider の背景色を w_set_value で変更できません。	w_slider の背景色を w_set_value で変更可能です。
6	色 ID256 が使用不可	色 ID256 が使用可能 (1~256 の範囲)。	V4 版以降は 8bit で色 ID を指定できるようになり、色 ID255 までしか使用できません。色 ID256 は使用不可です (1~255 の範囲。0 は透過色指定)。
7	太さ 2 以上の線種が有効	図形オブジェクト描画時に線幅 2 以上の時実線しか描画できません。	図形オブジェクト描画時に線幅 2 以上の線描画の線種が有効になっています。
8	図形描画イメージのずれ	—	座標投影精度調整のため、同じ WC 座標でも 1 ドット描画が異なる場合があります。
9	ミラー文字のサポートに関して	ミラー文字可能。	TrueType サポートにより、一部のフォント種別だけミラー文字不可となってしまうため、ミラー文字は使用不可として仕様統一しました。

No.	動作概要	A 版動作仕様	A 版互換モード動作仕様
10	縦横比率不一致時の回転 文字列描画	グラフィックウィンドウの DC サイズ (ピクセル単位のサイズ) の縦横比率と、WC サイズの縦横比率が異なる場合、文字列回転を行った複数文字の文字オブジェクトが 1 列にずれて描画します。	グラフィックウィンドウの DC サイズ (ピクセル単位のサイズ) の縦横比率と、WC サイズの縦横比率が異なる場合、文字列回転を行った複数文字の文字オブジェクトが 1 文字ごとにずれて描画します。
11	フィルターの制御点に対する色適用の差異	フィルターを使用した点変更モード時に、アクティブな制御点が図形オブジェクトの色でそれ以外の制御点が <code>g_open_filter</code> で指定した色で描画します。	フィルターを使用した点変更モード時に、アクティブな制御点が <code>g_open_filter</code> で指定した色、それ以外の制御点が図形オブジェクトの色で描画します。
12	パラメータ指定時の区切り文字の差異	パラメータ指定時に全角スペースも区切り文字として使用可能です。	パラメータ指定時のスペースを全角スペースにすると、エラーになります。
13	矩形検索時の円の判定領域	<code>g_draw_obj</code> で、矩形範囲を DC 指定 (<code>-dc</code> オプション) で描画した場合、矩形からはみ出た図形も描画します。	<code>g_draw_obj</code> で、矩形範囲を DC 指定 (<code>-dc</code>) で描画した場合、矩形内でクリッピングして描画します。
14	矩形検索時の円の判定領域	円の最大描画領域は、矩形で近似判定しています。	円の最大描画領域は、円の領域で判定しています。
15	文字オブジェクトの線種が有効	文字オブジェクトの線種指定は無効です。	文字オブジェクトの線種指定が有効です。
16	スプラインの線タイプの設定が無効	スプラインの線種指定が有効です。	スプラインの線種指定が無効です。
17	オプションパラメータ指定無し時	メモリ不正アクセスとしてエラー処理を実行します。	デフォルト値で正常動作します。
18	<code>w_text_box</code> のモード不正時の動作	モードを不正に指定した場合、2 行のマルチラインテキストとなります。	モードを不正に指定した場合、シングルラインテキストとなります。
19	ビットマップ指定不正時の動作	<code>w_button・w_arm_button・w_check_box・w_radio_button</code> で、不正なビットマップファイル名を指定すると、「def」と書いたビットマップを表示します。	<code>w_button ・ w_arm_button ・ w_check_box ・ w_radio_button</code> で、不正なビットマップファイル名を指定すると、キャプション文字列のデフォルトを表示します。

No.	動作概要	A 版動作仕様	A 版互換モード動作仕様
20	w_combo_box・w_list_box の幅指定不正時	w_combo_box・w_list_box で不正な幅を指定すると、ウィンドウを描画しません。	w_combo_box・w_list_box で不正な幅を指定すると、2・3 ドットのデフォルトの幅でウィンドウを表示します。
21	w_combo_box 高さ指定不正時	w_combo_box・w_list_box で不正な高さを指定すると、ウィンドウを描画しません。	w_combo_box・w_list_box で不正な高さを指定すると、デフォルトの1文字分の高さでウィンドウを表示します。
22	w_message_box の親ウィンドウ	ダイアログを親ウィンドウに指定するとエラーになります。	ダイアログを親ウィンドウに指定可能です。
23	w_combo_add で項目設定時の表示位置調整動作	w_combo_box のリストボックスを開いた時に、スクロールしてから開いています。	w_combo_box のリストボックスを開いた時に、反転している項目がトップに近くなるようにスクロールします。
24	w_text_ins_pos でテキストカーソル位置移動の動作	1 行目の改行位置にテキストカーソルあった場合、w_text_ins_pos の実行後テキストカーソルが2 行目の先頭に移動するケースがありました。	1 行目の改行位置にテキストカーソルあった場合、w_text_ins_pos の実行後テキストカーソルは正常な位置にあります。
25	w_message_box のメッセージ省略時の動作	メッセージボックスのメッセージ文字列指定をなしにすると、NULL となります。	メッセージボックスのメッセージ文字列指定をなしにすると、デフォルトの文字列 messege を表示します。
26	w_get_value で w_combo_box の-m オプションの動作	get_value に-m を指定した場合、マニュアルにない w_combo_box の値を取得します。	get_value に-m を指定した場合、A 版互換動作モードでは、w_combo_box の値を取得しません。
27	w_combo_del のアイテム指定不正時の動作	w_combo_del で、アイテム指定なしか、存在しないアイテムを指定した場合、最終項目を削除します。	w_combo_del で、アイテム指定なしか、存在しないアイテムを指定した場合、処理無効となります。
28	w_list_del でアイテムの指定不正時の動作	w_list_del で、アイテム指定なしか、存在しないアイテムを指定した場合、最終項目を削除します。	w_list_del で、アイテム指定なしか、存在しないアイテムを指定した場合、処理無効となります。
29	w_list_sel・w_list_desel でアイテム指定不正時の動作	w_list_sel・w_list_desel で、アイテム指定なしか、存在しないアイテムを指定した場合、最終項目が選択・選択解除します。	w_list_sel・w_list_desel で、アイテム指定なしか、存在しないアイテムを指定した場合、処理無効となります。

No.	動作概要	A 版動作仕様	A 版互換モード動作仕様
30	図形オブジェクト作成時のモード指定 (-m) の動作	-m オプションで、スプライン指定の 1 と塗りつぶしの 2 の論理和である 3 を指定しても、有効です。	-m オプションで、スプライン指定の 1 と塗りつぶしの 2 の論理和である 3 を指定しても、線を描画しません。
31	廃棄済みオブジェクトの再廃棄	g_close_context などオブジェクトの廃棄命令を同一のオブジェクトに対して 2 回コールした場合、エラーになります。	g_close_context などオブジェクトの廃棄命令を同一のオブジェクトに対して 2 回コールした場合、エラーになります。
32	変形時の制御点動作の差異	扇形の半径の線を伸ばしても半径より先は描画しません。	扇形の半径の線を伸ばした場合、伸ばした分も描画します。
33	テキストの負値回転角度指定時の動作	回転角度に負値を指定すると、テキストを描画しません。	回転角度に負値を指定すると、テキストを絶対値の回転角度で描画します。

2.3.5 A 版互換モード制限事項

A 版互換動作モードは、V5 版の標準モードに対して以下の機能制限を加えています。

- (1) スレッドを生成しないため、図形の描画・図形のメモリ開放時に処理性能が劣ります。

A 版はシングルスレッドのプログラムであり、マルチスレッドで A 版互換モードを実行すると、シングルスレッドを前提としたアプリケーションの処理シーケンスで、デッドロック等の排他制御の問題が発生する可能性があります。

このため、スレッドを生成し処理性能を向上させた図形の描画・図形のメモリ開放処理が実行を抑制しています。

- (2) 前景色・背景色を指定した塗りつぶしはできません。
- (3) A 版以降追加した Tcl/IF のマクロは使用できません。

本モード時は、V5 版の機能を抑制しているため、A 版以降追加した Tcl/IF のマクロコマンドを内部的なテーブルから削除して動作します。これは V5 版の API で作成した図形を、A 版の API で操作した場合、動作仕様に統一性がなくなることを回避するためです。

2.4 A 版統合モードとは？

A 版統合動作モードについて説明します。

2.4.1 概要

A 版統合動作モードとは、A 版と V5 版の API を統合した動作モードを示します。この動作モードは A 版ユーザ向けの機能であり、V5 版の機能をベースとして A 版の API も使用可能にした動作モードです。本動作モードは A 版で作成したアプリケーションで V5 版以降の標準機能を使用可能にすることを目的とします。

新規アプリケーション作成時は本機能をお奨めしません。新規アプリケーションは V5 版標準モードの API で作成願います。

2.4.2 A 版統合モードの起動方法

- (1) FE-COM.TclVersionOption プロパティに“-Z”を指定して、起動 HTML を作成してください。
- (2) A 版のマクロで最初に実行するマクロスクリプトを FE-COM.InitSource プロパティに設定してください。

2.4.3 A 版統合モードで使用可能となる機能

A 版統合動作モードは、A 版の API を利用する場合、A 版互換モードと同じ動作を行います。以下の V5 版機能の使用が可能^{注 3)}な点が異なります。

注 3) A 版の後で追加し、現在廃止している API は使用解除の対象外です。

以下に A 版統合と A 版互換モードの動作仕様の差を説明します。

- (1) G マクロで使用可能になる API

A 版統合モードは表 2-6 の G マクロ機能が使用可能になります。

表 2-6 使用可能となる G マクロ

No.	機能大分類	機能中分類	概 要
1	図形追加	ビットマップシンボル	ビットマップファイルを使用した、アイコン用の図形のサポート。 g_bitmapsymbol
2		鉄道線	JR 線や私鉄線など幅を持った特殊な線描画を行う図形のサポート。 g_RRline
3	マップ機能	マップ操作・制御	メッシュ単位を束ねて管理可能なマップオブジェクトの概念サポート。 g_map_element, g_map_type, g_map_attrib, g_map_range, g_map_margin, g_map_load
4		ベクトルマップ	ディスプレイリストをメッシュとして管理可能なベクトル図形専用のマップをサポート。 g_open_map, g_close_map
5		ラスターマップ	イメージファイルをメッシュとして表示可能なラスター地図専用のマップをサポート。 g_open_map, g_close_map
6	座標変換機能	旧日本測地系	横メルカートル図法により旧日本測地系の緯度経度と XY 座標の相互変換が可能。 g_yoko_mercator
7		新日本測地系	新日本測地系の緯度経度と XY 座標の変換および、旧日本測地系から新日本測地系への緯度経度変換が可能。 g_TBL2JBL, g_JBL2JXY, g_TKY2JGD_ParaFile
8	図形関連機能	各種図形操作機能	拡張した描画属性、オブジェクト属性が使用可能。またレイヤ操作の追加機能が使用可能。 g_layer_range, g_draw_attrib, g_obj_attrib, g_cur_draw_attrib, g_cur_obj_attrib, g_paint_pattern_file, g_obj_range
9	クリップボード	クリップボード転送機能	画面描画イメージをメタファイル形式でクリップボードに転送することが可能。転送データは一般的なアプリケーションで貼り付け可能。 g_set_clipboard
10	スケール	スケール制御	地図の表示サイズが実寸値の何分の 1 か表示縮尺で管理・制御する機能をサポート。 g_scale

(2) Wマクロで使用可能になるAPI

A版統合モードでは表 2-7のWマクロ機能が使用可能になります。

表 2-7 使用可能となる W マクロ

NO.	機能大分類	機能中分類	概 要
1	マウスカーソル	マウスカーソルの形状変更	マウスカーソルの形状が変更可能となり、操作種別をカーソルで識別可能。 w_cursor
2	フレーム	フレーム部品の追加	画面レイアウトを明確にするフレーム部品が使用可能。 w_frame
3	イベント	イベントのキャプチャ	他のウィンドウ部品のイベントをキャプチャし、自分のイベントとして処理可能。 w_grab
4	メッセージボックス	メッセージボックスの追加	メッセージボックスを使用可能。 w_message_box
5	IME	テキストボックスの仮名漢字変換モード	テキストボックスの仮名漢字変換モードの ON/OFF を制御可能。 w_text_fence
6	ツールチップ	ツールチップテキストの表示	マウスを上にした時、ツールチップテキストを表示可能。 w_tooltip_color, w_tooltip_enable, w_tooltip_init, w_tooltip_set
7	トップウィンドウ	トップウィンドウの制御追加	トップウィンドウのクローズ、サイズ変更時に実行するマクロの指定が可能。 w_top

(3) その他のマクロで使用可能になるAPI

A版統合モードでは表 2-8のその他のマクロ機能が使用可能になります。

表 2-8 使用可能となるその他のマクロ

No.	機能大分類	機能中分類	概 要
1	E マクロ	タイマー割り込み	タイマー割り込みのマクロを定義可能。 e_timer
2		スリープ実行	指定ミリ秒数処理を停止可能。 e_sleep
3		プラグインによる拡張	プラグインによるマクロコマンドの拡張が可能。 e_plugin
4		日付取得	システム日付の取得が可能。 e_date

2.4.4 A 版アプリケーションへの影響

A 版統合モードは、A 版で作成したアプリケーションで A 版以降の追加機能を利用できるメリットがありますが、以下の動作に影響が出ます。この影響を避けるために、あらかじめ A 版統合モード用に一部コード修正してください。

(1) Gマクロ上の影響

A版統合モードでは表 2-9のGマクロ機能で影響があります。

表 2-9 G マクロで影響が考えられる動作

No.	機能大分類	機能中分類	概 要
1	色 ID	色 ID の 256 が使用不可	色 ID を線色、背景色、前景色に分割したため、256 番目のパレット指定が使用不可能になっています。 対策としてはパレット 256 を 1~255 の範囲のパレットに割り当てなおしてください。
2	グループ	グループ処理が非推奨へ	グループとレイヤは、マルチスレッド動作時は同時使用を推奨しません。A 版統合モードは V5 版標準と同じくマルチスレッドで動作するため、グループ処理は V5 標準モードと同じく非推奨となります。 対策としては、レイヤとグループの同時使用を止めて、レイヤ処理を中心に置き換えてください。
3	図形全般	V5 版図形属性の影響	V5 版の属性設定 API と同時に使用した場合、A 版の API でサポートしていない図形属性を書き換えません。これにより A 版でなかった V5 版の表示が混在するなど、A 版としての表示に差異が出ます。 対策としては、描画属性に関しては A 版の属性設定 API (g_attrrib、g_mode) か V5 版の属性設定 API (g_draw_attrrib、g_obj_attrrib) のどちらかを選択して、使用してください。 例) g_draw_attrrib で塗り分けの描画属性を指定すると、g_attrrib で属性を指定しても、V5 版標準の塗り分け機能が動作するなど。
4	PreSerV 全般	マルチスレッド関連	A 版統合モードでは、描画関連の処理をマルチスレッドで実行します。したがって g_draw_obj の直後に描画対象図形を削除するなど、シングルスレッドの A 版動作を予定したアプリケーションに動作不具合などの影響が出る可能性があります。 対策の例としては、描画処理直後の描画リソースに関する処理は、一度” g_redraw -r ”を使用して、再描画設定を解除してから実行してください。

(2) Wマクロ上の影響

A版統合モードでは表 2-10のWマクロ機能で影響があります。

表 2-10 W マクロで影響が考えられる動作

No.	機能大分類	機能中分類	概 要
1	部品の外観	一部部品の外観変更	一部の部品を立体的に描画します。 陰影部の描画仕様の変更により、部品レイアウトに1ドットのレイアウトのずれがでることがあります。

2.5 Tcl インタプリタについて

Tcl 言語は、スクリプト言語の 1 つで Tool Command Language の略称です。

通常 Tk (Tool Kit) と称する GUI ツールキットとセット^{注 4)}です。

注 4) セットで Tcl/Tk と表記します。

PreSerV では、A 版開発時に、マルチプラットフォーム実現のためのプログラム言語として、Tcl インタプリタ部分を製品に取り込みました。また、Tk 相当の GUI コマンド部分を PreSerV の W マクロに置き換えました。

本節ではこの Tcl インタプリタ部分の仕様に関して、項目別に説明します。

2.5.1 Tcl の言語仕様

(1) Tcl 言語の特徴

Tcl 言語は基本的にインタプリタ内蔵のコマンドと文字列型ベースの変数から命令文が構成されるシンプルなスクリプト言語です。

リスト 2-2 命令文使用例 (変数 a に "1" を代入し、次行の命令の引数に使用した例)

```
set a 1
expr $a + 2
```

この命令文の実行の繰り返しで、1 つのアプリケーションを構築します。またシンプルな構造のため C 言語で命令を拡張しやすいという特徴があります。インタプリタ内蔵のコマンドは、「Tcl コマンドリファレンス」を参照してください。

PreSerV では C 言語で命令を拡張しやすい特徴を利用して、GUI 構築や GIS 用の各種グラフィックス操作機能を Tcl コマンドとして拡張しています。

(2) Tcl における数字の扱いについて

Tcl 言語では変数は文字列ですが、数字や一部のアルファベットを数値として評価する場合があります。

Tcl を使用する上で誤りやすい、数字の扱いについて説明します。

文字列が数字の場合、8 進数または 16 進数 → 実数の順に評価します。

最初の文字が '0' で、2 番目の文字が 'x' または 'X' のどちらでもない、文字列は 8 進整数と解釈します。

最初の文字が '0' で、2 番目の文字が 'x' または 'X' であると、文字列は 16 進整数と解釈します。

最初の文字が '0' でないと、10 進数と解釈します。

リスト 2-3 Tcl の数字の扱い

```
set a 8

    set b 0010
        if { $a == $b } {
            #OK なパターン
        } else {
            #NG なパターン
        }
```

b の 0010 は 8 進数とみなし、式の評価結果はコメント” OK なパターン” 側のステートメントを実行します。

(3) Tcl のエラー処理に関して

Tcl のインタプリタコマンドおよび、PreSerV のマクロの実行でエラー (TCL_ERROR をコマンドが返す) が発生した場合、Tcl インタプリタ内部の例外処理が発生し、スクリプトの実行を中断します。

エラーが発生してもスクリプトの実行を継続したい場合は、Tcl のインタプリタコマンド catch コマンドを使用して、リスト 2-4 の例のようにエラーに対するトラップを実装し、対応してください。

リスト 2-4 エラーをcatchする実装例^{※3}

```
set hostname "localhost"

catch { set hostname $user_set_host }
```

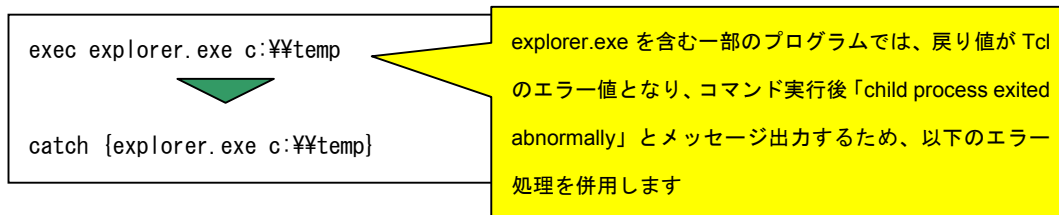
※3 \$user_set_host に値が設定されていない場合のエラーをトラップし、その場合は\$hostname にデフォルトの”localhost”を設定する実装例。

(4) 外部プログラムの実行

Tcl インタプリタの `exec` コマンドを使用することで、Windows アプリケーションを起動することができます（16bit アプリケーションは実行不可）。

以下はエクスプローラを特定のディレクトリを開くように起動したコマンド入力例です。※4

リスト 2-5 explorer.exe を実行する例



※4 このコマンドを実行するとエクスプローラは“C:¥¥temp”を開いた状態で起動します。

3 データ仕様

WebTcl 版で使用するファイルのデータ仕様を以下に示します。

3.1 ベクトルシンボル定義ファイル

ベクトルシンボル定義ファイルは、PreSerVのベクトルシンボルオブジェクトを構成する図形要素を定義するデータファイルです。通常このファイルは「PreSerV基本キット」に添付する「シンボル定義ツール」で作成・管理するため、データフォーマットをユーザが意識する必要はありません。^{注5)}

注5) シンボルファイルのデータ型はすべてビッグエンディアンです。アプリケーションでシンボル定義ファイルの入出力を行う際は、ビッグエンディアン⇄リトルエンディアンの変換を行ってください。

(1) ファイル構成

ベクトルシンボル定義ファイルは図 3-1のように、「管理レコード」と「シンボルデータレコード」から構成されます。

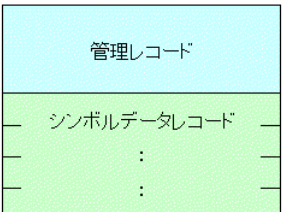


図 3-1 ベクトルシンボル定義ファイル構成

(2) 管理レコード構成

管理レコードはベクトルシンボル定義ファイルに登録しているシンボルの定義数と、各シンボルのシンボルNo、シンボルデータ開始アドレス、シンボルデータサイズがセットとなったシンボル単位のデータ管理情報で構成されます(図 3-2参照)。

各データ項目はすべてint型のデータです。^{注6)}

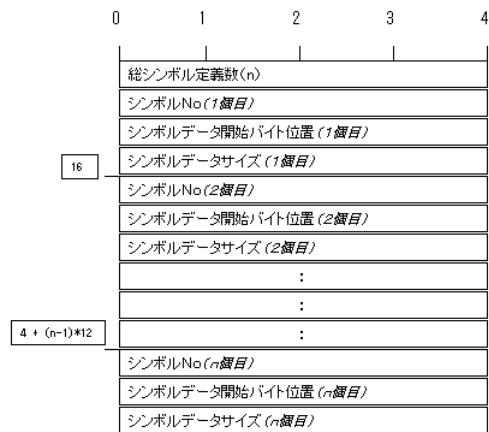


図 3-2 管理レコード詳細構造

注 6) 総シンボル定義数・シンボルNo の値は 1～99999 の範囲です。

(3) シンボルデータレコード構成

シンボルデータレコードは各シンボルのヘッダー情報を格納する個別管理レコードと、シンボルを構成する図形要素の情報を格納する図形要素レコードから構成されます（図 3-3参照）。

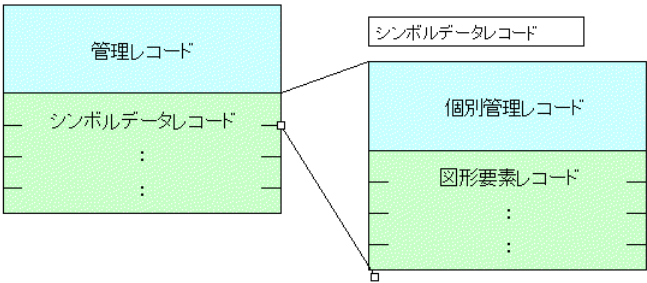


図 3-3 シンボルデータレコード構成

(4) 個別管理レコード

個別管理レコードには、「シンボル定義ツール」で埋め込まれるグリッドの情報や、外接長方形の矩形座標がローカル座標系を定義します（図 3-4参照）。各データ項目の詳細情報を表 3-1に示します。

0	1	2	3	4
シンボルNo.				
シンボルデータの図形要素数				
グリッドの縦の本数				
グリッドの横の本数				
グリッド間隔(WC系座標)				
シンボル基準点X座標(WC系座標)				
シンボル基準点Y座標(WC系座標)				
外接長方形領域の左下X座標(WC系座標)				
外接長方形領域の左下Y座標(WC系座標)				
外接長方形領域の右上X座標(WC系座標)				
外接長方形領域の右上Y座標(WC系座標)				

図 3-4 個別管理レコード詳細構造

表 3-1 個別管理レコードデータ項目一覧

データ項目名称	データ形式	データ定義内容
シンボル No.	int 型	定義対象のシンボルデータのシンボル No. を指定します。 指定する値の範囲は 1～99999 の範囲です。
シンボルデータの図形要素数	int 型	シンボルデータを構成する図形要素の数を指定します。 この値の数だけ、後述する図形要素レコードを定義します。
グリッドの縦の本数	int 型	「シンボル定義ツール」で表示する時の縦グリッドの本数を指定します。 グリッドの縦の本数＝外接長方形領域の高さ／グリッド間隔の関係が成立するように値を設定してください。
グリッドの横の本数	int 型	「シンボル定義ツール」で表示する際の横グリッドの本数を指定します。 グリッドの横の本数＝外接長方形領域の幅／グリッド間隔の関係が成立するように値を設定してください。
グリッドの間隔	int 型	グリッドの間隔を指定します。正の値で、外接長方形領域の高さ・幅より小さい値を設定してください。
シンボル基準点 X 座標	int 型	シンボルデータの座標系内で基準点の X 座標を指定します。値は外接長方形領域の範囲内に設定してください。
シンボル基準点 Y 座標	int 型	シンボルデータの座標系内で基準点の Y 座標を指定します。値は外接長方形領域の範囲内に設定してください。
外接長方形領域の 左下 X 座標	int 型	シンボルデータの図形要素全体を囲む外接長方形領域の左下 X 座標を指定します。
外接長方形領域の 左下 Y 座標	int 型	シンボルデータの図形要素全体を囲む外接長方形領域の左下 Y 座標を指定します。
外接長方形領域の 右上 X 座標	int 型	シンボルデータの図形要素全体を囲む外接長方形領域の右上 X 座標を指定します。
外接長方形領域の 右上 Y 座標	int 型	シンボルデータの図形要素全体を囲む外接長方形領域の右上 Y 座標を指定します。

(5) 図形要素レコード

図形要素レコードは、7 種類の図形種別の情報を格納します。7 種類の図形種別とは、折れ線、多角形、円、円弧、扇形、楕円、文字です。

この 7 種類の図形は 4 種類のデータ構造に分類されます（折線・多角形のようにデータ構造が同じものがあるため）。以下にデータ構造別に図形要素レコードの詳細構造を説明します。

① 折線・多角形図形要素レコードデータ構造

折線・多角形共通の図形要素レコードを図 3-5に示します。データ項目別の情報を表 3-2に示します。

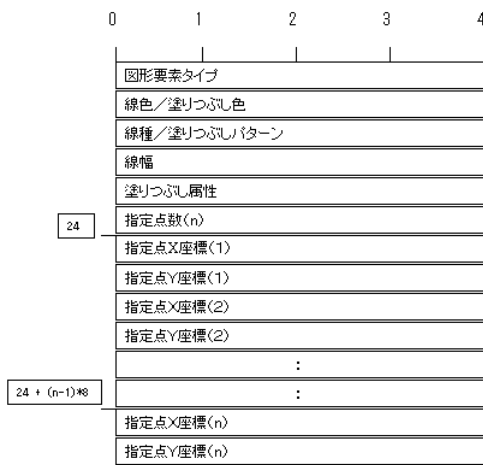


図 3-5 折線・多角形図形要素レコードデータ構造

表 3-2 折線・多角形図形要素レコードデータ項目一覧

データ項目名称	データ形式	データ定義内容
図形要素タイプ	int 型	図形要素のタイプを指定します。 折線 : 1、多角形 : 2
線色／塗りつぶし色	int 型	-1 を指定したときはシンボルオブジェクト全体の色で描画します。 カラーシンボルを作成する時は、1～255 の値を指定してください。
線種／塗りつぶしパターン	int 型	塗りつぶし無し時は線色の色 ID を、塗りつぶし有りの時は塗りつぶし色を指定します。
線幅	int 型	線幅を指定します。線幅は 0～15 の値で指定してください。
塗りつぶし属性	int 型	塗りつぶし属性を指定します。 塗りつぶし無し : 0、塗りつぶし有り : 1
指定点数	int 型	図形を構成する点座標の数を指定します。 このデータの後、指定点 X 座標・指定点 Y 座標のセットを指定点数個指定します。指定点数は 2～50 の範囲で指定してください。
指定点 X 座標	int 型	指定点の X 座標を指定します。 多角形指定時は始点と終点を同じ座標にする必要はありません。
指定点 Y 座標	int 型	指定点の Y 座標を指定します。 多角形指定時は始点と終点を同じ座標にする必要はありません。

② 円・円弧・扇形図形要素レコードデータ構造

円・円弧・扇形共通の図形要素レコードを図 3-6に示します。データ項目別の情報を表 3-3に示します。

0	1	2	3	4
図形要素タイプ				
線色／塗りつぶし色				
線種／塗りつぶしパターン				
線幅				
塗りつぶし属性				
中心点X				
中心点Y				
半径				
開始角				
終了角				

図 3-6 円・円弧・扇形図形要素レコードデータ構造

表 3-3 円・円弧・扇形図形要素レコードデータ項目一覧

データ項目名称	データ形式	データ定義内容
図形要素タイプ	int 型	図形要素のタイプを指定します。 円 : 3、円弧 : 4、扇形 : 5
線色／塗りつぶし色	int 型	-1 を指定したときはシンボルオブジェクト全体の色で描画します。カラーシンボルを作成する時は、1～255 の値を指定してください。
線種／塗りつぶしパターン	int 型	塗りつぶし無し時は線色の色 ID を、塗りつぶし有りの時は塗りつぶし色を指定します。
線幅	int 型	線幅を指定します。線幅は 0～15 の値で指定してください。
塗りつぶし属性	int 型	塗りつぶし属性を指定します。 塗りつぶし無し : 0、塗りつぶし有り : 1
中心点 X	int 型	円周の中心点の X 座標を指定します。
中心点 Y	int 型	円周の中心点の Y 座標を指定します。
半径	int 型	円周の半径を指定します。
開始角	int 型	円弧・扇形の円周の開始位置を 1/10 度単位の角度で指定します。 円の時は 0 を指定してください。
終了角	int 型	円弧・扇形の円周の終了位置を 1/10 度単位の角度で指定します。 円の時は 0 を指定してください。

③ 楕円図形要素レコードデータ構造

楕円の図形要素レコードを図 3-7に示します。データ項目別の情報を表 3-4に示します。

0	1	2	3	4
図形要素タイプ				
線色／塗りつぶし色				
線種／塗りつぶしパターン				
線幅				
塗りつぶし属性				
中心点X				
中心点Y				
X軸方向半径				
Y軸方向半径				

図 3-7 楕円図形要素レコードデータ構造

表 3-4 楕円図形要素レコードデータ項目一覧

データ項目名称	データ形式	データ定義内容
図形要素タイプ	int 型	図形要素の対応を指定します。 楕円 : 6
線色／塗りつぶし色	int 型	-1 を指定したときはシンボルオブジェクト全体の色で描画します。 カラーシンボルを作成する時は、1～255 の値を指定してください。
線種／塗りつぶしパターン	int 型	塗りつぶし無しの時は線色の色 ID を、塗りつぶし有りの時は塗りつぶし色を指定します。
線幅	int 型	線幅を指定します。線幅は 0～15 の値で指定してください。
塗りつぶし属性	int 型	塗りつぶし属性を指定します。 塗りつぶし無し : 0、塗りつぶし有り : 1
中心点 X	int 型	円周の中心点の X 座標を指定します。
中心点 Y	int 型	円周の中心点の Y 座標を指定します。
X 軸方向半径	int 型	楕円の X 軸方向の半径を指定します。
Y 軸方向半径	int 型	楕円の Y 軸方向の半径を指定します。

④ 文字列図形要素レコードデータ構造

文字列の図形要素レコードを図 3-8に示します。データ項目別の情報を表 3-5に示します。

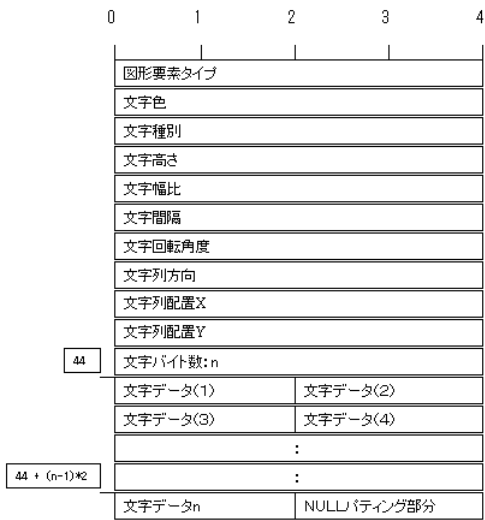


図 3-8 文字列図形要素レコードデータ構造

表 3-5 文字列図形要素レコードデータ項目一覧

データ項目名称	データ形式	データ定義内容
図形要素タイプ	int 型	図形要素の対応を指定します。 文字列：7
文字列色	int 型	-1 を指定したときはシンボルオブジェクト全体の色で描画します。 カラーシンボルを作成する時は、1～255 の値を指定してください。
文字列種別	int 型	現在固定値 0（ベクトルフォント）を指定します。
文字列高さ	int 型	文字列の高さを指定します。
文字幅比	float 型	文字幅と文字高さの比率を指定します。 半角幅にしたい場合は 0.5 を指定します。
文字回転角度	int 型	文字列の回転角度を 1/10 度単位で指定します。
文字列方向	int 型	文字列の描画方向を指定します。 横書き：0、縦書き：1
文字列配置 X	int 型	文字列を配置する基準点の X 座標を指定します。
文字列配置 Y	int 型	文字列を配置する基準点の Y 座標を指定します。
文字バイト数	int 型	文字データのバイト数を指定します。指定する値は 4 の倍数 で指定し、最大 32 バイトまで指定可能です。
文字データ	全角 EUC コード	文字データを指定します。文字データは全角 EUC コードで指定し、4 バイトバウンダリになるように NULL パディングして全体のバイト数が 4 の倍数になるように調整します。

3.2 ビットマップシンボル定義ファイル

ビットマップシンボル定義ファイルは、PreSerV のビットマップシンボルオブジェクトで表示するビットマップファイルのファイル名と、ビットマップシンボルのシンボル番号を関連付ける定義ファイルです。

通常このファイルは「PreSerV 基本キット」に添付する、「ビットマップシンボル定義ファイルの作成ツール」で作成します。

下図にビットマップシンボル定義ファイルのデータ構造を図 3-9に示します。データ項目別の情報を表 3-6に示します。

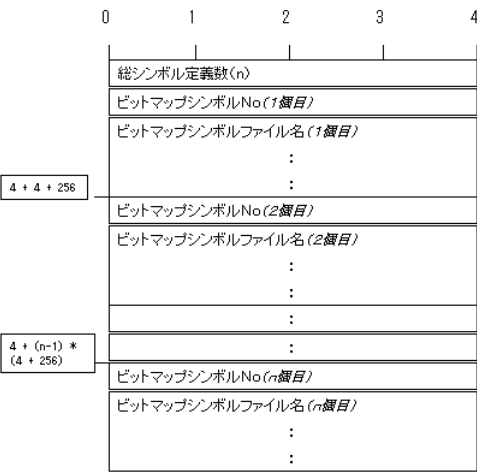


図 3-9 ビットマップシンボル定義ファイルデータ構造

表 3-6 ビットマップシンボル定義ファイルデータ項目一覧

データ項目名称	データ形式	データ定義内容
総シンボル定義数	int 型	ファイル内で定義するビットマップシンボルの定義数を指定します。 このデータ定義以下に、ビットマップシンボル No・ビットマップシンボルファイル名のセットを総シンボル定義数個分定義します。
ビットマップシンボル No	int 型	ビットマップシンボルのシンボル No を指定します。
ビットマップシンボル ファイル名	int 型	ビットマップシンボルで参照するビットマップファイル名を指定します。 ビットマップシンボル作成時に、ここで指定したパス名に Windows ビットマップ形式のファイルが存在するように定義ファイルを生成してください。

4 エラーメッセージ及びデバッグツール

4.1 Tcl インタプリタ独自のエラーメッセージ

Tcl のインタプリタ独自のエラーメッセージを表 4-1 に示します。

表 4-1 Tcl インタプリタ・エラーメッセージ一覧

エラーメッセージ	概要	備考
malformed expression word	Tcl では、“}”の後は半角スペースを記述する必要があります。本メッセージは“}”の後に“{”を続けて記述した場合、また間違えて全角スペースを記述した時に出力します。	エラー : <pre>if {\$i==0} {return \$i}</pre> 正常 : <pre>if {\$i==0} {return \$i}</pre>
invalid command name	マクロ名に相当する Tcl インタプリタコマンド、PreServ マクロ、ユーザ定義のプロシジャが存在しない場合、このメッセージを出力します。 ただし、unknown ハンドラを定義している場合は、このエラーにならず unknown ハンドラを実行します。	エラー : <pre>foo</pre> 正常 : <pre>pwd</pre>
missing close-brace	このエラーは中括弧 (“{”、“}”) の対応が取れていない場合に発生するエラーです。 実行スクリプト中の中括弧の対応を確認してください。	エラー : <pre>if {\$i == 1} {return \$i}}</pre> 正常 : <pre>if {\$i == 1} {return \$i}</pre>
syntax error in expression	このエラーは expr で使用する数式の文法エラーが発生した場合に発生するエラーです。 expr 内の変数名に“\$”を付け忘れているケースや、文字列を囲む“””の対応が間違っているケース、数値が入る変数部分に数値以外の値が入っているケースなど expr 内の文法を確認してください。	エラー : <pre>set cval "test_str1" set result [expr \$cval+1]</pre>

4.2 PreSerV の Tcl/IF 独自のエラーメッセージ

PreSerVのTcl/IF独自のエラーメッセージを表 4-2に示します。

表 4-2 Tcl/IF 独自のエラーメッセージ一覧

エラーメッセージ	概要	備考
invalid parent	引数で指定した、親ウィンドウ ID が不正な場合このメッセージを出力します。 親ウィンドウ ID を確認してください。	親ウィンドウを必要とするウィンドウ部品作成マクロで出力します。
invalid object	処理対象として引数で指定した、対象ウィンドウの ID が不正な場合このメッセージを出力します。 対象ウィンドウの ID を確認してください。	引数に各種オブジェクトの ID 指定があるマクロで出力します。
invalid parameter	引数の値が不正な値である場合、このメッセージを出力します。 対象のコマンドの API リファレンスを参考に、マクロ引数の指定値を確認してください。	主に引数の値指定部分の省略時に出力します。
invalid argument	引数の指定形式が不正な場合、このメッセージを出力します。 対象のコマンドの API リファレンスを参考にマクロのコール形式を確認してください。	主に使用不可能なオプション文字列を指定した場合に出力します。
invalid gwinID	グラフィックスウィンドウの引数値が不正な値である場合、このメッセージを出力します。 また、引数省略時に、カレントのグラフィックスウィンドウ値が不正な値である場合もこのエラーになります。 グラフィックスウィンドウの引数値を確認してください。	グラフィックスウィンドウ操作関連のマクロと、一時描画命令のマクロで出力します。
invalid dlistID	ディスプレイリストメモリの引数値が不正な値である場合、このメッセージを出力します。 また、引数省略時に、カレントのディスプレイリストメモリ値が不正な値である場合もこのエラーとなります。 ディスプレイリストメモリの引数値を確認してください。	ディスプレイリスト関連マクロと、図形オブジェクト操作関連のマクロで出力します。

4 章エラーメッセージ及びデバッグツール

エラーメッセージ	概要	備考
invalid groupID	グループオブジェクトの引数値が不正な値である場合、このメッセージを出力します。 また、引数省略時に、カレントのグループオブジェクト値が不正な値である場合もこのエラーとなります。 グループオブジェクトの引数値を確認してください。	グループオブジェクトに関連するマクロで出力します。
invalid layerID	レイヤオブジェクトの引数値が不正な値である場合、このメッセージを出力します。 また、引数省略時に、カレントのレイヤオブジェクト値が不正な値である場合もこのエラーとなります。 レイヤオブジェクトの引数値を確認してください。	レイヤオブジェクトに関連するマクロで出力します。
invalid filterID	フィルターオブジェクトの引数値が不正な値である場合、このメッセージを出力します。 また、引数省略時に、カレントのフィルターオブジェクト値が不正な値である場合もこのエラーとなります。 フィルターオブジェクトの引数値を確認してください。	フィルターオブジェクトに関連するマクロで出力します。
invalid objID	図形オブジェクトの引数値が不正な値である場合、このメッセージを出力します。 また、引数省略時に、カレントの図形オブジェクト値が不正な値である場合もこのエラーとなります。 図形オブジェクトの引数値を確認してください。	図形オブジェクトの操作関連のマクロで出力します。
scale is zero	ラインシンボルのスケール値で 0 を指定した場合、このエラーとなります。 引数 scale の値を確認してください。	g_linesymbol マクロで出力します。
no such symbol	g_symbol で、存在しないシンボル No を指定した場合、このエラーとなります。 引数 symNo の値と、g_symbol_file で設定したシンボル定義ファイルの内容を確認してください。	g_symbol マクロで出力します。
usage	マクロ引数の形式が誤っている場合、このエラーとなります。 必須指定の引数の指定を省略した場合に出力します。 マニュアルを参考にマクロのコーリングシーケンスを再度確認してください。	必須指定引数があるマクロ全般で出力します。

4 章エラーメッセージ及びデバッグツール

エラーメッセージ	概要	備考
tooltip initialize error	ツールチップの初期化に失敗しました。 PC上の comctl32.dll のバージョンを確認してください。 ツールチップ機能は、comctl32.dll のバージョン 4.71 以上の環境で動作確認しています。	w_tooltip_init マクロで出力し ます。
invalid tip text	ツールチップテキストのテキスト文字列の指定が不正で す。	w_tooltip_set マクロで出力しま す。
tooltip set error happened	ツールチップテキストの設定処理でエラーが発生しまし た。 対象ウィンドウの種別や、初期化の有無を確認してくだ さい。	w_tooltip_set マクロで出力しま す。
tooltip color error happened	ツールチップテキストの色指定でエラーが発生しまし た。 対象ウィンドウの種別や、初期化の有無を確認してくだ さい。	w_tooltip_color マクロで出力し ました。
tooltip enable error happened	ツールチップのアクティブ化、非アクティブ化でエラー が発生しました。 対象ウィンドウの種別や、初期化の有無を確認してくだ さい。	w_tooltip_enable マクロで出力

4.3 デバッグツールによるエラー情報の取得

(1) デバッグツールの使い方

Tcl インタプリタの\$errorInfo 変数には最後に発生したエラーの情報を格納します。

この変数を参照し、迅速に WebTcl アプリケーションの動作の問題箇所を確認するには、デバッグツールのデバッグコンソールを使用します。デバッグコンソールはプラグイン形式で提供しました。今後、徐々に機能強化を行う予定です。本書では初版のデバッグコンソールの使用方法を記述します。

初期起動マクロにデバッグツールのプラグイン宣言およびデバッグコンソールの実行を追加してください。

例) `e_plugin "debugTools.dll" debugConsole`
`debugConsole`

デバッグコンソールが実行されるとタスクトレイにアイコンを表示します (図 4-1 参照)。



図 4-1 デバッグコンソールの初期表示

① コンソールの表示

タスクトレイのアイコンをクリックすることで、デバッグコンソール本体を表示します (図 4-2 参照)。

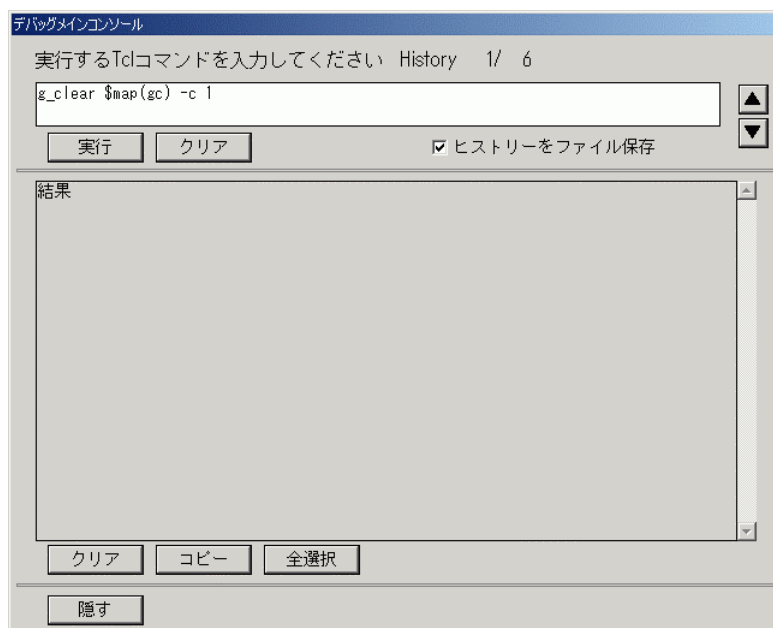


図 4-2 デバッグコンソールウィンドウ表示イメージ

② Tcl コマンドの実行

エラー発生後、「コマンド入力欄」に、以下のコマンド入力し、「実行」をクリックします。

set errorInfo

\$errorInfo変数の内容を「結果」のウィンドウに追加形式で表示します（図 4-3参照）^{注 7)}。

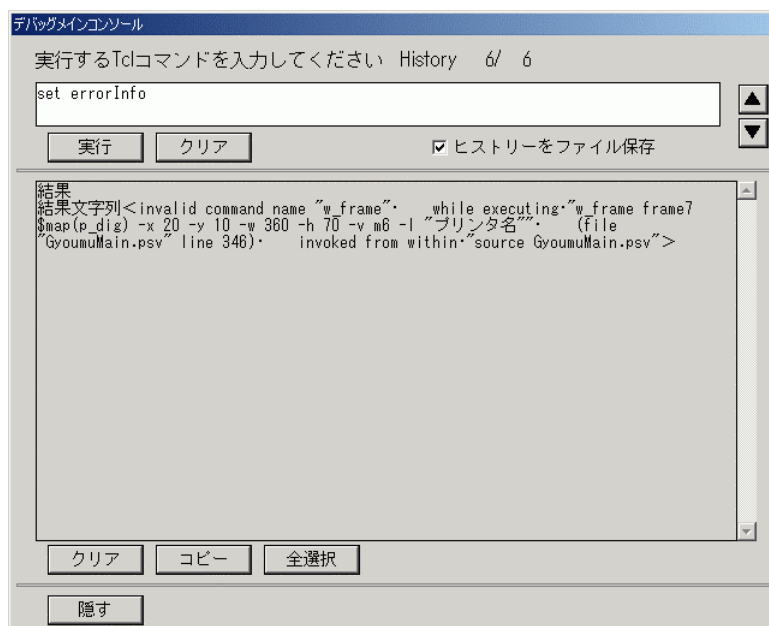


図 4-3 A 版互換モードで V5 版標準モード用のマクロを実行した場合のエラー例

注 7) デバッグコンソールは WebTcl エンジン側のインタプリタのストリームを直接リダイレクトして実現した機能です。このため、従来の psvlogin.exe のようにクライアントとして動作するツールとは結果取得の方法が違います。また、実行結果を取得する時に通信を伴う従来の“e_send \$errorInfo”の形式は実行できません。

③ エラー結果を取り出し

エラー結果をマウスでドラッグして選択・反転し、「コピー」ボタンか、「CTRL+C」でクリップボードにコピー^{注 8)}してください。

注 8) 保守または開発者サポートを通じてお問い合わせいただく場合は、この情報をワードパッド等にコピーしてテキスト形式で保存し、添付をお願いします。

(2) エラー情報の見方

取得可能なエラー情報は以下のとおりです。

エラー情報には、エラー原因の箇所から上位のプロシジャまでのスタックが下位から順に表示されます。

後述の例の場合は、`sym_create_menu` → `FS_CreateFileSelection` の順にコールされるプロシジャ内で、`"}`の後の半角スペースがないエラーを発生させた場合のエラーメッセージです。

エラーメッセージには、メッセージの意味別に色をつけて例示しています。

赤字はエラーメッセージ、紫字は処理状態、青字は対象行、緑字はエラー部分の位置を示しています。

リスト 4-1 エラーメッセージ出力例

```
malformed expression word
("if" test expression)
while compiling
"if [ [FS_CheckID $id] = 0 ] { ..."
(compiling body of proc "FS_CreateFileSelection", line 8)
invoked from within
"FS_CreateFileSelection $sym(top) $fg $bg $col3 $col2 $col2"
(procedure "sym_create_menu" line 311)
invoked from within
"sym_create_menu "
```

このメッセージからエラーを解析する場合、

- `FS_CreateFileSelection` プロシジャの定義位置を探す。
- そこから 8 行目にある行がエラー対象行となります。

この行の命令を赤字のエラーメッセージ部分から、エラーメッセージ表を探し、デバッグ作業に役立ててください。

5 サンプルに関して

製品に同封するサンプルファイルに関して説明します。

サンプルファイルはマニュアル中のサンプルコードと重複^{注 8)}していますが、単独で動作可能なため、動かしながらマクロの使い方を覚えてください。

注 8) 『PreSerV V5 基本パック』と同一ファイル名のサンプルは e_send 部分を削除したファイルを掲載しています。

5.1 サンプルの動かし方

sample は『PreSerV WebTcl 基本パック』インストールディレクトリ下の“sample”ディレクトリ下にファイルを格納しています。

動作するためには、“sample”ディレクトリごと Web サーバ上に配置してください。

後述の 3 種類のサンプルは起動 HTML から実行します（図 5-1 参照）。

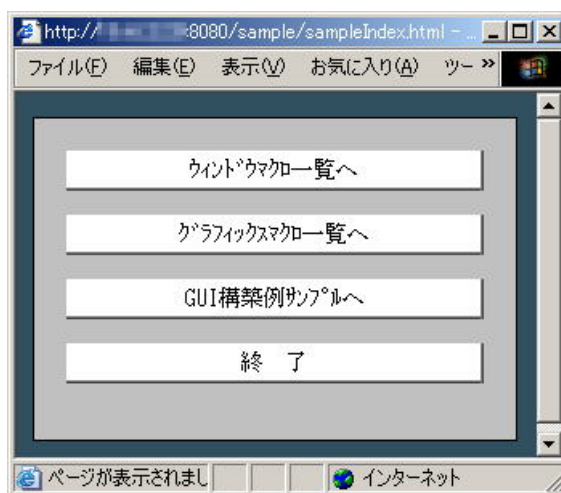


図 5-1 サンプルの起動 HTML 表示

起動 HTML には Web サーバ名とポート番号の記述部分がありますので、その部分を実行環境に合わせて修正してください。

ウィンドウマクロとグラフィックマクロは、サンプルを続けて起動した場合、変数の重複により前回に起動したサンプルへの操作ができなくなります。サンプル画面を閉じることもできなくなりますが、IE を閉じると、サンプル画面も一緒に閉じます。

5.2 Wマクロサンプル

起動 HTML 中の「ウィンドウマクロ一覧へ」をクリックすると Wマクロのサンプル一覧を表示します。各サンプルファイルの概要を「Wマクロサンプル一覧」に示します（図 5-2表 5-1 参照）。

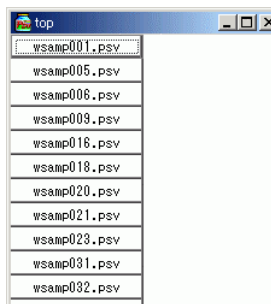


図 5-2 windex.psv を実行した画面

表 5-1 W マクロサンプル一覧

サンプルファイル	概 要	主に使用している W マクロ命令
wsamp001.psv	アームボタンを使用したイベント処理の連続実行のサンプルです。絵の部分はビットマップのラベルを使用しています。	w_arm_button w_combo_box w_label
wsamp005.psv	チェックボックスの作成例です。	w_check_box
wsamp006.psv	各種部品のクラス名、タグ名、親ウィンドウ ID を取得する例です。リストから部品名称を選択すると、情報を取得して表示します。	w_class w_tag w_parent
wsamp009.psv	コンボボックスの各種制御マクロを使用したサンプルです。	w_combo_box w_combo_add w_combo_count w_combo_del w_combo_item w_combo_str
wsamp016.psv	ダイアログのモーダル、モードレスの違いを示したサンプルです。	w_dialog
wsamp018.psv	W マクロ部品のフォーカス設定のサンプルです。	w_focus
wsamp020.psv	W マクロ部品から設定値の取得、設定を行うサンプルです。	w_get_value w_set_value
wsamp021.psv	ラベルのフォント指定と表示位置の設定でどのように表示するかを示したサンプルです。 フォントと表示位置の指定にはラジオボタンを使用しています。	w_label w_radio_button

5 章サンプルに関して

サンプルファイル	概 要	主に使用している W マクロ命令
wsamp023. psv	リストボックスの各種制御マクロを使用したサンプルです。 サンプルでは、左側のリストから右側のリストへ項目を移すサンプルです。	w_list_add w_list_box w_list_count w_list_del w_list_deselection w_list_item w_list_selection w_list_selection_item w_list_set
wsamp031. psv	各種部品のマップ(表示)とアンマップ(非表示)の動作を確認するサンプルです。	w_map w_unmap
wsamp032. psv	メニューバーのメニューを作成するサンプルです。 このサンプルでは一般的な Windows アプリケーションのメニューバー形式に合わせて作成しています。	w_menu_bar w_cascade w_pull_down_menu
wsamp033. psv	メッセージボックスの表示形式を確認するサンプルです。	w_message_box
wsamp034. psv	ウィンドウの移動を確認するサンプルです。	w_move
wsamp037. psv	ポップアップメニューを作成するサンプルです。 画面上でマウス右クリックを行うとポップアップメニューを表示します。	w_popup w_popup_menu w_event_handler w_get_event
wsamp042. psv	ウィンドウのサイズ変更を確認するサンプルです。	w_resize
wsamp043. psv	ディスプレイの表示解像度に合わせて、ウィンドウ部品の表示サイズを可変にするサンプルです。	w_root_size w_def_size
wsamp044. psv	ボタンのセンシティブを切り替えた状態を確認するサンプルです。	w_sense
wsamp045. psv	セパレータを使用して表を描画したサンプルです。	w_separator
wsamp047. psv	スライダーを使用したサンプルです。 スライダーの値により RGB 値が変化し、背景色に適用します。	w_slider

5 章サンプルに関して

サンプルファイル	概 要	主に使用している W マクロ命令
wsamp049. psv	テキストボックスの各種制御マクロを使用したサンプルです。 サンプルはエディタ用のウィンドウを作成します。	w_text_box w_text_str w_text_select w_text_ins_pos w_text_replace w_text_clr_select
wsamp055. psv	トップウィンドウの 3 種類の生成状態を確認するサンプルです。	w_top
wsamp056. psv	DC 座標 (相対座標) と、WC 座標 (絶対座標) の差を確認するサンプルです。 相対座標と絶対座標では、Y 軸の正の方向が逆になっていることが確認できます。	w_trans_coords
wsamp057. psv	カーソルの変更を確認するサンプルです。 ラジオボタンとラベルの連係動作も行っています。	w_cursor w_label w_radio_button
wsamp060. psv	イベントハンドラをカスタマイズしてツールチップ的な表示を行ったサンプルです。 注) 現在の版では、ツールチップを標準でサポートしています。	w_grab w_event_handler
wsamp062. psv	マウスイベントの情報を取得するサンプルです。黒地のキャンバス上でマウス操作を行うと、取得した情報を表示します。 イベントの情報を取得することで、イベントハンドラの高度なカスタマイズも可能です。	w_event_handler w_get_event
wsamp100. psv	フレームを利用したプリントダイアログ用のウィンドウを作成するサンプルです。 本サンプル中では、変数未定義のエラーハンドリングを利用して、画面の重複起動をチェックしています。	w_frame
wsamp101. psv	ツールチップの各種制御マクロを使用したサンプルです。	w_tooltip_init w_tooltip_set w_tooltip_color w_tooltip_enable
wsamp103. psv	ビットマップを利用したボタンを作成するサンプルです。サンプルでは、ボタンのラベルを別定義にして変更を簡易にしています。	w_button

5.3 Gマクロサンプル

起動HTML中の「グラフィックスマクロ一覧へ」をクリックするとGマクロのサンプル一覧を表示します。各サンプルファイルの概要を「Gマクロサンプル一覧」に示します（表 5-2参照）。

表 5-2 G マクロサンプル一覧

サンプルファイル	概 要	主に使用している G マクロ命令
gsamp001. psv	色 ID の操作サンプルです。RGB 値を入力すると画面に表示するテキストの表示色に反映します。	g_alloc_color
gsamp002. psv	円弧オブジェクトの作成サンプルです。再描画は g_draw_obj をコールして行う形式です。	g_arc
gsamp003. psv	検索と属性取得の組合せサンプルです。w_event_handler によるマウス移動検知を検索のトリガーとしています。	g_attrib g_search_obj g_draw_obj
gsamp004. psv	円オブジェクトの作成サンプルです。再描画は g_draw_obj をコールして行う形式です。	g_circle
gsamp014. psv	WC座標とDC座標の座標投影パラメータを再計算するサンプルです。スクロールバーをスライドするとキャンバスのサイズを変更して、図形描画時の座標投影パラメータ（内部的なもの）を変更するため、図形のサイズも変化します。	g_compute_size
gsamp023. psv	画面スクロールのサンプルです。青十字は DC 座標を使用した一時描画で、スクロールする赤十字とは同期せず表示位置が固定です。	g_scroll
gsamp024. psv	ディスプレイリストに投入した円を削除していくサンプルです。マウス左クリックにより近くの図形を検索して、削除していきます。	g_del_obj
gsamp026. psv	円弧一時描画のサンプルです。gsamp002. psv と違って、図形オブジェクトを作成せず、一時的に画面に描画を行います。	g_draw_arc
gsamp027. psv	円一時描画のサンプルです。gsamp004. psv と違って、図形オブジェクトを作成せず、一時的に画面に描画を行います。	g_draw_circle
gsamp028. psv	楕円一時描画のサンプルです。	g_draw_ellipse
gsamp029. psv	線分一時描画のサンプルです。	g_draw_line
gsamp031. psv	扇形一時描画のサンプルです。	g_draw_pie
gsamp032. psv	ポリゴン一時描画のサンプルです。	g_draw_polygon
gsamp033. psv	ポリライン一時描画のサンプルです。ポリラインは-f を付けるとスプラインとして描画します。	g_draw_polyline

5 章サンプルに関して

サンプルファイル	概 要	主に使用している G マクロ命令
gsamp034. psv	矩形一時描画のサンプルです。矩形という図形オブジェクトは存在しませんが、矩形のポリゴンを書く頻度が高いため、一時描画命令で矩形をサポートしています。	<code>g_draw_rect</code>
gsamp035. psv	正多角形一時描画のサンプルです。	<code>g_draw_regpoly</code>
gsamp036. psv	角丸長方形一時描画のサンプルです。	<code>g_draw_round_box</code>
gsamp037. psv	ベクトルシンボル一時描画のサンプルです。	<code>g_draw_symbol</code>
gsamp038. psv	文字一時描画のサンプルです。	<code>g_draw_text</code>
gsamp039. psv	図形情報のダンプのサンプルです。図形にはレイヤかグループを指定して、マウス移動を検知して属性をダンプする処理を行っています。またカレント値を利用したパラメータ指定省略も行っています。 注) サンプルのようにレイヤとグループは排他的に使用してください。	<code>g_dump_obj</code> <code>g_cur_attr</code> <code>g_cur_color_id</code> <code>g_cur_dlist</code> <code>g_cur_group</code> <code>g_cur_gwin</code> <code>g_cur_layer</code> <code>g_cur_tx_attr</code>
gsamp040. psv	楕円オブジェクト作成のサンプルです。再描画は <code>g_draw_obj</code> をコールして行う形式です。	<code>g_ellipse</code>
gsamp041. psv	gsamp039. psv に図形に色を割り当てる処理を追加したサンプルです。図形作成後図形単位で色換えする場合に使用してください。	<code>g_color_id</code>
gsamp045. psv	ユーザ定義データ取得のサンプルです。マウス移動を検知して、図形に設定したユーザ定義データの文字列を取得します。サンプルでは、キー“type”で各図形の種別を示す文字列を格納しています。	<code>g_get_usrdata</code>
gsamp046. psv	グループ操作のサンプルです。グループ単位で色を設定する処理も行っています。	<code>g_group_attr</code> <code>g_make_group</code> <code>g_group_no</code> <code>g_group_color_id</code>
gsamp048. psv	ユーザ定義データ削除のサンプルです。マウスクリックによる検知を行い、対象を反転後「del_usrdata」ボタンをクリックすると対象図形のユーザ定義データを削除します。	<code>g_del_usrdata</code>
gsamp049. psv	グラフィックスウインドウに属性を設定するサンプルです。グラフィックスウインドウの属性は一時描画命令で属性指定を省略した場合に適用します。	<code>g_gwin_attr</code> <code>g_gwin_color</code> <code>g_gwin_tx_attr</code>

5 章サンプルに関して

サンプルファイル	概 要	主に使用している G マクロ命令
gsamp056. psv	レイヤ操作のサンプルです。レイヤのオブジェクト属性、色 ID をレイヤ単位で設定する操作を行っています。	g_layer_attr g_layer_no g_layer_color_id
gsamp059. psv	線分オブジェクトの作成サンプルです。 再描画は g_draw_obj をコールして行う形式です。	g_line
gsamp061. psv	旧 g_mode 命令を g_draw_attr に置き換えたサンプルです。	g_mode g_draw_attr
gsamp063. psv	フィルターを使用した図形のドラッグ移動のサンプルです。ドラッグ時に図形は xor で描画します。文字オブジェクトはそのままでは描画時に論理演算ができないため、矩形で表現します。TrueType フォントはラスタとしての特性があるため、xor の論理演算が適用できません。このため xor の描画時に文字オブジェクトを矩形で表現する仕様に統一しました。	g_open_filter g_filter_add g_filter_draw_obj g_filter_remove g_filter_set_param g_close_filter g_translate_obj g_raster
gsamp066. psv	ハッシュオブジェクトの操作サンプルです。検索の母集団が大規模な場合や、空間検索以外で図形を検索したい場合は、ハッシュオブジェクトを図形管理に応用することで、対象図形の取り出しが高速に行えます。	g_open_hash g_close_hash g_hash_fetch g_hash_store g_hash_del
gsamp068. psv	グラフィックスウインドウの矩形領域を WC 座標で指定するサンプルです。中心点が固定になるように座標値を計算すると、ズームと同じ動作ができます。	g_ortho
gsamp069. psv	扇形オブジェクトの作成サンプルです。再描画は g_draw_obj をコールして行う形式です。	g_pie
gsamp070. psv	ポリゴンオブジェクトの作成サンプルです。再描画は g_draw_obj をコールして行う形式です。	g_polygon
gsamp071. psv	ポリラインオブジェクトの作成サンプルです。再描画は g_draw_obj をコールして行う形式です。オブジェクトではスプラインにする時はオブジェクト属性のスプラインフラグ (0x200) を立てます。	g_polyline
gsamp074. psv	正多角形オブジェクトの作成サンプルです。再描画は g_draw_obj をコールして行う形式です。	g_regpoly

5 章サンプルに関して

サンプルファイル	概 要	主に使用している G マクロ命令
gsamp075. psv	図形オブジェクトのディスプレイリスト投入解除のサンプルです。 投入解除の場合、参照が残っていると図形自体は削除しませんので、再投入することで元に戻ります。	g_rm_obj
gsamp076. psv	画面全体を回転するサンプルです。	g_rotate
gsamp077. psv	図形単独で回転するサンプルです。	g_rotate_obj
gsamp078. psv	角丸長方形オブジェクトを作成するサンプルです。再描画は g_draw_obj をコールして行う形式です。	g_round_box
gsamp079. psv	フィルターを使用した図形変形サンプルです。	g_open_filter g_filter_add g_filter_draw_obj g_filter_remove g_filter_set_param g_close_filter g_transform_obj g_raster
gsamp084. psv	距離に関するマクロと検索を組み合わせたサンプルです。マウスの移動を検知して図形との距離を算出し、距離順にソートして一番近い図形の距離を表示しています。 検索結果は最近点検索と同じになります。	g_sort g_search_obj g_dist
gsamp085. psv	ベクトルシンボルオブジェクトの作成サンプルです。再描画は g_draw_obj をコールして行う形式です。	g_symbol
gsamp088. psv	文字オブジェクトの作成を行うサンプルです。再描画は g_draw_obj をコールして行う形式です。	g_text
gsamp089. psv	gsamp079. psv の改良バージョンです。対象図形確定時に、変形の基準となる位置に円を描画しています。	g_transform_obj
gsamp090. psv	図形の外接長方形を取得するサンプルです。長方形の座標を取得後、長方形と中心点まで 1/2 の長方形を描画しています。	g_region_obj
gsamp091. psv	描画処理をマクロで定義した、ユーザ定義オブジェクトの作成サンプルです。ユーザ定義オブジェクトは描画範囲だけをもつ空のオブジェクトで、描画処理は Tcl のプロシジャを指定して、プロシジャ内で行います。	g_usr_def

5 章サンプルに関して

サンプルファイル	概 要	主に使用している G マクロ命令
gsamp092. psv	1つのキャンバスに2つのグラフィックスウインドウを作成し、ビューポートで表示領域を分けたサンプルです。チェックボックスの ON/OFF を切り分けると、グラフィックスウインドウが別々にズーム動作を行います。	g_viewport
gsamp093. psv	クリップボードにキャンバスの描画イメージを転送するサンプルです。クリップボードに転送する形式はメタファイル形式であるため、貼り付けるアプリケーションのメディア形式に合った変換を行います。例) ペイントに貼り付けるとイメージ、MS PowerPoint に貼り付けるとベクトル図形に変換されます。	g_set_clipboard
gsamp097. psv	図形オブジェクトのオブジェクト属性取得のサンプルです。マウス移動を検知して、対象図形を選択し、オブジェクト属性を取得します。	g_cur_obj_attrib g_obj_attrib
gsamp098. psv	図形オブジェクトの描画属性取得のサンプルです。ペイントパターンはユーザ定義を行って拡張しています。	g_paint_pattern_file g_draw_attrib g_cur_draw_attrib
gsamp099. psv	ラスターマップにイメージ領域としてイメージファイルを貼り付けたサンプルです。サポートするフォーマットである bmp、TIF、JPEG を表示しています。	g_map_element g_close_image_region g_close_map g_open_map g_open_image_region
gsamp100. psv	ラインシンボルオブジェクトの作成サンプルです。再描画は g_draw_obj をコールして行う形式です。	g_linesymbol
gsamp101. psv	座標変換 (WC 座標→緯度経度) のサンプルです。サンプルでは9系の WC 座標として、緯度経度に変換して表示しています。	g_yoko_mercator
gsamp102. psv	マップオブジェクトの操作に関連したマクロのサンプルです。	g_open_map g_close_map g_map_element g_map_type g_map_attrib g_map_range
gsamp103. psv	スケール分母値を利用して画面のズームを制御するサンプルです。g_ortho と違い表示中心位置の変更はできませんが、ズーム処理の場合こちらの方が簡単に実装できます。	g_scale

5 章サンプルに関して

サンプルファイル	概 要	主に使用している G マクロ命令
gsamp104. psv	レイヤ単位で表示するスケール範囲を設定するサンプルです。	<code>g_layer_range</code>
gsamp105. psv	ビットマップシンボルオブジェクトの作成サンプルです。ビットマップシンボルは表示サイズ固定の場合、9 箇所の基準点を設定できます。	<code>g_bitmapsymbol</code> <code>g_bitmapsymbol_file</code>
gsamp106. psv	図形オブジェクトの表示サイズ範囲を設定するサンプルです。	<code>g_obj_range</code>
gsamp107. psv	鉄道線オブジェクトの作成サンプルです。鉄道線オブジェクトは、JR 線、私鉄線、国道などの幅を持った線を描画する場合に使用します。描画パターンは4種類提供しています。	<code>g_RRline</code>
gsamp108. psv	新測地系(現在の日本測地系)対応のマクロを実行するサンプルです。高精度な計算を行う場合は、国土地理院から変換パラメータファイルをダウンロードしてサンプルに読み込ませる必要があります。ない場合は誤差が大きい簡易計算を行います。	<code>g_TKY2JGD_ParaFile</code> <code>g_TBL2JBL</code> <code>g_JBL2JXY</code> <code>g_TBL2TXY</code>
gsamp120. psv	レイヤ情報を取得するサンプルです。マウスカーソルに最も近いオブジェクトのレイヤ情報を取得して表示します。	<code>g_layer_name</code>

5.4 その他サンプル

起動HTML中の「GUI構築例サンプルへ」をクリックするとアプリケーションの画面を作成したサンプルを表示します（図 5-3参照）。このサンプルは、ビットマップを利用した簡単な地図操作画面の構築例です。ダイアログの部品化や、標準部品の使用、ホイールイベントの使用例として参考にしてください。

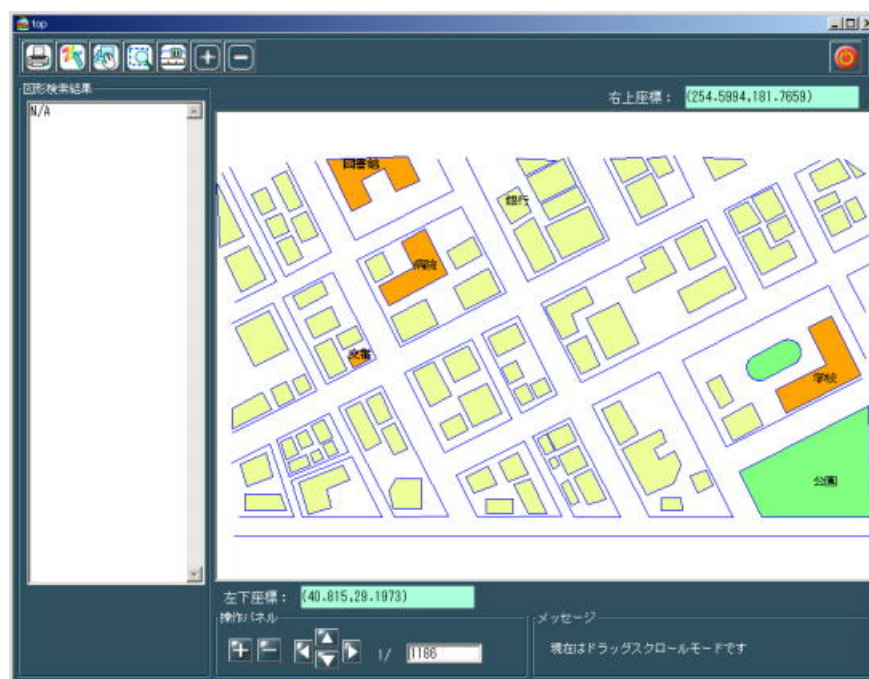


図 5-3 GUI 構築例サンプル画面

その他にも、『PreSerV WebTcl 基本パック』にはサンプルが同封しています。

表 5-3にサンプルの概要と、配置場所の一覧を記載します。^{注9)}

注9) 配置場所では、インストールディレクトリを *HOME* と表記しています。

表 5-3 その他サンプル一覧

サンプルファイル	概 要	配置場所	主に使用している命令
charlist.psv	文字オブジェクトの文字コードリストを表示します。フォント間で共通の表示ができない外字領域の確認が可能です。これを参考に、アプリケーション作成時のコード範囲を考慮して、フォントの選択を行ってください。	HOME¥sample¥ ディレクトリ下	g_text
custom_wd.psv	カスタマイズ部品のサンプルです。source 実行後、cu_create_mag プロシジャを実行すると虫眼鏡を表示します。cu_create_keypad プロシジャを実行するとテンキーを生成します。	"	g_scroll st_operation g_ortho
esamp002.psv	タイマーの割り込みプロシジャのサンプルです。	"	e_timer
esamp003.psv	起動後の経過時間を表示するサンプルです。	"	e_clock
esamp004.psv	現在のシステム日付を表示するサンプルです。	"	e_date
esamp005.psv	スレッド実行のサンプルです。	"	e_thread_begin
esamp006.psv	スレッド制御のサンプルその 1 です。	"	e_thread_loop e_thread_begin e_thread_end e_thread_stop
esamp007.psv	スレッド制御のサンプルその 2 です。	"	e_thread_loop e_thread_begin e_thread_end e_thread_stop
sample1.exe	起動確認サンプルです。簡単な AP サーバとの通信を行うサンプルで WebTcl アプリの基本動作の確認に使用します。自己解凍 EXE 形式でインストール CD のランチャ画面から起動します。	CD¥sample1¥ ディレクトリ下	
pri_list.exe	プラグイン作成のサンプルです。VC++ のプロジェクトです。本サンプルでは Windows の API を実行し、PC で使用可能なプリンタ名を列挙して、", " 区切り文字列として返します。自己解凍 EXE 形式でインストール CD のランチャ画面から起動します。	CD¥sample2¥ ディレクトリ下	