

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 API リファレンス(その他)

Ver. 6.0

マニュアル

はじめに

はじめに

本書では、PreSerV WebTcl の E マクロ、標準部品、FE-COM の API に関する動作仕様を説明します。

目次

1	APIリファレンスに関して	5
1.1	表記に関して	5
1.2	用語説明	6
2	Eマクロ・標準部品	7
2.1	Eマクロ	7
2.1.1	e_clock	8
2.1.2	e_close	9
2.1.3	e_connect	10
2.1.4	e_date	11
2.1.5	e_error	12
2.1.6	e_id	13
2.1.7	e_plugin	14
2.1.8	e_recv	15
2.1.9	e_send	16
2.1.10	e_sleep	17
2.1.11	e_thread_begin	18
2.1.12	e_thread_end	19
2.1.13	e_thread_loop	20
2.1.14	e_thread_stop	21
2.1.15	e_timer	22
2.1.16	e_external_send	23
2.2	標準部品 (st) 索引	24
2.2.1	st_open	24
2.2.2	st_close	25
2.2.3	st_operation	26
2.2.4	st_reset	34
2.2.5	st_set_grid	35
2.2.6	st_grid_ctl	37
2.2.7	st_ndeflist	38
2.3	標準部品操作説明	39
2.3.1	マウス操作	39
2.3.2	図形の選択方法	40

2.3.3	画面の拡大	40
2.3.4	画面の縮小	41
2.3.5	画面のスクロール	41
2.3.6	範囲指定のプリント	41
2.3.7	最近点検索	41
2.3.8	線作成	42
2.3.9	ポリライン作成	42
2.3.10	ポリゴン作成	42
2.3.11	矩形作成	43
2.3.12	円作成	43
2.3.13	円弧作成	43
2.3.14	楕円作成	44
2.3.15	文字作成	44
2.3.16	スプライン作成	45
2.3.17	フリーライン作成	45
2.3.18	角丸長方形作成	46
2.3.19	正多角形作成	46
2.3.20	移動	46
2.3.21	削除	47
2.3.22	変形	47
2.3.23	回転	48
2.3.24	コピー	48
2.3.25	拡大縮小	49
2.3.26	点追加	49
2.3.27	点削除	50
2.3.28	点移動	50
2.4	標準部品カスタマイズ方法	51
2.4.1	カスタマイズの考え方	51
2.4.2	カスタマイズ例	52
2.4.3	外部変数一覧	56
2.4.4	アクションごとの外部変数設定内容	58
3	WebTcl FE-COM API	62
3.1	WebTcl FE-COM API概要	62
3.2	WebTclFE-COM APIリファレンス	64

3.3	FE-COMメソッド詳細	65
3.3.1	execCmd.....	65
3.3.2	init_e_external_send.....	65
3.4	FE-COMプロパティ詳細	66
3.4.1	AskRequestURL.....	66
3.4.2	CertRetryCount.....	66
3.4.3	CodeBase.....	66
3.4.4	DebugPolicy.....	67
3.4.5	EndActionURL.....	67
3.4.6	EndModifyString.....	68
3.4.7	InitActionURL.....	68
3.4.8	InitModifyString.....	69
3.4.9	InitSource.....	69
3.4.10	ModifyString.....	70
3.4.11	TclVersionOption.....	70
3.4.12	TimeOutExecCmd.....	71
3.4.13	TimeOutExecInterval.....	71
3.4.14	TimeOutVal.....	72
3.4.15	TclIndexURL.....	72
3.4.16	ActionCodeBase.....	73
3.4.17	DuplicationErrorMessage.....	73
3.4.18	HideMode.....	74
3.4.19	StandardRecvFlag.....	74
3.5	WebTcl FE-COM APIエラーメッセージ	75
3.5.1	エラーメッセージ取得方法.....	75
3.5.2	エラーメッセージ一覧.....	76
4	その他コマンド	81
4.1	C/S版スクリプト	81
4.1.1	インデックス生成コマンド(createIndex).....	81
5	付録	86
5.1	標準部品内部マクロ	86

1 API リファレンスに関して

1.1 表記に関して

(1) 索引の使い方に関して

API は接頭語 (“w_” など)を除いたアルファベット順に記載しています。
また 2 章の先頭には機能別索引の表を提示しています。

(2) API 更新のラベルに関して

3 章 API リファレンスでは、仕様変更を確認しやすいように、機能一覧表に以下のラベルを添付しています。

Ver. 3更新 Ver. 3 版で API 自体もしくは API のオプションを追加、削除、変更しました。

Ver. 4更新 Ver. 4 版で API 自体もしくは API のオプションを追加、削除、変更しました。

Ver. 5更新 V5 版で API 自体もしくは API のオプションを追加、変更しました。

WebTcl更新 WebTcl 版で API 自体もしくは API のオプションを追加、変更しました。

API の更新に関する詳細な履歴は、各 API のページ下部に記述されていますので、そちらを参照してください。

(3) 使用例の表記

「XXX.psv 参照」：XXX.psv は本体付属のサンプルスクリプトファイルを示しています。

1.2 用語説明

WebTcl API リファレンス中の用語に関しては、別冊の「PreSerV WebTcl 用語集」を参照してください。

2 E マクロ・標準部品

2.1 E マクロ

E マクロは、PreSerV 上で実行するアプリケーション(スクリプト)が、AP サーバ上のアプリケーションと通信を行うためのマクロや、タイマーによるマクロ実行、スレッドでのイベント待ちなどを行うマクロを提供しています。本節ではE マクロのAPI 仕様を説明します。

表 2-1 E マクロ機能索引

E マクロ			
e_clock	PreSerV を起動してからの時間の取得	Ver. 3更新	2.1.1
e_close	セッションのクローズ	WebToI更新	2.1.2
e_connect	セッションの接続	WebToI更新	2.1.3
e_date	システム日付の取得	Ver. 3更新	2.1.4
e_error	セッションのエラー時における実行マクロの登録	WebToI更新	2.1.5
e_id	セッション ID の取得	WebToI更新	2.1.6
e_plugin	C 言語によるマクロの拡張	Ver. 3更新 WebToI更新	2.1.7
e_recv	セッションからのデータ受信	WebToI更新	2.1.8
e_send	セッションに文字列を送信	WebToI更新	2.1.9
e_sleep	スリープ	Ver. 3更新	2.1.10
e_thread_begin	新しいスレッドの起動	Ver. 3更新	2.1.11
e_thread_end	スレッドの終了	Ver. 3更新	2.1.12
e_thread_loop	スレッドでのイベント待ち	Ver. 3更新	2.1.13
e_thread_stop	スレッドの強制終了	Ver. 3更新	2.1.14
e_timer	タイマー割り込み実行	Ver. 3更新	2.1.15
e_external_send	外部関係用マクロ	WebToI更新	2.1.16

2.1.1 e_clock

PreSerV を起動してからの時間の取得

(1) 形式

e_clock

(2) 説明

e_clock マクロは、PreSerV の起動からの時間を取得します。

(3) 戻り値

PreSerV が起動されてからの時間（単位：秒）を少数点表記で返します。

(4) 関連項目

なし

(5) 使用例

esamp003.psv 参照（以下画面例）

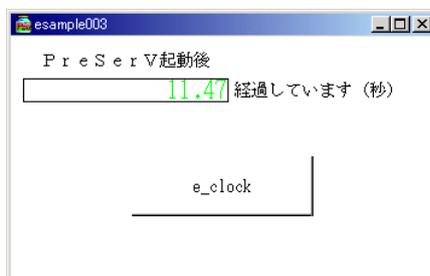


図 2-1 esamp003.psv 実行画面

(6) 更新履歴

① Ver 3.0

Ver. 3.0 で新規に追加したマクロです。

2.1.2 e_close

セッションのクローズ

(1) 形式

e_close [-s]

(2) 説明

e_closeマクロは、*sessionID*で指定されたセッションをクローズします。

① -s *sessionID*

セッション ID を指定します。省略するとカレントのセッション ID を使用します。

※ カレントのセッション ID が未設定の場合、通信でを使用したセッション ID を設定します。

(3) 戻り値

なし

(4) 関連項目

e_id, e_send, e_recv, e_error, e_connect

(5) 使用例

動作確認アプリケーションの CheckMenu.psv 参照

(6) 更新履歴

① WebTcl

WebTcl で FE-COM を経由して、AP サーバと通信する機能に変更しました。

2.1.3 e_connect

セッションの接続

(1) 形式

`e_connect hostName flag [services|port]`

(2) 説明

`e_connect`マクロは、*hostName*で指定されたマシンのアプリケーションとのセッションを開始します。

① *hostName*

TCP/IP のホスト名前を指定します。

※ WebTcl ではこの指定は無効です。下位互換性のため指定を残しています。

② *flag*

以後 PreSerV のディスパッチルーチンに登録する場合は on、特に指定しない場合は、off を指定します。

※ WebTcl ではこの指定は無効です。e_send で受信したレスポンスはコマンド実行、e_recv で受信したレスポンスはデータとして扱います。

③ *services|port*

サービスを指定します。サービス名またはポート番号で 指定します。省略した場合は、PreSerV とセッションを開始します。

※ WebTcl ではこの指定は無効です。下位互換性のため指定を残しています。

(3) 戻り値

セッション ID、0 以下の値の場合はエラーとなります。0: 通信エラー全般、-1: 予期せぬ通信障害、-2: 偽接続モードによる一時的接続状態(直後に切断処理を実行し、FE-COM#EndActionURL が実行されます)。

(4) 関連項目

`e_id`, `e_recv`, `e_error`, `e_send`, `e_close`

(5) 使用例

動作確認アプリケーションの CheckMenu.psv 参照

(6) 更新履歴

① WebTcl

WebTcl で FE-COM を経由して、AP サーバと通信する機能に変更しました。

2.1.4 e_date

システム日付の取得

(1) 形式

e_date

(2) 説明

e_date マクロは、現在のシステム日付を取得します。

(3) 戻り値

システム日付（年 月 日 曜日 時 分 秒）をリスト形式で返します。

年：10進数で表す4桁の西暦

月：10進数で表す月（01～12）

日：10進数で表す日（01～31）

曜日：曜日の正式名（英語）例）” Sunday”、” Monday”

時：24時間表記の時間（00～23）

分：10進数で表す分（00～59）

秒：10進数で表す秒（00～59）

(4) 関連項目

なし

(5) 使用例

esamp004.psv 参照（以下画面例）



図 2-2 esamp004.psv 実行画面

(6) 更新履歴

① Ver 3.0

Ver. 3.0 で新規に追加したマクロです。

2.1.5 e_error

セッションのエラー時における実行マクロの登録

(1) **形式**

`e_error [-s] macro_string`

(2) **説明**

`e_error` マクロは、*sessionID*で指定されたセッションが`e_send`実行時と`e_close`実行時に通信エラーとなった時に実行するマクロを指定します。

`e_error` 指定時は、`e_send` と `e_close` の通信エラー処理(強制終了)は実行されず、指定したマクロの処理だけが実行されます。

① `-s sessionID`

セッション ID を指定します。省略するとカレントのセッション ID を使用します。

※ カレントのセッション ID が未設定の場合、通信で使用したセッション ID を設定します。

② `macro_string`

実行するマクロを指定します。

(3) **戻り値**

なし

(4) **関連項目**

`e_id`, `e_recv`, `e_send`, `e_connect`, `e_close`

(5) **使用例**

なし

(6) **更新履歴**

① WebTcl

WebTcl で FE-COM を経由して、AP サーバと通信する機能に変更しました。

2.1.6 e_id

セッション ID の取得

(1) **形式**

e_id

(2) **説明**

e_id マクロはカレントのセッション ID を取得します。

※ カレントのセッション ID が未設定の場合、通信でを使用したセッション ID を設定します。

(3) **戻り値**

セッション ID

(4) **関連項目**

e_send, e_recv, e_error, e_connect, e_close

(5) **使用例**

なし

(6) **更新履歴**

① WebTcl

WebTcl で FE-COM を経由して、AP サーバと通信する機能に変更しました。

2.1.7 e_plugin

C 言語によるマクロの拡張

(1) 形式

`e_plugin libName macroName [macroName...]`

(2) 説明

e_plugin マクロは、C 言語で新しいマクロコマンドを定義し、WebTcl の機能を拡張します。

① libName

新しいマクロを格納しているライブラリ名を指定します。ライブラリは、DLL 形式で作成してください。

② macroName [macroName...]

新しく定義するマクロ名を指定します。マクロ名と関数名は同一にする必要があります。

(3) 戻り値

ライブラリのファイル ID

(4) 関連項目

なし

(5) 使用例

print.psv 参照 (以下画面例)

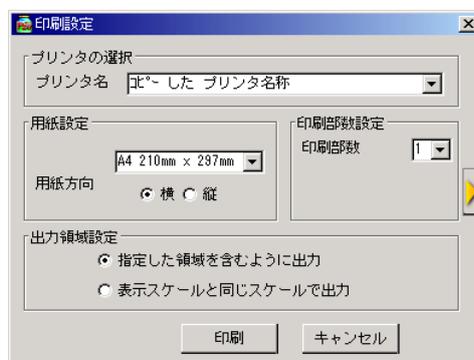


図 2-3 print.psv 実行画面

(6) 更新履歴

① WebTcl

WebTcl で FE-COM を経由して、Web サーバから DLL をダウンロードする機能に変更しました。

2.1.8 e_recv

セッションからのデータ受信

(1) **形式**

`e_recv sessionID [timeVal]`

(2) **説明**

e_recvマクロは、*sessionID*で指定したセッションから送信されるデータを受信します。

① *sessionID*

受信するセッション ID を指定します。

② *timeVal*

タイムアウト値を秒単位で指定します。

(3) **戻り値**

受信した文字列。タイムアウトの場合は、“timeout”の文字列となります。

(4) **関連項目**

`e_id`, `e_connect`, `e_error`, `e_send`, `e_close`

(5) **使用例**

なし

(6) **更新履歴**

① WebTcl

WebTcl で FE-COM を経由して、AP サーバと通信する機能に変更しました。

2.1.9 e_send

セッションに文字列を送信

(1) **形式**

`e_send [-s] string`

(2) **説明**

`e_send`マクロは、*sessionID*で指定するセッションへ文字列を送信します。

① `-s sessionID`

セッション ID を指定します。省略するとカレントのセッション ID を使用します。

※ カレントのセッション ID が未設定の場合、通信で使したセッション ID を設定します。

② `string`

送信する文字列を指定します。

(3) **戻り値**

なし

(4) **関連項目**

`e_id`, `e_recv`, `e_connect`, `e_error`, `e_close`

(5) **使用例**

動作確認アプリケーションの `CheckMenu.psv` 参照

(6) **更新履歴**

① WebTcl

WebTcl で FE-COM を経由して、AP サーバと通信する機能に変更しました。

2.1.10 e_sleep

スリープ

(1) 形式

`e_sleep timeVal`

(2) 説明

`e_sleep` マクロは、指定された時間だけスリープします。

① *timeVal*

スリープする時間（ミリ秒）を指定します。スレッド単位でスリープします。

(3) 戻り値

なし

(4) 関連項目

なし

(5) 使用例

`sampleIndex.psv` 参照（以下画面例）

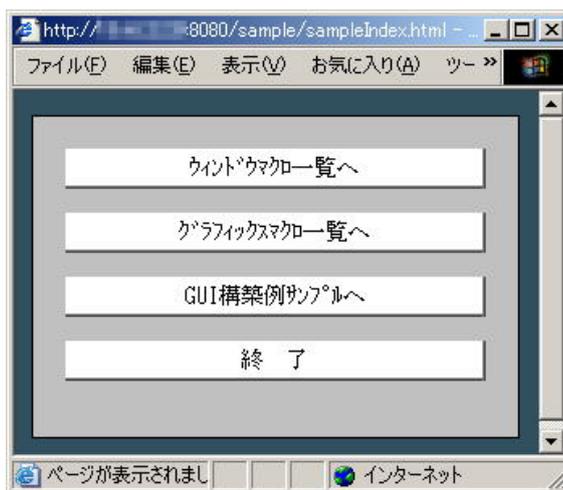


図 2-4 `sampleIndex.psv` 実行画面

(6) 更新履歴

① Ver. 3.0

Ver. 3.0 で新規に追加したマクロです。

2.1.11 e_thread_begin

新しいスレッドの起動

(1) **形式**

`e_thread_begin` *command*

(2) **説明**

`e_thread_begin` マクロは、新しいスレッドを起動して指定のマクロを実行します。

① *command*

新しいスレッドで実行するマクロを指定します。

(3) **戻り値**

スレッドオブジェクト ID

(4) **関連項目**

`e_thread_end`, `e_thread_loop`, `e_thread_stop`

(5) **使用例**

esamp006.psv 参照 (以下画面例)

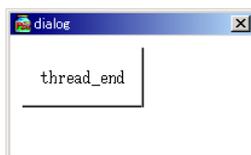


図 2-5 esamp006.psv 実行画面

(6) **更新履歴**

① Ver. 3.0

Ver. 3.0 で新規に追加したマクロです。

2.1.12 e_thread_end

スレッドの終了

(1) **形式**

`e_thread_end` *exit_code*

(2) **説明**

`e_thread_end` マクロは、`e_thread_begin` で起動されたスレッドを終了します。

① *exit_code*

OS にリターンする終了値を指定します。

(3) **戻り値**

なし

(4) **関連項目**

`e_thread_begin`, `e_thread_loop`, `e_thread_stop`

(5) **使用例**

esamp006.psv参照 (画面例は2.1.11の画面参照)

(6) **更新履歴**

① Ver. 3.0

Ver. 3.0 で新規に追加したマクロです。

2.1.13 e_thread_loop

スレッドでのイベント待ち

(1) **形式**

e_thread_loop

(2) **説明**

e_thread_loop マクロは、e_thread_begin で起動されたスレッドに対してイベント待ちを行います。本マクロを呼ばない場合は、起動時に指定したマクロの実行が終了すると自動的にスレッドは終了します。

イベント待ちはウィンドウを生成していなければいけません。スレッドを終了する時は、スレッド内で e_thread_end を呼ぶか、e_thread_stop を呼んで強制的に終了してください。

(3) **戻り値**

なし

(4) **関連項目**

e_thread_begin, e_thread_end, e_thread_stop

(5) **使用例**

esamp006.psv参照 (画面例は2.1.11の画面参照)

(6) **更新履歴**

① Ver. 3.0

Ver. 3.0 で新規に追加したマクロです。

2.1.14 e_thread_stop

スレッドの強制終了

(1) **形式**

`e_thread_stop threadObjID exit_code`

(2) **説明**

e_thread_stop マクロは、e_thread_begin で起動したスレッドを強制的に終了します。

① *threadObjID*

強制終了させるスレッドオブジェクトの ID を指定します。

② *exit_code*

OS にリターンする終了値を指定します。

(3) **戻り値**

なし

(4) **関連項目**

e_thread_begin, e_thread_end, e_thread_loop

(5) **使用例**

esamp006.psv参照 (画面例は2.1.11の画面参照)

(6) **更新履歴**

① Ver. 3.0

Ver. 3.0 で新規に追加したマクロです。

2.1.15 e_timer

タイマー割り込み実行

(1) **形式**

`e_timer timeVal macro_string`

(2) **説明**

`e_timer` マクロは、指定した割り込み間隔でマクロを実行します。

① *timeVal*

割り込み間隔（ミリ秒）を指定します。

② *macro_string*

指定した割り込み間隔で実行するマクロを指定します。割り込みを解除する時には、*timeVal*に0をセットし、*macro_string*に割り込みオブジェクト ID を設定します。

(3) **戻り値**

割り込みオブジェクト ID

(4) **関連項目**

なし

(5) **使用例**

esamp002.psv 参照 (`e_timer` で下のトップウィンドウが右に移動していきま
す)

(6) **更新履歴**

① Ver. 3.0

Ver. 3.0 で新規に追加したマクロです。

2.1.16 e_external_send

外部関係用マクロ

(1) **形式**

`e_external_send mode message`

(2) **説明**

`e_external_send` マクロは、Tcl のスクリプトから JavaScript への関係を可能にします。本マクロ実行後、FE-COM の `FECallback` イベントが発生します。JavaScript 側は FE-COM オブジェクトの `FECallback` イベントを実装しておくことで、Tcl のスクリプトからの、`mode` と `message` を受け取り、処理を実行可能です。

注) 本マクロ実行前には必ず FE-COM の `init_e_external_send` メソッドを実行してください。

① *mode*

JavaScript 側の処理モードのキーとなる情報を数値で渡します。数値変換不可能な値を指定した場合必ず 0 として `FECallback` に通知します。

② *message*

送信する文字列を指定します。

(3) **戻り値**

なし

(4) **関連項目**

FE-COM オブジェクトの `init_e_external_send` メソッド、`FECallback` イベント参照

(5) **使用例**

なし

(6) **更新履歴**

① WebTcl

WebTcl で追加したマクロです。

2.2 標準部品 (st) 索引

2.2.1 st_open

標準部品の初期処理

(1) 形式

`st_open canvasWindowID`

(2) 説明

`st_open`は、標準部品で使用する変数を初期化し、イベントハンドラを `canvasWindowID`で 指定されたキャンバスへセットします。`st_open`は、1 キャンバスにつき 1 回有効です。

他の標準部品では、`st_open` の戻り値である標準部品 ID を引数として指定します。

`st_open` を実行して標準部品を使用するには、事前にマクロで標準部品マクロのロード (source `st_op.psv`) を行っておく必要があります。また、少なくとも以下のマクロを実行しておく必要があります。

- `w_canvas`
- `g_context`
- `g_open_gwin`
- `g_open_dlist`

① `canvasWindowID`

`w_canvas` で作成したウィンドウ ID を指定します。

(3) 戻り値

標準部品 ID

(4) 関連項目

`st_close`

(5) 使用例

なし

(6) 更新履歴

なし

2.2.2 st_close

標準部品の終了処理

(1) 形式

`st_close st/D`

(2) 説明

`st_close` は、標準部品で使った変数をクリアし、標準部品の処理を終了します。

① `st/D`

クローズする標準部品 ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`st_open`

(5) 使用例

なし

(6) 更新履歴

なし

2.2.3 st_operation

画面操作／図形作成／図形編集の実行

(1) 形式

```
st_operation stID contextID gwinID dlistID [dlistID ...] action [eproc
arg...]
```

(2) 説明

st_operationは、*action*で指定されたアクションをイベントハンドラに割り当てます。イベントハンドラへの割り当てにより、キャンバスの画面操作および、キャンバスへの図形作成、図形編集を実行できます。

たとえばキャンバス上に矩形を作成する場合、アクションとして矩形作成（機能名 *rect*）を指定して st_operation を実行します。これによりイベントハンドラに矩形作成機能が割り当てられます。

ユーザーがキャンバス上でマウスクリックすると、始点が指定され、ラバーバンドが表示されます。次のマウスクリックで終点が指定され、矩形が描画されます。

一度指定したアクションは、他のアクションを指定して st_operation を実行するか、st_reset を実行するまで有効です。すなわち矩形の作成であれば、キャンバス上でマウスクリックを繰り返すことによって、連続して矩形が描画されます。

アクションの1回の処理が終了する（たとえば一つの矩形が描画される）ごとに、*eproc*で指定した関数が実行されます。*eproc*には、引数として各アクションの戻り値が渡されますので、これを参照して任意の後処理を行ってください。

たとえば、st_operationの図形作成のアクションでは、キャンバスへの図形の描画は行いますが、図形オブジェクトは作成されません。必要に応じて *eproc* 中で、グラフィックスマクロによる図形オブジェクトの作成や、ディスプレイリストメモリへの図形オブジェクトの格納などの処理を行ってください。

アクションごとの操作方法については、操作説明を参照してください。

① *stID*

st_open で作成した標準部品 ID を指定します。

② *contextID*

g_context で作成したコンテキスト ID を指定します。

③ *gwinID*

図形を描画するグラフィックスウィンドウを、g_open_gwin で作成したグラフィックスウィンドウ ID で指定します。

なお、図形作成のアクションでは、グラフィックスウィンドウに設定している描

画色、属性を使用します。

④ *dlistID* [*dlistID* ...]

ディスプレイリストメモリを指定します。一つのグラフィックスウィンドウに複数のディスプレイリストメモリを設定している場合は、ディスプレイリストメモリを複数指定します。

図形編集および画面制御のアクションでは、ここで指定したディスプレイリストメモリ中の図形オブジェクトが、編集や表示の対象となります。

ただし、図形オブジェクトのコピーにおいて、コピー先図形オブジェクトを *contextID* のカレントのディスプレイリストメモリに格納します。

⑤ *action*

アクションの機能名を指定します。アクションに引数があるときは一つの文字列にして渡します。

例)

(a) 画面の拡大

```
st_operation stID contextID gwinID dlistID enlarge
```

(b) 正多角形（六角形）の作成

```
st_operation stID contextID gwinID dlistID {regpoly 6}
```

(c) 図形の移動（単一選択）

```
st_operation stID contextID gwinID dlistID move_obj
```

各機能の機能名、引数、戻り値を以下に示します。

戻り値中の座標(X, Y)は、すべて WC 座標です。

表 2-2 enlarge 引数説明

画面の拡大	キャンバスの表示倍率を拡大します。
機能名	enlarge
引数	なし
戻り値	enlarge 拡大後の g_ortho の値(左下 X 左下 Y 右上 X 右上 Y)

表 2-3 shrink 引数説明

画面の縮小	キャンバスの表示倍率を縮小します。
機能名	shrink
引数	なし
戻り値	shrink 縮小後の g_ortho の値(左下 X 左下 Y 右上 X 右上 Y)

表 2-4 scroll 引数説明

画面のスクロール	キャンバスの表示をスクロールします。
機能名	scroll
引数	なし
戻り値	scroll スクロール後の g_ortho の値(左下 X 左下 Y 右上 X 右上 Y)

表 2-5 print 引数説明

範囲指定のプリント	キャンバス上の矩形範囲のポストスクリプトファイルを生成します。作成したファイルは、UNIX や Windows のプリンタ出力コマンドを使用して、プリンタに出力します。 <small>注)WebTcl では Windows プリンタ出力以外は未サポートです。</small>
機能名	print
引数	紙タイプ : ポストスクリプトファイル名 紙タイプには次のいずれかを指定します。 a4l : A4 横, a3l : A3 横, a4p : A4 縦, a3p : A3 縦
戻り値	なし

表 2-6 nearest 引数説明

最近点検索	キャンバス上でクリックした座標に最も近い図形の図形要素 ID を返します。
機能名	nearest
引数	なし
戻り値	nearest 図形オブジェクト ID 座標 X 座標 Y

表 2-7 line 引数説明

線作成	直線を作成します。
機能名	line
引数	なし
戻り値	line 始点 X 始点 Y 終点 X 終点 Y

表 2-8 polyline 引数説明

ポリライン作成	連続線を作成します。
機能名	polyline
引数	なし
戻り値	polyline 点数 X1 Y1 ... Xn Yn

表 2-9 polygon 引数説明

ポリゴン作成	多角形を作成します。
機能名	polygon
引数	なし
戻り値	polygon 点数 X1 Y1 ... Xn Yn

表 2-10 rect 引数説明

矩形作成	矩形を作成します。
機能名	rect
引数	なし
戻り値	rect 始点 X 始点 Y 終点 X 終点 Y

表 2-11 circle 引数説明

円作成	円を作成します。
機能名	circle
引数	なし
戻り値	circle 中心点 X 中心点 Y 半径

表 2-12 arc 引数説明

円弧作成	円弧を作成します。
機能名	arc
引数	なし
戻り値	arc 中心点 X 中心点 Y 半径 開始角度 終了角度

表 2-13 ellipse 引数説明

楕円作成	楕円を作成します。
機能名	ellipse
引数	なし
戻り値	ellipse 中心点 X 中心点 Y 長径 短径 回転角度

表 2-14 text 引数説明

文字作成	文字を入力します。
機能名	text
引数	なし
戻り値	text 左下 X 左下 Y 文字数 文字列 高さ 幅 回転角度 文字間隔

表 2-15 spline 引数説明

スプライン作成	スプライン曲線を作成します。
機能名	spline
引数	なし
戻り値	spline 点数 X1 Y1 ... Xn Yn

表 2-16 freeline 引数説明

フリーライン作成	自由曲線を作成します。
機能名	freeline
引数	なし
戻り値	freeline 点数 X1 Y1 ... Xn Yn

表 2-17 roundbox 引数説明

角丸長方形作成	角丸長方形を作成します。
機能名	roundbox
引数	なし
戻り値	roundbox 始点 X 始点 Y 終点 X 終点 Y 半径 角度

表 2-18 regpoly 引数説明

正多角形作成	正多角形を作成します。
機能名	regpoly
引数	頂点数(3点以上)
戻り値	regpoly 中心点 X 中心点 Y 第1頂点 X 第1頂点 Y 角数

表 2-19 move_obj/move_obj_h/move_obj_r 引数説明

移動	図形を移動します。
機能名	move_obj(単一選択)/move_obj_h(複数選択)/move_obj_r(範囲選択)
引数	なし
戻り値	機能名 図形オブジェクト ID ...

表 2-20 del_obj/del_obj_h/del_obj_r 引数説明

削除	図形を削除します。
機能名	del_obj(単一選択)/del_obj_h(複数選択)/del_obj_r(範囲選択)
引数	なし
戻り値	機能名 図形オブジェクト ID ...

表 2-21 transform_obj/transform_obj_h/transform_obj_r 引数説明

変形	図形を変形します。 変形の基準点は選択図形の中心です。複数選択時の基準点は、選択した全図形に外接する矩形の中心になります(カスタマイズで変更できます)。図形のうち、円とシンボルは位置とサイズだけが変更し、形は変わりません。
機能名	transform_obj(単一選択)/transform_obj_h(複数選択) /transform_obj_r(範囲選択)
引数	なし
戻り値	機能名 図形オブジェクト ID ...

表 2-22 rotate_obj/rotate_obj_h/rotate_obj_r 引数説明

回転	図形を回転します。 回転の中心点は、選択した図形の外接矩形の中心です。 複数選択時の中心点は、選択した全図形に外接する矩形の中心になります(カスタマイズで変更できます)。
機能名	rotate_obj(単一選択)/rotate_obj_h(複数選択)/rotate_obj_r(範囲選択)
引数	なし
戻り値	機能名 図形オブジェクト ID ...

表 2-23 copy_obj/copy_obj_h/copy_obj_r 引数説明

コピー	図形を複製します。
機能名	copy_obj(単一選択)/copy_obj_h(複数選択)/copy_obj_r(範囲選択)
引数	なし
戻り値	機能名 {コピー元図形オブジェクト ID ... } {コピー先図形オブジェクト ID ... }

表 2-24 resize_obj/resize_obj_h/resize_obj_r 引数説明

拡大縮小	図形を拡大または縮小します。 拡大、縮小の基準点は、選択図形の中心です。複数選択時の基準点は、選択した全図形に外接する矩形の中心になります(カスタマイズで変更できます)。
機能名	resize_obj(単一選択)/resize_obj_h(複数選択)/resize_obj_r(範囲選択)
引数	なし
戻り値	機能名 図形オブジェクト ID ...

表 2-25 poly_edit_a 引数説明

点追加	ポリラインに点を追加します。
機能名	poly_edit_a
引数	なし
戻り値	poly_edit_a 図形オブジェクト ID

表 2-26 poly_edit_x 引数説明

点削除	ポリラインの点を削除します。
機能名	poly_edit_x
引数	なし
戻り値	poly_edit_x 図形オブジェクト ID

表 2-27 poly_edit_r

点移動	ポリラインの点を移動します。
機能名	poly_edit_r
引数	なし
戻り値	poly_edit_r 図形オブジェクト ID

⑥ *eproc*

ユーザー関数を指定します。この関数を指定したアクションの 1 回の処理が終了するごとに実行します。アクションの戻り値は *eproc* の第 3 引数で返ります（使用例を参照）。

(3) 戻り値

「説明」欄を参照

(4) 関連項目

なし

(5) 使用例

（例 1）ユーザー関数のコーディング例

```
# stID   : 標準部品 ID
# winID  : 対象のウィンドウ ID
# gpoint : アクションの戻り値
# args   : ユーザーの引数
#
proc usr_eproc { stID gwin gpoint args } {
  ...
}
```

(6) 更新履歴

なし

2.2.4 st_reset

標準部品のリセット

(1) 形式

`st_reset stID`

(2) 説明

`st_reset`は、`stID`で指定された標準部品に対する、`st_operation`の処理をリセットします（`st_operation`で使った内部変数のクリアとイベントハンドラの削除）。

① `stID`

標準部品 ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`st_operation`

(5) 使用例

なし

(6) 更新履歴

なし

2.2.5 st_set_grid

グリッドの生成

(1) 形式

`st_set_grid stID contextID gwinID xmin ymin xmax ymax width height`

(2) 説明

`st_set_grid`は、`stID`で指定されたキャンバスにグリッドを生成します。生成したグリッドの表示／非表示、座標指定の補正あり／なしは、`st_grid_ctl`で切り換えられます。生成時の設定は、表示、補正ありです。

① `stID`

標準部品 ID を指定します。

② `contextID`

`g_context` で作成したコンテキスト ID を指定します。

グリッドの図形オブジェクトを、指定したコンテキストオブジェクトのカレントのディスプレイリストメモリに格納します。

③ `gwinID`

グリッドを生成するグラフィックスウィンドウを、`g_open_gwin`で作成したグラフィックスウィンドウ ID で指定します。

④ `xmin`

グリッドを生成する最小 X 座標 (WC) を指定します。

⑤ `ymin`

グリッドを生成する最小 Y 座標 (WC) を指定します。

⑥ `xmax`

グリッドを生成する最大 X 座標 (WC) を指定します。

⑦ `ymax`

グリッドを生成する最大 Y 座標 (WC) を指定します。

⑧ `width`

1 グリッドの横幅 (WC) を指定します。

⑨ `height`

1 グリッドの縦幅 (WC) を指定します。

(3) 戻り値

なし

(4) 関連項目

`st_grid_ctl`

(5) 使用例

なし

- (6) 更新履歴
なし

2.2.6 st_grid_ctl

グリッドの制御

(1) 形式

`st_grid_ctl stID contextID gwinID [-h] [-g]`

(2) 説明

`st_grid_ctl` は、グリッドに対して引数で指定された制御を行います。

① *stID*

標準部品 ID を指定します。

② *contextID*

`g_context` で作成したコンテキスト ID を指定します。

③ *gwinID*

グリッドを操作するグラフィックスウィンドウを、`g_open_gwin` で作成したグラフィックスウィンドウ ID で指定します。

④ `-h on|off`

グリッドを表示する／しないを切り換えます。

`on` : グリッドを表示します。

`off` : グリッドを非表示にします。

⑤ `-g on|off`

図形作成時および編集時の指定座標を、グリッド位置に補正するかどうかを指定します。

`on` : グリッド位置に合わせます。

`off` : グリッド位置に合わせません。

(3) 戻り値

なし

(4) 関連項目

`st_set_grid`

(5) 使用例

なし

(6) 更新履歴

なし

2.2.7 st_ldeflist

標準部品のオペレーション対象レイヤ制御

(1) 形式

```
st_ldeflist st/D -l [ layer/D ] layerNo [ [ [ layer/D ] layerNo] ... ]  
st_ldeflist st/D -l  
st_ldeflist st/D
```

(2) 説明

st_ldeflist は、st_operation 図形編集機能（最近点検策・移動・削除・変形・回転・コピー・拡大縮小・点追加・点削除・点移動）の検索対象レイヤ ID・NO の指定をします。

① st/D

標準部品 ID を指定します。

② -l [layer/D] LayerNo [[[layer/D] LayerNo] ...]

レイヤ番号を指定して、ディスプレイリストメモリから図形オブジェクトを検索します。レイヤオブジェクトIDやオブジェクト (*layer/D*) の指定がない場合は、コンテキストオブジェクトのカレントのレイが対象となります。

③ -l

現在の検索対象レイヤ ID・NO 変更せず戻り値の出力をします。

④ -l 省略時

検索対象レイヤ ID・NO の設定を解除します

(3) 戻り値

現在の検索対象レイヤ ID・NO リスト形式

(4) 関連項目

st_operation

(5) 使用例

なし

(6) 更新履歴

なし

2.3 標準部品操作説明

st_operation の各アクションの操作方法を説明します。機能名、引数、戻り値についてもあわせて示しています。

表 2-28 標準部品操作名称一覧

操作分類	操作名称
共通操作	マウス操作, 図形の選択方法
画面制御・印刷	画面の拡大, 画面の縮小, 画面のスクロール, 範囲指定のプリント
図形描画	線作成, ポリライン作成, ポリゴン作成, 矩形作成, 円作成, 円弧作成, 楕円作成, 文字作成, スプライン作成, フリーライン作成, 角丸長方形作成, 正多角形作成
図形編集	最近点検索, 移動, 削除, 変形, 回転, コピー, 拡大縮小, 点追加, 点削除, 点移動

2.3.1 マウス操作

マウスのボタンは以下のように定義します。

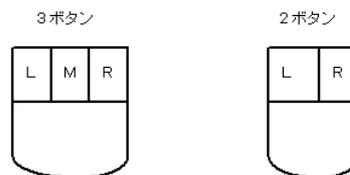


図 2-6 マウスボタンの定義

- 操作説明中では、マウス左ボタンをLボタン、マウス右ボタンをRボタン、3ボタンマウスの真中のボタンをMボタンと表記します。
- 3ボタンのMボタンは使用しません。
- オペレーションはLボタンクリックで確定し、Rボタンクリックでポップアップメニューを表示する操作が基本です。
- ポップアップメニューでは、[確定]、[キャンセル]、[リジェクト]のどれかを選択できます。
- [確定]は操作結果の確定、[キャンセル]は操作全体の取消し、[リジェクト]は直前の操作の取消しを行います。

2.3.2 図形を選択方法

- (1) 単一選択
 - ① ポインタを選択したい図形に移動し、図形の色が変わったらLボタンクリックします。
 - ② 図形を選択します。
- (2) 複数選択
 - ① ポインタを選択したい図形に移動し、図形の色が変わったらLボタンクリックします。
 - ② 選択したい図形をすべてLボタンクリックします。
 - ③ 選択した図形をもう一度Lボタンクリックすると選択が取り消されます。
 - ④ Rボタンクリックしてポップアップメニューを表示し、[確定]を選択します。
 - ⑤ 図形を複数選択します。
- (3) 範囲選択
 - ① 図形を囲む矩形範囲の左上、右下をLボタンクリックします。
 - ② 範囲が矩形で囲まれます。図形の一部でも範囲に入っていれば選択されます。
 - ③ Rボタンクリックしてポップアップメニューを表示し、[確定]を選択します。
 - ④ 図形が複数選択されます。
- (4) 複数選択および範囲選択で、図形を選択後にRボタンクリックし、ポップアップメニューから[キャンセル]を選択すると選択が取り消されます。
- (5) 範囲選択後は自動的に複数選択のモードになります。
- (6) 複数選択中にポップアップメニューから[範囲選択]を選択すると、選択方法を切り換えられます。

2.3.3 画面の拡大

キャンバスの表示倍率を拡大します。使用形式は `st_operation` の説明を参照してください。

- (1) 拡大したい範囲の左上、右下をLボタンクリックします。
指定範囲が全体に表示されるように拡大表示されます。

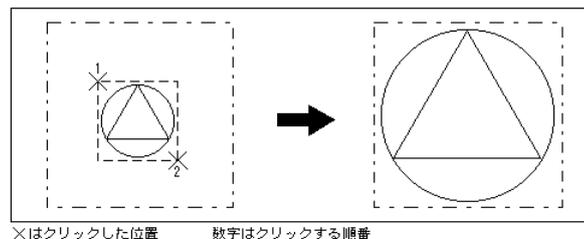


図 2-7 指定範囲と画面の拡大

2.3.4 画面の縮小

キャンバスの表示倍率を縮小します。使用形式は `st_operation` の説明を参照してください。

- (1) 縮小したい範囲の左上、右下を L ボタンクリックします。
全体が指定範囲に表示されるように縮小表示されます。

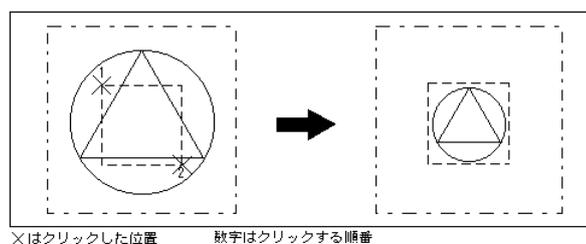


図 2-8 指定範囲と画面の縮小

2.3.5 画面のスクロール

キャンバスの表示をスクロールします。使用形式は `st_operation` の説明を参照してください。

- (1) L ボタンクリックし、ポインタをスクロールしたい方向に移動します。
表示内容がポインタの動きに合わせてスクロールします。
- (2) スクロールを終わらせたい位置で L ボタンクリックします。

2.3.6 範囲指定のプリント

キャンバス上の矩形範囲のポストスクリプトファイルを生成します。作成したファイルは、UNIX や Windows のプリンタ出力コマンドを使用して、プリンタに出力します。

- (1) 範囲の左上、右下を L ボタンクリックします。

2.3.7 最近点検索

キャンバス上のクリックした座標に最も近い図形の図形要素 ID を返します。

- (1) ポインタを移動します。
ポインタに最も近い図形の色が変わります。
- (2) 目的の図形の色が変わった位置で、L ボタンクリックします。

2.3.8 線作成

直線を作成します。

- (1) 始点、終点をLボタンクリックします。
線分が描画されます。

2.3.9 ポリライン作成

連続線を作成します。

- (1) 始点、各頂点をLボタンクリックします。
- (2) 終点でLボタンをダブルクリックします。または終点をLボタンクリック後、Rボタンをクリックしてポップアップメニューから[確定]を選択します。
ポリラインが描画されます。

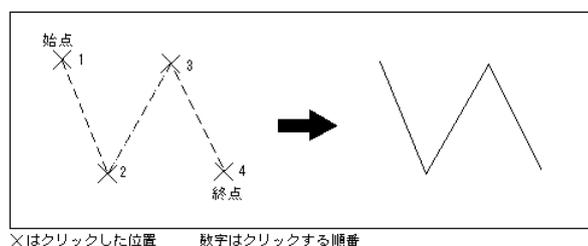


図 2-9 ポリラインの作成

- (3) 頂点をクリック後、ポップアップメニューから[キャンセル]を選択すると、各頂点を取り消されます。また[リジェクト]を選択すると直前の頂点を取り消されます。

2.3.10 ポリゴン作成

多角形を作成します。

- (1) 始点、各頂点をLボタンクリックします。
- (2) 終点でLボタンをダブルクリックします。または終点をLボタンクリック後、Rボタンをクリックしてポップアップメニューから[確定]を選択します。
ポリゴンが描画されます。
- (3) 頂点をクリック後、ポップアップメニューから[キャンセル]を選択すると、各頂点を取り消されます。また[リジェクト]を選択すると直前の頂点を取り消されます。

2.3.11 矩形作成

矩形を作成します。

- (1) 矩形の左上、右下をLボタンクリックします。
矩形が描画されます。

2.3.12 円作成

円を作成します。

- (1) 円の中心をLボタンクリックします。
- (2) 円周上の点をLボタンクリックします（円の半径を指定します）。
円が描画されます。

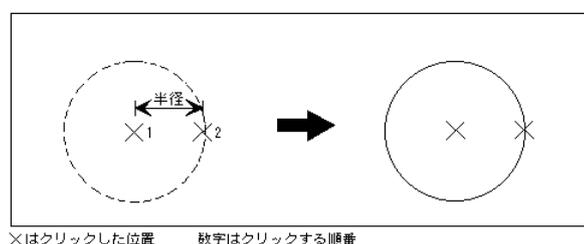


図 2-10 円の作成

2.3.13 円弧作成

円弧を作成します。

- (1) 円の中心をLボタンクリックします。
- (2) 円周上の点をLボタンクリックします（円の半径を指定します）。
- (3) 円弧の開始点、終了点をLボタンクリックします（開始点、終了点は反時計回りで指定します）。
円弧が描画されます。

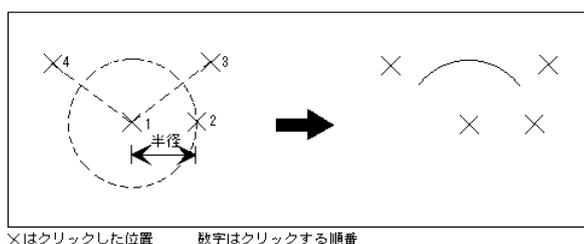


図 2-11 円弧の作成

2.3.14 楕円作成

楕円を作成します。

- (1) 楕円に外接する矩形の左上、右下をLボタンクリックします。
楕円が描画されます。

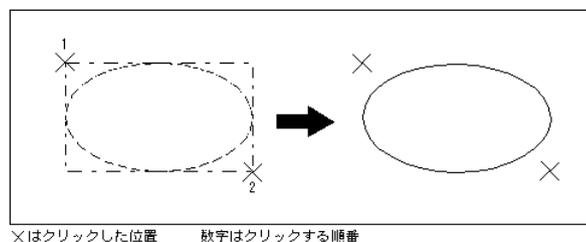


図 2-12 楕円の作成

2.3.15 文字作成

文字を入力します。

- (1) 文字を入力する範囲の左上、右下をLボタンクリックします。
これにより、文字の入力位置と大きさが決まります。

※ 横幅は文字サイズ、文字数で自動的に決まります。ここでは、どのような横幅を指定してもかまいません。

矩形が表示されます。

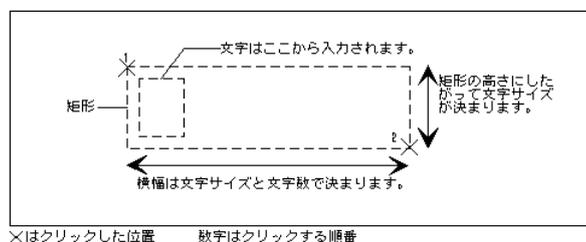


図 2-13 文字の作成

- (2) 矩形をLボタンクリックします。
文字入力ウィンドウが表示されます。
- (3) 文字を全角文字で入力し、エンターキーを押します。
矩形が消え、文字が入力されます。

2.3.16 スプライン作成

スプライン曲線を作成します。

- (1) 始点、通過点をLボタンクリックします。
各点が線で結ばれます。
- (2) 終点でLボタンをダブルクリックします。または終点をLボタンクリック後、Rボタンをクリックしてポップアップメニューから「確定」を選択します。
スプライン曲線が描画されます。

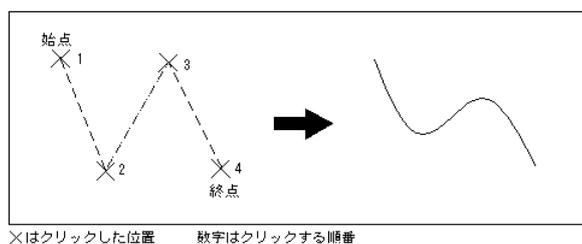


図 2-14 スプラインの作成

- (3) 通過点をクリック後、ポップアップメニューから「キャンセル」を選択すると、各通過点が取り消されます。また「リジェクト」を選択すると直前の通過点が取り消されます。

2.3.17 フリーライン作成

自由曲線を作成します。

- (1) キャンバスをLボタンでドラッグします。
ドラッグした軌跡に曲線が描画されます。

2.3.18 角丸長方形作成

角丸長方形を作成します。

- (1) 長方形の左上、右下をLボタンクリックします。
- (2) 長方形内にポインタを移動します。角に付ける円弧の通過点をLボタンクリックします。

角丸長方形が描画されます。

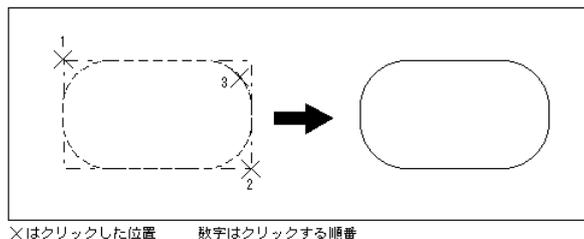


図 2-15 角丸長方形の作成

2.3.19 正多角形作成

正多角形を作成します。

- (1) 正多角形の中心点をLボタンクリックします。
- (2) 一つ目の頂点にする位置をLボタンクリックします。

正多角形が描画されます。

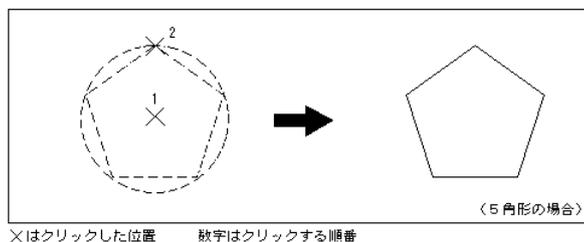


図 2-16 正多角形の作成

2.3.20 移動

図形を移動します。

- (1) 図形を選択します（共通：図形を選択）。
- (2) マウスを移動します。
ポインタにあわせて図形が移動します。
- (3) 移動したい位置でLボタンクリックします。
図形を移動します。

2.3.21 削除

図形を削除します。

- (1) 図形を選択します（共通：図形を選択）。
選択した図形が削除されます。

2.3.22 変形

図形を変形します。

変形の基準点は選択図形の中心です。複数選択時の基準点は、確定したときの点になります。なお図形のうち、円とシンボルは位置とサイズだけを変更して、形は変わりません。

- (1) 図形を選択します（共通：図形を選択）。
- (2) マウスを移動します。
ポインタにあわせて図形が変形します。
- (3) 変形したい位置でLボタンをクリックします。
図形を変形します。

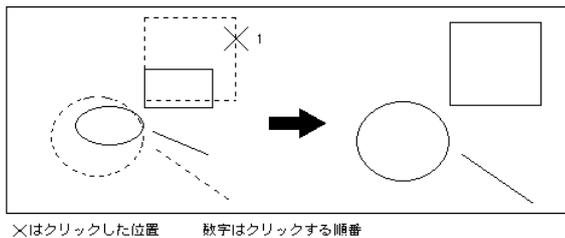


図 2-17 指定範囲と変形

2.3.23 回転

図形を回転します。

回転の中心点は、選択した図形の外接矩形の中心です。複数選択時の中心点は、選択した全図形に外接する矩形の中心になります（カスタマイズで変更できます）。

- (1) 図形を選択します（共通：図形を選択）。
- (2) マウスを移動します。
ポインタにあわせて図形が回転します。
- (3) 回転したい位置でLボタンクリックします。
図形を回転します。

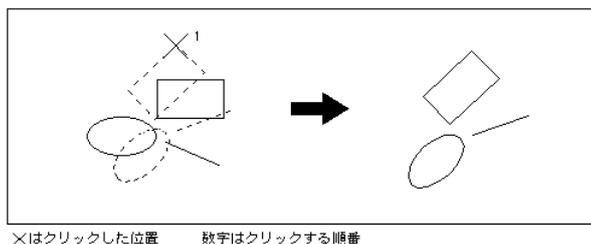


図 2-18 指定範囲と回転

2.3.24 コピー

図形を複写します。

- (1) 図形を選択します（共通：図形を選択）。
- (2) マウスを移動します。
ポインタにあわせて図形が移動します。
- (3) コピーしたい位置でクリックします。
図形をコピーします。

2.3.25 拡大縮小

図形を拡大または縮小します。

拡大、縮小の基準点は、選択図形の中心です。複数選択時の基準点は、確定したときの点になります。

- (1) 図形を選択します（共通：図形を選択）。
- (2) マウスを移動します。
ポインタにあわせて図形の大きさが変わります。
- (3) 拡大、縮小したい位置でLボタンクリックします。
図形を拡大、縮小します。

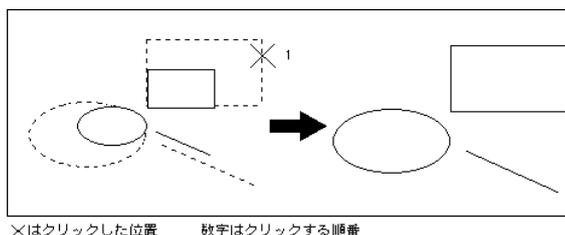


図 2-19 選択範囲と拡大縮小

2.3.26 点追加

ポリラインに点を追加します。

- (1) ポリラインを選択します（共通：図形を選択。単一選択のみ）。
- (2) マウスを移動します。ポインタにあわせて線が色替え表示されます。
- (3) 点追加する線の色が変わったらLボタンクリックします。
ラバーバンドを表示します。
- (4) マウスを移動し、点追加したい位置でLボタンクリックします。
ポリラインに点を追加します。

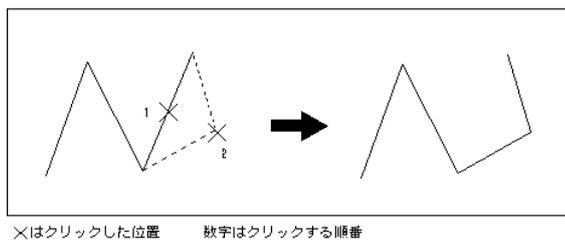


図 2-20 点の追加

2.3.27 点削除

ポリラインの点を削除します。

- (1) ポリラインを選択します（共通：図形を選択。単一選択のみ）。

- (2) マウスを移動します。

ポインタにあわせて点が色替え表示されます。

- (3) 削除する点の色が変わったらLボタンクリックします。

ポリラインの点を削除します。

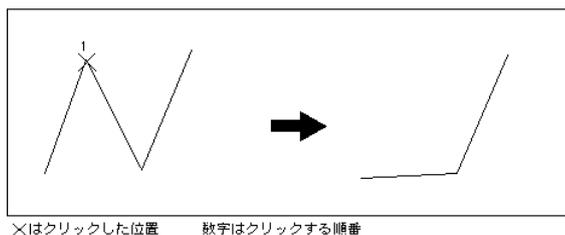


図 2-21 点の削除

2.3.28 点移動

ポリラインの点を移動します。

- (1) 図形を選択します（共通：図形を選択。単一選択のみ）。

- (2) マウスを移動します。

ポインタにあわせて点が色替え表示されます。

- (3) 移動する点の色が変わったらLボタンクリックします。

ラバーバンドを表示します。

- (4) マウスを移動し、移動したい位置でLボタンクリックします。

ポリラインの点を移動します。

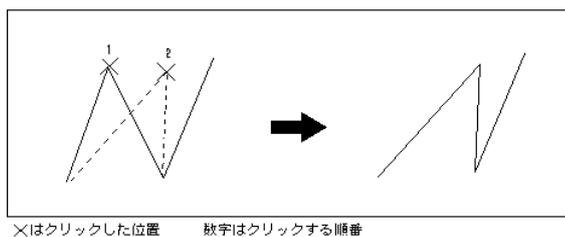


図 2-22 点の移動

2.4 標準部品カスタマイズ方法

st_operation の各アクションをカスタマイズする方法について説明します。

2.4.1 カスタマイズの考え方

st_operation のカスタマイズの基本的な考え方を説明します。

(1) st_operation の実現方式

st_operation は、次の 3 つのイベントに処理を割り当てて、図形入力、図形編集、画面制御を実現しています。

① マウスクリック・イベント

キャンバス上でマウスクリックが行われたときに発生するイベントです。

② マウスモーション・イベント

キャンバス上でマウスポインタが移動したときに発生するイベントです。

③ 最大クリック回数イベント

キャンバス上でのマウスクリックが指定回数に達したときに発生するイベントです。

これらのイベントに対応する処理は、外部変数 stval () の配列で管理されます。

st_operation が呼び出されると、外部変数 stval () に、各イベントで実行する関数（手続き）として、標準部品の内部マクロがセットされます。同時に、各アクションで使用する内部データの初期値も stval () に設定されます。

マウス操作によってイベントが発生すると、外部変数 stval () が参照され、イベントに対応した処理が実行されます。

外部変数 stval () の内容は、st_reset が実行されるまで保持されます。

(2) カスタマイズの方法

3つのイベントに割り当てられた処理（標準部品内部マクロ）を、ユーザー作成の関数（手続き）に置き換えるか、または追加することによって、標準部品を独自の機能にカスタマイズできます。これは、st_operation を実行後、外部変数 stval () の内容を変更することで行います（カスタマイズ例）。

また、外部変数 stval () を変更することによって、たとえばラバーバンド表示色の変更などのカスタマイズも行えます。

さらに、ユーザー独自のアクションを st_operation に追加することもできます。これは、アクションを管理する外部変数 stval (opetype) にアクションの定義を追加することで行います。これにより st_operation で、他のアクションと同様に、独自のアクションを実行できるようになります（アクションごとの外部変数設定内容）。

2.4.2 カスタマイズ例

カスタマイズの例を示します。

(1) 直線描画 (line) の実現方式

① 使用する主な外部変数

使用する主な外部変数は次のとおりです。

- (a) `stval(stid, pcom)` : マウスクリック・イベントにより実行するマクロ
- (b) `stval(stid, mcom)` : マウスモーショント・イベントにより実行するマクロ
- (c) `stval(stid, ecom)` : 最大クリック回数イベントにより実行するマクロ
- (d) `stval(stid, ucom)` : `st_operation`の引数`eproc`で指定されたユーザー関数
- (e) `stval(stid, pmax)` : `ecom`イベントを発生するマウスクリック数
- (f) `stval(stid, pnum)` : マウスクリック数のカウンタ
- (g) `stval(stid, gpoint)` : マウスクリックされた座標値(X Y X Y…)

② 変数の設定値

各変数の初期設定値は次のとおりです。

- (a) `stval(stid, pcom)` : `st_p_point`
- (b) `stval(stid, mcom)` : `st_m_line`
- (c) `stval(stid, ecom)` : `st_e_line`
- (d) `stval(stid, ucom)` : `eproc`で指定されたユーザー関数
- (e) `stval(stid, pmax)` : 2
- (f) `stval(stid, pnum)` : 0
- (g) `stval(stid, gpoint)` : "" (空の文字列)

③ 処理の流れ

`st_operation` 実行時の処理の流れは次のとおりです。

- (a) 1点目のマウスクリック・イベントによって、`st_p_point`が実行されます。
`st_p_point`では、クリックした座標値を`stval(stid, gpoint)`に格納して、`stval(stid, pnum)`に1を加え、`stval(stid, pmax)`と比較します。
- (b) マウスモーショント・イベントによって、`st_m_line`が実行されます。
`st_m_line`では、`stval(stid, gpoint)`の座標値と、現在のポインタ位置を結ぶ線を描画します(ラバーバンド描画)。

※ 1点目のマウスクリック前にも、ポインタがキャンバス上を移動することによりマウスモーショント・イベントが発生しますが、`stval(stid, gpoint)`が空の場合、線は描画しません。

- (c) 2点目のマウスクリック・イベントによって、`st_p_point`が実行され、1点目と同様に、`stval(stid, gpoint)`への座標値格納と`stval(stid, pnum)`のインクリメントが行われます。

`stval(stid, pnum)`が`stval(stid, pmax)`と比較して、クリック回数が最大値に達した時、`st_p_point`から`st_e_line`を実行します。

- (d) `st_e_line`では、ラバーバンドを消去して線を確定し、始点、終点座標をセットして`stval(stid, ucom)`のユーザー関数を実行します。

`stval(stid, gpoint)`、`stval(stid, pnum)`はクリアします。

(2) カスタマイズ的具体例

シンボルの中心を結ぶ直線を描画するようにカスタマイズします。

このためマウスモーション・イベントとマウスクリック・イベントのマクロに、最近点検索処理を行って、シンボルが検索されたときにその中心座標をセットする処理を追加します。これによりクリックされた座標ではなく、選択されたシンボルの中心点が直線の始点、終点となるようにします。

① カスタマイズ内容

`pcom` と `mcom` に割り付けるユーザー関数を作成します。

- (a) `ww_pole_set` : `pcom` に追加

最近点検索で選択した図形オブジェクトをダンプ (`g_dump_obj`) し、中心点の座標を `stval(stid, x)`、`stval(stid, y)` にセットします。

- (b) `ww_check_zuobj` : `mcom` に追加

最近点検索した図形オブジェクトがシンボルか否かをチェックします。

`st_operation` をコールしてから `pcom`、`mcom` に、ユーザー関数および最近点検索の標準部品内部マクロを追加します。

② カスタマイズ方法

リスト 2-1 ww_pole_set のカスタマイズ例

```
[ww_pole_set の定義]
#シンボルの中心座標をセット
proc ww_pole_set { stid } {
    global stval

    set zudt [lindex [g_dump_obj $stval($stid, anydata)] 0]

    set stval(x) [lindex $zudt 1]
    set stval(y) [lindex $zudt 2]
}
```

リスト 2-2 ww_check_zuobj のカスタマイズ例

```
[ww_check_zuobj の定義]

#図形チェック
proc ww_check_zuobj { stid } {
    global stval

    set zudt [lindex [g_dump_obj $stval($stid, anydata)] 0]

    if { [lindex $zudt 0] != "symbol" } {
echo <ww_check_zuobj> symbol 以外は対象外
#anydata に olddata をセットすることによりシンボル以外は選択されません
        set stval($stid, anydata) $stval($stid, olddata)
    }
}

[st_operaton の実行]

#st_operation をコールする (機能名は L I N E)
    st_operation $stid $gc $gwin $dlist line eproc

#pcom にシンボル座標セット関数を追加
    set stval($stid, pcom) [concat ww_pole_set $stid st_p_point $stid]

#mcom に最近点検索／図形チェック関数を追加
    set stval($stid, mcom) [concat st_m_nearest $stid ¥; ww_check_zuobj $stid ¥; ¥
        st_m_chgcol $stid ¥; $stval($stid, mcom) ]
```

2.4.3 外部変数一覧

st_operation の処理に使用する外部変数の種類と意味を説明します。

標準部品で使用する外部変数はstval()の配列で管理します。さらにstid(標準部品ID)による2次元配列で管理する情報もあります。stval()の配列の添字(要素名)と内容を以下に示します。

表 2-29 標準部品外部変数一覧

変数要素名	内容
x	マウスクリック・イベント、マウスモーション・イベント発生時のカーソル X 座標 (座標は DC から WC に変換済み)
y	マウスクリック・イベント、マウスモーション・イベント発生時のカーソル Y 座標 (座標は DC から WC に変換済み)
b	マウスクリック・イベント発生時のボタン種別 (l : 左, m : 中, r : 右)
h	イベント発生箇所 (p : マウスクリック, m : マウスモーション)
stids	オープン中の標準部品 ID 群
rubber_color	ラバーバンド表示の描画色および、最近点検索での図形オブジェクト表示色 (色 ID で指定)
opetype	標準部品アクション初期設定値
stid, pnum	マウスクリック数
stid, pmax	マウスクリック数の最大値
stid, gpoint	マウスクリックにより格納した座標群 (X Y X Y...)
stid, param	ラバーバンド描画時のワーク変数
stid, anydata	現在検索中の図形オブジェクト ID
stid, olddata	検索後の図形オブジェクト ID (最近点色を元に戻す時に使用)
stid, zuobjjs	マウスクリックにより格納した図形オブジェクト ID 群
stid, rubber	ラバーバンド表示、最近点検索中に 1 をセット (st_reset の処理で、このフラグが 1 のときに色をリセットする)
stid, ope	st_operation でセットした action の機能名
stid, pcom	マウスクリックによるイベントで実行する関数 (手続き)
stid, mcom	マウスモーション (移動) によるイベントで実行する関数 (手続き)
stid, ecom	マウスクリックが指定回数に達した時のイベントで実行する関数 (手続き)
stid, ucom	st_operation でセットした後処理用ユーザー関数 (eproc)
stid, gc	st_operation でセットしたコンテキスト ID
stid, gw	st_operation でセットしたグラフィックスウィンドウ ID
stid, dl	st_operation でセットしたディスプレイリストメモリ ID
stid, filter	図形編集処理で使用するフィルタ ID

2.4.4 アクションごとの外部変数設定内容

st_operation のアクションごとに、外部変数の設定内容を示します。

アクションごとの外部変数の初期設定は st_op.psv のマクロ内の stval(opetype) で管理しています。アクションごとにリスト構造になっており opetype (機能名)、pcom、mcom、ecom、pmax の5つの項目があります。

st_operation の処理中で、機能名から値を検索して、それぞれ対応する外部変数に初期設定します。

また、ユーザー独自のアクションを stval(opetype) に追加することにより、ユーザー独自の機能を実現できます (アクション追加の例)。

※ ユーザー独自のアクションを定義する場合、そのアクションで使用するユーザー定義関数は、アクションを呼び出す前に定義 (proc) しておく必要があります。

(1) stval(opetype) の構造

リスト 2-3 stval(opetype) のコード

```
#標準部品オペレーション定数
# opetype pcom mcom ecom pmax
set stval(opetype) {
#---- 図形入力 -----
    { line {st_p_point $stid} {st_m_line $stid} {st_e_line $stid} 2 }
    { polyline {st_p_point $stid ; st_p_poly $stid} {st_m_line $stid}
      {st_e_polyline $stid} 2 }
    { polygon {st_p_point $stid ; st_p_poly $stid} {st_m_line $stid}
      {st_e_polyline $stid} 2 }
    { spline {st_p_point $stid ; st_p_poly $stid} {st_m_line $stid}
      {st_e_polyline $stid} 2 }
    { rect {st_p_point $stid} {st_m_rect $stid} {st_e_rect $stid} 2 }
    { circle {st_p_point $stid} {st_m_circle $stid} {st_e_circle $stid} 2 }
    { arc {st_p_point $stid} {st_m_circle $stid} {st_e_arc1 $stid} 2 }
    { ellipse {st_p_point $stid} {st_m_ellipse $stid}
      {st_e_ellipse $stid} 2 }
    { text {st_p_point $stid} {st_m_rect $stid} {st_e_text1 $stid} 2 }
    { freeline {st_p_point $stid} {st_m_freeline $stid}
      {st_e_freeline $stid} 2 }
    { roundbox {st_p_point $stid} {st_m_rect $stid} {st_e_roundbox1 $stid}
2 }
```

```
        { regpoly {st_p_point $stid} {st_m_regpoly $stid $targ}
          {st_e_regpoly $stid} 2 }
#---- 画面制御 -----
        { nearest {st_p_data $stid ; st_p_point $stid}
          {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_nearest $stid}
1 }
        { enlarge {st_p_point $stid} {st_m_rect $stid} {st_e_chgview $stid {} }
2 }
        { shrink {st_p_point $stid} {st_m_rect $stid} {st_e_chgview $stid -s } 2 }
        { scroll {st_p_point $stid} {st_m_scroll $stid} {st_e_scroll $stid} 2 }
        { print {st_p_point $stid} {st_m_rect $stid} {st_e_print $stid $targ }
2 }
#---- 図形編集 -----
        { move_obj {st_p_data $stid ; st_p_point $stid}
          {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
        { move_obj_h {st_p_data $stid ; st_p_sel $stid}
          {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
        { move_obj_r {st_p_point $stid} {st_m_rect $stid}
          {st_e_edit_objr $stid } 2 }
        { del_obj {st_p_data $stid ; st_p_point $stid}
          {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
        { del_obj_h {st_p_data $stid ; st_p_sel $stid}
          {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
        { del_obj_r {st_p_point $stid} {st_m_rect $stid} {st_e_edit_objr $stid }
2 }
        { transform_obj {st_p_data $stid ; st_p_point $stid}
          {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
        { transform_obj_h {st_p_data $stid ; st_p_sel $stid}
          {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
```

```
{ transform_obj_r {st_p_point $stid} {st_m_rect $stid}
    {st_e_edit_objr $stid } 2 }
{ rotate_obj {st_p_data $stid ; st_p_point $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
{ rotate_obj_h {st_p_data $stid ; st_p_sel $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
{ rotate_obj_r {st_p_point $stid} {st_m_rect $stid}
    {st_e_edit_objr $stid } 2 }
{ copy_obj {st_p_data $stid ; st_p_point $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
{ copy_obj_h {st_p_data $stid ; st_p_sel $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
{ copy_obj_r {st_p_point $stid} {st_m_rect $stid}
    {st_e_edit_objr $stid } 2 }
{ resize_obj {st_p_data $stid ; st_p_point $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
{ resize_obj_h {st_p_data $stid ; st_p_sel $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
{ resize_obj_r {st_p_point $stid} {st_m_rect $stid}
    {st_e_edit_objr $stid } 2 }
{ poly_edit_a {st_p_data $stid ; st_p_point $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
{ poly_edit_x {st_p_data $stid ; st_p_point $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
{ poly_edit_r {st_p_data $stid ; st_p_point $stid}
    {st_m_nearest $stid ; st_m_chgcol $stid} {st_e_edit_obj1 $stid}
1 }
}
```

(2) アクション追加の例

カスタマイズ例で示した「シンボルの中心を結ぶ線を引く」アクションを定義します。

標準部品のロード (source st_op.psv) を行った後、stval (opetype) に、追加する機能の初期設定値をセットします。

リスト 2-4 アクション追加の例

```
set stval(opetype) [concat $stval(opetype)
    { sym_line {ww_pole_set $stid ; st_p_point $stid}
    {st_m_nearest $stid ; ww_check_zuobj $stid ; st_m_chgcol $stid ;
st_m_line $stid}
    {st_e_line $stid } 2 ]
```

上記のカスタマイズを行うことにより、st_operation の機能名に“sym_line”を指定するだけで、シンボルの中心から線を引く機能を実現できます。

```
st_operation $stid $gc $gwin $dlist sym_line eproc
```

3 WebTcl FE-COM API

3.1 WebTcl FE-COM API 概要

FE-COM は Internet Explorer (以下 IE) のアドオンとして動作し、AP サーバとの通信のフロント・エンドの役割を行う COM オブジェクトです。

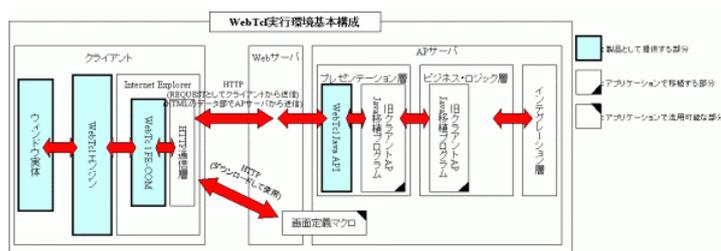


図 3-1 WebTcl 実行環境基本構成

FE-COM では以下の機能を提供します。これらの機能は FE-COM オブジェクトのメソッド・プロパティから制御します。

(1) 初期起動マクロ実行までの初期処理

WebTcl エンジンの起動、FE-COM. InitSource プロパティに URL 指定される初期起動マクロのダウンロードと実行を行います。

(2) HTTP 通信の代行

WebTcl エンジンとプロセス間通信により密に連携し、アプリケーションへのリクエスト送信と、返信の実行マクロの受信を行います。FE-COM が通信を代行するのは、IE と同一プロセスで動作し、IE の HTTP 通信層を共用することで、通信処理の統合を行うためです。

(3) IE との関係

FE-COM は「ウィンドウ付きコントロール」としての特徴があり、IE 上にベースとなるウィンドウを作成します。このウィンドウは一部のウィンドウ資源参照にも使用されます。このウィンドウ上に、w_top のウィンドウを -a オプションで貼り付けることも可能です (図 3-2 は初期起動マクロのメニュー画面を統合した例)。

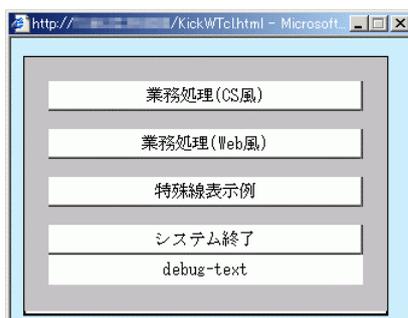


図 3-2 メニュー画面を IE に統合した例

また JavaScript からのマクロ実行も可能で、HTML からレイヤ操作などを行うことも可能です（例は配列に入ったレイヤ番号のレイヤ属性を変更する例）。

例)	<pre>for (lpcnt = 0 ; lpcnt < cnt ; lpcnt++) { if (mode == 1) { parent.wtcl_body.WTclIFCom1.execCmd("g_layer_attrib \$map2(gc) " + layerNo[lpcnt] + " " + attrib + "; g_clear \$map2(gc); g_draw_obj \$map2(gc)"); } else if (mode == 2) { parent.wtcl_body.WTclIFCom1.execCmd("g_layer_attrib \$map(gc) " + layerNo[lpcnt] + " " + attrib + "; g_clear \$map(gc); g_draw_obj \$map(gc)"); } }</pre>
----	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

3.2 WebTcl FE-COM API リファレンス

WebTcl FE-COM API のリファレンスです。FE-COM で使用可能なメソッド、プロパティ、コールバックを以下に示します。

表 3-1 FE-COM メソッド・プロパティ・コールバック一覧

No.	種別	名称	概要	必須
1	メソッド	execCmd	外部マクロ実行メソッド	—
16		init_e_external_send	e_external_send マクロ用初期化メソッド	—
2	プロパティ	AskRequestURL	サーバ要求自動問合せ URL プロパティ	—
3		CertRetryCount	認証リトライ回数指定プロパティ	—
4		CodeBase	HTTP コードベース URL プロパティ	○
5		DebugPolicy	デバッグ・ポリシー指定プロパティ	—
6		EndActionURL	AP サーバ切断 URL プロパティ	○
7		EndModifyString	EndActionURL のリクエスト時の後方修飾文字列指定プロパティ	—
8		InitActionURL	AP サーバ接続 URL プロパティ	○
9		InitModifyString	InitActionURL のリクエスト時の後方修飾文字列指定プロパティ	—
10		InitSource	初期起動マクロ指定プロパティ	○
11		ModifyString	後方修飾文字列指定プロパティ	—
12		TclVersionOption	Tcl 互換モードオプション指定プロパティ	—
13		TimeoutExecCmd	通信タイムアウト実行マクロ指定プロパティ	—
14		TimeoutExecInterval	タイムアウト時マクロ間隔指定プロパティ	—
15		TimeoutVal	通信タイムアウト時間指定プロパティ	—
17		TclIndexURL	インデックスファイル名プロパティ	—
18		ActionCodeBase	アクションマップ加工用 URL 指定プロパティ	—
19		DuplicationErrorMessage	重複起動エラーメッセージ指定プロパティ	—
20		HideMode	非表示モード指定プロパティ	—
21		StandardRecvFlag	標準 e_recv サポートフラグ	—

3.3 FE-COM メソッド詳細

3.3.1 execCmd

名 称	外部マクロ実行メソッド				
コーディング・シケンス	ret = objectid.execCmd(cmd, sync)				
機 能	<p>外部マクロ実行メソッド</p> <p>起動用 HTML の JavaScript から FE-COM や、WebTcl エンジンへメッセージを送る際に使用します。</p> <p>cmd には実行するマクロコマンドを設定します。</p> <p>同期・非同期フラグを sync に設定します。</p> <p>sync フラグで同期に設定した場合はコマンドの実行結果を ret に戻します。</p>				
入出カインターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	cmd	可変	文字列	
	[1/]	sync	可変	文字列	同期 : true 非同期 : false
	戻り値	ret	可変	文字列	

3.3.2 init_e_external_send

名 称	e_external_send マクロ用初期化メソッド				
コーディング・シケンス	ret = objectid.init_e_external_send()				
機 能	e_external_send マクロ実行前に、FE-COM と JavaScript 間の接続を確立し、通信可能にします。				
入出カインターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	戻り値	ret	4 バイト	Long	0:初期化成功 1:初期化失敗

3.4 FE-COM プロパティ詳細

3.4.1 AskRequestURL

名 称	サーバ要求自動問合せ URL プロパティ				
コーディング・シケンス	object id. AskRequestURL="URL"				
機 能	サーバ要求自動問合せ URL プロパティ ・URL には e_recv でサーバ要求の問合せを送信するサーブレットのアドレスを指定します。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	URL	可変	文字列	

3.4.2 CertRetryCount

名 称	認証リトライ回数指定プロパティ				
コーディング・シケンス	object id. CertRetryCount="Count"				
機 能	認証リトライ回数指定プロパティ ・ Count には認証ダイアログで認証失敗時のリトライ回数を指定します。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	Count	可変	文字列	正の値が有効。 デフォルト 10

3.4.3 CodeBase

名 称	HTTP コードベース URL プロパティ				
コーディング・シケンス	object id. CodeBase="URL"				
機 能	HTTP コードベース URL プロパティ ・ URL にはマクロファイルを格納している Web 上のディレクトリの位置を指定します。 この URL をベースとして、マクロファイルの相対ディレクトリからマクロを配置している URL を合成します。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	URL	可変	文字列	

3.4.4 DebugPolicy

名 称	デバッグ・ポリシー指定プロパティ				
コーディング・シケンス	objectid.DebugPolicy="Number"				
機 能	デバッグ・ポリシー指定プロパティ ・エラー時のデバッグ情報の出力方式を指定します。 0:出力なし 1:追加形式ファイル出力あり 2:逐次出力ダイアログ表示あり 3:追加形式ファイル出力+逐次出力ダイアログ表示あり 注) FE-COM の致命的エラーが発生した場合、未設定・設定値に関わらず、 設定値:3 の動作を行います。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	Number	可変	文字列	0~3 の値が有効。 デフォルト 0

3.4.5 EndActionURL

名 称	AP サーバ切断 URL プロパティ				
コーディング・シケンス	objectid.EndActionURL="URL"				
機 能	AP サーバ切断 URL プロパティ ・ URL には e_close で切断要求を送信するサーブレットの URL を指定します。 ・ 切断要求は IE 強制終了時にも実行します。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	URL	可変	文字列	

3.4.6 EndModifyString

名 称	EndActionURL リクエスト時の後方修飾文字列指定プロパティ				
コーディング・シケンス	objectid.EndModifyString = "String"				
機 能	EndActionURL リクエスト時の後方修飾文字列指定プロパティ ・ AP サーバに切断要求送信 (e_close) する際に URL の後ろに付加する文字列を設定します。 ・ 本プロパティを使用する場合は e_close を行う前に設定してください。 ・ 後方修飾文字列 (String) は以下のクエリー文字列形式で URL に付加します。 URL?String ・ 後方修飾文字列の設定は、先頭 '?' がある場合、ない場合何れも使用可能とする。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	String	可変	文字列	

3.4.7 InitActionURL

名 称	AP サーバ接続 URL プロパティ				
コーディング・シケンス	objectid.InitActionURL="URL"				
機 能	AP サーバ接続 URL プロパティ ・ URL には e_connect で接続するサーブレットの URL を指定してください。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	URL	可変	文字列	

3.4.8 InitModifyString

名 称	InitActionURL リクエスト時の後方修飾文字列指定プロパティ				
コーディング・シケンス	objectid. InitModifyString = "String"				
機 能	<p>InitActionURL リクエスト時の後方修飾文字列指定プロパティ</p> <ul style="list-style-type: none"> ・ AP サーバに接続 (e_connect) する際に URL の後ろに付加する文字列を設定します。 ・ 本プロパティを使用する場合は e_conect を行う前に設定してください。 ・ 後方修飾文字列 (String) は以下のクエリー文字列形式で URL に付加します。 URL?String ・ 後方修飾文字列の設定は、先頭 '?' がある場合、ない場合何れも使用可能とします。 				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	String	可変	文字列	

3.4.9 InitSource

名 称	初回起動マクロ指定プロパティ				
コーディング・シケンス	objectid. InitSource = "URL"				
機 能	<p>初回起動マクロ指定プロパティ</p> <ul style="list-style-type: none"> ・ URL には初回起動用マクロファイルのアドレスを指定します。 ・ URL にローカルファイルを指定する場合、指定方法は以下のとおりとします。 file://ファイル名 (絶対パス) 注) ローカルファイルの指定時ディレクトリの区切りは '/' で指定。 ・ 本プロパティの設定は、プロパティ設定の最後に指定してください。 				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	URL	可変	文字列	

3.4.10 ModifyString

名 称	後方修飾文字列指定プロパティ				
コーディング・シケンス	objectid.ModifyString = "String"				
機 能	後方修飾文字列指定プロパティ ・ AP サーバに要求送信 (e_send、e_recv) する際に URL の後ろに付加する文字列を設定します。 ・ 本プロパティを使用する場合は e_send、e_recv を行う前に設定してください。 ・ 後方修飾文字列 (String) は以下のクエリー文字列形式で URL に付加します。 URL?String ・ 後方修飾文字列の設定は、先頭 '?' がある場合、ない場合何れも使用可能とする。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	String	可変	文字列	

3.4.11 TclVersionOption

名 称	Tcl 互換モードオプション指定プロパティ				
コーディング・シケンス	objectid.TclVersionOption = "-A" "-Z"				
機 能	Tcl 互換モードオプション指定プロパティ ・ 下位バージョンのマクロ命令をエミュレーションする互換モードを指定します。 "-A" 指定時は A 版互換モードで動作します。 "-Z" 指定時は A 版統合モードで動作します。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	-A	可変	文字列	A 版互換モード、A 版統合モードで動作する場合に設定します。 本プロパティ未設定時は、V5 版標準モードで動作します。

3.4.12 TimeOutExecCmd

名 称	通信タイムアウト実行マクロ指定プロパティ				
コーディング・シケンス	objectid.TimeOutExecCmd = "Macro"				
機 能	通信タイムアウト実行マクロ指定プロパティ ・ TimeOutVal 秒通信処理がなかった場合、実行する Tcl マクロを指定します。 ・ 上記 Tcl マクロ実行後、TimeOutVal (TimeOutExecInterval 指定なし時) or TimeOutExecInterval (TimeOutExecInterval 指定あり時) 秒後までに通信処理がなかった場合、再度本マクロを実行します。 主に、通信タイムアウトの警告や、データの自動保存目的に使用します。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	Macro	可変	文字列	

3.4.13 TimeOutExecInterval

名 称	タイムアウト時マクロ間隔指定プロパティ				
コーディング・シケンス	objectid.TimeOutExecInterval = "Second"				
機 能	タイムアウト時マクロ間隔指定プロパティ ・ TimeOutExecCmd プロパティで指定するマクロの2回目以降の実行間隔を TimeOutVal から変更したいときに指定します。 途中で通信処理が行われた場合、秒数のカウントをリセットします。				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	Second	可変	文字列	

3.4.14 TimeOutVal

名 称	通信タイムアウト時間指定プロパティ				
コーディング・シケンス	objectid.TimeOutVal = "Second"				
機 能	<p>通信タイムアウト時間指定プロパティ</p> <ul style="list-style-type: none"> ・通信終了後、及び、通信タイムアウト発生後、本プロパティで指定する秒数の間に通信がなかった場合、TimeOutExecCmd プロパティで指定するマクロを実行します。 <p>途中で通信処理が行われた場合、秒数のカウントをリセットします。</p>				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	Second	可変	文字列	

3.4.15 TclIndexURL

名 称	インデックスファイル名プロパティ				
コーディング・シケンス	objectid.TclIndexURL = "URL"				
機 能	<p>インデックス機能で利用する、インデックスファイルのロケーションを、URL形式で指定します。</p> <p>本プロパティで指定するインデックスファイルは、プロパティ設定直後はFE-COM 内部に値を格納するだけです。</p> <p>インデックス読込処理自体は、InitSource プロパティの設定時(内部的にsetterの関数が実行される時)に、初期処理として一括して実行されます。</p> <p>インデックスファイルの生成に関しては、支援ツールとしてcreateIndex.psv (PreSerV C/S 版上で実行)を提供します。</p> <p>インデックス機能の設定と動作に関しては、4.1.1インデックス生成コマンド(createIndex)を参照してください。</p>				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	URL	可変	文字列	http://から始まる完全なURL形式の文字列を指定します。

3.4.16 ActionCodeBase

名 称	アクション・マップ加工用 URL 指定プロパティ				
コーディング・トークス	objectid. ActionCodeBase = "URL"				
機 能	<p>アクション・マップ加工の際に使用するコードベース URL を指定します。</p> <p>SSO を利用して URL の書き換えを行うシステムに WebTcl を適用する場合は、本プロパティの設定が必要です。</p> <p>本プロパティを設定しない場合は、InitActionURL のホスト名+ポート番号をコードベースとして使用します。</p>				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	URL	可変	文字列	

3.4.17 DuplicationErrorMessage

名 称	重複起動エラーメッセージ指定プロパティ				
コーディング・トークス	objectid. DuplicationErrorMessage = "String"				
機 能	<p>重複起動時のエラーメッセージを指定します。</p> <p>"¥r¥n"で改行、"¥t"でタブを指定することができます。</p> <p>指定できる文字列は 1024 バイトまでで、それより大きな文字列を指定した場合は動作の保障ができません。</p> <p>本プロパティを設定しない場合に重複起動すると、デフォルトのメッセージ "WebTcl の重複起動はできません。"を表示します。</p>				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	String	可変	文字列	

3.4.18 HideMode

名 称	非表示モード指定プロパティ				
コーディング・シケンス	objectid.HideMode = "Number"				
機 能	<p>起動 HTML の非表示モードを指定します。</p> <p>0: IE ウィンドウを表示します。</p> <p>1: IE ウィンドウをタスクバーにアイコン化して表示します。</p> <p>2: IE ウィンドウを通知領域にアイコン化して表示します。</p> <p>上記の 0~2 以外の値を指定した場合は、エラーメッセージを表示して、デフォルト"0"の動作をします。</p> <p>非表示モードが 1, 2 の場合には、w_top コマンドの、ウィンドウの統合機能が使用できません。</p>				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	Number	可変	文字列	0~2 の値が有効。 デフォルト 0

3.4.19 StandardRecvFlag

名 称	標準 e_recv サポートフラグ指定プロパティ				
コーディング・シケンス	objectid.StandardRecvFlag = "Number"				
機 能	<p>e_recv の通信効率化に伴い、従来と同様に e_recv のポーリング処理を行う場合は、このプロパティに" 1" を設定してください。</p> <p>1: 従来同様一定時間ごとに、FE-COM#AskRequestURL プロパティで指定されるアクションへ POST 動作を繰り返します。</p> <p>1 以外の値 : e_recv で受信する内容を e_send のレスポンスで行い、FE-COM#AskRequestURL プロパティで指定されるアクションへ POST 動作を行いません。</p> <p>本プロパティ未設定時は、" 0" を指定したとみなし、1 以外の値を指定した場合の動作をします。</p>				
入出インターフェース	入出力	変数シンボル	サイズ	タイプ	概要
	[1/]	Number	可変	文字列	1 かそれ以外の値 デフォルト" 0"

3.5 WebTcl FE-COM API エラーメッセージ

WebTcl FE-COM API が出力するエラーメッセージを説明します。

3.5.1 エラーメッセージ取得方法

FE-COM API では“DebugPolicy”プロパティの値に 1~3 を設定することで、エラーメッセージの出力が可能となっています。エラーメッセージの出力方法は以下の 2 種類の方法があります。

(1) 逐次ダイアログ表示

エラーの度にダイアログでメッセージを表示します。以下にダイアログの出力例を示します。



図 3-3 逐次ダイアログ表示例

(2) ファイル出力

エラー情報をファイル (WebTclError.log) に追加形式で出力します。

WebTclError.log は環境変数“TEMP”ディレクトリに格納されます。

環境変数“TEMP”が存在しない場合、環境変数“TMP”を探し、両方とも見つからない場合、WebTclError.log へのログの出力は行いません。

以下にログの出力例を示します。

```
Sun Sep 30 17:19:55 2006 [invalid command name "w_frame"
while executing
"w_frame frame7 $map(p_dig) -x 20 -y 10 -w 360 -h 70 -v m6
-l "プリンタ名"
(file "GyoumuMain.psv" line 346)
invoked from within
"source GyoumuMain.psv"]
```

3.5.2 エラーメッセージ一覧

FE-COM API が出力するエラーメッセージの一覧を以下に示します。

Tcl 実行時のエラーはTclのエラーメッセージ一覧を参照してください。

表 3-2 エラーメッセージ一覧

エラーメッセージ	エラー内容	対処法	致命的エラー	エラー後処置	備考
WTFE0033 error: command error	コマンド不正	execCmd に入力したコマンドの確認を行うこと。	—	処理継続	
WTFE0033 error: tcl send error	Tcl メッセージ送信失敗	WebTclServer との接続エラーを確認すること。	—	処理継続	
WTFE0033 error: message message	Tcl メッセージ QUIT	quit の使用を止めること。	—	処理継続	
WTFE0033 error: e_connect error	AP サーバコネクタ失敗	プロパティ (起動 url) の確認、AP サーバの状態確認を行うこと。	—	処理継続	
WTFE0012 error: e_close error	AP サーバ切断失敗	AP サーバの状態確認を行うこと。	○	強制終了	
WTFE0014 error: e_error error	e_error 失敗	エラーとなった時に実行するマクロの登録に失敗しています。 *発生要因はメモリの確保エラー時。	○	強制終了	
WTFE0009 error: cd error	cd 失敗	アクセス不能 URL への移動を行いました。	○	処理継続	
WTFE0010 error: pwd error	pwd 失敗	カレント URL が未設定。	○	処理継続	
WTFE0033 error: e_send error	e_send エラー	接続状態、送信内容を確認すること。	—	処理継続	
WTFE0020 error: e_send error [url]	e_send エラー	url を確認すること。	○	処理継続	
WTFE0033 error: e_recv error	e_recv エラー	起動用 HTML のプロパティのコードベースを確認すること。	—	処理継続	
WTFE0017 error: time out	タイムアウト	AP サーバ、又は EG からの応答がない。 AP サーバ、又は EG の状態を確認すること。	○	強制終了	

エラーメッセージ	エラー内容	対処法	致命的エラー	エラー後処置	備考
WTFE0033 error: file"fileName" not found	ファイルなし	ファイルの有無、アクセス権を確認すること。	—	処理継続	
WTFE0001 error: create thread error	スレッド作成失敗	メモリ状態を確認すること。	○	強制終了	
WTFE0013 error: Unsetting Init ActionURL	プロパティ(起動 url 未設定)	起動用 HTML のプロパティを確認すること。	○	強制終了	
WTFE0033 error: run InitSource error	プロパティ(起動 url 未設定)	起動用 HTML のプロパティを確認すること。	—	処理継続	
WTFE0015 error: Unsetting Recv URL	プロパティ (recv url 未設定)	起動用 HTML のプロパティを確認すること。	○	強制終了	
WTFE0021 error: WebTclServer start up error	WebTclServer 起動失敗	System32 ディレクトリに WebTclServer.exe がダウンロードしていることを確認すること。 System32 ディレクトリに WebTclServer.exe が存在することが確認できた場合、System32 までのパスを環境変数に設定していることを確認する。	○	強制終了	
WTFE0002 error: connect WebTcl Server port[portNo]	WebTclServer 接続失敗	FE と EG との接続失敗。 portNo が既に使用していないか確認すること。	○	強制終了	
WTFE0025 error: connect APServer error	AP サーバ接続失敗	プロパティ(起動 url)の確認、AP サーバの状態確認を行うこと。	○	強制終了	
WTFE0026 error: connect Internet	インターネット接続失敗	インターネットの接続環境を確認すること。	○	強制終了	

エラーメッセージ	エラー内容	対処法	致命的エラー	エラー後処置	備考
WTFE0016 error: Unconnection server	AP サーバ未接続	e_connect で接続済みの状態で、e_recv を実行してください。	○	強制終了	
WTFE0023 error: request error	不正リクエスト	リクエストの送信方法が間違っている。 使用できるリクエスト方法は POST GET のみとする。	○	強制終了	
WTFE0027 error: create request error[errorCode]	リクエスト作成失敗	errorCode を確認すること。	○	強制終了	※1
WTFE0028 error: create header error[errorCode]	ヘッダ作成失敗	errorCode を確認すること。	○	強制終了	※1
WTFE0029 error: send request error[errorCode]	リクエスト送信失敗	errorCode を確認すること。	○	強制終了	※1
WTFE0030 error: get status error[errorCode][url]	リクエスト送信失敗	errorCode と url を確認すること。	○	強制終了	※1
WTFE0032 error: status code [HTTPStatusCode][url]	リクエスト送信失敗	HTTPStatusCode を確認すること。	—	処理継続	※1
WTFE0024 error: unknown url[url]	url 不明	url を確認すること。	○	処理継続	
WTFE0005 error: alloc error	メモリ確保エラー	メモリ状態を確認すること。	○	強制終了	
WTFE0031 error: sock timeout	ソケットタイムアウト(同期時)	ソケットからの応答がない。 送信したコマンドを確認すること。	○	強制終了	

エラーメッセージ	エラー内容	対処法	致命的エラー	エラー後処置	備考
WTFE0006 error: Unknown TclVersionOption	A 版互換オプション指定不正	A 版互換オプションの指定は“-A”のみを指定すること。 指定不正の場合、A 版互換オプションは設定しません。V5 標準モードで動作を行います。	○	強制終了	
WTFE0003 インデックスファイル[%s]のダウンロードに失敗しました。インデックス機能は動作しません。	インデックスファイル指定不正	FE-COM#TclIndexURL プロパティの設定を確認願います。	○	処理継続	
WTFE0004 インデックスファイル[%s]は書式不正です。インデックス機能は動作しません。	インデックスファイル書式不正	インデックスファイルの書式を確認願います。	○	処理継続	
WTFE0007 内部通信用の未定義コマンド専用をそのまま出力	WebTcl エンジン-FE-COM 間内部プロトコルエラー	基本モジュールのインストールが正常に行われているか確認願います。	○	強制終了	
WTFE0008 error: %ld	予測しないエラー	エラー値と WebTcl の動作設定、エラー発生時の状況を確認し、製品問合せ窓口へ連絡ください。	○	強制終了	
WTFE0018 error: unsetting uri<%s>	サブレット未登録	アクションキーワードと対応づいた URL の一部のパス名を定義するか、デフォルト値を登録してください。	○	処理継続	
WTFE0019 WebTcl 以外の通信を受信しました。処理を終了します。	e_send レスポンス書式不正	現在確認しているケースとしては SSO の通信タイムアウトがあります。 再度 SSO ログイン後処理を継続してください。	○	強制終了	

エラーメッセージ	エラー内容	対処法	致命的エラー	エラー後処置	備考
WTFE0022 error: Unsetting EndActionURL	EndActionURL 未設定	FE-COM#EndActionURL プロパティの設定を確認願います。 またアプリケーションの処理シーケンスを確認し、本来不要な e_connect 処理が実行されていないか確認願います。	○	強制終了	
WTFE0011 error: Unconnection server	AP サーバ未接続	e_close、e_send は e_connect 後発行してください。	○	強制終了	

※1：出力のエラーコードは WininetAPI のエラーコード

※2：上記表「致命的エラー」は、対象のエラーが発生したときのエラー表示・エラーログ出力の動作を示す。

「○」：FE-COM プロパティ「DebugPolicy」の未設定・設定値に関わらず、エラー表示・エラーログ出力を行う。

「ー」：FE-COM プロパティ「DebugPolicy」の未設定・設定値により、エラー表示・エラーログ出力を行う。

※3：上記表「エラー後処置」は、エラー発生後の WebTcl アプリケーションの動作を示す。

「強制終了」：エラー発生後、WebTcl アプリケーションを強制終了させる。

「処理継続」：エラー発生後、WebTcl アプリケーションを処理継続させる。

4 その他コマンド

本章では、スクリプト形式などで提供されるコマンドに関して説明します。

表 4-1に説明するコマンドの一覧を記述します。

表 4-1 その他コマンド一覧

No.	種別	名称	概要	必須
1	C/S 版スクリプト	createIndex.psv	インデックスファイルの作成を支援するツールです。本ツールにはインデックス生成コマンドである、createIndex プロシジャが定義されています。	—

4.1 C/S 版スクリプト

4.1.1 インデックス生成コマンド(createIndex)

(1) インデックス生成コマンド概要

インデックス生成コマンドはインデックス機能で使用するインデックスファイルを作成するツールです。インデックスファイルはファイル内容のスキャンが必要なため、ファイル操作に制限がないC/S版のTclエンジンを利用して動作します。

インデックス生成コマンドはWeb上に構築する、コンテンツと同じディレクトリ構造を、開発機上に作成し、WebTclのスクリプトのディレクトリをスキャンして、インデックスファイルを生成します。

図 4-1にインデックス機能全体の概要を示します。

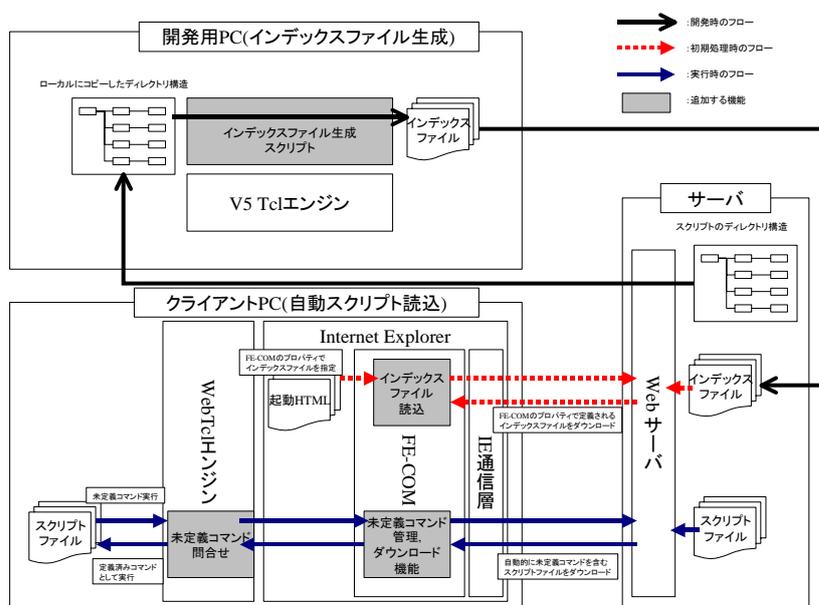


図 4-1 インデックス機能概要

(2) インターフェース

インデックスファイル生成コマンドのインターフェースについて説明します。

表 4-2 インデックス生成コマンドインターフェース

API 名称	インデックスファイル生成コマンド	シンボル	createIndex	
呼び出し形式	createIndex targetDir outputFile [extension]			
ファイル	createIndex.psv			
機能	<p>targetDir で指定するインデックス作成ディレクトリ以下の、対象ファイルを調査して、プロシジャ定義ルールに一致するスクリプトとプロシジャ名の一覧表(インデックス)を作成します。インデックスは調査がすべて正常に終了後、outputFile で指定するインデックス出力ファイル名のファイルに出力します。</p> <p>対象ファイルはデフォルトで、拡張子”psv”のファイルに限定します。拡張子の指定を変更する場合は、extension に拡張子を指定してください。(例：*.tcl ファイルを対象ファイルとする場合は、”tcl”を extension に指定。)複数の拡張子を対象ファイルに指定することもできます。(例：*.tcl ファイルと*.psv ファイルを対称にする場合 “tcl psv”を extension に指定。)</p> <p>注) 拡張子の一致は前方部分一致です。”tcl”を extension に指定した場合、*.tclbak も対象となりますので、3文字以上の拡張子を指定したファイルがある場合は、事前に調査ディレクトリから除外してください。</p> <p>createIndexはcreateIndex.psvファイル内で定義されるプロシジャの1つで、createIndex.psvをTclコマンドsourceで読み込み後、プロシジャの実行で機能します。</p> <p>createIndexはローカルファイルアクセスを前提とした機能であるため、WebTclエンジン上で動作せず、PreSerV V5のインタプリタ・シェル(psvlogin.exe)から実行することが前提です。</p>			
入出力	入出力	変数シンボル	必須	内容
	[1/]	targetDir	必須	インデックス作成ディレクトリパス名 絶対パス名で指定
	[1/]	outputFile	必須	インデックス出力ファイル名
	[1/]	extension	省略可能	対象ファイルの拡張子を”psv”以外を指定する場合に、指定
	エラー	—		エラー時にTclコマンドの”error”を使用して、エラーを通知し、処理を中断します。

(3) 実行方法

createIndex を実行するには、PreSerV サーバ (preserv.exe) を起動し、PreSerV V5 のインタプリタ・シェル (psvlogin.exe) を実行します。

インタプリタ・シェルでは日本語を使用できないため、createIndex.psv ファイルと対象のディレクトリを、日本語を使用していないディレクトリにコピーする必要があります。createIndex.psv ファイルは、以下のディレクトリに格納されています。

```
" C:\Program Files\三菱電機株式会社\PreSerV WebTcl Basic Media\TclTool"
```

リスト 4-1にコマンド入力ウィンドウに入力するコマンドの例を示します。

リスト 4-1 createIndex コマンド例

```
hostname-> ①
% source createIndex.psv ②
% createIndex C:/test/macro C:/test/macro/index.txt psv tcl
③ ④ ⑤
```

- ① エンターキーを押下します。
- ② createIndex.psv を読み込みます。
- ③ createIndex プロシジャを実行します。第一引数に対象のディレクトリ指定します。
- ④ 第二引数に出力するインデックスファイルの名称を指定します。作成モードは上書きモードなので、同一名称のファイルが存在する場合は上書きされます。
- ⑤ 第三引数は対象ファイルの拡張子を指定します。指定しない場合はデフォルトとして”psv” ファイルを対象とします。上記の例では”psv” ファイルと”tcl” ファイルを対象に指定しています。

注) ②、③、④、⑤については、日本語を使用できません。

(4) インデックスファイル

インデックス作成コマンドが出力するインデックスファイルについて説明します。

インデックスファイルは、「プロシジャ名」と「スクリプト名」を、1行1定義で、以下の書式で定義します。

プロシジャ名: スクリプト名

注) プロシジャ名に日本語を使用している場合は、正しくインデックスファイルに出力できない場合があります。

また、同一プロシジャ名が複数存在する場合は、ひとつ目のプロシジャに対してのみインデックスファイルに出力し、二つ目以降はインデックスファイルに出力しません。
重複したプロシジャ名とスクリプト名を Warning メッセージで表示します

表 4-3 インデックスファイル中の定義情報

No.	分類	定義	説明
1	プロシジャ名	スクリプト中の先頭の” proc プロシジャ名 { “で始まる情報から切り出した名称	インデックスの対象となるプロシジャの名称を示す。 基本的に Tcl でプロシジャ名として定義可能な任意の文字列で構成される。
2	スクリプト名	スクリプトの相対パス名	処理対象となるディレクトリからの相対パス名でスクリプトのパス名を表現する。 実行時は処理対象のディレクトリ名を FE-COM の CodeBase プロパティで指定される URL に置換して、ダウンロード対象の URL を合成する。

リスト 4-2 インデックスファイル出力例

createIndex:createIndex.psv
searchDir:createIndex.psv
extractIndex:createIndex.psv
outputIndex:createIndex.psv
g_attrib:pop/bcomp.psv
g_cur_attrib:pop/bcomp.psv
g_mode:pop/bcomp.psv
g_open_image:pop/bcomp.psv
g_close_image:pop/bcomp.psv
g_add_image:pop/bcomp.psv
win_pop1:pop/cmbtst.psv
win_pop2:pop/cmbtst.psv
win_pop3:pop/popupfree.psv
bmp_change3:pop/popupfree.psv
win_destroy3:pop/popupfree.psv
win_pop:pop/rdbtnfree.psv
bmp_change:pop/rdbtnfree.psv
win_destroy:pop/rdbtnfree.psv

5 付録

5.1 標準部品内部マクロ

st_operation で使用している内部マクロについて説明します

表 5-1 st_p_point コーリングシーケンス

st_p_point	マウスクリック座標の格納
形式	st_p_point stid
説明	st_p_point は stval(x)、stval(y) の値を stval(stid, gpoint) に追加し、stval(stid, pnum) を インクリメントします。 もし stval(stid, pnum) が stval(stid, pmax) に達したら stval(stid, ecom) を実行します。
引数	stid: 標準部品IDを指定します。
参照 外部変数	stval(x) stval(y) stval(stid, pmax)
出力 外部変数	stval(stid, gpoint) stval(stid, pnum)
戻り値	なし

表 5-2 st_p_poly コーリングシーケンス

st_p_poly	ポリライン、ポリゴン、スプライン描画時の確定ポップアップ表示
形式	st_p_poly stid
説明	st_p_poly は、stval(b) が右(r)の時に、マウスクリックした場所へ確定のポップアップを表示します。
引数	stid: 標準部品IDを指定します。
参照 外部変数	stval(b)
出力 外部変数	なし
戻り値	なし

表 5-3 st_p_sel コーリングシーケンス

st_p_sel	図形編集における複数選択確定ポップアップ表示
形式	st_p_sel stid
説明	st_p_poly は stval (b) が右 (r) の時に、マウスクリックした場所へ確定のポップアップを表示します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (b)
出力 外部変数	なし
戻り値	なし

表 5-4 st_p_data コーリングシーケンス

st_p_data	最近点検索における選択図形オブジェクトの格納
形式	st_p_data stid
説明	st_p_data は stval (stid, anydata) の図形オブジェクトを選択色に変更し、stval (stid, zuobjs) に追加します。ただし、同じ図形オブジェクトは格納しません。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, anydata)
出力 外部変数	stval (stid, zuobjs)
戻り値	なし

表 5-5 st_m_nearest コーリングシーケンス

st_m_nearest	最近点検索の実行
形式	st_m_nearest stid
説明	st_m_nearest は、stval(x)、stval(y) から最近点図形オブジェクトを取得し、stval(stid, anydata) にセットします。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(x) stval(y)
出力 外部変数	stval(stid, anydata)
戻り値	なし

表 5-6 st_m_chgcol コーリングシーケンス

st_m_chgcol	最近点検索における図形表示色の変更
形式	st_m_chgcol stid
説明	st_m_chgcol は、stval(stid, olddata) の図形オブジェクトの色を元に戻して、stval(stid, anydata) の図形オブジェクトの色を stval(rubber_color) に変更します。 stval(stid, anydata) を stval(stid, olddata) にセットします。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, anydata) stval(stid, olddata) stval(rubber_color)
出力 外部変数	stval(stid, olddata)
戻り値	なし

表 5-7 st_m_scroll コーリングシーケンス

st_m_scroll	スクロール
形式	st_m_scroll stid
説明	st_m_scroll は、stval (stid, gpoint) と、stval (x), stval (y) の差分だけスクロールします。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, gpoint) stval (x) stval (y)
出力 外部変数	なし
戻り値	なし

表 5-8 st_m_line コーリングシーケンス

st_m_line	線ラバーバンド描画
形式	st_m_line stid
説明	st_m_line は、stval (stid, gpoint) の最終点から、stval (x)、stval (y) で指定する点まで、線のラバーバンドを描画します。線入力、ポリライン入力、ポリゴン入力、スプライン入力の mcom に使用します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, gpoint) stval (x) stval (y)
出力 外部変数	なし
戻り値	なし

表 5-9 st_m_rect コーリングシーケンス

st_m_rect	長方形ラバーバンド描画
形式	st_m_rect stid
説明	st_m_rect は、stval (stid, gpoint) の最終点から、stval (x)、stval (y) で指定する点まで、長方形のラバーバンドを描画します。長方形入力、拡大縮小、図形編集の範囲選択、印刷の mcom に使用します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, gpoint) stval (x) stval (y)
出力 外部変数	なし
戻り値	なし

表 5-10 st_m_circle コーリングシーケンス

st_m_circle	円ラバーバンド描画
形式	st_m_circle stid
説明	st_m_circle は、stval (stid, gpoint) の最終点から、stval (x)、stval (y) で指定する点まで、円のラバーバンドを描画します。円入力、円弧入力の mcom に使用します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, gpoint) stval (x) stval (y)
出力 外部変数	なし
戻り値	なし

表 5-11 st_m_arc コーリングシーケンス

st_m_arc	円弧ラバーバンド描画
形式	st_m_arc stid
説明	st_m_arc は、stval (stid, gpoint) の中心点、半径、開始角から、stval (x)、stval (y) で指定する終了角まで、円弧のラバーバンドを描画します。円弧入力の mcom に使用します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, gpoint) stval (x) stval (y)
出力 外部変数	なし
戻り値	なし

表 5-12 st_m_ellipse コーリングシーケンス

st_m_ellipse	楕円ラバーバンド描画
形式	st_m_ellipse stid
説明	st_m_ellipse は、stval (stid, gpoint) の最終点から、stval (x)、stval (y) で指定する点まで、楕円のラバーバンドを描画します。楕円のラバーバンドは外接方形内に収まるように描画します。楕円入力の mcom に使用します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, gpoint) stval (x) stval (y)
出力 外部変数	なし
戻り値	なし

表 5-13 st_m_freeline コーリングシーケンス

st_m_freeline	自由線描画
形式	st_m_freeline stid
説明	st_m_freeline は、stval (stid, gpoint) の最終点から、stval (x)、stval (y) で指定する点まで、stval (stid, gpoint) に座標を追加し、stval (stid, pnum) をインクリメントします。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, gpoint) stval (stid, pnum) stval (x) stval (y)
出力 外部変数	stval (stid, gpoint) stval (stid, pnum)
戻り値	なし

表 5-14 st_m_roundbox コーリングシーケンス

st_m_roundbox	角丸長方形ラバーバンド描画
形式	st_m_roundbox stid
説明	st_m_roundbox は、stval (stid, gpoint) の長方形の頂点から、stval (x)、stval (y) で指定する点まで、半径を算出し、角丸長方形のラバーバンドを描画します。角丸長方形入力の mcom に使用します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval (stid, gpoint) stval (x) stval (y)
出力 外部変数	なし
戻り値	なし

表 5-15 st_m_regpoly コーリングシーケンス

st_m_regpoly	正多角形ラバーバンド描画
形式	st_m_regpoly stid
説明	st_m_regpoly は、stval(stid, gpoint)の最終点から、stval(x)、stval(y)で指定する点まで、正多角形のラバーバンドを描画します。正多角形入力の mcom に使用します。
引数	stid: 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint) stval(x) stval(y)
出力 外部変数	なし
戻り値	なし

表 5-16 st_m_edit_obj コーリングシーケンス

st_m_edit_obj	図形編集ドラッキング描画
形式	st_m_edit_obj stid
説明	st_m_edit_obj は、stval(stid, filter)で定義する図形編集の内容を、stval(x)、stval(y)で指定する点までドラッキング描画します。図形編集の mcom に使用します。
引数	stid: 標準部品IDを指定します。
参照 外部変数	stval(stid, filter) stval(x) stval(y)
出力 外部変数	なし
戻り値	なし

表 5-17 st_e_nearest コーリングシーケンス

st_e_nearest	最近点検索の終了
形式	st_e_nearest stid
説明	st_e_nearest は、stval(stid, anydata) から stval(stid, ucom) に渡す引数を生成し、stval(stid, ucom) を実行します。
引数	stid: 標準部品IDを指定します。
参照 外部変数	stval(stid, anydata) stval(stid, ucom)
出力 外部変数	なし
戻り値	なし

表 5-18 st_e_chgview コーリングシーケンス

st_e_chgview	拡大・縮小の終了
形式	st_e_chgview stid flg
説明	st_e_chgview は、stval(stid, gpoint) の矩形座標から、拡大・縮小 (g_ortho) を行います。拡大・縮小後、現在のビューの座標を取得し (g_ortho) stval(stid, ucom) に渡す引数を生成し、stval(stid, ucom) を実行します。
引数	stid: 標準部品IDを指定します。 flg: 拡大のときは "" (NULL)、縮小のときは "-S" を指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-19 st_e_scroll コーリングシーケンス

st_e_scroll	スクロールの終了
形式	st_e_scroll stid
説明	st_e_scroll は、スクロール後のビュー座標を取得し、stval (stid, ucom) に渡す引数を生成後、stval (stid, ucom) を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	なし
戻り値	なし

表 5-20 st_e_print コーリングシーケンス

st_e_print	図面印字の終了
形式	st_e_print stid kami fname
説明	st_e_print は、stval (stid, gpoint) で指定する矩形範囲からポストスクリプトファイルを生成します。 ※WebTcl ではこの機能は未サポートです。
引数	stid : 標準部品IDを指定します。 kami : 紙の種類を指定します (a4l : A4 横, a3l : A3 横, a4p : A4 縦, a3p : A3 縦)。 fname : 出力するファイル名を指定します。
参照 外部変数	stval (stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-21 st_e_line コーリングシーケンス

st_e_line	線入力の終了
形式	st_e_line stid
説明	st_e_line は、stval(stid, gpoint) から stval(stid, ucom) に渡す引数を生成後、stval(stid, ucom) を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-22 st_e_polyline コーリングシーケンス

st_e_polyline	ポリライン・ポリゴン・スプライン入力の終了
形式	st_e_polyline stid
説明	st_e_polyline は、stval(stid, pnum)、stval(stid, gpoint) から stval(stid, ucom) に渡す引数を生成後、stval(stid, ucom) を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, pnum) stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-23 st_e_rect コーリングシーケンス

st_e_rect	長方形入力の終了
形式	st_e_rect stid
説明	st_e_rect は、stval(stid, gpoint) から stval(stid, ucom) に渡す引数を生成後、stval(stid, ucom) を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-24 st_e_circle コーリングシーケンス

st_e_circle	円入力の終了
形式	st_e_circle stid
説明	st_e_circle は、stval(stid, gpoint) から stval(stid, ucom) に渡す引数を生成後、stval(stid, ucom) を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-25 st_e_arc1 コーリングシーケンス

st_e_arc1	円弧入力の円描画終了
形式	st_e_arc1 stid
説明	st_e_arc1 は、円ラバーバンド設定後、stval(stid, gpoint)から中心点、半径を得ます。 stval(x), stval(y)から開始角度を得ます。 stval(stid, gpoint)に中心点、半径、開始角をセットし mcom に st_m_arc 、 pcom に st_e_arc2 、 pmax に 1 をセットします。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint) stval(x) stval(y)
出力 外部変数	stval(stid, gpoint) stval(stid, pcom) stval(stid, mcom) stval(stid, pmax)
戻り値	なし

表 5-26 st_e_arc2 コーリングシーケンス

st_e_arc2	円弧入力の終了
形式	st_e_arc2 stid
説明	st_e_arc2 は、stval(stid, gpoint)から stval(stid, ucom)に渡す引数を生成後、stval(stid, ucom)を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-27 st_e_ellipse コーリングシーケンス

st_e_ellipse	楕円入力の終了
形式	st_e_ellipse stid
説明	st_e_ellipse は、stval(stid, gpoint) から stval(stid, ucom) に渡す引数を生成後、stval(stid, ucom) を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-28 st_e_text1 コーリングシーケンス

st_e_text1	文字入力の長方形描画終了
形式	st_e_text1 stid
説明	st_e_text1 は、長方形ラバーバンド設定後、mcom、pcom をクリアし、文字入力ダイアログを表示します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pcom) stval(stid, mcom)
戻り値	なし

表 5-29 st_e_text2 コーリングシーケンス

st_e_text2	文字入力の終了
形式	st_e_text2 stid mj_text
説明	st_e_text2 は、stval(stid, gpoint)から、文字列表示位置、文字高、文字幅を算出し、stval(stid, gpoint)にセットします。stval(stid, ucom)を実行します。
引数	stid : 標準部品IDを指定します。 mj_text : 表示する文字列を指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	stval(stid, gpoint)
戻り値	なし

表 5-30 st_e_freeline コーリングシーケンス

st_e_freeline	自由線入力の終了
形式	st_e_freeline stid
説明	st_e_freeline は、stval(stid, gpoint)から stval(stid, ucom)に渡す引数を生成後、stval(stid, ucom)を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-31 st_e_roundbox1 コーリングシーケンス

st_e_roundbox1	角丸長方形入力の長方形描画終了
形式	st_e_roundbox1 stid
説明	st_e_roundbox1 は、長方形ラバーバンド設定後、長方形座標を stval(stid, gpoint) にセットし、mcom に st_m_roundbox 、 pcom に st_e_roundbox2 、 pmax に 1 をセットします。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint) stval(x) stval(y)
出力 外部変数	stval(stid, gpoint) stval(stid, pcom) stval(stid, mcom) stval(stid, pmax)
戻り値	なし

表 5-32 st_e_roundbox2 コーリングシーケンス

st_e_roundbox2	角丸長方形入力の終了
形式	st_e_roundbox2 stid
説明	st_e_roundbox2 は、stval(stid, gpoint) から stval(stid, ucom) に渡す引数を生成後、stval(stid, ucom) を実行します。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-33 st_e_regpoly コーリングシーケンス

st_e_regpoly	正多角形入力の終了
形式	st_e_regpoly stid
説明	st_e_regpoly は、stval(stid, gpoint) から stval(stid, ucom) に渡す引数を生成後、stval(stid, ucom) を実行します。
引数	stid : 標準部品 ID を指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	なし
戻り値	なし

表 5-34 st_e_edit_obj1 コーリングシーケンス

st_e_edit_obj1	図形編集の対象図形決定
形式	st_e_edit_obj1 stid
説明	st_e_edit_obj1 は、確定した図形オブジェクトから、それぞれの図形編集（移動、変形、回転、コピー、拡大・縮小、ポリライン編集）に対応したフィルタを生成し、値を stval(stid, filter) にセットします。その後、pcom に st_p_point、mcom に st_m_edit_obj、ecom に st_e_edit_obj2 をセットします。 stval(stid, ope) が "del_obj" のときは確定した図形オブジェクトを削除します。
引数	stid : 標準部品 ID を指定します。
参照 外部変数	stval(stid, gpoint) stval(stid, zuobjs) stval(stid, ope)
出力 外部変数	stval(stid, filter) stval(stid, pcom) stval(stid, mcom) stval(stid, ecom)
戻り値	なし

表 5-35 st_e_edit_obj2 コーリングシーケンス

st_e_edit_obj2	図形編集の終了
形式	st_e_edit_obj2 stid
説明	st_e_edit_obj2 は、それぞれの図形編集処理（移動、変形、回転、コピー、拡大・縮小、ポリライン編集）を確定し、フィルターをクローズします。 stval(stid, ucom)に渡す引数を生成後、stval(stid, ucom)を実行します。
引数	stid: 標準部品IDを指定します。
参照 外部変数	stval(stid, filter) stval(stid, zuobjjs) stval(stid, ope)
出力 外部変数	なし
戻り値	なし

表 5-36 st_e_edit_objr コーリングシーケンス

st_e_edit_objr	図形編集の範囲選択
形式	st_e_edit_objr stid
説明	st_e_edit_objr は、長方形ラバーバンド確定後、stval(stid, gpoint)の長方形座標に含まれる図形オブジェクトを検索して、選択色に変更し、stval(stid, zuobjjs)に、検索した図形オブジェクトをセットします。
引数	stid: 標準部品IDを指定します。
参照 外部変数	stval(stid, gpoint)
出力 外部変数	stval(stid, zuobjjs)
戻り値	なし

表 5-37 st_e_edit_objp コーリングシーケンス

st_e_edit_objp	ポリライン編集の図形決定
形式	st_e_edit_objp stid
説明	st_e_edit_objp は、ポリライン編集の対象図形オブジェクトを確定し ecom に st_e_edit_obj2 をセットします。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, filter) stval(stid, gpoint)
出力 外部変数	stval(stid, ecom)
戻り値	なし

表 5-38 st_reset_nearest コーリングシーケンス

st_reset_nearest	最近点検索における図形データリセット
形式	st_reset_nearest stid
説明	st_reset_nearest は以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, anydata) "" stval(stid, zuobjjs) ""
引数	stid : 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) stval(stid, anydata) stval(stid, zuobjjs)
戻り値	なし

表 5-39 st_reset_chgcol コーリングシーケンス

st_reset_chgcol	最近点検索における図形表示色のリセット
形式	st_reset_chgcol stid
説明	st_reset_chgcol は、stval(stid, olddata) の図形オブジェクトを元の色に戻して、stval(stid, olddata) の値を NULL クリアします。
引数	stid : 標準部品IDを指定します。
参照 外部変数	stval(stid, olddata)
出力 外部変数	stval(stid, olddata)
戻り値	なし

表 5-40 st_reset_scroll コーリングシーケンス

st_reset_scroll	スクロールにおける外部変数のリセット
形式	st_reset_scroll stid
説明	st_reset_scroll は以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid : 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) stval(stid, gpoint)
戻り値	なし

表 5-41 st_reset_line コーリングシーケンス

st_reset_line	線ラバーバンドの消去
形式	st_reset_line stid
説明	st_reset_line は、線ラバーバンドの軌跡を消去します。ポリライン、ポリゴン、スプラインのときは、その軌跡も消去します。 以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid: 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) 0 stval(stid, gpoint) ""
戻り値	なし

表 5-42 st_reset_rect コーリングシーケンス

st_reset_rect	長方形ラバーバンドの消去
形式	st_reset_rect stid
説明	st_reset_rect は、長方形ラバーバンドの軌跡を消去し、以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid: 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) 0 stval(stid, gpoint) ""
戻り値	なし

表 5-43 st_reset_circle コーリングシーケンス

st_reset_circle	円ラバーバンドの消去
形式	st_reset_circle stid
説明	st_reset_circle は、円ラバーバンドの軌跡を消去し、以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid : 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) 0 stval(stid, gpoint) ""
戻り値	なし

表 5-44 st_reset_arc コーリングシーケンス

st_reset_arc	円弧ラバーバンドの消去
形式	st_reset_arc stid
説明	st_reset_arc は、円弧ラバーバンドの軌跡を消去し、以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid : 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) 0 stval(stid, gpoint) ""
戻り値	なし

表 5-45 st_reset_ellipse コーリングシーケンス

st_reset_ellipse	楕円ラバーバンドの消去
形式	st_reset_ellipse stid
説明	st_reset_ellipse は、楕円ラバーバンドの軌跡を消去し、以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid 標準部品 ID を指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) 0 stval(stid, gpoint) ""
戻り値	なし

表 5-46 st_reset_freeline コーリングシーケンス

st_reset_freeline	自由線ラバーバンドの消去
形式	st_reset_freeline stid
説明	st_reset_freeline は、自由線ラバーバンドの軌跡を消去します。 以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid : 標準部品 ID を指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) 0 stval(stid, gpoint) ""
戻り値	なし

表 5-47 st_reset_roundbox コーリングシーケンス

st_reset_roundbox	角丸長方形ラバーバンドの消去
形式	st_reset_roundbox stid
説明	st_reset_roundbox は、角丸長方形ラバーバンドの軌跡を消去します。 以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid: 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) 0 stval(stid, gpoint) ""
戻り値	なし

表 5-48 st_reset_regpoly コーリングシーケンス

st_reset_regpoly	正多角形ラバーバンドの消去
形式	st_reset_regpoly stid
説明	st_reset_regpoly は、正多角形ラバーバンドの軌跡を消去し、以下の変数をリセットします。 stval(stid, pnum) 0 stval(stid, gpoint) ""
引数	stid: 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, pnum) 0 stval(stid, gpoint) ""
戻り値	なし

表 5-49 st_reset_edit_obj コーリングシーケンス

st_reset_edit_obj	図形編集ドラッキング図形の消去
形式	st_reset_edit_obj stid
説明	st_reset_edit_obj は、図形編集のドラッキング図形の軌跡を消去して、フィルタをクローズし、以下の変数をリセットします。 stval(stid, zuobjjs) ""
引数	stid: 標準部品IDを指定します。
参照 外部変数	なし
出力 外部変数	stval(stid, zuobjjs) ""
戻り値	なし