

MITSUBISHI

三菱Web地理情報システム構築パッケージ



PreSerV WebTcl API リファレンス (G マクロ)

Ver. 6.0

マニュアル

はじめに

はじめに

本書では、PreSerV WebTcl のグラフィックスマクロ (G マクロ) で提供する API の仕様を説明します。

目次

1	APIリファレンスに関して.....	1
1.1	表記に関して.....	1
1.2	用語説明.....	3
2	グラフィックスマクロ (Gマクロ).....	4
2.1	A~Bで始まるコマンド.....	12
2.1.1	g_add_image.....	12
2.1.2	g_alloc_color.....	13
2.1.3	g_arc.....	15
2.1.4	g_attrib.....	17
2.1.5	g_bitmapsymbol.....	20
2.1.6	g_bitmapsymbol_file.....	22
2.2	Cで始まるコマンド.....	23
2.2.1	g_circle.....	23
2.2.2	g_clear.....	25
2.2.3	g_close_context.....	26
2.2.4	g_close_dlist.....	27
2.2.5	g_close_filter.....	28
2.2.6	g_close_group.....	29
2.2.7	g_close_gwin.....	30
2.2.8	g_close_hash.....	31
2.2.9	g_close_image.....	32
2.2.10	g_close_image_region.....	33
2.2.11	g_close_layer.....	34
2.2.12	g_close_map.....	35
2.2.13	g_color_id.....	36
2.2.14	g_compute_size.....	38
2.2.15	g_context.....	39
2.2.16	g_cur_attrib.....	40
2.2.17	g_cur_color_id.....	42
2.2.18	g_cur_dlist.....	43
2.2.19	g_cur_draw_attrib.....	44
2.2.20	g_cur_group.....	46
2.2.21	g_cur_gwin.....	47
2.2.22	g_cur_layer.....	48
2.2.23	g_cur_obj_attrib.....	49

2.2.24	<code>g_cur_tx_attrib</code>	51
2.3	Dで始まるコマンド.....	53
2.3.1	<code>g_dc2wc</code>	53
2.3.2	<code>g_del_obj</code>	55
2.3.3	<code>g_del_usrdata</code>	57
2.3.4	<code>g_dist</code>	58
2.3.5	<code>g_draw_arc</code>	59
2.3.6	<code>g_draw_attrib</code>	61
2.3.7	<code>g_draw_circle</code>	64
2.3.8	<code>g_draw_ellipse</code>	66
2.3.9	<code>g_draw_line</code>	68
2.3.10	<code>g_draw_obj</code>	70
2.3.11	<code>g_draw_pie</code>	72
2.3.12	<code>g_draw_polygon</code>	74
2.3.13	<code>g_draw_polyline</code>	76
2.3.14	<code>g_draw_rect</code>	78
2.3.15	<code>g_draw_regpoly</code>	80
2.3.16	<code>g_draw_round_box</code>	82
2.3.17	<code>g_draw_symbol</code>	84
2.3.18	<code>g_draw_text</code>	86
2.3.19	<code>g_dump_obj</code>	88
2.4	E~Gで始まるコマンド.....	90
2.4.1	<code>g_ellipse</code>	90
2.4.2	<code>g_filter_add</code>	92
2.4.3	<code>g_filter_draw_obj</code>	93
2.4.4	<code>g_filter_remove</code>	95
2.4.5	<code>g_filter_set_param</code>	96
2.4.6	<code>g_get_usrdata</code>	98
2.4.7	<code>g_group_attrib</code>	99
2.4.8	<code>g_group_color_id</code>	101
2.4.9	<code>g_group_no</code>	102
2.4.10	<code>g_gwin_attrib</code>	103
2.4.11	<code>g_gwin_color</code>	105
2.4.12	<code>g_gwin_tx_attrib</code>	106
2.5	H~Lで始まるコマンド.....	108
2.5.1	<code>g_hash_del</code>	108

2.5.2	g_hash_fetch.....	109
2.5.3	g_hash_store.....	110
2.5.4	g_image_dsp.....	111
2.5.5	g_ins_obj.....	112
2.5.6	g_JBL2JXY.....	113
2.5.7	g_layer_attrib.....	114
2.5.8	g_layer_color_id.....	116
2.5.9	g_layer_name.....	117
2.5.10	g_layer_no.....	119
2.5.11	g_layer_range.....	120
2.5.12	g_line.....	122
2.5.13	g_linesymbol.....	124
2.5.14	g_linesymbol_file.....	127
2.6	Mで始まるコマンド.....	128
2.6.1	g_make_group.....	128
2.6.2	g_map_attrib.....	130
2.6.3	g_map_element.....	132
2.6.4	g_map_load.....	133
2.6.5	g_map_margin.....	134
2.6.6	g_map_range.....	135
2.6.7	g_map_type.....	136
2.6.8	g_mode.....	137
2.6.9	mkbsym.exe (MS_DOSコマンド).....	139
2.7	0で始まるコマンド.....	141
2.7.1	g_obj_attrib.....	141
2.7.2	g_obj_range.....	143
2.7.3	g_open_dlist.....	145
2.7.4	g_open_filter.....	146
2.7.5	g_open_group.....	148
2.7.6	g_open_gwin.....	149
2.7.7	g_open_hash.....	152
2.7.8	g_open_image.....	153
2.7.9	g_open_image_region.....	154
2.7.10	g_open_map.....	156
2.7.11	g_open_layer.....	157
2.7.12	g_ortho.....	158

2.8	P~Rで始まるコマンド	160
2.8.1	g_paint_pattern_file	160
2.8.2	g_pie	161
2.8.3	g_polygon	163
2.8.4	g_polyline	165
2.8.5	g_raster	167
2.8.6	g_redraw	168
2.8.7	g_region_obj	170
2.8.8	g_regpoly	172
2.8.9	g_rm_obj	174
2.8.10	g_rotate	176
2.8.11	g_rotate_obj	178
2.8.12	g_round_box	180
2.8.13	g_RRline	182
2.9	Sで始まるコマンド	185
2.9.1	g_scale	185
2.9.2	g_scale_obj	187
2.9.3	g_scroll	189
2.9.4	g_search_obj	191
2.9.5	g_seqsearch_obj	194
2.9.6	g_seqsearch_next	196
2.9.7	g_set_clipboard	197
2.9.8	g_set_usrdata	200
2.9.9	g_showpage	201
2.9.10	g_sort	202
2.9.11	g_symbol	203
2.9.12	g_symbol_file	205
2.10	T~Yで始まるコマンド	206
2.10.1	g_tag_id	206
2.10.2	g_TBL2JBL	207
2.10.3	g_TBL2TXY	208
2.10.4	g_TKY2JGD_ParaFile	209
2.10.5	g_text	210
2.10.6	g_transform_obj	213
2.10.7	g_translate_obj	216
2.10.8	g_usr_def	217

2.10.9	<code>g_viewport</code>	220
2.10.10	<code>g_wc2dc</code>	222
2.10.11	<code>g_yoko_mercator</code>	223

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(“g_”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、一覧画面と API の説明画面には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) API の表記に関して

API全般で共通の表記を表 1-1で説明します。

表 1-1 API 共通表記

表 記	表記の説明
<i>macro_string</i>	指定されたタイミング(イベント)で実行される Tcl コマンドの集合(マクロ)を示します。Tcl コマンド実行時は各種情報をシステム変数(psv_size(w)など)から取得し、利用することが可能です。 例) w_top のサンプル中の-d オプション以下の {} で括られる部分の Tcl コマンドは、トップウィンドウクローズ時に実行されます。
<i>color_name</i>	カラー名称で定義される色の名称です。名称を指定すると、対応付けられた RGB 値を使用します。

(4) 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。

(5) 単位について

単位の説明が省略されている場合、高さ、幅、長さ、円弧の半径は画面表示の画素(ピクセル)単位です。また時間の単位はミリ秒です。

スケールについては、「PreSerV WebTcl 用語集」を参照ください。

(6) 各コマンドの形式の表記について

各コマンドの形式部分および形式の説明での表記で [] で閉じられている部分は省略可能です。また、| で区切られている部分は | の左右どちらかが使用可能です。

例)

`g_attrib obj/D [attribVal]` *attribVal*は省略可能です。

`-o [h|v]` *h*もしくは*v*は使用可能です。

(7) 「DC」と「WC」の表記について

「DC」「WC」が明示的に表記されていない場合は、「WC」として扱われます。

用語説明

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

2 グラフィックスマクロ (G マクロ)

G マクロは、グラフィックス操作を行うマクロです。図形の描画や管理、図形操作を行います。

本章では G マクロを API のアルファベット順に説明します。API の機能的な分類は、以下の表に示す分類となります。

表 2-1 G マクロ機能索引

コンテキストオブジェクトの作成と設定			
g_context	コンテキストオブジェクトの作成		2. 2. 15
g_close_context	コンテキストオブジェクトの消去		2. 2. 3
g_cur_gwin	グラフィックスウィンドウのカレント値 取得／変更		2. 2. 21
g_cur_dlist	ディスプレイリストメモリのカレント値 取得／変更		2. 2. 18
g_cur_group	グループオブジェクトのカレント値取得 ／変更		2. 2. 20
g_cur_layer	レイヤオブジェクトのカレント値取得／ 変更		2. 2. 22
g_cur_attr	図形オブジェクト属性値のカレント値取 得／変更	Ver. 4更新	2. 2. 16
g_cur_obj_attr	図形オブジェクト属性値のカレント値取 得／変更	Ver. 4更新	2. 2. 23
g_cur_draw_attr	図形オブジェクト描画属性値のカレント 値取得／変更	Ver. 4更新	2. 2. 19
g_cur_color_id	色 ID のカレント値取得／変更		2. 2. 17
g_cur_tx_attr	文字属性値のカレント値取得／変更	Ver. 5更新	2. 2. 24
グラフィックスウィンドウの作成と操作			
g_open_gwin	グラフィックスウィンドウのオープン	Ver. 3更新 Ver. 4更新	2. 7. 6
g_close_gwin	グラフィックスウィンドウの消去		2. 2. 7
g_redraw	キャンバスの自動再描画モードの設定	Ver. 4更新	2. 8. 6
g_clear	グラフィックスウィンドウのクリア	Ver. 4更新	2. 2. 2
g_ortho	グラフィックスウィンドウの表示位置取 得		2. 7. 12
g_rotate	グラフィックスウィンドウの回転		2. 8. 10

2 章グラフィックスマクロ (G マクロ)

グラフィックスウィンドウの作成と操作		
<code>g_scroll</code>	グラフィックスウィンドウのスクロール	2. 9. 3
<code>g_viewport</code>	ビューポート座標の設定	2. 10. 9
<code>g_raster</code>	ラスタオペレーションの操作	2. 8. 5
<code>g_compute_size</code>	グラフィックスウィンドウの DC、WC の比率の再設定	2. 2. 14
<code>g_scale</code>	グラフィックスウィンドウの表示縮尺の取得／変更	2. 9. 1 Ver. 4更新
<code>g_gwin_attrib</code>	グラフィックスウィンドウの図形属性値の取得／変更	2. 4. 10
<code>g_gwin_color</code>	グラフィックスウィンドウの色 ID の取得／変更	2. 4. 11
<code>g_gwin_tx_attrib</code>	グラフィックスウィンドウの文字列属性値の取得／変更	2. 4. 12 Ver. 5更新
<code>g_showpage</code>	印刷の終了	2. 9. 9
図形オブジェクトの作成		
<code>g_line</code>	線オブジェクトの作成	2. 5. 12 Ver. 4更新
<code>g_polyline</code>	折れ線(スプライン)オブジェクトの作成	2. 8. 4 Ver. 4更新
<code>g_circle</code>	円オブジェクトの作成	2. 2. 1 Ver. 4更新
<code>g_ellipse</code>	楕円オブジェクトの作成	2. 4. 1 Ver. 4更新
<code>g_arc</code>	円弧オブジェクトの作成	2. 1. 3 Ver. 4更新
<code>g_pie</code>	扇形オブジェクトの作成	2. 8. 2 Ver. 4更新
<code>g_polygon</code>	多角形オブジェクトの作成	2. 8. 3 Ver. 4更新
<code>g_round_box</code>	角丸長方形オブジェクトの作成	2. 8. 12 Ver. 4更新
<code>g_regpoly</code>	正多角形オブジェクトの作成	2. 8. 8 Ver. 4更新
<code>g_text</code>	文字列オブジェクトの作成	2. 10. 5 Ver. 3更新 Ver. 4更新
<code>g_symbol</code>	シンボルオブジェクトの作成	2. 9. 11 Ver. 4更新
<code>g_usr_def</code>	ユーザ定義オブジェクトの作成	2. 10. 8 Ver. 4更新
<code>g_bitmapsymbol</code>	ビットマップシンボルオブジェクトの作成	2. 1. 5 Ver. 4更新
<code>g_linesymbol</code>	ラインシンボルオブジェクトの作成	2. 5. 13 Ver. 4更新
<code>g_RRline</code>	鉄道線オブジェクトの作成	2. 8. 13 Ver. 5更新

2 章グラフィックスマクロ (G マクロ)

図形オブジェクトの操作			
g_draw_obj	グラフィックスウィンドウへの図形オブジェクト描画	Ver. 4更新	2. 3. 10
g_search_obj	図形オブジェクトの検索	Ver. 3更新	2. 9. 4
g_seqsearch_obj	図形オブジェクトの順次検索の開始	Ver. 3更新	2. 9. 5
g_seqsearch_next	図形オブジェクトの順次検索	Ver. 3更新	2. 9. 6
g_ins_obj	ディスプレイリストメモリへの図形オブジェクト格納		2. 5. 5
g_rm_obj	ディスプレイリストメモリからの図形オブジェクト削除		2. 8. 9
g_group_no	図形オブジェクトのグループ番号を取得／変更		2. 4. 9
g_make_group	図形オブジェクトのグループ化		2. 6. 1
g_layer_no	図形オブジェクトのレイヤ番号を取得／変更		2. 5. 10
g_layer_name	レイヤ名の取得／設定、レイヤ番号の取得	WebTool更新	2. 5. 9
g_del_obj	図形オブジェクトの消去		2. 3. 2
g_rotate_obj	図形オブジェクトの回転移動		2. 8. 11
g_scale_obj	図形オブジェクトの拡大／縮小		2. 9. 2
g_transform_obj	図形オブジェクトの変形		2. 10. 6
g_translate_obj	図形オブジェクトの移動		2. 10. 7
g_attrib	図形オブジェクトの属性値の取得／変更	Ver. 4更新	2. 1. 4
g_obj_attrib	図形オブジェクトの属性値の取得／変更	Ver. 4更新	2. 7. 1
g_draw_attrib	図形オブジェクトの描画属性値の取得／変更	Ver. 4更新	2. 3. 6
g_mode	図形オブジェクトのモードの取得／変更	Ver. 4更新	2. 6. 8
g_color_id	図形オブジェクトの色 ID の取得／変更	Ver. 4更新	2. 2. 13
g_dump_obj	図形オブジェクトの内容取得	Ver. 3更新 Ver. 5更新	2. 3. 19
g_tag_id	図形オブジェクトのタグ ID の取得／変更	Ver. 3更新	2. 10. 1
g_get_usrdata	図形オブジェクトのユーザ定義データ取得	Ver. 3更新	2. 4. 6
g_set_usrdata	図形オブジェクトのユーザ定義データ設定		2. 9. 8
g_del_usrdata	図形オブジェクトのユーザ定義データ削除		2. 3. 3

2 章グラフィックスマクロ (G マクロ)

図形オブジェクトの操作		
g_region_obj	図形オブジェクトの描画範囲取得	2. 8. 7
g_dist	指定点から図形オブジェクトへの距離の取得	2. 3. 4
g_sort	図形オブジェクトの距離順ソート	2. 9. 10
g_set_clipboard	図形オブジェクトのクリップボードへのコピー	2. 9. 7 <small>Ver. 3更新</small>
図形の描画		
g_draw_obj	グラフィックスウィンドウへの図形オブジェクト描画	2. 3. 10 <small>Ver. 4更新</small>
g_draw_line	線の描画	2. 3. 9
g_draw_polyline	折れ線(スプライン)の描画	2. 3. 13
g_draw_circle	円の描画	2. 3. 7
g_draw_ellipse	楕円の描画	2. 3. 8
g_draw_arc	円弧の描画	2. 3. 5
g_draw_pie	扇形の描画	2. 3. 11
g_draw_polygon	多角形の描画	2. 3. 12
g_draw_rect	長方形の描画	2. 3. 14
g_draw_regpoly	正多角形の描画	2. 3. 15
g_draw_round_box	角丸長方形の描画	2. 3. 16
g_draw_text	文字列の描画	2. 3. 18 <small>Ver. 4更新 Ver. 5更新</small>
g_draw_symbol	シンボルの描画	2. 3. 17
ディスプレイリストメモリの作成と操作		
g_open_dlist	ディスプレイリストメモリのオープン	2. 7. 3
g_close_dlist	ディスプレイリストメモリの消去	2. 2. 4
g_cur_dlist	ディスプレイリストメモリのカレント値の取得/変更	2. 2. 18
g_ins_obj	ディスプレイリストメモリへの図形オブジェクト格納	2. 5. 5
g_rm_obj	ディスプレイリストメモリからの図形オブジェクト削除	2. 8. 9
g_draw_obj	グラフィックスウィンドウへの図形オブジェクト描画	2. 3. 10 <small>Ver. 4更新</small>

2 章グラフィックスマクロ (G マクロ)

ディスプレイリストメモリの作成と操作			
g_search_obj	図形オブジェクトの検索	Ver. 3更新	2. 9. 4
g_seqsearch_obj	図形オブジェクトの順次検索の開始	Ver. 3更新	2. 9. 5
g_seqsearch_next	図形オブジェクトの順次検索	Ver. 3更新	2. 9. 6
グループオブジェクトの作成と操作			
g_open_group	グループオブジェクトの作成		2. 7. 5
g_close_group	グループオブジェクトの消去		2. 2. 6
g_group_attrb	グループオブジェクトの属性値の取得／ 変更		2. 4. 7
g_group_color_id	グループオブジェクトの色 ID の取得／変 更		2. 4. 8
g_make_group	図形オブジェクトのグループ化		2. 6. 1
g_group_no	図形オブジェクトのグループ番号を取得 ／変更		2. 4. 9
レイヤオブジェクトの作成と操作			
g_open_layer	レイヤオブジェクトの作成		2. 7. 11
g_close_layer	レイヤオブジェクトの消去		2. 2. 11
g_layer_attrb	レイヤオブジェクトの属性値の取得／変 更		2. 5. 7
g_layer_color_id	レイヤオブジェクトの色 ID の取得／変更		2. 5. 8
g_layer_range	レイヤオブジェクトの表示縮尺範囲の取 得／変更	Ver. 4更新	2. 5. 11
g_obj_range	図形オブジェクト表示サイズの上下限值 の取得／変更	Ver. 4更新	2. 7. 2
g_layer_no	図形オブジェクトのレイヤ番号の取得／ 変更		2. 5. 10
ハッシュオブジェクトの作成と操作			
g_open_hash	ハッシュオブジェクトの作成		2. 7. 7
g_close_hash	ハッシュオブジェクトの消去		2. 2. 8
g_hash_store	ハッシュオブジェクトへのデータ挿入		2. 5. 3
g_hash_fetch	ハッシュオブジェクトのデータ検索		2. 5. 2
g_hash_del	ハッシュオブジェクトのデータ削除		2. 5. 1

2 章グラフィックスマクロ (G マクロ)

属性に関するマクロ			
<code>g_attrib</code>	図形オブジェクト属性値の取得／変更	Ver. 4更新	2. 1. 4
<code>g_obj_attrib</code>	図形オブジェクト属性値のカレント値取得／変更	Ver. 4更新	2. 7. 1
<code>g_draw_attrib</code>	図形オブジェクト描画属性値の取得／変更	Ver. 4更新	2. 3. 6
<code>g_gwin_attrib</code>	グラフィックスウィンドウの図形属性値の取得／変更		2. 4. 10
<code>g_group_attrib</code>	グループオブジェクト属性値の取得／変更		2. 4. 7
<code>g_layer_attrib</code>	レイヤオブジェクト属性値の取得／変更		2. 5. 7
<code>g_paint_pattern_file</code>	塗りつぶしパターンビットマップファイルの指定	Ver. 4更新	2. 8. 1
色に関するマクロ			
<code>g_alloc_color</code>	色 ID への色の割付け／解除		2. 1. 2
<code>g_color_id</code>	図形オブジェクト色 ID の取得／変更	Ver. 4更新	2. 2. 13
<code>g_cur_color_id</code>	図形オブジェクト色 ID のカレント値取得／変更		2. 2. 17
<code>g_gwin_color</code>	グラフィックスウィンドウ色 ID の取得／変更		2. 4. 11
<code>g_gourp_color_id</code>	グループオブジェクト色 ID の取得／変更		2. 4. 8
<code>g_layer_color_id</code>	レイヤオブジェクト色 ID の取得／変更		2. 5. 8
文字属性に関するマクロ			
<code>g_cur_tx_attrib</code>	文字列属性値のカレント値の取得／変更	Ver. 5更新	2. 2. 24
<code>g_gwin_tx_attrib</code>	グラフィックスウィンドウ文字列属性値の取得／変更	Ver. 5更新	2. 4. 12
シンボルに関するマクロ			
<code>g_symbol</code>	シンボルオブジェクトの作成	Ver. 4更新	2. 9. 11
<code>g_bitmapsymbol</code>	ビットマップシンボルオブジェクトの作成	Ver. 4更新	2. 1. 5
<code>g_linesymbol</code>	ラインシンボルオブジェクトの作成	Ver. 4更新	2. 5. 13
<code>g_draw_symbol</code>	グラフィックスウィンドウへのシンボルの描画	Ver. 4更新	2. 3. 17

2 章グラフィックスマクロ (G マクロ)

シンボルに関するマクロ			
<code>g_symbol_file</code>	シンボル定義ファイルの設定	Ver. 4更新	2. 9. 12
<code>g_bitmapsymbol_file</code>	ビットマップシンボル定義ファイルの設定	Ver. 4更新	2. 1. 6
<code>g_linesymbol_file</code>	ラインシンボル定義ファイルの設定	Ver. 4更新	2. 5. 14
座標の変換			
<code>g_dc2wc</code>	DC 座標を WC 座標に変換		2. 3. 1
<code>g_wc2dc</code>	WC 座標を DC 座標に変換		2. 10. 10
<code>g_yoko_mercator</code>	緯度経度座標と XY 座標との変換	Ver. 4更新	2. 10. 11
<code>g_TKY2JGD_ParaFile</code>	Tokyo97 の緯度・経度を JGD2000 の緯度・経度に変換するパラメータファイルを指定	Ver. 5更新	2. 10. 4
<code>g_JBL2JXY</code>	JGD2000 の緯度・経度を平面直交座標系に変換	Ver. 5更新	2. 5. 6
<code>g_TBL2JBL</code>	Tokyo97 の緯度・経度を JGD2000 の緯度・経度に変換	Ver. 5更新	2. 10. 2
<code>g_TBL2TXY</code>	Tokyo97 の緯度・経度を平面直交座標系に変換	Ver. 5更新	2. 10. 3
フィルタ操作			
<code>g_open_filter</code>	フィルタオブジェクトの作成	Ver. 3更新	2. 7. 4
<code>g_filter_set_param</code>	フィルタオブジェクトのパラメータ作成	Ver. 3更新	2. 4. 5
<code>g_filter_add</code>	図形オブジェクトへのフィルタオブジェクトの登録	Ver. 3更新	2. 4. 2
<code>g_filter_remove</code>	図形オブジェクトからのフィルタオブジェクトの消去	Ver. 3更新	2. 4. 4
<code>g_filter_draw_obj</code>	フィルタを用いたグラフィックスウィンドウへの図形オブジェクト描画	Ver. 3更新	2. 4. 3
<code>g_close_filter</code>	フィルタオブジェクトの消去	Ver. 3更新	2. 2. 5

2 章グラフィックスマクロ (G マクロ)

イメージの操作			
g_open_image	イメージオブジェクトの作成	Ver. 3更新 Ver. 4更新	2. 7. 8
g_add_image	イメージオブジェクトへのイメージデータの追加	Ver. 3更新 Ver. 4更新	2. 1. 1
g_image_dsp	イメージオブジェクトの指定／取消し	Ver. 3更新 Ver. 4更新	2. 5. 4
g_close_image	イメージオブジェクトの消去	Ver. 3更新 Ver. 4更新	2. 2. 9
g_open_image_region	イメージオブジェクトの作成	Ver. 4更新	2. 7. 9
g_close_image_region	イメージオブジェクトの消去	Ver. 4更新	2. 2. 10
マップオブジェクトの操作			
g_open_map	マップオブジェクトの作成	Ver. 4更新	2. 7. 10
g_close_map	マップオブジェクトの消去	Ver. 4更新	2. 2. 12
g_map_element	マップオブジェクトの構成要素の取得／変更	Ver. 4更新	2. 6. 3
g_map_type	マップオブジェクトの種別の取得	Ver. 4更新	2. 6. 7
g_map_attrib	マップオブジェクト属性値の取得／変更	Ver. 4更新	2. 6. 2
g_map_range	マップオブジェクト表示縮尺範囲の取得／変更	Ver. 4更新	2. 6. 6
g_map_margin	マップオブジェクト先読み範囲の取得／変更	Ver. 4更新	2. 6. 5
g_map_load	マップオブジェクトの地図データの読み出し	Ver. 4更新	2. 6. 4
ツール類			
mkbsym. exe	ビットマップシンボル定義ファイルの作成	Ver. 4更新	2. 6. 9

2.1 A~B で始まるコマンド

2.1.1 g_add_image

イメージオブジェクトへのイメージデータの追加

注) Ver. 4 でマップ概念導入のため廃止した API です。

(1) 形式

```
g_add_image contextID imageID filename x1 y1 x2 y2
```

(2) 説明

g_add_image マクロは、イメージオブジェクトにイメージデータを追加します。設定範囲が以前に追加したイメージデータと重なる場合は、先に追加したイメージデータを優先します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② imageID

g_open_image で作成したイメージ ID を指定します。

③ filename

追加するイメージデータを収めたファイル名を指定します。Windows ビットマップ (***.bmp)、TIFF 形式 (***.tiff または***.tif) が指定できます。

④ x1 y1 x2 y2

イメージデータの設定範囲を指定します。左下座標 (x1、y1)、右上座標 (x2、y2) で指定します。

(3) 戻り値

イメージ ID

(4) 関連項目

g_open_image, g_open_gwin, g_close_image, g_image_dsp

(5) 使用例

なし

(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 Ver. 4.0

Ver. 4.0 で削除したマクロです。互換性用マクロファイル bcomp.psv を参考に g_open_image_region マクロ関連の機能に置き換えてください。

2.1.2 g_alloc_color

色 ID への色の割付け／解除

(1) 形式

```
g_alloc_color contextID [-w] [a|x] colorID [colorVal|colorName]
```

(2) 説明

g_alloc_color マクロは、色 ID への色の割付けと、割付けの解除を行います。色はカラー名称または RGB 値で指定します。

色の指定を省略すると、色 ID に現在割り付けている RGB 値を取得できます（カラー名称で色を割り付けた場合も RGB 値に変換します）。

① contextID

g_context で作成したコンテキスト ID を指定します。

② -w gwinID

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外を指定する場合に、g_open_gwin で作成したグラフィックスウィンドウ ID を指定します。色 ID は複数のグラフィックスウィンドウ間で共有するため、任意のグラフィックスウィンドウ指定が可能です。

a|x

色の割付けまたは解除を指定します。省略すると色の割付けになります。

a : 色の割付けを行います。

x : 色の割付けを解除します。

③ colorID

色 ID を指定します。範囲は 1～255 です。

1 から 8 の初期値は以下の RGB 値を設定しています。変更する場合は、1 から 8 の色 ID に対して、本マクロを実行して新しい色を割り付けてください。

色 ID	カラー名称	RGB 値
1	red	&ff0000
2	green	&00ff00
3	blue	&0000ff
4	cyan	&00ffff
5	yellow	&ffff00
6	magenta	&ff00ff
7	white	&ffffff
8	black	&000000

④ *colorVal* | *colorName*

割り付ける色を指定します。

colorVal: RGB値 (例: r=255, g=0, b=10 → &ff000a)

colorName: カラー名称 (例: red)

(3) 戻り値

色 (RGB 値)

(4) 関連項目

g_context, *g_open_gwin*

(5) 使用例

gsamp001.psv 参照 (以下画面例)

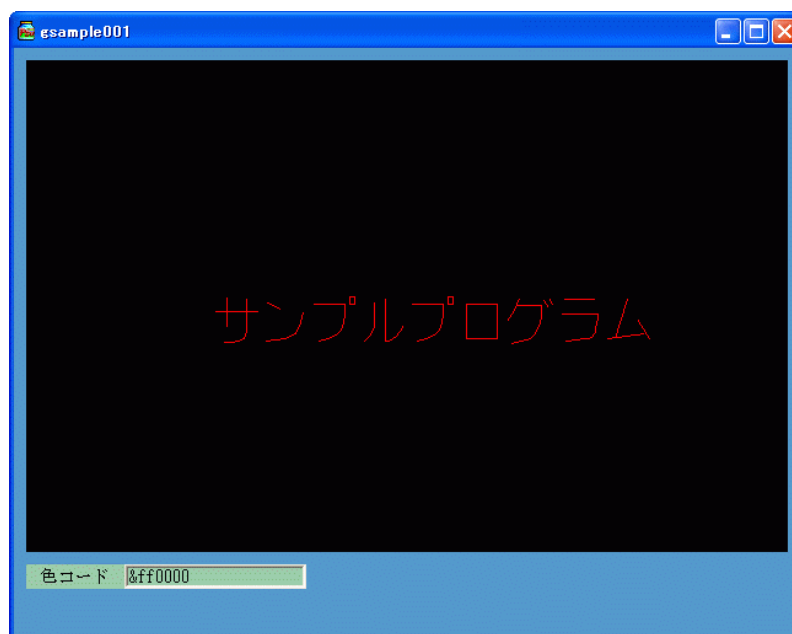


図 2-1 gsamp001.psv 実行画面

(6) 更新履歴

なし

2.1.3 g_arc

円弧オブジェクトの作成

(1) 形式

```
g_arc context/D [-dc] cx cy radius sdeg edeg [[-t] [-c] [-a] [-g] [-l] [-m]]  
[-u]
```

(2) 説明

g_arc マクロは円弧オブジェクトを作成します。

① *context/D*

g_context で作成したコンテキスト ID を指定します。

② *-dc*

座標を DC 座標で指定する時に指定します。

③ *cx cy radius sdeg edeg*

円弧の中心座標 (*cx*、*cy*)、半径 (*radius*)、開始角 (*sdeg*)、終了角 (*edeg*) を指定します。角度は反時計回り、1/10 度単位で指定します。

④ *-t tag/D*

図形オブジェクトのタグ ID を数字で指定します。

⑤ *-c color/D*

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑥ *-a attribVal* (Ver. 4 以降削除 A版互換モードの場合)

図形オブジェクトの属性を数字で指定します (*g_attr* 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑦ *-a objAttribVal drawAttribVal*

図形オブジェクトの属性値と描画属性値を数字で指定します (*g_obj_attr*、*g_draw_attr* 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ *-g [group/D] groupNo*

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (*group/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グループオブジェクトが対象となります。

⑨ *-l [layer/D] layerNo*

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID (*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ヤオブジェクトが対象となります。

⑩ *-m mode* (Ver. 4 以降削除 A版互換モード時有効)

図形オブジェクトのモードを数字で指定します (*g_mode* 参照)。

⑪ `-u key body [key body]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

円弧の図形オブジェクト ID

(4) 関連項目

`g_attrb`, `g_context`, `g_draw_obj`, `g_mode`, `g_pie`, `g_open_group`, `g_open_layer`,
`g_obj_attrb`, `g_draw_attrb`

(5) 使用例

`gsamp002.psv` 参照 (以下画面例)

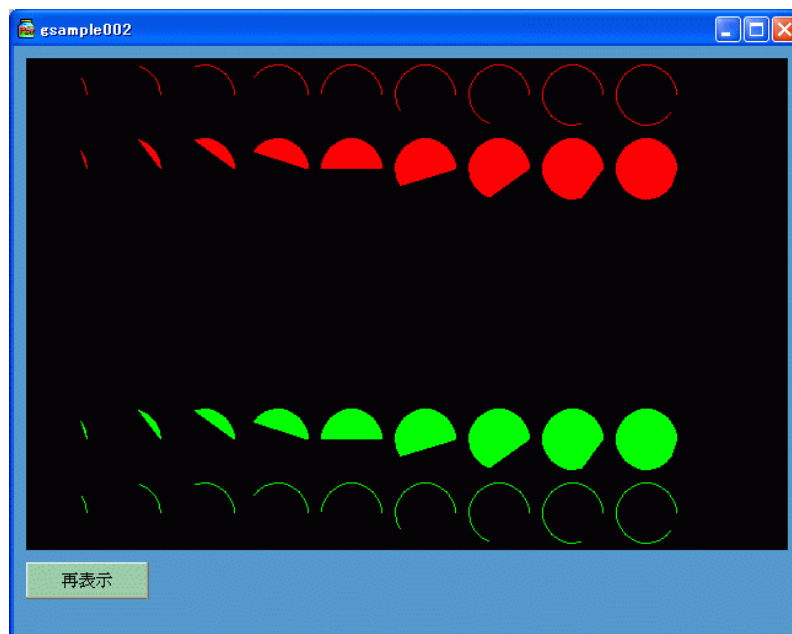


図 2-2 `gsamp002.psv` 実行画面

(6) 更新履歴

① Ver. 4.0

属性値のオプションの指定を変更しました (`-a`)。

モード指定のオプションを削除しました (`-m`)。

2.1.4 g_attrib

図形オブジェクトの属性値の取得／変更

注) Ver. 4 で図形属性を強化したため廃止した API です。

(1) 形式

`g_attrib objID [attribVal]`

(2) 説明

`g_attrib` マクロは、図形オブジェクトの属性値を取得／変更します。引数に属性値を指定しない時は、現在の属性値を取得します。指定した場合は属性値を変更して、変更後の値が返ります。

① `objID`

作成した図形オブジェクトの ID を指定します。

② `attribVal`

次の 32 ビットの整数で指定します。

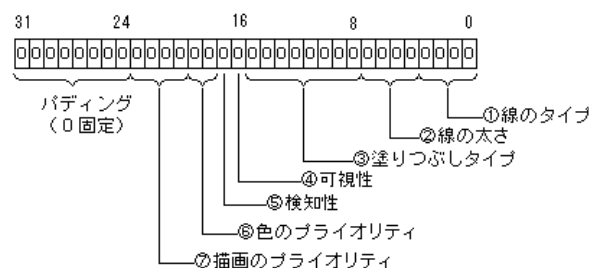


図 2-3 図形オブジェクト属性値ビット割り当て

(a) 線のタイプ (ビット 0~3)

0~4 の値の範囲で、線の種類を次の中から指定します。

- 0 : 実線
- 1 : 破線
- 2 : 点線
- 3 : 一点鎖線
- 4 : 二点鎖線

(b) 線の太さ (ビット 4~7)

0~15 の値の範囲で、線の太さを指定します。値が大きいくほど太くなります。

(c) 塗りつぶしタイプ (ビット 8~15)

図形の塗りつぶしタイプを指定します。

- 0 : ベタ
- 1 : 不透明 (指定色の輪郭線を描き、内部を背景色で塗りつぶし)
- 2 : 横線
- 3 : 斜め線 (右上がり)
- 4 : 斜め線 (右下がり)

- 5 : 縦線
- 6 : 縦横格子
- 7 : 斜め格子

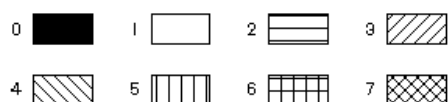


図 2-4 塗りつぶしタイプ

なお、図形オブジェクトを塗りつぶすかは、図形オブジェクトのモードの設定に従います (g_mode 参照)。

(d) **可視性** (ビット 16)

図形オブジェクトの可視性を指定します。不可視とすると、g_draw_obj で描画を実行してもその図形オブジェクトを描画しません。

0 : 不可視

1 : 可視

(e) **マウスの検知性** (ビット 17)

g_search_obj マクロで図形オブジェクトを検知するかを指定します。

0 : 検知しない

1 : 検知する

(f) **色のプライオリティ** (ビット 18~19)

図形オブジェクト描画時の、色の指定の優先順位を指定します。

a : 図形オブジェクトの設定色

b : 図形オブジェクトのグループ番号の設定色

c : 図形オブジェクトのレイヤ番号の設定色

優先順位にしたがって色の指定有無をチェックして、指定があればその色で描画し、指定がなければ次の指定の有無をチェックします。

0 : 優先順位が高い順より a、b、c の順 (既定値)

1 : 優先順位が高い順より b、a、c の順

2 : 優先順位が高い順より c、a、b の順

(g) **描画のプライオリティ** (ビット 20~23)

0~15 の範囲で値を指定します。複数の図形オブジェクトを一括して描画する時、どの図形から先に描画するかプライオリティを指定します。数値が大きいものほど後から描画するため、画面上では前面に表示します。

(3) **戻り値**

対象図形オブジェクトの属性値

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_round_box`, `g_regpoly`,

(5) 使用例

gsamp003.psv 参照 (以下画面例)

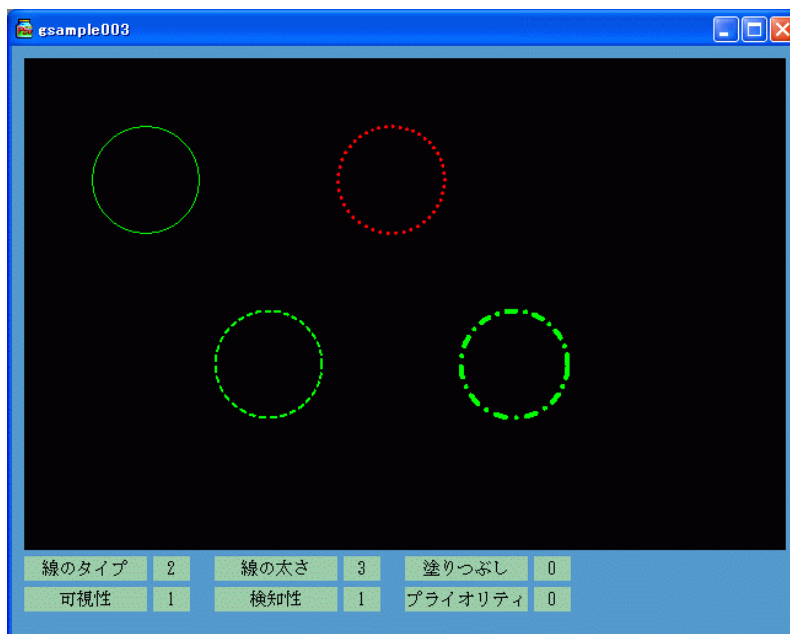


図 2-5 gsamp003.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で削除したマクロです。互換性用マクロファイル `bcomp.psv` を参考にし
て、`g_obj_attrib`、`g_draw_attrib` マクロに置き換えてください。

2.1.5 g_bitmapsymbol

ビットマップシンボルオブジェクトの作成

(1) 形式

```
g_bitmapsymbol context/D [-dc] sx sy ex ey symNo [pos] [[-t] [-c] [-a] [-g]
[-l]] [-u]
```

(2) 説明

g_bitmapsymbol マクロはビットマップシンボルオブジェクトを作成します。

① context/D

g_context で作成したコンテキスト ID を指定します。

② -dc

座標を DC 座標で指定する時に指定します。

③ sx sy ex ey

シンボルの左下座標 (sx, sy)、右上座標 (ex, ey) を指定します。

可変サイズ指定時は、左下座標、右上座標の中心が基準点となります。

④ symNo

g_bitmapsymbol_file で指定したシンボルファイル中のシンボル番号を数字で指定します。

⑤ pos

0 の時は可変サイズとなります。1~9 の場合は固定サイズとなり、グラフィックスウィンドウの縮尺に関係なく、常に同じ表示サイズです。省略すると 0 (可変サイズ) となります。

基準点は以下となります。

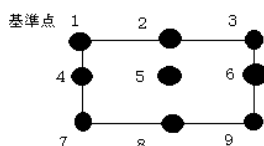


図 2-6 pos の値と基準点の関係

注意) 固定サイズ指定時は、g_map_attrib のアクティブサイズを「1: 再計算を行う」に設定してください。

⑥ -t tag/D

図形オブジェクトのタグ ID を数字で指定します。

⑦ -c color/D

ビットマップシンボルでは、色 ID 指定は無効です。

⑧ -a objAttribVal drawAttribVal

図形オブジェクトの属性値と描画属性値を数字で指定します (g_obj_attrib、g_draw_attrib 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑨ `-g [groupID] groupNo`

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクトID (*groupID*) 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グループオブジェクトが対象となります。

⑩ `-l [layerID] layerNo`

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクトID (*layerID*) 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ヤオブジェクトが対象となります。

⑪ `-u key body [key body]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

シンボルの図形オブジェクト ID

(4) 関連項目

`g_context`, `g_draw_obj`, `g_bitmapsymbol_file`, `g_open_group`, `g_open_layer`,
`g_obj_attrib`, `g_draw_attrib`

(5) 使用例

`gsamp105.psv` 参照 (以下画面例)

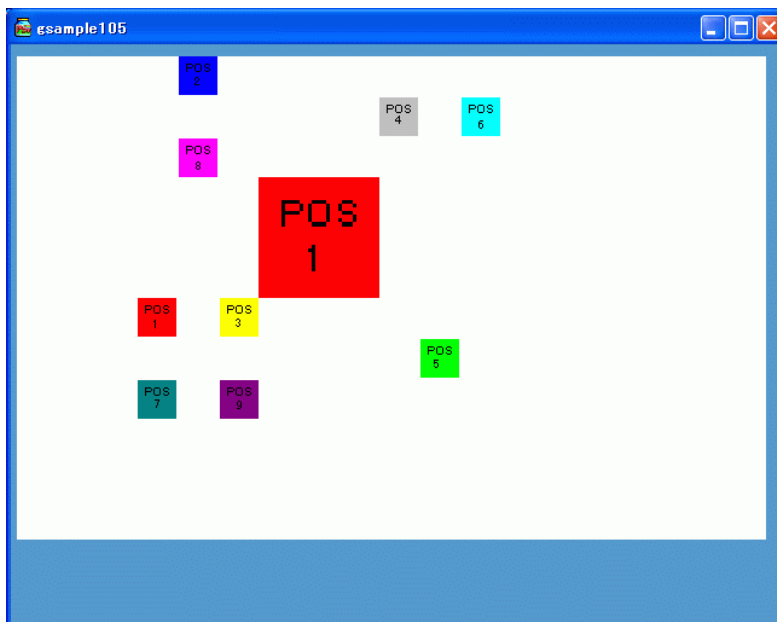


図 2-7 `gsamp105.psv` 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.1.6 g_bitmapsymbol_file

ビットマップシンボル定義ファイルの設定

(1) **形式**

`g_bitmapsymbol_file [fileName] | [-s FileID]`

(2) **説明**

`g_bitmapsymbol_file` マクロは、ビットマップシンボル定義ファイルを指定／取得します。ビットマップシンボル定義ファイルは付属のツール `mkbsym.exe` を利用して作成してください。

① *fileName*

ビットマップシンボル定義ファイルを指定します。引数にファイル名を指定しない時はビットマップシンボル定義ファイルの現在の登録 ID が返ります。

② `-s FileID`

既に登録済みのシンボル定義ファイルを指定する場合は、登録時の戻り値である登録 ID を指定します。

(3) **戻り値**

ビットマップシンボル定義ファイルの登録 ID

(4) **関連項目**

`g_bitmapsymbol`

(5) **使用例**

`gsamp105.psv`参照 (画面例は2.1.5の画面参照)

(6) **更新履歴**

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.2 G で始まるコマンド

2.2.1 g_circle

円オブジェクトの作成

(1) 形式

```
g_circle contextID [-dc] cx cy radius [[-t] [-c] [-a] [-g] [-l] [-m]] [-u]
```

(2) 説明

`g_circle` マクロは、円オブジェクトを作成します。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-dc`

座標を DC 座標で指定する時に指定します。

③ `cx cy radius`

円の中心座標 (`cx`, `cy`)、半径 (`radius`) を指定します。

④ `-t tagID`

図形オブジェクトのタグ ID を数字で指定します。

⑤ `-c colorID`

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑥ `-a attribVal` (Ver. 4 以降削除 A版互換モードの場合)

図形オブジェクトの属性を数字で指定します (`g_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑦ `-a objAttribVal drawAttribVal`

図形オブジェクトの属性値と描画属性値を数字で指定します (`g_obj_attrib`、`g_draw_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ `-g [groupID] groupNo`

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクトID (`groupID`) 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グループオブジェクトが対象となります。

⑨ `-l [layerID] layerNo`

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクトID (`layerID`) 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ヤオブジェクトが対象となります。

⑩ `-m mode` (Ver. 4 以降削除 A版互換モード時有効)

図形オブジェクトのモードを数字で指定します (`g_mode` 参照)。

⑪ `-u key body [key body]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) **戻り値**

円の図形オブジェクト ID

(4) **関連項目**

`g_attrb`, `g_context`, `g_draw_obj`, `g_mode`, `g_open_group`, `g_open_layer`,
`g_obj_attrb`, `g_draw_attrb`

(5) **使用例**

gsamp003.psv参照 (画面例は2. 1. 4の画面参照)

(6) **更新履歴**

① Ver. 4. 0

属性値のオプションの指定を変更しました (-a) 。

モード指定のオプションを削除しました (-m) 。

2.2.2 g_clear

グラフィックスウィンドウのクリア

(1) 形式

`g_clear context/D [-w] [[-c] [-t]]`

(2) 説明

`g_clear` マクロはグラフィックスウィンドウの表示をクリアします。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwin/D`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外を指定する場合には、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-c color/D`

グラフィックスウィンドウのクリアする色を変える場合に、色 ID を数字で指定します。`-1` を指定した場合は、透明（背景色無し）となります。

④ `-t`

一時的な塗りつぶしの場合に指定します。本引数を指定した場合、クリアに指定した色 ID はクリア時だけで、次のグラフィックスウィンドウ再描画時には、元の色でクリアします。

(3) 戻り値

グラフィックスウィンドウの色 ID

(4) 関連項目

`g_context`, `g_open_gwin`

(5) 使用例

`gsamp002.psv` 参照 (画面例は2.1.3の画面参照)

(6) 更新履歴

① Ver. 4.0

透明な背景色の指定が可能となりました。

2.2.3 g_close_context

コンテキストオブジェクトの消去

(1) **形式**

`g_close_context context/D [-r]`

(2) **説明**

`g_close_context` マクロは、コンテキストオブジェクトを消去します。

① *context/D*

`g_context` で作成したコンテキスト ID を指定します。

② `-r`

コンテキストオブジェクトに設定しているカレントのディスプレイリストメモリ、グラフィックスウィンドウ、レイヤオブジェクト、グループオブジェクトを同時に消去する場合に指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_group`, `g_open_gwin`, `g_open_group`, `g_open_layer`

(5) **使用例**

`gsamp002.psv`参照 (画面例は2.1.3の画面参照)

(6) **更新履歴**

なし

2.2.4 g_close_dlist

ディスプレイリストメモリの消去

(1) **形式**

`g_close_dlist contextID [-d]`

(2) **説明**

`g_close_dlist` マクロは、ディスプレイリストメモリを消去します。

ディスプレイリストに格納している図形オブジェクトは、自動的に削除します。
ただし、図形オブジェクトが、他のディスプレイリストメモリ、グループオブジェクトに格納されている場合は、削除しません。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② `-d dlistID`

コンテキストオブジェクトに設定しているディスプレイリストメモリ以外のディスプレイリストメモリを消去する場合に、`g_open_dlist` で作成したディスプレイリストメモリ ID を指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_dlist`

(5) **使用例**

`gsamp002.psv` 参照 (画面例は2.1.3の画面参照)

(6) **更新履歴**

なし

2.2.5 g_close_filter

フィルタオブジェクトの消去

(1) 形式

```
g_close_filter contextID filterID [filterID...]
```

(2) 説明

フィルタオブジェクトを消去します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② filterID [filterID...]

g_open_filter で作成したフィルタオブジェクト ID を指定します。

(3) 戻り値

なし

(4) 関連項目

g_open_filter, g_filter_set_param, g_filter_add, g_filter_remove,
g_filter_draw_obj

(5) 使用例

gsamp063.psv 参照 (以下画面例)

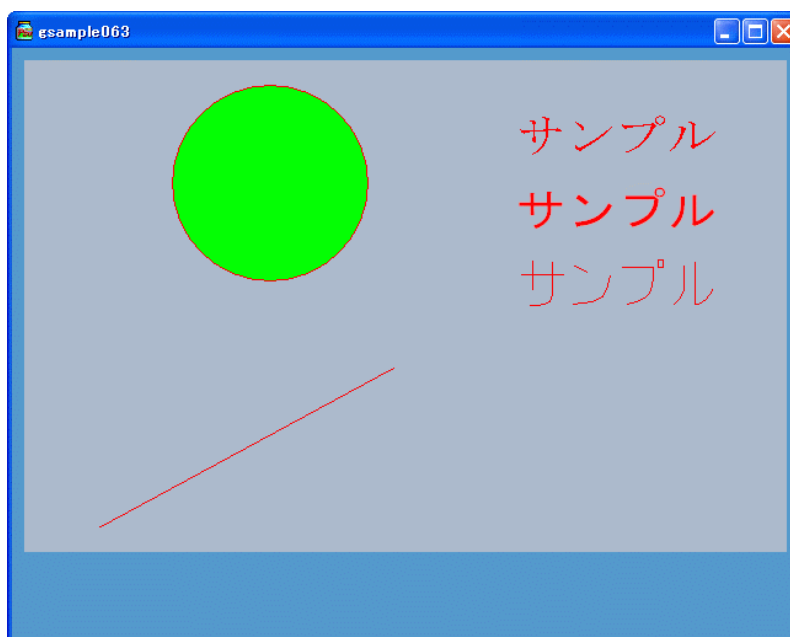


図 2-8 gsamp063.psv 実行画面

(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。

2.2.6 g_close_group

グループオブジェクトの消去

(1) 形式

`g_close_group contextID [-g]`

(2) 説明

`g_close_group` マクロは、グループオブジェクトを消去します。

グループオブジェクトに格納していた図形オブジェクトは、自動的に削除します。ただし、図形オブジェクトが、他のディスプレイリストメモリ、グループオブジェクトに格納されている場合は、削除しません。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-g groupID*

コンテキストオブジェクトに設定しているグループオブジェクト以外を指定する場合に、`g_open_group` で作成したグループオブジェクト ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_group`

(5) 使用例

`gsamp002.psv`参照 (画面例は2.1.3の画面参照)

(6) 更新履歴

なし

2.2.7 g_close_gwin

グラフィックスウィンドウの消去

(1) **形式**

`g_close_gwin context/D [-w]`

(2) **説明**

`g_close_gwin` マクロは、グラフィックスウィンドウを消去します。

① *context/D*

`g_context` で作成したコンテキスト ID を指定します。グラフィックスウィンドウの背景にイメージオブジェクトを指定している場合、本マクロではイメージオブジェクトは消去しません。`g_close_image` で明示的に消去してください。

② *-w gwin/D*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外を消去する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_gwin`

(5) **使用例**

`gsamp002.psv`参照 (画面例は2.1.3の画面参照)

(6) **更新履歴**

なし

2.2.8 g_close_hash

ハッシュオブジェクトの消去

(1) 形式

`g_close_hash hashID [hashID...]`

(2) 説明

ハッシュオブジェクトを消去します。

① `hashID [hashID...]`

`g_open_hash` で作成したハッシュオブジェクト ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_open_hash`, `g_hash_store`, `g_hash_fetch`, `g_hash_del`

(5) 使用例

gsamp066.psv 参照 (以下画面例)



図 2-9 gsamp066.psv 実行画面

(6) 更新履歴

なし

2.2.9 g_close_image

イメージオブジェクトの消去

注) Ver. 4 でマップ概念導入のため廃止した API です。

(1) **形式**

`g_close_image imageID`

(2) **説明**

`g_close_image` マクロは、イメージオブジェクトを消去します。

① *imageID*

`g_open_image` で作成したイメージオブジェクト ID を指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_add_image`, `g_open_gwin`, `g_open_image`, `g_image_dsp`

(5) **使用例**

なし

(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 Ver. 4.0

Ver. 4.0 で削除したマクロです。互換性用マクロファイル `bcomp.psv` を参考に `g_open_image_region` マクロ関連の機能に置き換えてください。

2.2.10 g_close_image_region

イメージオブジェクトの消去

(1) 形式

```
g_close_image_region imageID [imageID ...]
```

(2) 説明

g_close_image_region マクロは、イメージオブジェクトを消去します。

① imageID [imageID]

g_open_image_region で作成したイメージオブジェクト ID を指定します。

(3) 戻り値

なし

(4) 関連項目

g_open_image_region, g_open_map

(5) 使用例

gsamp099.psv 参照 (以下画面例)



図 2-10 gsamp099.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.2.11 g_close_layer

レイヤオブジェクトの消去

(1) **形式**

`g_close_layer contextID [-l]`

(2) **説明**

`g_close_layer` マクロは、レイヤオブジェクトを消去します。

複数のレイヤオブジェクト ID を `g_open_layer` で作成した場合、`g_close_layer` で、個々にレイヤオブジェクト ID を消去してください。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-l layerID...*

コンテキストオブジェクトに設定しているレイヤオブジェクト以外のレイヤオブジェクトを指定する場合、`g_open_layer` で作成したレイヤオブジェクト ID を指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_layer`

(5) **使用例**

gsamp002.psv参照 (画面例は2.1.3の画面参照)

(6) **更新履歴**

なし

2.2.12 g_close_map

マップオブジェクトの消去

(1) 形式

`g_close_map mapID`

(2) 説明

`g_close_map` マクロは、マップオブジェクトを消去します。

構成要素のディスプレイリストメモリやイメージ領域は削除しません。明示的に `g_close_dlist` や `g_close_image_region` を使用して削除してください。

注意： マップオブジェクトを自動描画モードに設定している場合、`g_redraw` を解除してから実行してください。

① `mapID`

`g_open_map` で作成したマップオブジェクト ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_open_map`, `g_close_dlist`, `g_close_image_region`

(5) 使用例

`gsamp102.psv` 参照 (以下画面例)

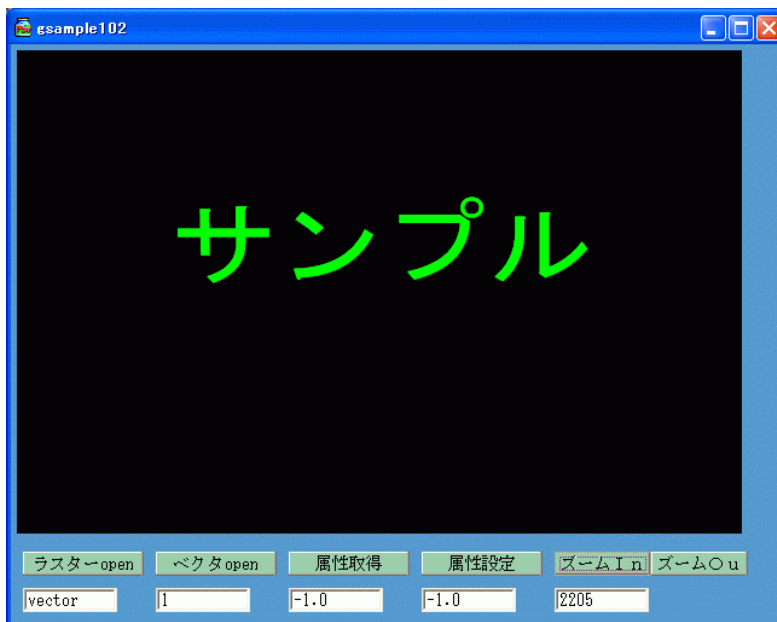


図 2-11 `gsamp102.psv` 実行画面

`gsamp099.psv` 参照 (画面例は 2.2.10 の画面参照)

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.2.13 g_color_id

図形オブジェクトの色 ID の取得／変更

(1) 形式

`g_color_id objID [colorID]`

(2) 説明

`g_color_id` マクロは、図形オブジェクトの色 ID を取得／変更します。

引数に色 ID を指定しない時は現在の色 ID を返します。指定した場合は色 ID が変更され、変更後の色 ID を返します。

① *objID*

作成した図形オブジェクトの ID を指定します。

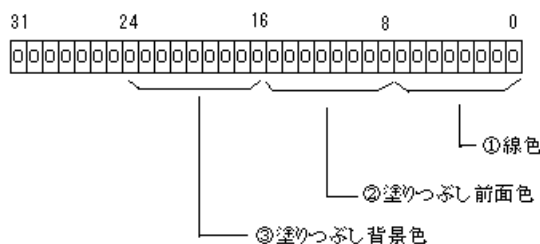
② *colorID*`g_alloc_color` で作成した色 ID を数字で指定します。塗りつぶし色と線色の個別指定は `g_draw_attrib` で図形の塗りつぶしモードが個別指定の場合有効です。一括指定の場合は、塗りつぶしにも線色を使用します。

図 2-12 色 ID のビットの割り当て

(a) 線色 (ビット 0～ビット 7)

0～255 の範囲の値で、図形オブジェクトの線色を指定します。

(b) 塗りつぶし前面色 (ビット 8～ビット 15)

0～255 の範囲の値で、図形オブジェクトの塗りつぶし前面色を指定します。

(c) 塗りつぶし背景色 (ビット 16～ビット 23)

0～255 の範囲の値で、図形オブジェクトの塗りつぶし背景色を指定します。

`g_draw_attrib` で図形の塗りつぶしモードが不透明の時に有効です。

(3) 戻り値

色 ID

(4) 関連項目

`g_alloc_color`, `g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`,
`g_ellipse`, `g_text`, `g_symbol`, `g_usr_def`, `g_round_box`, `g_regpoly`,
`g_draw_attrib`

(5) 使用例

gsamp041.psv 参照 (以下画面例)

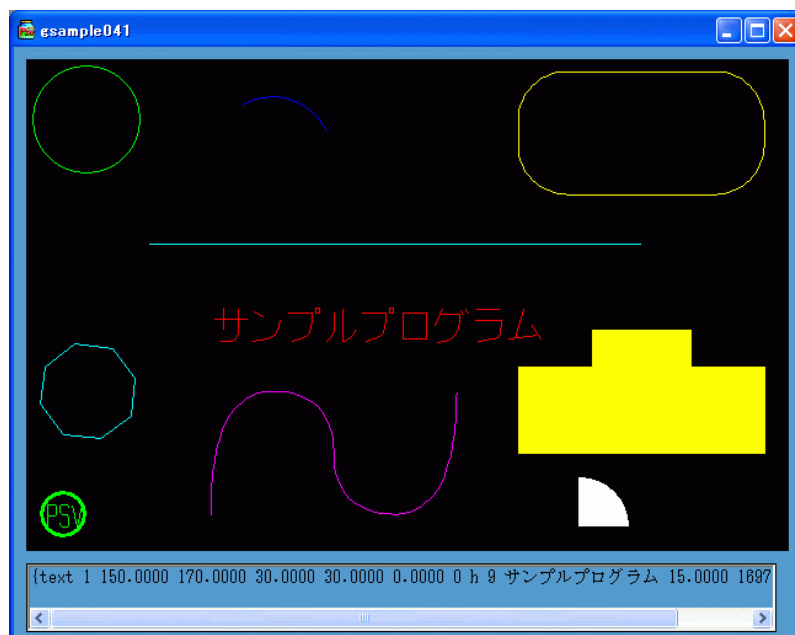


図 2-13 gsamp041.psv 実行画面

(6) 更新履歴

① Ver. 4.0

塗りつぶし色と線色が個別に指定可能となりました。

2.2.14 g_compute_size

グラフィックスウィンドウの DC、WC の比率の再設定

(1) 形式

`g_compute_size context/D [-w]`

(2) 説明

`g_compute_size` マクロは、グラフィックスウィンドウとキャンバス（ビューポート）の DC 座標範囲と、WC 座標範囲の比率を再計算し、設定します。

本マクロは、キャンバスのサイズを変更した場合に使用します。グラフィックスウィンドウのサイズ変更、ビューポートの設定時には使用する必要はありません。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwin/D`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外を指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_viewport`

(5) 使用例

`gsamp014.psv` 参照（以下画面例）

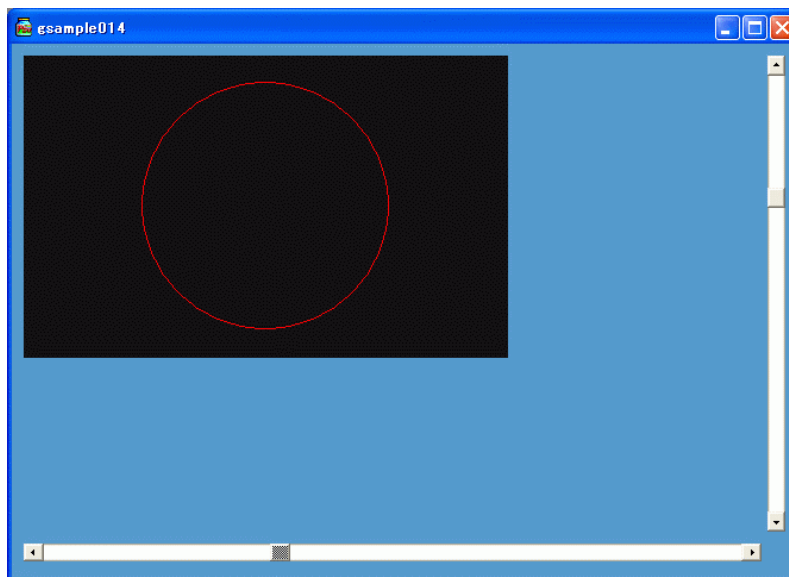


図 2-14 `gsamp014.psv` 実行画面

(6) 更新履歴

なし

2.2.15 g_context

コンテキストオブジェクトの作成

(1) **形式**

`g_context`

(2) **説明**

`g_context` マクロはコンテキストオブジェクトを作成します。コンテキストオブジェクトは、以下のオブジェクト ID と値をカレント値として保持します。これらの値を、マクロを実行する際の指定省略値として適用します。

- ディスプレイリストメモリ ID
- グラフィックスウィンドウ ID
- グループオブジェクト ID
- レイヤオブジェクト ID
- 図形オブジェクトの属性値
- 図形オブジェクトの色 ID
- 図形オブジェクトの文字列属性

(3) **戻り値**

コンテキストオブジェクトのコンテキスト ID

(4) **関連項目**

`g_open_gwin`, `g_open_dlist`, `g_open_group`, `g_open_layer`, `g_cur_attr`,
`g_cur_color_id`, `g_cur_dlist`, `g_cur_gwin`, `g_cur_layer`, `g_cur_tx_attr`

(5) **使用例**

`gsamp002.psv`参照 (画面例は2.1.3の画面参照)

(6) **更新履歴**

なし

2.2.16 g_cur_attrb

コンテキストオブジェクトの図形オブジェクト属性値の取得／変更

注) Ver. 4 で図形属性を強化したため廃止した API です。

(1) **形式**

`g_cur_attrb contextID [attribVal]`

(2) **説明**

`g_cur_attrb` マクロは、コンテキストオブジェクトのカレント図形オブジェクト属性値を取得／変更します。引数に属性値を指定しない時は、カレント値を返します。指定する場合はカレント値を変更して、変更後の値を返します。

本マクロの設定値は、図形オブジェクト作成時の既定値となります。描画時の設定値は、グラフィックスウィンドウで保持します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *attribVal*

変更する属性値を数字で指定します (`g_attrb` 参照)。

(3) **戻り値**

図形オブジェクト属性値

(4) **関連項目**

`g_context`, `g_attrb`, `g_cur_color_id`, `g_cur_tx_attrb`, `g_cur_dlist`,
`g_cur_gwin`, `g_cur_layer`, `g_cur_group`

(5) 使用例

gsamp039.psv 参照 (以下画面例)



図 2-15 gsamp039.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で削除したマクロです。互換性用マクロファイル bcomp.psv を参考に g_cur_obj_attr、g_cur_draw_attr マクロに置き換えてください。

2.2.17 g_cur_color_id

コンテキストオブジェクトの色 ID の取得／変更

(1) **形式**

`g_cur_color_id contextID [colorID]`

(2) **説明**

`g_cur_color_id` マクロは、コンテキストオブジェクトのカレントの色 ID を取得／変更します。引数に色 ID を指定しない時は、カレント値を返します。指定した場合はカレント値を変更して、変更後の値を返します。

本マクロの設定値は、図形オブジェクト作成時の既定値となります。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *colorID*

`g_alloc_color` で作成した色 ID を数字で指定します。

(3) **戻り値**

色 ID

(4) **関連項目**

`g_context`, `g_alloc_color`, `g_cur_attr`, `g_cur_tx_attr`, `g_cur_dlist`,
`g_cur_gwin`, `g_cur_layer`, `g_cur_group`

(5) **使用例**

gsamp039.psv参照 (画面例は2.2.16の画面参照)

(6) **更新履歴**

なし

2.2.18 g_cur_dlist

コンテキストオブジェクトのディスプレイリストメモリ ID の取得／変更

(1) **形式**

`g_cur_dlist contextID [dlistID]`

(2) **説明**

`g_cur_dlist` マクロは、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を取得／変更します。引数にディスプレイリストメモリ ID を指定しない時は、カレント値を返します。指定した場合はカレント値を変更して、変更後の値を返します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *dlistID*

ディスプレイリストメモリ ID を指定します。

(3) **戻り値**

ディスプレイリストメモリ ID

(4) **関連項目**

`g_context`, `g_open_dlist`, `g_cur_attrb`, `g_cur_tx_attrb`, `g_cur_color_id`,
`g_cur_gwin`, `g_cur_layer`, `g_cur_group`, `g_open_dlist`

(5) **使用例**

`gsamp039.psv`参照 (画面例は2.2.16の画面参照)

(6) **更新履歴**

なし

2.2.19 g_cur_draw_attrb

コンテキストオブジェクトの図形オブジェクト描画属性値の取得／変更

(1) **形式**

`g_cur_draw_attrb contextID [attribVal]`

(2) **説明**

`g_cur_draw_attrb` マクロは、コンテキストオブジェクトのカレント図形オブジェクト描画属性値を取得／変更します。引数に属性値を指定しない時は、カレント値を返します。指定した場合はカレント値を変更して、変更後の値を返します。本マクロの設定値は、図形オブジェクト作成時の既定値となります。描画時の設定値は、グラフィックスウィンドウで保持します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *attribVal*

変更する属性値を数字で指定します (`g_draw_attrb` 参照)。

(3) **戻り値**

属性値

(4) **関連項目**

`g_context`, `g_attrb`, `g_cur_color_id`, `g_cur_tx_attrb`, `g_cur_dlist`,
`g_cur_gwin`, `g_cur_layer`, `g_cur_group`, `g_cur_attrb`

(5) 使用例

gsamp098.psv 参照 (以下画面例)

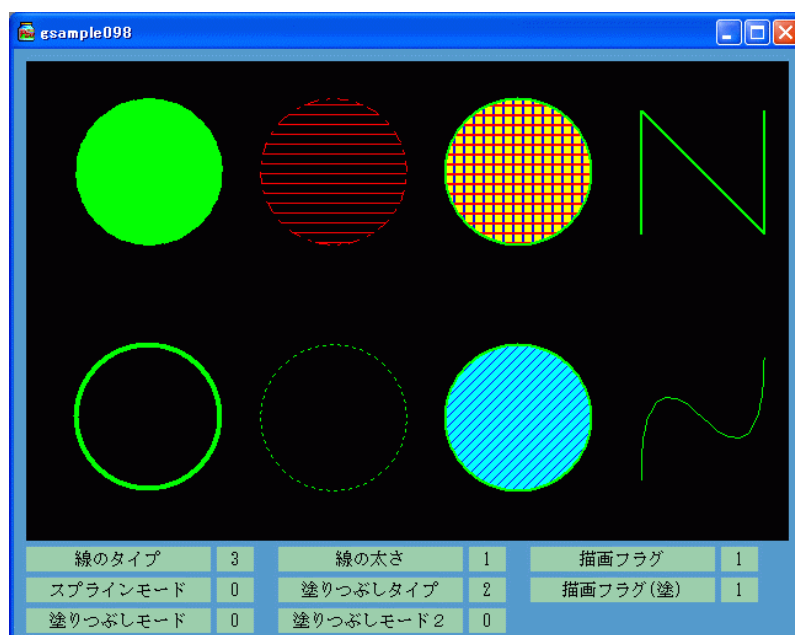


図 2-16 gsamp098.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2. 2. 20 g_cur_group

コンテキストオブジェクトのグループオブジェクトの取得／変更

(1) 形式

`g_cur_group context/D [group/D]`

(2) 説明

`g_cur_group` マクロは、コンテキストオブジェクトのカレントのグループオブジェクトを取得／変更します。引数にグループオブジェクト ID を指定しない時は、カレント値を返します。指定した場合はカレント値を変更して、変更後の値を返します。

① *context/D*

`g_context` で作成したコンテキスト ID を指定します。

② *group/D*

グループオブジェクト ID を指定します。

(3) 戻り値

グループオブジェクト ID

(4) 関連項目

`g_context`, `g_open_group`, `g_cur_attr`, `g_cur_tx_attr`, `g_cur_color_id`,
`g_cur_dlist`, `g_cur_gwin`

(5) 使用例

`gsamp039.psv` 参照 (画面例は 2. 2. 16 の画面参照)

(6) 更新履歴

なし

2.2.21 g_cur_gwin

コンテキストオブジェクトのグラフィックスウィンドウの取得／変更

(1) **形式**

`g_cur_gwin contextID [gwinID]`

(2) **説明**

`g_cur_gwin` マクロは、コンテキストオブジェクトのカレントのグラフィックスウィンドウを取得／変更します。引数にグラフィックスウィンドウの ID を指定しない時は、カレント値を返します。指定した場合はカレント値を変更して、変更後の ID を返します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *gwinID*

グラフィックスウィンドウ ID を指定します。

(3) **戻り値**

グラフィックスウィンドウ ID

(4) **関連項目**

`g_context`, `g_open_gwin`, `g_cur_attr`, `g_cur_tx_attr`, `g_cur_color_id`,
`g_cur_dlist`, `g_cur_layer`, `g_cur_group`

(5) **使用例**

gsamp039.psv参照 (画面例は2.2.16の画面参照)

(6) **更新履歴**

なし

2. 2. 22 g_cur_layer

コンテキストオブジェクトのレイヤオブジェクトの取得／変更

(1) **形式**

`g_cur_layer contextID [/layerID]`

(2) **説明**

`g_cur_layer` マクロは、コンテキストオブジェクトのカレントのレイヤオブジェクトを取得／変更します。引数にレイヤオブジェクト ID を指定しない時は、カレント値を返します。指定した場合はカレント値を変更して、変更後の値を返します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② */layerID*

レイヤオブジェクト ID を指定します。

(3) **戻り値**

レイヤオブジェクト ID

(4) **関連項目**

`g_context`, `g_open_layer`, `g_cur_attrib`, `g_cur_tx_attrib`, `g_cur_color_id`,
`g_cur_dlist`, `g_cur_gwin`, `g_cur_group`

(5) **使用例**

`gsamp039.psv`参照 (画面例は2. 2. 16の画面参照)

(6) **更新履歴**

なし

2.2.23 g_cur_obj_attrib

コンテキストオブジェクトの図形オブジェクト属性値の取得／変更

(1) **形式**

`g_cur_obj_attrib contextID [attribVal]`

(2) **説明**

`g_cur_obj_attrib` マクロは、コンテキストオブジェクトのカレント図形オブジェクト属性値を取得／変更します。引数に属性値を指定しない時は、カレント値が取得されます。指定する場合はカレント値が変更され、変更後の値が返ります。本マクロの設定値は、図形オブジェクト作成時の既定値となります。描画時の設定値は、グラフィックスウィンドウで保持します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *attribVal*

変更する属性値を数字で指定します (`g_obj_attrib` 参照)。

(3) **戻り値**

属性値

(4) **関連項目**

`g_context`, `g_attrib`, `g_cur_color_id`, `g_cur_tx_attrib`, `g_cur_dlist`,
`g_cur_gwin`, `g_cur_layer`, `g_cur_group`, `g_cur_attrib`

(5) 使用例

gsamp097.psv 参照 (以下画面例)

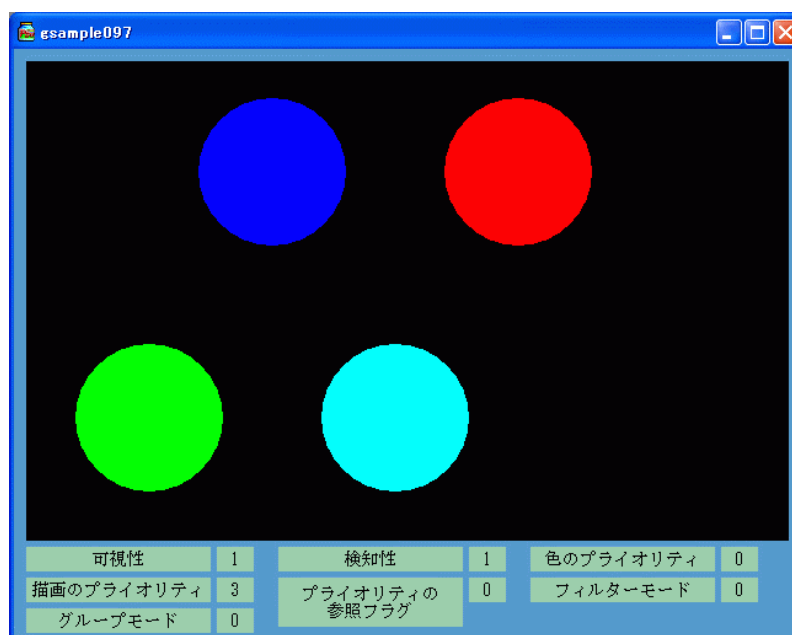


図 2-17 gsamp097.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.2.24 g_cur_tx_attrib

コンテキストオブジェクトの文字列属性値の取得／変更

(1) 形式

`g_cur_tx_attrib contextID [-h] [-w] [-r] [-o] [-s] [-f]`

(2) 説明

`g_cur_tx_attrib` マクロは、カレント値がコンテキストオブジェクトのカレント文字列属性値を取得／変更します。各引数で属性値を指定しない時は、カレント値が取得されます。指定する場合はカレント値が変更され、変更後の値が返ります。本マクロの設定値は、図形オブジェクト作成時の既定値となります。描画時の設定値は、グラフィックスウィンドウで保持します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② `-h [heightVal]`

文字高さを数字で指定します (WC 座標単位)。

③ `-w [widthVal]`

文字幅を数字で指定します (WC 座標単位)。

④ `-r [deg]`

文字列の角度を反時計回り、1/10 度単位で指定します。

⑤ `-o [h|v]`

文字列方向を指定します。

h: 横書き

v: 縦書き

⑥ `-s [spaceVal]`

文字間隔を数字で指定します (WC 座標単位)。

⑦ `-f [fontType]`

フォントタイプを数字で指定します (0: ベクトルフォント, 1: MS ゴシック, 2: MS 明朝)。

(3) 戻り値

カレントの属性値。`-h` 以降の指定した引数に対応する値が、リスト形式で返ります。

(4) 関連項目

`g_context`, `g_cur_color_id`, `g_cur_attrib`, `g_cur_dlist`, `g_cur_gwin`,
`g_cur_layer`, `g_cur_group`

(5) **使用例**

gsamp039.psv参照 (画面例は2. 2. 16の画面参照)

(6) **更新履歴**

① Ver. 5. 0

フォントタイプ用のオプション“-f”を追加しました。

2.3 D で始まるコマンド

2.3.1 g_dc2wc

DC 座標を WC 座標に変換

(1) 形式

`g_dc2wc contextID [-w] dx dy`

(2) 説明

`g_dc2wc` マクロは、グラフィックスウィンドウに対応するキャンバス（ビューポート）の DC 座標を WC 座標に変換します。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwinID`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外を指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `dx dy`

DC座標の点 (`dx`, `dy`) を指定します。

(3) 戻り値

WC 座標の X 座標、Y 座標（リスト形式）

(4) 関連項目

`g_context`, `g_wc2dc`, `g_open_gwin`, `g_scroll`, `g_rotate`, `g_viewport`, `g_ortho`

(5) 使用例

gsamp023.psv 参照 (以下画面例)

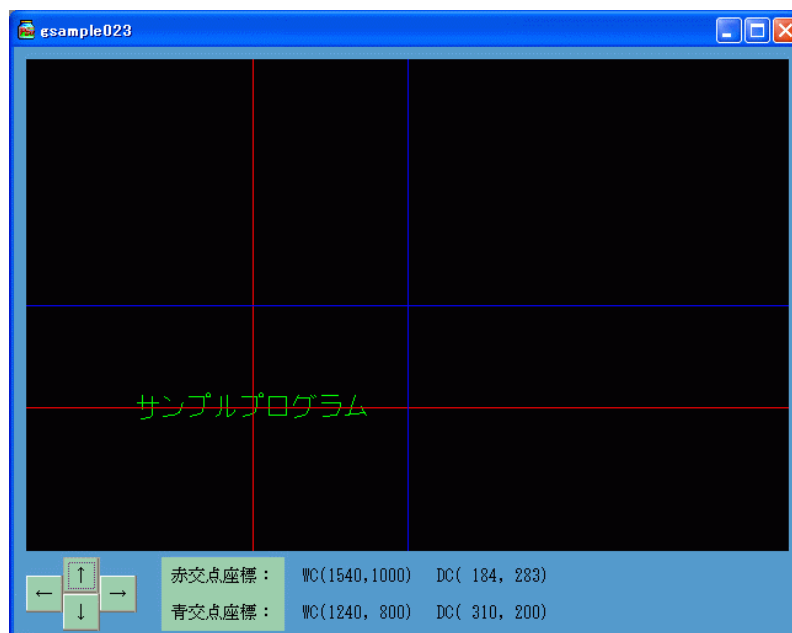


図 2-18 gsamp023.psv 実行画面

(6) 更新履歴

なし

2.3.2 g_del_obj

図形オブジェクトの消去

(1) 形式

```
g_del_obj contextID objID [objID ....]
```

```
g_del_obj contextID [-d] -g|-l
```

(2) 説明

g_del_obj マクロは、図形オブジェクトを消去します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② objID [objID...]

削除する図形オブジェクト ID を指定します。

③ -d dlist ...

-l 指定の時、検索するディスプレイリストメモリを指定します。省略した場合は、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となります。

④ -g [groupID] groupNo...

グループ番号を指定して、グループ番号単位で図形オブジェクトを消去します。グループオブジェクトID (groupID) 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グループオブジェクトが対象となります。

⑤ -l [layerID] layerNo

レイヤ番号を指定して、レイヤ番号単位で図形オブジェクトを消去します。ただし、レイヤ番号指定時の検索対象は、ディスプレイリストに格納している図形オブジェクトです。

レイヤオブジェクトID (layerID) 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ヤオブジェクトが対象となります。

(3) 戻り値

なし

(4) 関連項目

g_context, g_line, g_polyline, g_polygon, g_circle, g_arc, g_pie, g_ellipse,
g_text, g_symbol, g_usr_def, g_open_group, g_open_layer, g_round_box,
g_regpoly

(5) 使用例

gsamp024.psv 参照 (以下画面例)

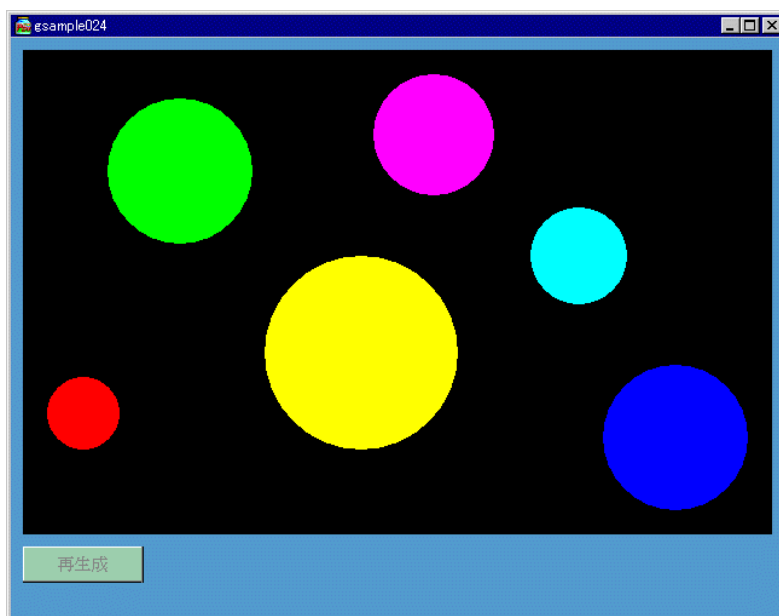


図 2-19 gsamp024.psv 実行画面

(6) 更新履歴

なし

2.3.3 g_del_usrdata

ユーザ定義データの削除

(1) 形式

`g_del_usrdata objID key [key...]`

(2) 説明

`g_del_usrdata` マクロは、指定した図形オブジェクトのキー (*key*) と、キーに対応するデータ (*body*) を削除します。

① *objID*

対象の図形オブジェクトの ID を指定します。

② *key [key...]*

ユーザ定義データのキー (*key*) を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_get_usrdata`, `g_set_usrdata`

(5) 使用例

gsamp048.psv 参照 (以下画面例)



図 2-20 gsamp048.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。

2.3.4 g_dist

指定点から図形オブジェクトへの距離の取得

(1) 形式

`g_dist contextID [-dc] xVal/ yVal/ objID [objID...]`

(2) 説明

`g_dist` マクロは、指定した点から図形オブジェクトまでの最短距離を取得します。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-dc`

座標を DC 座標で指定する時に指定します。

③ `xVal/ yVal/`

距離計測の基準点となる座標を指定します。

④ `objID [objID...]`

距離取得の対象となる、1 個以上の図形オブジェクト ID を指定します。

(3) 戻り値

指定点からの距離（図形オブジェクトの指定数分リスト形式）

(4) 関連項目

`g_context`, `g_sort`

(5) 使用例

`gsamp084.psv` 参照（以下画面例）



図 2-21 `gsamp084.psv` 実行画面

(6) 更新履歴

なし

2.3.5 g_draw_arc

グラフィックスウィンドウへの円弧の描画

(1) 形式

`g_draw_arc contextID [-w] [-dc] cx cy rarious sdeg edeg [-f] [-c] [-a]`

(2) 説明

`g_draw_arc` マクロは、グラフィックスウィンドウに円弧を描画します。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwinID`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウに描画する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-dc`

座標を DC 座標で指定する時に指定します。

④ `cx cy rarious sdeg edeg`

円弧の中心座標 (`cx`, `cy`)、半径 (`raious`)、開始角 (`sdeg`)、終了角 (`edeg`) を指定します。角度は 1/10 度単位で指定します。

⑤ `-f`

塗りつぶしを行う時に指定します。

⑥ `-c colorID`

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

⑦ `-a attribVal`

描画図形の属性を数字で指定します (`g_attrib` 参照)。塗りつぶしタイプ、線の太さ、線のタイプのみ有効です。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_attrib`

(5) 使用例

gsamp026.psv 参照 (以下画面例)

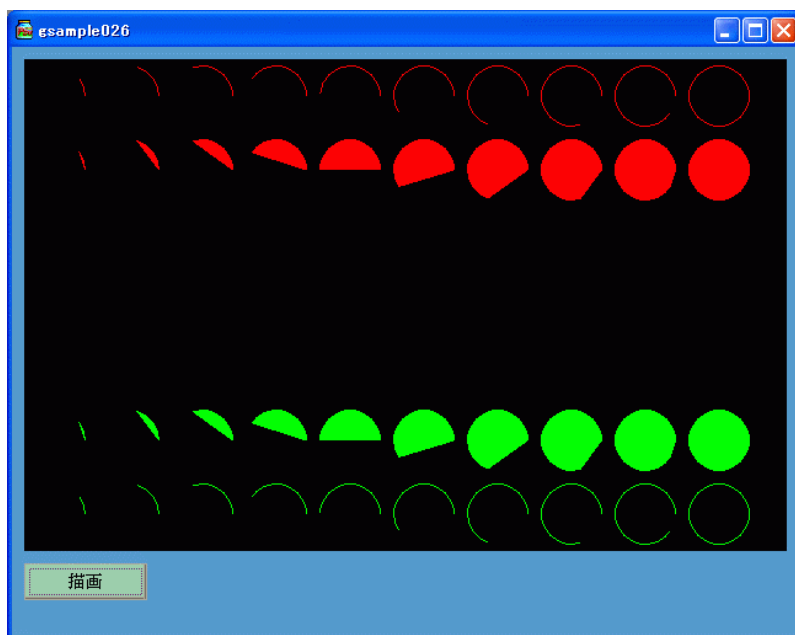


図 2-22 gsamp026.psv 実行画面

(6) 更新履歴

なし

2.3.6 g_draw_attrb

図形オブジェクトの描画属性値の取得／変更

(1) 形式

`g_draw_attrb objID [attribVal]`

(2) 説明

`g_draw_attrb` マクロは、図形オブジェクトの描画属性値を取得／変更します。引数に属性値を指定しない時は、現在の属性値を取得します。指定した場合は、属性値を変更して、変更後の値を返します。

① *objID*

作成した図形オブジェクトの ID を指定します。

② *attribVal*

次の 32 ビットの整数で指定します。

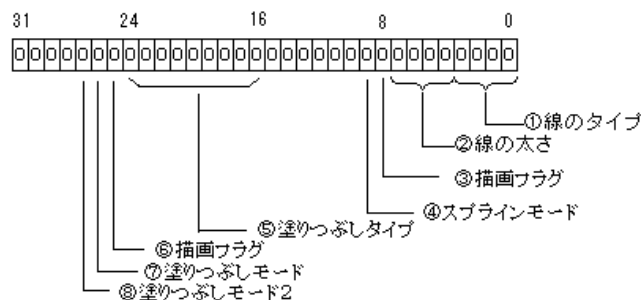


図 2-23 描画属性値ビット割り当て

(a) 線のタイプ (ビット 0～3)

線の種類を次の中から指定します。

- 0 : 実線
- 1 : 破線
- 2 : 点線
- 3 : 一点鎖線
- 4 : 二点鎖線

(b) 線の太さ (ビット 4～7)

線の太さを指定します。値が大きいほど太くなります。

(c) 描画フラグ (ビット 8)

線を描画するかどうかを指定します。

- 0 : 描画なし (空の線)
- 1 : 描画有り

(d) スプラインモード (ビット 9)

折れ線 (g_polyline) にのみ有効です。折れ線をそのまま描画するか、スプライン曲線として描画するかを指定します。

0 : 折れ線

1 : スプライン曲線

(e) 塗りつぶしタイプ (ビット 16~23)

図形の塗りつぶしタイプを指定します。

0 : ベタ

1 : 不透明 (指定色の輪郭線を描き、内部を背景色で塗りつぶし)

2 : 横線

3 : 斜め線 (右上がり)

4 : 斜め線 (右下がり)

5 : 縦線

6 : 縦横格子

7 : 斜め格子

8~ : g_paint_pattern_file で指定した ビットマップファイルのパターン



図 2-24 塗りつぶしパターン

(f) (塗りつぶし)描画フラグ (ビット 24)

円 (g_circle)、円弧 (g_arc)、楕円 (g_ellipse)、多角形 (g_polygon)、扇型 (g_pie)、角丸長方形 (g_roundbox)、正多角形 (g_regpoly) だけで有効です。

図形オブジェクトを塗りつぶしの有無を指定します。

0 : 塗りつぶさない (空の塗りつぶし)

1 : 塗りつぶす

(g) 塗りつぶしモード (ビット 25)

塗りつぶしのモード (透明・不透明) を指定します

0 : 透明

1 : 不透明

(h) 塗りつぶしモード 2 (ビット 26)

塗りつぶしのモード (個別指定・一括指定) を指定します。

0 : 線色と塗りつぶし色は同じ色になります (Ver. 3.0 互換)

1 : 線色と塗りつぶし色が別々に指定できます

(3) **戻り値**

描画属性値

(4) **関連項目**

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_round_box`, `g_regpoly`, `g_attrib`, `g_mode`

(5) **使用例**

gsamp098.psv参照 (画面例は2. 2. 19の画面参照)

(6) **更新履歴**

① Ver. 4. 0

Ver. 4. 0 で新規に追加したマクロです。

2.3.7 g_draw_circle

グラフィックスウィンドウへの円の描画

(1) 形式

```
g_draw_circle contextID [-w] [-dc] cx cy radius [-f] [-c] [-a]
```

(2) 説明

`g_draw_circle` マクロはグラフィックスウィンドウに円を描画します。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwinID`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外に描画する場合は、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-dc`

座標を DC 座標で指定する時に指定します。

④ `cx cy radius`

円の中心座標 (`cx`, `cy`)、半径 (`radius`) を指定します。

⑤ `-f`

塗りつぶしを行う時に指定します。

⑥ `-c colorID`

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウで設定したカレントの値を使用します。

⑦ `-a attribVal`

描画図形の属性を数字で指定します (`g_attr` 参照)。塗りつぶしタイプ、線の太さ、線のタイプのみ有効です。省略した場合には、グラフィックスウィンドウに設定したカレントの値を使用します。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_attr`

(5) 使用例

gsamp027.psv 参照 (以下画面例)

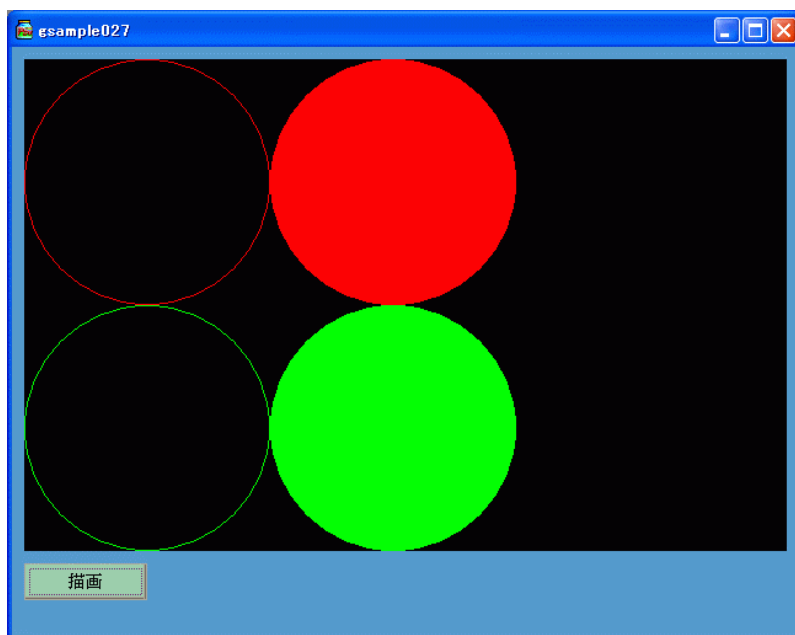


図 2-25 gsamp027.psv 実行画面

(6) 更新履歴

なし

2.3.8 g_draw_ellipse

グラフィックスウィンドウへの楕円の描画

(1) 形式

`g_draw_ellipse contextID [-w] [-dc] cx cy lr sr deg [-f] [-c] [-a]`

(2) 説明

`g_draw_ellipse` マクロは、グラフィックスウィンドウに楕円を描画します。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwinID`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外に描画する場合は、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-dc`

座標を DC 座標で指定する時に指定します。

④ `cx cy lr sr deg`

中心座標 (`cx`, `cy`)、横軸 (`lr`)、縦軸 (`sr`)、角度 (`deg`) を指定します。角度は 1/10 度単位、反時計回りで指定します。

⑤ `-f`

塗りつぶしを行う時に指定します。

⑥ `-c colorID`

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウに設定したカレントの値を使用します。

⑦ `-a attribVal`

描画図形の属性を数字で指定します (`g_attrib` 参照)。塗りつぶしタイプ、線の太さ、線のタイプだけ有効です。省略した場合には、グラフィックスウィンドウに設定したカレントの値を使用します。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_attrib`

(5) 使用例

gsamp028.psv 参照 (以下画面例)

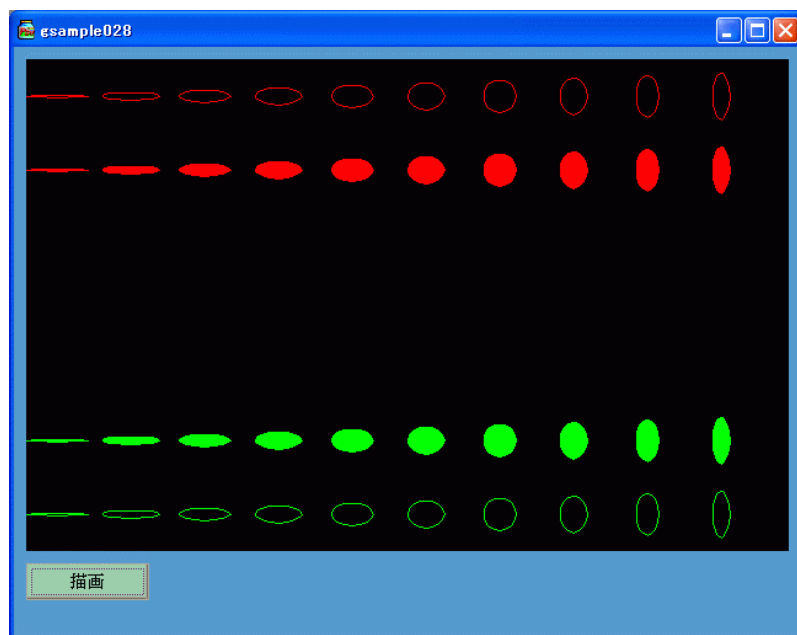


図 2-26 gsamp028.psv 実行画面

(6) 更新履歴

なし

2.3.9 g_draw_line

グラフィックスウィンドウへの線の描画

(1) **形式**

```
g_draw_line context/D [-w] [-dc] x1 y1 x2 y2 [-c] [-a]
```

(2) **説明**

g_draw_line マクロはグラフィックスウィンドウに線を描画します。

① *context/D*

g_context で作成したコンテキスト ID を指定します。

② *-w gwin/D*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外に描画する場合は、g_open_gwin で作成したグラフィックスウィンドウ ID を指定します。

③ *-dc*

座標を DC 座標で指定する時に指定します。

④ *x1 y1 x2 y2*

線の座標を指定します。始点 (*x1, y1*)、終点 (*x2, y2*) の順に指定します。

⑤ *-c color/D*

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

⑥ *-a attribVal*

描画図形の属性を数字で指定します (g_attrib 参照)。線の太さ、線のタイプのみ有効です。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

(3) **戻り値**

なし

(4) **関連項目**

g_context, g_open_gwin, g_attrib

(5) 使用例

gsamp029.psv 参照 (以下画面例)

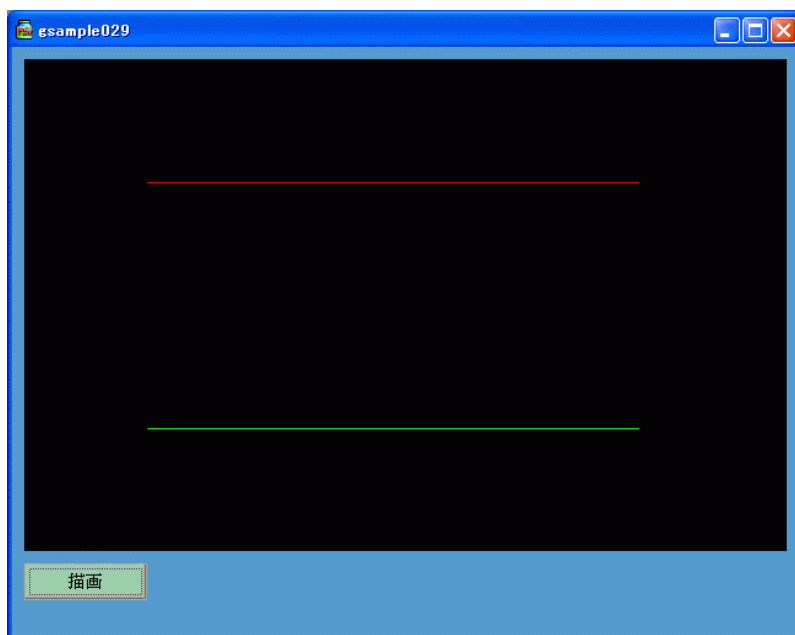


図 2-27 gsamp029.psv 実行画面

(6) 更新履歴

なし

2.3.10 g_draw_obj

グラフィックスウィンドウへの図形オブジェクト描画

(1) 形式

```
g_draw_obj contextID [-c] [-w] objID [objID...]
```

```
g_draw_obj contextID [-c] [-w] [-d dlistID [dlistID...]] [-g|-l|-r|-p]
```

```
g_draw_obj contextID [-w] [-b] -m
```

(2) 説明

g_draw_obj マクロは、図形オブジェクトをグラフィックスウィンドウに描画します。実行形式は3つあり、図形オブジェクト ID を個々に指定する方法と、グループ番号やレイヤ番号、座標範囲を指定して一括描画する方法、マップオブジェクト単位で一括描画する方法があります。

描画対象となった、個別の図形オブジェクトを描画するかは、次の属性値での可視性の設定にしがいます (g_attrib 参照)。これらのすべてが「可視」に設定されている場合だけ、描画します。

- 図形オブジェクトに設定されている属性
- 図形オブジェクトに設定されているグループ番号に設定されている属性
- 図形オブジェクトに設定されているレイヤ番号に設定されている属性

① contextID

g_context で作成したコンテキスト ID を指定します。

② -c colorID

色を指定する場合に、g_alloc_color で作成した色 ID を指定します。

省略するか、-1 を指定すると、図形オブジェクトの属性で指定した優先順位にしたがって、次のいずれかの色で描画します (g_attrib 参照)。

- 図形オブジェクトに設定されている色
- 図形オブジェクトに設定されているグループ番号に設定されている色
- 図形オブジェクトに設定されているレイヤ番号に設定されている色

③ -w gwinID

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外に描画する場合に、g_open_gwin で作成したグラフィックスウィンドウ ID を指定します。

④ objID [objID...]

図形オブジェクトを個別に指定する場合に、図形オブジェクト ID を指定します。

⑤ -d dlistID ...

-l、-r、-p 指定の時、検索するディスプレイリストメモリを指定します。省略すると、コンテキストオブジェクトに設定しているディスプレイリストメモリを対象とします。

⑥ `-g [groupID] groupNo ...`

グループ番号 (*groupNo*) を指定して図形オブジェクトを描画します。グループオブジェクトID (*groupID*) の指定がない場合は、コンテキストオブジェクトに設定しているグループオブジェクトが対象となります。

⑦ `-l [layerID] layerNo`

レイヤ番号 (*layerNo*) を指定して、ディスプレイリストメモリ内の図形オブジェクトを描画します。レイヤオブジェクトID (*layerID*) 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ヤオブジェクトが対象となります。

⑧ `-r [-dc] x1 y1 x2 y2`

矩形の座標範囲を指定して、ディスプレイリストメモリ内の図形オブジェクトを描画します。矩形範囲に含む図形オブジェクトが描画対象です。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標として扱います。

⑨ `-p num [-dc] x1 y1 x2 y2`

多角形の座標範囲を指定して、ディスプレイリストメモリ内の図形オブジェクトを描画します。多角形の頂点の座標を (*x1*, *y1*)、(*x2*, *y2*) ... の順に指定します。範囲に部分的にかかる図形オブジェクトも対象となります。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標として扱います。

⑩ `-b`

描画の前にグラフィックスウィンドウの背景の塗りつぶし `g_clear` を実行します。

⑪ `-m`

`g_redraw` でグラフィックスウィンドウに登録したマップオブジェクトを利用して図形を描画します。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_make_group`, `g_group_attrib`, `g_group_color_id`, `g_layer_no`,
`g_layer_attrib`, `g_layer_color_id`, `g_alloc_color`, `g_line`, `g_polyline`,
`g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`, `g_symbol`, `g_usr_def`,
`g_round_box`, `g_regpoly`, `g_clear`, `g_redraw`, `g_open_map`

(5) 使用例

`gsamp003.psv` 参照 (画面例は 2.1.4 の画面参照)

(6) 更新履歴

① Ver. 4.0

指定のグラフィックスウィンドウの塗りつぶし指定が可能になりました (`-b` オプション)。

マップオブジェクトを利用した描画が可能になりました (`-m` オプション)。

2.3.11 g_draw_pie

グラフィックスウィンドウへの扇型の描画

(1) **形式**

`g_draw_pie contextID [-w] [-dc] cx cy radius sdeg edeg [-f] [-c] [-a]`

(2) **説明**

`g_draw_pie` マクロは、グラフィックスウィンドウに扇型を描画します。円弧の始点と終点が、円弧の中心座標と線分で結ばれ、扇型が描画されます。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwinID`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外に描画する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-dc`

座標を DC 座標で指定する時に指定します。

④ `cx cy radius sdeg edeg`

扇型の中心座標 (`cx`, `cy`)、半径 (`radius`)、開始角 (`sdeg`)、終了角 (`edeg`) を指定します。角度は 1/10 度単位で指定します。

⑤ `-f`

塗りつぶしを行う時に指定します。

⑥ `-c colorID`

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウで設定したカレントの値を使用します。

⑦ `-a attribVal`

描画図形の属性を数字で指定します (`g_attr` 参照)。塗りつぶしタイプ、線の太さ、線のタイプのみ有効です。省略した場合には、グラフィックスウィンドウで設定したカレントの値を使用します。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_gwin`, `g_attr`

(5) 使用例

gsamp031.psv 参照 (以下画面例)

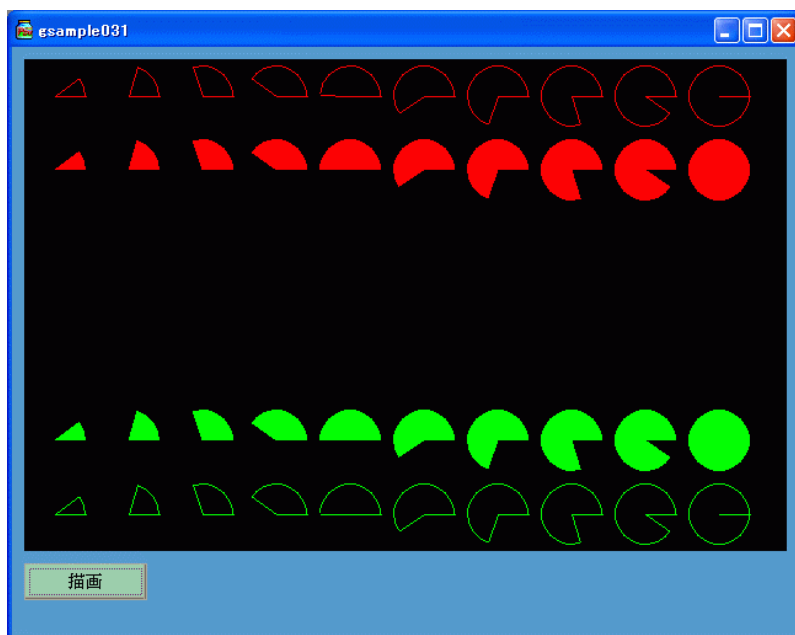


図 2-28 gsamp031.psv 実行画面

(6) 更新履歴

なし

2.3.12 g_draw_polygon

グラフィックスウィンドウへの多角形の描画

(1) **形式**

`g_draw_polygon contextID [-w] num [-dc] x1 y1 x2 y2.... [-f] [-c] [-a]`

(2) **説明**

`g_draw_polygon` マクロはグラフィックスウィンドウに多角形を描画します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-w gwinID*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外に描画する場合は、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *num*

多角形の頂点数を指定します。

④ *-dc*

座標を DC 座標で指定する時に指定します。

⑤ *x1 y1 x2 y2....*

頂点の座標を指定します。

⑥ *-f*

塗りつぶしを行う時に指定します。

⑦ *-c colorID*

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウで設定したカレントの値を使用します。

⑧ *-a attribVal*

描画図形の属性を数字で指定します (`g_attr` 参照)。塗りつぶしタイプ、線の太さ、線のタイプだけ有効です。省略した場合には、グラフィックスウィンドウで設定したカレントの値を使用します。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_gwin`, `g_attr`

(5) 使用例

gsamp032.psv 参照 (以下画面例)

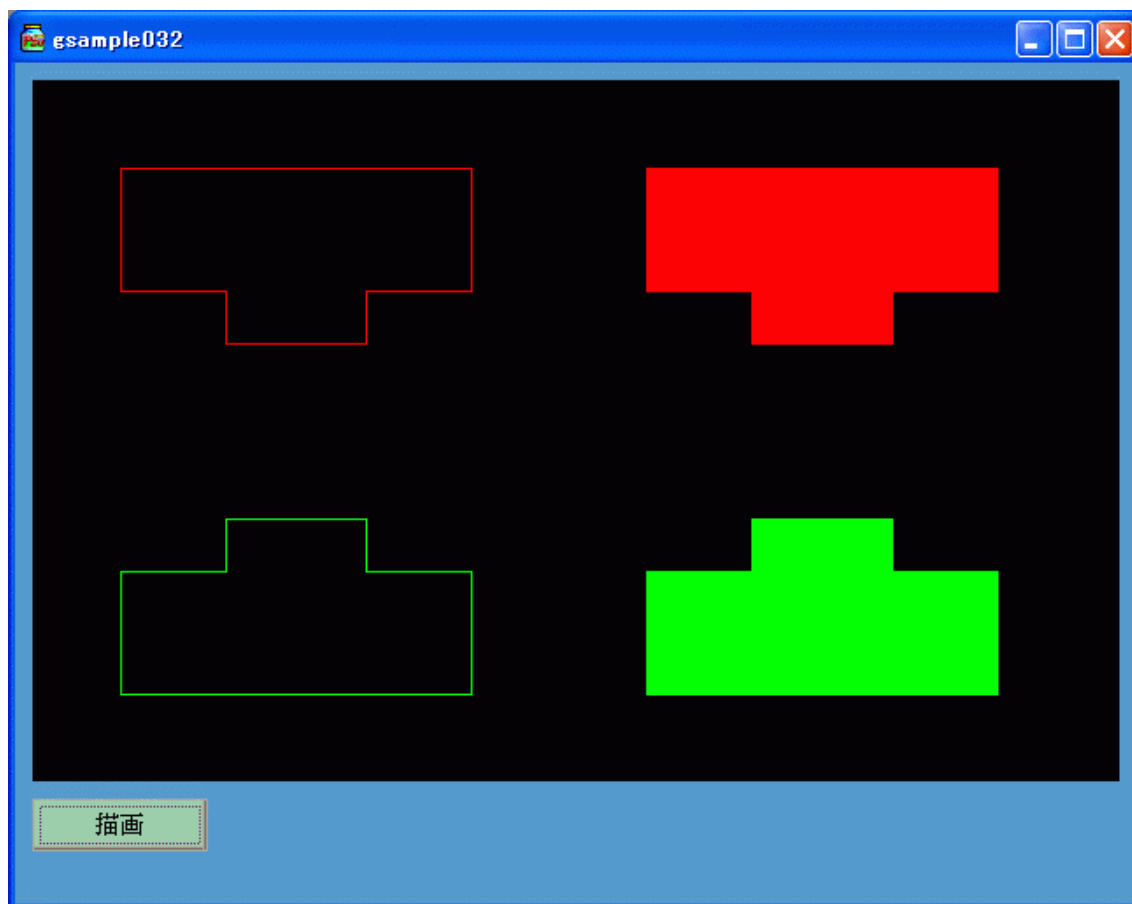


図 2-29 gsamp032.psv 実行画面

(6) 更新履歴

なし

2.3.13 g_draw_polyline

グラフィックスウィンドウへの折れ線、スプライン曲線の描画

(1) **形式**

`g_draw_polyline contextID [-w] num [-dc] x1 y1 x2 y2 [-f] [-c] [-a]`

(2) **説明**

`g_draw_polyline` マクロは、グラフィックスウィンドウに折れ線、またはスプライン曲線を描画します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-w gwinID*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外に描画する場合には、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *num*

折れ線の点数を指定します。

④ *-dc*

座標を DC 座標で指定する時に指定します。

⑤ *x1 y1 x2 y2*

折れ線の座標を指定します。

⑥ *-f*

スプライン曲線を描画する時に指定します。

⑦ *-c colorID*

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウで設定したカレントの値を使用します。

⑧ *-a attribVal*

描画図形の属性を数字で指定します (`g_attrib` 参照)。線の太さ、線のタイプのみ有効です。省略した場合には、グラフィックスウィンドウで設定したカレントの値を使用します。

(3) **戻り値**

なし

(4) **関連項目**

`g_attrib`, `g_context`, `g_open_gwin`

(5) 使用例

gsamp033.psv 参照 (以下画面例)

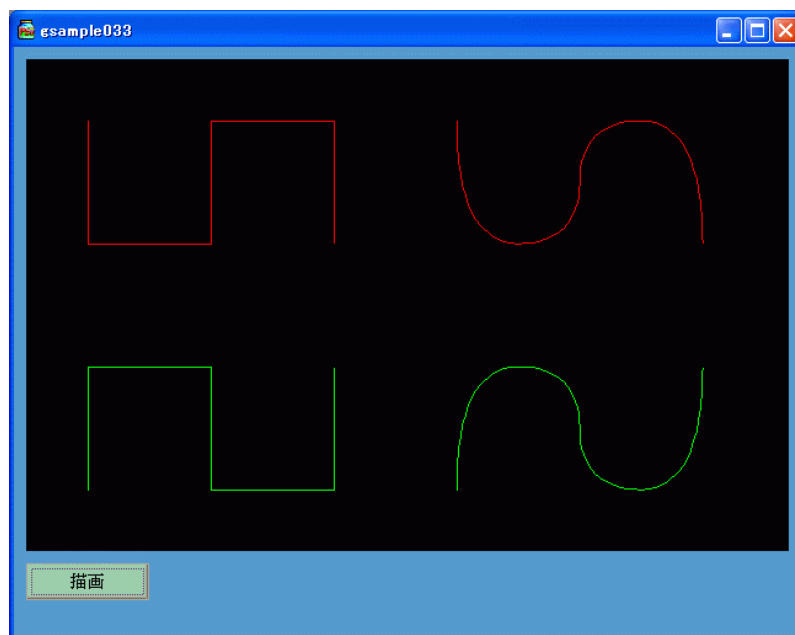


図 2-30 gsamp033.psv 実行画面

(6) 更新履歴

なし

2.3.14 g_draw_rect

グラフィックスウィンドウへの長方形の描画

(1) 形式

```
g_draw_rect context/D [-w] [-dc] x1 y1 x2 y2 [-f] [-c] [-a]
```

(2) 説明

`g_draw_rect` マクロは、グラフィックスウィンドウに長方形を描画します。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwin/D`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウに描画する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-dc`

座標を DC 座標で指定する時に指定します。

④ `x1 y1 x2 y2`

長方形の左上角、右下角または、左下角、右上角の座標を指定します。

⑤ `-f`

塗りつぶしを行う時に指定します。

⑥ `-c color/D`

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

⑦ `-a attribVal`

描画図形の属性を数字で指定します (`g_attr` 参照)。塗りつぶしタイプ、線の太さ、線のタイプのみ有効です。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_attr`

(5) 使用例

gsamp034.psv 参照 (以下画面例)

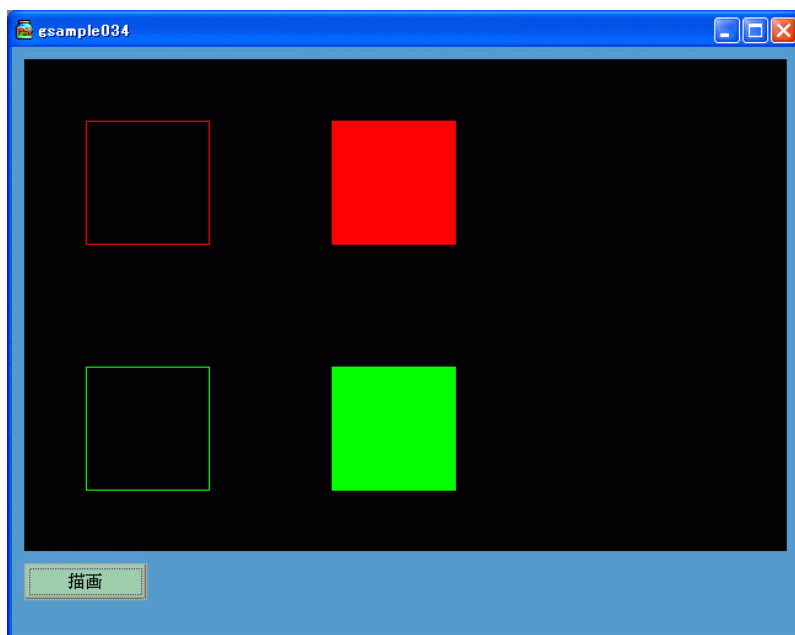


図 2-31 gsamp034.psv 実行画面

(6) 更新履歴

なし

2.3.15 g_draw_regpoly

グラフィックスウィンドウへの正多角形の描画

(1) **形式**

`g_draw_regpoly contextID [-dc] cx cy sx sy num [-f] [-c] [-a]`

(2) **説明**

`g_draw_regpoly` マクロは、グラフィックスウィンドウに正多角形オブジェクトを描画します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-dc*

座標を DC 座標で指定する時に指定します。

③ *cx cy sx sy*

正多角形の中心座標 (*cx*, *cy*)、最初の頂点の開始座標 (*sx*, *sy*) を指定します。

④ *num*

角数を指定します。

⑤ *-f*

塗りつぶしを行う時に指定します。

⑥ *-c colorID*

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

⑦ *-a attribVal*

描画図形の属性を数字で指定します (`g_attr` 参照)。塗りつぶしタイプ、線の太さ、線のタイプのみ有効です。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_gwin`, `g_attr`

(5) 使用例

gsamp035.psv 参照 (以下画面例)

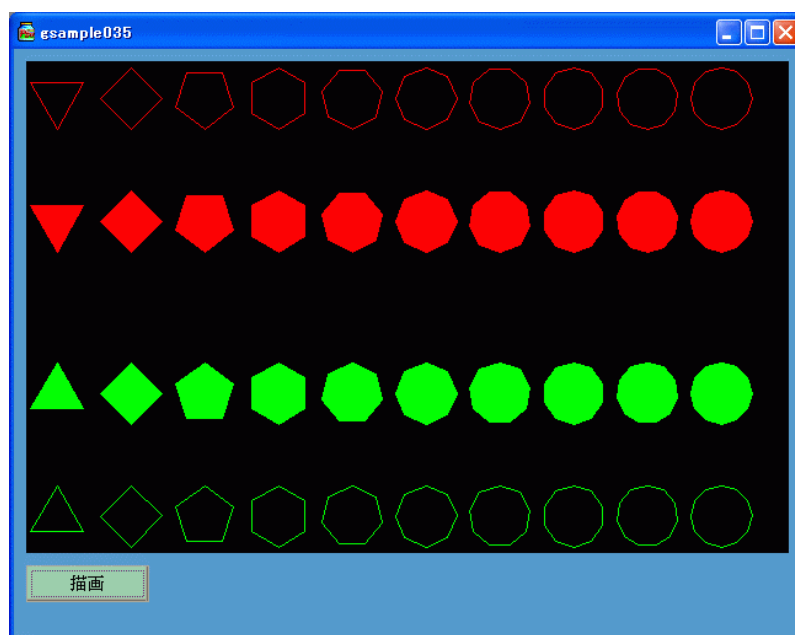


図 2-32 gsamp035.psv 実行画面

(6) 更新履歴

なし

2.3.16 g_draw_round_box

グラフィックスウィンドウへの角丸長方形の描画

(1) 形式

`g_draw_round_box context/D [-dc] x1 y1 x2 y2 rad degree [-f] [-c] [-a]`

(2) 説明

`g_draw_round_box` マクロは、グラフィックスウィンドウに角丸長方形オブジェクトを描画します。

① *context/D*

`g_context` で作成したコンテキスト ID を指定します。

② *-dc*

座標を DC 座標で指定する時に指定します。

③ *x1 y1 x2 y2*

角丸長方形の座標を指定します。左下点 (*x1*, *y1*) と右上点 (*x2*, *y2*) または、左上点 (*x1*, *y1*) と右下点 (*x2*, *y2*) を指定します。

④ *rad*

角の円弧の半径を指定します。

⑤ *degree*

角丸長方形の回転角度を 1/10 度単位で指定します。

⑥ *-f*

塗りつぶしを行う時に指定します。

⑦ *-c color/D*

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

⑧ *-a attribVal*

描画図形の属性を数字で指定します (`g_attrib` 参照)。塗りつぶしタイプ、線の太さ、線のタイプのみ有効です。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_attrib`

(5) 使用例

gsamp036.psv 参照 (以下画面例)

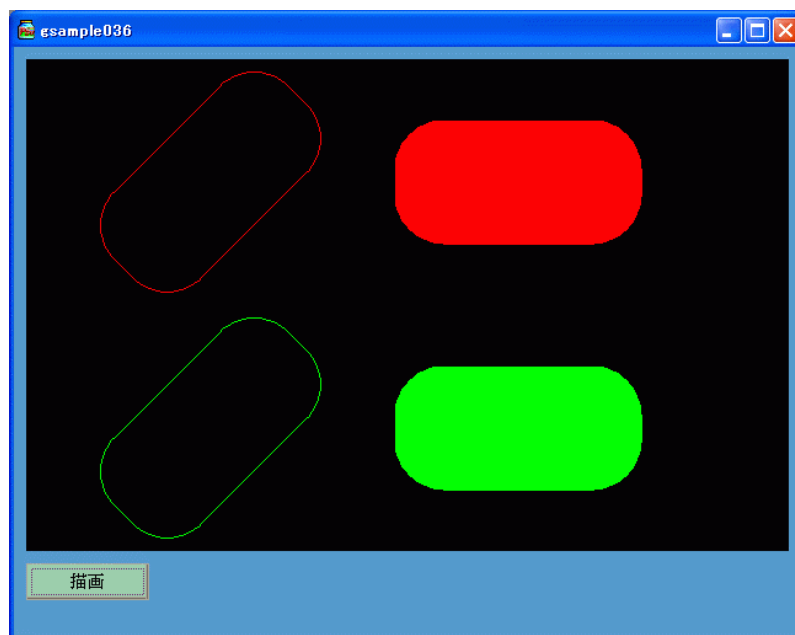


図 2-33 gsamp036.psv 実行画面

(6) 更新履歴

なし

2.3.17 g_draw_symbol

グラフィックスウィンドウへのシンボルの描画

(1) 形式

`g_draw_symbol contextID [-w] [-dc] cx cy symNo deg scale [-c] [-a]`

(2) 説明

`g_draw_symbol` マクロはグラフィックスウィンドウにシンボルを描画します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-w gwinID*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウに描画する場合には、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *-dc*

座標を DC 座標で指定する時に指定します。

④ *cx cy*

シンボルの中心座標 (*cx*, *cy*) を指定します。

⑤ *symNo*

`g_symbol_file` で設定したシンボルファイル中のシンボル番号を指定します。

⑥ *deg scale*

角度 (*deg*)、倍率 (*scale*) を指定します。角度は反時計回り、1/10 度単位で指定します。

⑦ *-c colorID*

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います (カラーシンボル指定時は無効となります)。

⑧ *-a attribVal*

描画図形の属性を数字で指定します (`g_attr` 参照)。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

注) 現版ではシンボル描画時はこの属性は無効です。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_attr`, `g_symbol_file`

(5) 使用例

gsamp037.psv 参照 (以下画面例)



図 2-34 gsamp037.psv 実行画面

(6) 更新履歴

なし

2.3.18 g_draw_text

グラフィックスウィンドウへの文字列の描画

(1) 形式

```
g_draw_text contextID [-w] [-dc] x y num string [-c] [-a] [-h] [-w] [-r] [-o] [-s]
```

(2) 説明

`g_draw_text` マクロはグラフィックスウィンドウに文字列を描画します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-w gwinID*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウに描画する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *-dc*

座標を DC 座標で指定する時に指定します。

④ *x y*

文字列の座標を指定します。横書きの時は文字列の左下、縦書きの時は文字列の左上になります。

⑤ *num*

文字数を指定します。

⑥ *string*

文字列を指定します。文字列は 2 バイトのシフト JIS か EUC のみ有効となります。

⑦ *-c colorID*

描画色の色 ID を数字で指定します。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

⑧ *-a attribVal*

描画図形の属性を数字で指定します (`g_attrib` 参照)。線の太さの指定のみ有効です。省略した場合には、グラフィックスウィンドウに設定したカレントの値を用います。

⑨ *-h hightVal*

文字高さを数字で指定します (WC 座標単位)。

⑩ *-w widthVal*

文字幅を数字で指定します (WC 座標単位)。

⑪ *-r deg*

文字列の角度を反時計回り、1/10 度単位で指定します。

⑫ -o h|v

文字列方向を指定します。

h : 横書き

v : 縦書き

⑬ -s *spaceVal*

文字間隔を数字で指定します (WC 座標単位)。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_attr`

(5) 使用例

`gsamp038.psv` 参照 (以下画面例)

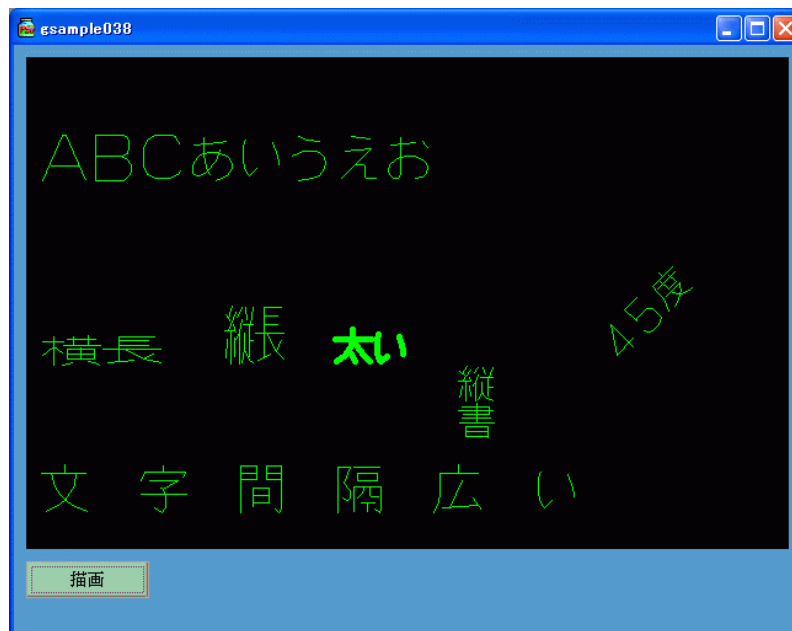


図 2-35 `gsamp038.psv` 実行画面

(6) 更新履歴

① Ver. 4.0

半角記号の指定が可能になりました。

② Ver. 5.0

半角カナ・ローマ数字大文字・小文字・全角スペースの指定が可能になりました。

2.3.19 g_dump_obj

図形オブジェクトの内容取得

(1) 形式

`g_dump_obj objID [objID...]`

(2) 説明

`g_dump_obj` マクロは、図形オブジェクトの座標などの内容を取得します。

① `objID [objID...]`

作成した図形オブジェクトの ID を指定します。

(3) 戻り値

図形の内容（1 図形オブジェクトごとにリスト形式となります）

図形オブジェクトごとの内容は以下のとおりです。

線

`line` 点数 座標 属性 色 モード

折れ線

`polyline` 点数 座標 属性 色 モード

多角形

`polygon` 点数 座標 属性 色 モード

円

`circle` 中心座標 半径 属性 色 モード

扇型

`pie` 中心座標 半径 開始角 終了角 属性 色 モード

円弧

`arc` 中心座標 半径 開始角 終了角 属性 色 モード

シンボル

`symbol` シンボル番号 中心座標 角度 倍率 属性 色 モード

楕円

`ellipse` 中心座標 X 軸長さ Y 軸長さ 角度 属性 色 モード

文字列

`text` 点数 基準点座標 文字高さ 文字幅 文字間隔 文字角度 文字列方向
文字数 文字コード [行間隔]^{注1)} 属性 色 モード

注1) 行間隔は「A 版互換モード」、「A 版統合モード」の時は A 版との互換性重視のため、出力しません。

正多角形

`regpoly` 中心座標 開始座標 点数 属性 色 モード

角丸長方形

`round_box` 矩形左下座標 矩形右上座標 丸め度 角度 属性 色 モード

ユーザ線種

applines 点数 座標 ユーザ線種番号 属性 色 モード

鉄道線

RRline 点数 座標 鉄道線種番号 線幅 属性 色 モード

(4) 関連項目

g_line, g_polyline, g_polygon, g_circle, g_arc, g_pie, g_ellipse, g_text,
g_symbol, g_round_box, g_regpoly

(5) 使用例

gsamp039.psv参照 (画面例は2. 2. 16の画面参照)

(6) 更新履歴

① Ver. 3. 0

文字列の行間隔を追加しました。

② Ver. 5. 2

鉄道線の実力フォーマットを追加しました。

2.4 E~G で始まるコマンド

2.4.1 g_ellipse

楕円オブジェクトの作成

(1) 形式

```
g_ellipse contextID [-dc] cx cy /r sr deg [[-t] [-c] [-a] [-g] [-l] [-m]]  
[-u]
```

(2) 説明

g_ellipse マクロは、楕円オブジェクトを作成します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② -dc

座標を DC 座標で指定する時に指定します。

③ cx cy /r sr deg

楕円の中心座標 (cx, cy)、横軸 (/r)、縦軸 (sr)、回転角度 (deg) を指定します。角度は反時計回り、1/10 度単位で指定します。

④ -t tagID

図形オブジェクトのタグ ID を数字で指定します。

⑤ -c colorID

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑥ -a attribVal (Ver. 4 以降削除 A版互換モードの場合)

図形オブジェクトの属性を数字で指定します (g_attrib 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑦ -a objAttribVal drawAttribVal

図形オブジェクトの属性値と描画属性値を数字で指定します (g_obj_attrib、g_draw_attrib 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ -g [groupID] groupNo

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクトID (groupID) 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グループオブジェクトが対象となります。

⑨ -l [layerID] layerNo

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクトID (layerID) 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ヤオブジェクトが対象となります。

⑩ `-m mode` (Ver. 4 以降削除 A版互換モード時有効)

図形オブジェクトのモードを数字で指定します (g_mode 参照)。

⑪ `-u key body [key body]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

楕円の図形オブジェクト ID

(4) 関連項目

g_attrb, g_context, g_draw_obj, g_mode, g_open_group, g_open_layer,
g_obj_attrb, g_draw_attrb

(5) 使用例

gsamp040.psv 参照 (以下画面例)



図 2-36 gsamp040.psv 実行画面

(6) 更新履歴

① Ver. 4.0

属性値のオプションの指定を変更しました (-a)。

モード指定のオプションを削除しました (-m)。

2.4.2 g_filter_add

図形オブジェクトへのフィルタオブジェクトの登録

(1) 形式

`g_filter_add contextID filterID objID [objID...]`

(2) 説明

`g_filter_add` マクロは、フィルタオブジェクトを図形オブジェクトに登録します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *filterID*

`g_open_filter` で作成したフィルタオブジェクト ID を指定します。

③ *objID [objID...]*

フィルタを登録する図形オブジェクトの図形オブジェクト ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_open_filter`, `g_filter_set_param`, `g_filter_remove`, `g_filter_draw_obj`,

(5) 使用例

`gsamp063.psv`参照 (画面例は2.2.5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。

2.4.3 g_filter_draw_obj

フィルタを用いたグラフィックスウィンドウへの図形オブジェクト描画

(1) 形式

```
g_filter_draw_obj context/D [-c] [-w] obj/D [obj/D...]
```

(2) 説明

`g_filter_draw_obj` マクロは、フィルタを用いて 図形オブジェクトをグラフィックスウィンドウに描画します。図形オブジェクトが描画されるかどうかは、次の属性値での可視性の設定にしがいます (`g_attr` 参照)。これらのすべてを「可視」に設定している場合だけ、描画します。

- 図形オブジェクトに設定している属性
- 図形オブジェクトに設定しているグループ番号に設定している属性
- 図形オブジェクトに設定しているレイヤ番号に設定している属性

① context/D

`g_context` で作成したコンテキスト ID を指定します。

② -c color/D

色を指定する場合に、`g_alloc_color` で作成した色 ID を指定します。

省略するか、-1 を指定すると、図形オブジェクトの属性で指定した優先順位にしたがって、次のいずれかの色で描画します (`g_attr` 参照)。

- 図形オブジェクトに設定している色
- 図形オブジェクトに設定しているグループ番号に設定している色
- 図形オブジェクトに設定しているレイヤ番号に設定している色

③ -w gwin/D

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外の グラフィックスウィンドウに描画する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

④ obj/D [obj/D...]

図形オブジェクトを個別に指定する場合に、図形オブジェクト ID を指定します。

注) フィルタ処理はベクトル図形専用の処理であるため、ラスタ図形であるビットマップシンボルの図形オブジェクト ID は指定できません。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_alloc_color`, `g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`,
`g_pie`, `g_ellipse`, `g_text`, `g_symbol`, `g_usr_def`, `g_round_box`, `g_regpoly`,
`g_open_filter`, `g_close_filter`, `g_filter_set_param`, `g_filter_add`,
`g_filter_remove`

(5) **使用例**

gsamp063.psv参照 (画面例は2. 2. 5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。

2.4.4 g_filter_remove

図形オブジェクトからのフィルタオブジェクトの消去

(1) **形式**

`g_filter_remove contextID filterID objID [objID...]`

(2) **説明**

フィルタオブジェクトを図形オブジェクトより消去します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *filterID*

`g_open_filter` で作成したフィルタオブジェクト ID を指定します。

③ *objID [objID...]*

フィルタを消去する図形オブジェクトの図形オブジェクト ID を指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_open_filter`, `g_close_filter`, `g_filter_set_param`, `g_filter_add`,
`g_filter_draw_obj`

(5) **使用例**

gsamp063.psv参照 (画面例は2.2.5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。

2.4.5 g_filter_set_param

フィルタオブジェクトのパラメータ設定

(1) 形式

```
g_filter_set_param contextID filterID -g
```

```
g_filter_set_param contextID filterID [-s|-p|-m polygon] x y
```

(2) 説明

g_filter_set_param マクロは、フィルタオブジェクトのパラメータを取得／設定します。パラメータの内容は g_open_filter によって作られたフィルタオブジェクトの種類によって異なります。

① contextID

g_context で作成したコンテキスト ID を指定します。

② filterID

g_open_filter で作成したフィルタオブジェクト ID を指定します。

③ -g

フィルタの現在の状態を取得します。取得する内容はフィルタオブジェクトの種類によって異なります。

移動用フィルタ：移動分 (x , y)

回転用フィルタ：回転の中心座標 (x , y)、回転角度

スケール変更用フィルタ：中心座標 (x , y)、スケール (x 軸、 y 軸)

ポリゴン／ポリライン編集用フィルタ：編集した頂点の番号 (0～)、
変更後の座標 (x , y)

(a) x y (移動用)

移動用のフィルタの場合、座標 (x , y) を指定します。移動分は g_open_filter により作成した時に指定した座標 (sx , sy) と (x , y) の差となります。

(b) x y (回転用)

回転用のフィルタの場合、座標 (x , y) を指定します。回転角度は g_open_filter により作成した時に指定した座標 (rx , ry) と (x , y) の角度となります。

(c) [-s] x y (スケール用)

スケール変更用フィルタ座標 (x , y) を指定します。スケールは g_open_filter により作成した時に指定した座標 (sx , sy) と (x , y) の比となります。X 軸と Y 軸を等しいスケールにする場合は -s を指定します。この場合は Y 軸のスケールは無視して、X 軸と同じスケールになります。

(d) `-p x y -m polygon x y` (ポリゴン／ポリライン編集用)

ポリゴン／ポリライン編集用フィルタの場合、いずれかの形式で座標 (x , y) を指定します。ラバーバンドをポリゴンモード (最後の点と最初の点を結ぶ) で表示する場合には `[-m polygon]` を指定します。

点追加モード : 追加する点の座標

点変更モード : 変更する点の座標

点削除モード : 削除する点に最も近い角が削除対象となります。

(3) **戻り値**

`-g` 指定時、現在のフィルタの設定値がリスト形式で返ります。

(4) **関連項目**

`g_context`, `g_open_filter`, `g_close_filter`, `g_filter_add`, `g_filter_remove`

(5) **使用例**

`gsamp063.psv` 参照 (画面例は 2. 2. 5 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。

2.4.6 g_get_usrdata

図形オブジェクトのユーザ定義データ取得

(1) 形式

```
g_get_usrdata objID key [key...]
```

(2) 説明

g_get_usrdata マクロは、図形オブジェクトに格納されたユーザ定義データを取得します。

① objID

作成した図形オブジェクトの ID を指定します。

② key [key...]

ユーザ定義データのキーを指定します。

(3) 戻り値

ユーザ定義データ (リスト形式)

(4) 関連項目

g_line, g_polyline, g_polygon, g_circle, g_arc, g_pie, g_ellipse, g_text, g_symbol, g_usr_def, g_set_usrdata, g_round_box, g_regpoly

(5) 使用例

gsamp045.psv 参照 (以下画面例)



図 2-37 gsamp045.psv 実行画面

(6) 更新履歴

なし

2.4.7 g_group_attrib

グループオブジェクトの属性値の取得／変更

(1) 形式

g_group_attrib *contextID* [-g] *groupNo* [*attribVal*]

(2) 説明

g_group_attrib マクロは、グループオブジェクトのグループ番号ごとの属性値を取得／変更します。引数に属性値を指定しない場合は、属性値が取得されます。指定する場合は、属性値が変更され、変更後の属性値が返ります。属性のうち検知性、可視性のみ有効です。設定した属性は、g_search_obj での図形オブジェクトの検索、g_draw_obj での図形オブジェクトの描画の際に参照されます。

① *contextID*

g_context で作成したコンテキスト ID を指定します。

② -g *groupID*

コンテキストオブジェクトに設定しているグループオブジェクト以外のグループオブジェクトを指定する場合に、g_open_group で作成したグループオブジェクト ID を指定します。

③ *groupNo*

グループ番号を指定します。

④ *attribVal*

次の 32 ビットを 10 進の整数で指定します。

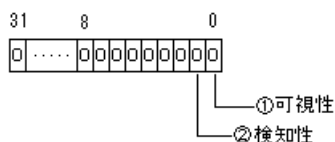


図 2-38 グループオブジェクト属性値ビット割り当て

(a) 可視性 (ビット 0)

図形オブジェクトの可視性を指定します。不可視とすると、g_draw_obj で描画を実行してもその図形オブジェクトは描画しません。

0 : 不可視

1 : 可視

(b) マウスの検知性 (ビット 1)

g_search_obj マクロで図形オブジェクトを検知可能とするかどうかを指定します。

0 : 検知しない

1 : 検知する

(3) 戻り値

属性値

(4) 関連項目

`g_context`, `g_draw_obj`, `g_search_obj`, `g_attrib`, `g_open_group`, `g_make_group`,
`g_group_no`, `g_group_color_id`

(5) 使用例

`gsamp046.psv` 参照 (以下画面例)

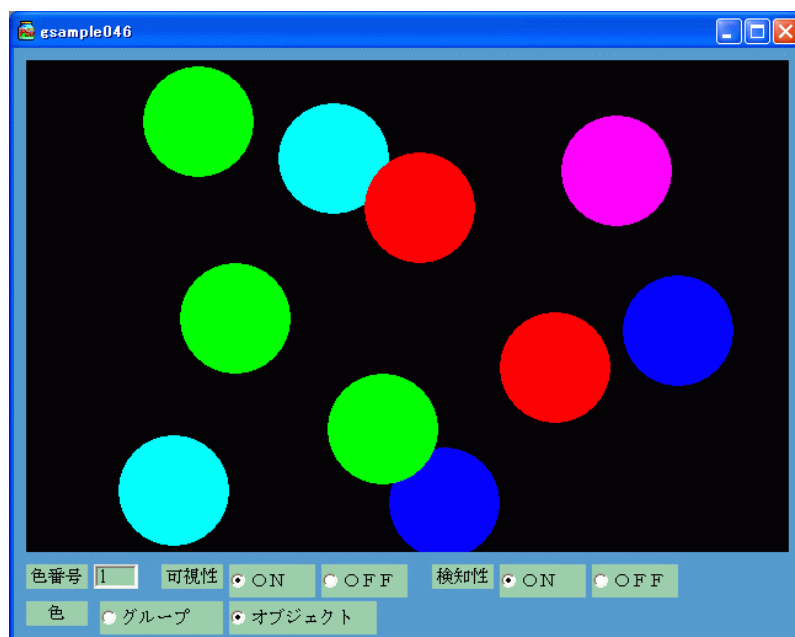


図 2-39 gsamp046.psv 実行画面

(6) 更新履歴

なし

2.4.8 g_group_color_id

グループオブジェクトの色 ID の取得／変更

(1) 形式

`g_group_color_id contextID [-g] groupNo [colorID]`

(2) 説明

`g_group_color_id` マクロは、グループオブジェクトのグループ番号ごとの色 ID を取得／変更します。引数に色 ID を指定しない場合は、色 ID が取得されます。指定する場合は、色 ID が変更され、変更後の値が返ります。設定した色 ID は、`g_draw_obj` での図形オブジェクトの描画時に参照されます。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-g groupID*

コンテキストオブジェクトに設定しているグループオブジェクト以外のグループオブジェクトを指定する場合に、`g_open_group` で作成したグループオブジェクト ID を指定します。

③ *groupNo*

グループ番号を指定します。

④ *colorID*

`g_alloc_color` で作成した色 ID を指定します (`g_color_id` 参照)。

(3) 戻り値

色 ID

(4) 関連項目

`g_context`, `g_draw_obj`, `g_alloc_color`, `g_open_group`, `g_make_group`,
`g_group_no`, `g_group_attrb`

(5) 使用例

gsamp046.psv参照 (画面例は2.4.7の画面参照)

(6) 更新履歴

なし

2.4.9 g_group_no

図形オブジェクトのグループ番号を取得／変更

(1) 形式

`g_group_no contextID [-g] objID [groupNo]`

(2) 説明

`g_group_no` マクロは、図形オブジェクトのグループ番号を取得／変更します。引数にグループ番号を指定しない場合はグループ番号が取得されます。指定する場合は、グループ番号が変更され、変更後の値が返ります。グループ番号の設定については、`g_make_group` の説明を参照してください。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-g groupID*

コンテキストオブジェクトに設定しているグループオブジェクト以外のグループオブジェクトを指定する場合に、`g_open_group` で作成したグループオブジェクト ID を指定します。

③ *objID*

作成した図形オブジェクトの ID を指定します。

④ *groupNo*

変更するグループ番号を数字で指定します。

(3) 戻り値

グループ ID グループ番号（リスト形式）

(4) 関連項目

`g_context`, `g_open_group`, `g_make_group`, `g_line`, `g_polyline`, `g_polygon`,
`g_regpoly`, `g_round_box`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`, `g_symbol`,
`g_usr_def`

(5) 使用例

gsamp046.psv参照 (画面例は2.4.7の画面参照)

(6) 更新履歴

なし

2.4.10 g_gwin_attrib

グラフィックスウィンドウの図形属性値の取得／変更

(1) 形式

```
g_gwin_attrib contextID [-w] [attribVal]
```

(2) 説明

`g_gwin_attrib` マクロは、グラフィックスウィンドウのカレントの図形属性値を取得／変更します。引数で属性値を指定しない時は、カレント値が取得されます。指定する場合はカレント値が変更され、変更後の値が返ります。本マクロの設定値は、図形描画時の既定値になります。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-w gwinID*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *attribVal*

変更する属性値を数字で指定します (`g_attrib` 参照)。

(3) 戻り値

カレントの属性値

(4) 関連項目

`g_attrib`, `g_context`, `g_open_gwin`

(5) 使用例

gsamp049.psv 参照 (以下画面例)

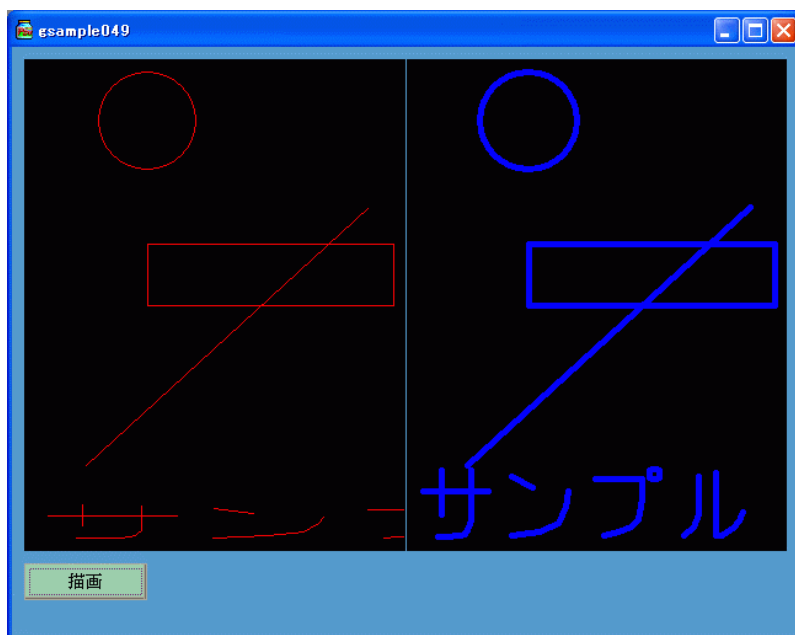


図 2-40 gsamp049.psv 実行画面

(6) 更新履歴

なし

2.4.11 g_gwin_color

グラフィックスウィンドウの色 ID の取得／変更

(1) **形式**

`g_gwin_color context/D [-w] [color/D]`

(2) **説明**

`g_gwin_color` マクロは、グラフィックスウィンドウのカレントの色 ID を取得／変更します。引数で色 ID を指定しない時は、カレント値が取得されます。指定する場合はカレント値が変更され、変更後の値が返ります。本マクロの設定値は、図形描画時の既定値になります。

① *context/D*

`g_context` で作成したコンテキスト ID を指定します。

② *-w gwin/D*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *color/D*

`g_alloc_color` で作成した色 ID を数字で指定します。

(3) **戻り値**

色 ID

(4) **関連項目**

`g_alloc_color`, `g_context`, `g_open_gwin`

(5) **使用例**

`gsamp049.psv`参照 (画面例は2.4.10の画面参照)

(6) **更新履歴**

なし

2.4.12 g_gwin_tx_attrb

グラフィックスウィンドウの文字列属性値の取得／変更

(1) 形式

`g_gwin_tx_attrb context/D [-w] [-h] [-w] [-r] [-o] [-s] [-f]`

(2) 説明

`g_gwin_tx_attrb` マクロは、グラフィックスウィンドウのカレントの文字列属性値を取得／変更します。引数で属性値を指定しない時は、カレント値が取得されます。指定する場合はカレント値が変更されます。カレント値の取得時はオプションを 1 個だけ指定してください。本マクロの設定値は、図形描画時の既定値になります。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwin/D`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する 場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-h [heightVal]`

文字高さを数字で指定します (WC 座標単位)。

④ `-w [widthVal]`

文字幅を数字で指定します (WC 座標単位)。

⑤ `-r [deg]`

文字列の角度を反時計回りで 1/10 度単位で指定します。

⑥ `-o [h|v]`

文字列方向を指定します。

h : 横書き

v : 縦書き

⑦ `-s [spaceVal]`

文字間隔を数字で指定します (WC 座標単位)。

⑧ `-f [fontType]`

フォントタイプを数字で指定します (0 : ベクトルフォント, 1 : MS ゴシック, 2 : MS 明朝)。

(3) 戻り値

取得時に指定したオプション文字列に対応するカレントの属性値。

(4) 関連項目

`g_context`, `g_open_gwin`

(5) **使用例**

gsamp049.psv参照 (画面例は2. 4. 10の画面参照)

(6) **更新履歴**

① **Ver. 5. 0**

フォントタイプ用のオプション“-f”を追加しました。

2.5 H~L で始まるコマンド

2.5.1 g_hash_del

ハッシュオブジェクトのデータ削除

(1) **形式**

`g_hash_del hashID key [key...]`

(2) **説明**

`g_hash_del` マクロは、ハッシュオブジェクトからデータを削除します。

① *hashID*

`g_open_hash` で作成したハッシュオブジェクト ID を指定します。

② *key [key...]*

削除するデータのキーを指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_open_hash`, `g_hash_store`, `g_hash_fetch`, `g_close_hash`

(5) **使用例**

gsamp066.psv参照 (画面例は2. 2. 8の画面参照)

(6) **更新履歴**

なし

2.5.2 g_hash_fetch

ハッシュオブジェクトのデータ検索

(1) **形式**

`g_hash_fetch hashID key [key...]`

(2) **説明**

`g_hash_fetch` マクロは、ハッシュオブジェクトから、キーを指定してデータを検索します。キーに該当するデータが戻り値に設定されます。

① *hashID*

`g_open_hash` で作成したハッシュオブジェクト ID を指定します。

② *key [key...]*

検索するデータのキーを指定します。

(3) **戻り値**

データ (リスト形式)

(4) **関連項目**

`g_open_hash`, `g_hash_store`, `g_hash_del`, `g_close_hash`

(5) **使用例**

gsamp066.psv参照 (画面例は2.2.8の画面参照)

(6) **更新履歴**

なし

2.5.3 g_hash_store

ハッシュオブジェクトへのデータ挿入

(1) **形式**

`g_hash_store hashID r|i key body [key body...]`

(2) **説明**

`g_hash_store` マクロは、ハッシュオブジェクトにデータを挿入します。

① *hashID*

`g_open_hash` で作成したハッシュオブジェクト ID を指定します。

② *r|i*

挿入モードを指定します。

r : 上書きモード (同一キーが存在すると上書き)

i : 通常モード (同一キーが存在するとエラー)

③ *key body [key body...]*

key はデータのキーを指定します。*body* はデータを指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_open_hash`, `g_hash_fetch`, `g_hash_del`, `g_close_hash`

(5) **使用例**

`gsamp066.psv` 参照 (画面例は 2.2.8 の画面参照)

(6) **更新履歴**

なし

2.5.4 g_image_dsp

グラフィックスウィンドウのイメージオブジェクトの指定／取り消し

注) Ver. 4 でマップ概念導入のため廃止した API です。

(1) **形式**

`g_image_dsp contextID gwinID off|on imageID`

(2) **説明**

`g_image_dsp` マクロは、`g_open_gwin` で作成したグラフィックスウィンドウの背景のイメージオブジェクトを指定します。本マクロは、作成時にイメージオブジェクトを指定されたグラフィックスウィンドウのみ有効です。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *gwinID*

イメージオブジェクトを指定するグラフィックスウィンドウの ID を指定します。

③ *off*

イメージオブジェクトの指定を取り消します。

④ *on imageID*

背景のイメージを `g_open_image` で作成したイメージオブジェクトのイメージ ID を指定します。

(3) **戻り値**

off の場合には指定されていたイメージオブジェクト ID

(4) **関連項目**

`g_add_image`, `g_open_gwin`, `g_open_image`, `g_close_image`

(5) **使用例**

なし

(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 Ver. 4.0

Ver. 4.0 で削除したマクロです。互換性用マクロファイル `bcomp.psv` を参考に `g_open_image_region` マクロ関連の機能に置き換えてください。

2.5.5 g_ins_obj

ディスプレイリストメモリへの図形オブジェクト格納

(1) **形式**

`g_ins_obj contextID [-d] objID [objID...]`

(2) **説明**

`g_ins_obj` マクロは、図形オブジェクトをディスプレイリストメモリに格納します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-d dlistID*

コンテキストオブジェクトに設定しているディスプレイリストメモリ以外のディスプレイリストメモリを指定する場合に、ディスプレイリストメモリ ID を指定します。`g_open_dlist` で作成したディスプレイリストメモリ ID を指定します。

③ *objID [objID...]*

格納する図形オブジェクトの図形オブジェクト ID を指示します。

(3) **戻り値**

格納した図形オブジェクト ID

(4) **関連項目**

`g_context`, `g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`,
`g_text`, `g_symbol`, `g_usr_def`, `g_rm_obj`, `g_search_obj`, `g_round_box`, `g_regpoly`

(5) **使用例**

gsamp002.psv参照 (画面例は2.1.3の画面参照)

(6) **更新履歴**

なし

2.5.6 g_JBL2JXY

JGD2000 の緯度・経度を平面直交座標系に変換

(1) 形式

`g_JBL2JXY context/D JB JL`

(2) 説明

`g_JBL2JXY` マクロは日本測地系 2000 の緯度・経度座標を、GRS-80 楕円体パラメータを使用して 19 系座標に変換します。

変換後の XY 座標はリスト形式で出力されます。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `JB`

日本測地系 2000 の緯度をラディアン値で指定します。指定する変数の精度は単数精度以上の精度で指定しないと、変換誤差の原因となります。

③ `JL`

日本測地系 2000 の経度をラディアン値で指定します。指定する変数の精度は単数精度以上の精度で指定しないと、変換誤差の原因となります。

(3) 戻り値

19 系座標の XY 座標を“X 座標 Y 座標”の並びのリスト構造で返します。数値の単位は“m”で、精度は倍数精度で返します。

(4) 関連項目

`g_TBL2JBL`, `g_TBL2TXY`

(5) 使用例

`gsamp108.psv` 参照 (以下画面例)

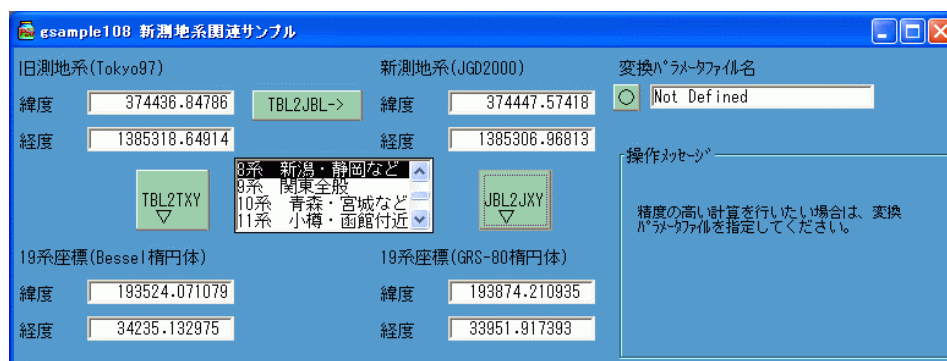


図 2-41 `gsamp108.psv` 実行画面

(6) 更新履歴

① Ver. 5.2

Ver. 5.2 で新規に追加したマクロです。

2.5.7 g_layer_attrb

レイヤオブジェクトの属性値の取得／変更

(1) 形式

g_layer_attrb contextID [-l] layerNo [attribVal]

(2) 説明

g_layer_attrb マクロは、レイヤオブジェクトのレイヤ番号ごとの属性値を取得／変更します。引数に属性値を指定しない場合は、属性値が取得されます。属性値を指定する場合は、属性値が変更され、変更後の属性値が返ります。

属性のうち検知性、可視性のみ有効です。設定した属性は、g_search_obj での図形オブジェクトの検索、g_draw_obj での図形オブジェクトの描画の際に参照されます。

① contextID

g_context で作成したコンテキスト ID を指定します。

② -l layerID

コンテキストオブジェクトに設定しているレイヤオブジェクト以外のレイヤオブジェクトを指定する場合に、g_open_layer で作成したレイヤオブジェクト ID を指定します。

③ layerNo

レイヤ番号を指定します。

④ attribVal

次の 32 ビットを 10 進の整数で指定します。

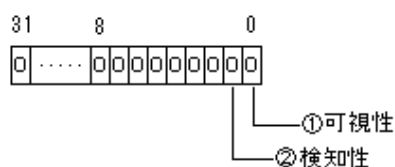


図 2-42 レイヤオブジェクト属性値ビット割り当て

(a) 可視性 (ビット 0)

図形オブジェクトの可視性を指定します。不可視とすると、g_draw_obj で描画を実行してもその図形オブジェクトを描画しません。

0 : 不可視

1 : 可視

(b) マウスの検知性 (ビット 1)

g_search_obj マクロで図形オブジェクトを検知可能とすることが指定します。

0 : 検知しない

1 : 検知する

(3) 戻り値

属性値

(4) 関連項目

`g_context`, `g_draw_obj`, `g_search_obj`, `g_attrib`, `g_open_layer`, `g_layer_no`,
`g_layer_color_id`

(5) 使用例

`gsamp056.psv` 参照 (以下画面例)

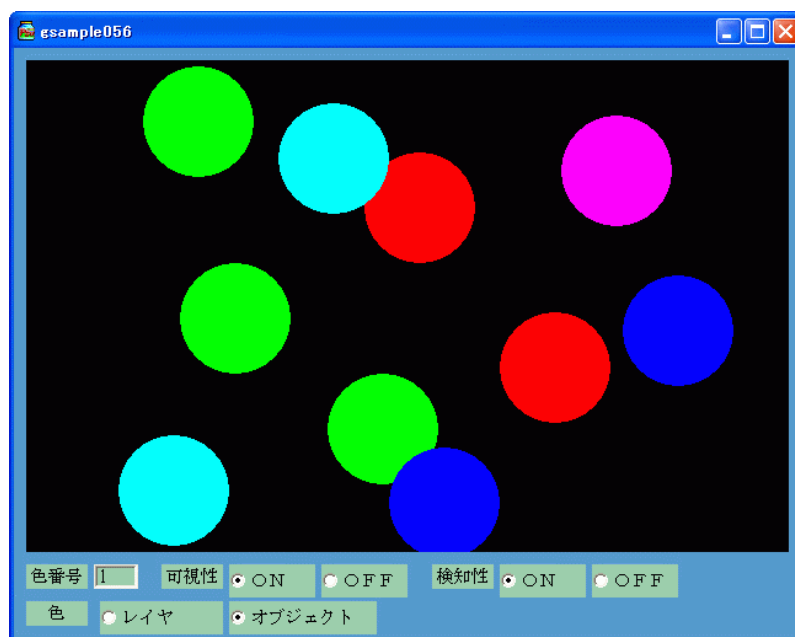


図 2-43 `gsamp056.psv` 実行画面

(6) 更新履歴

なし

2.5.8 g_layer_color_id

レイヤオブジェクトの色 ID の取得／変更

(1) **形式**

`g_layer_color_id context/ID [-l] /layerNo [color/ID]`

(2) **説明**

`g_layer_color_id` マクロは、レイヤオブジェクトのレイヤ番号ごとの色 ID を取得／変更します。引数に色 ID を指定しない場合は、色 ID が取得されます。指定する場合は、色 ID が変更され、変更後の値が返ります。

設定した色 ID は、`g_draw_obj` での図形オブジェクトの描画時に参照されます。

① *context/ID*

`g_context` で作成したコンテキスト ID を指定します。

② *-l layer/ID*

コンテキストオブジェクトに設定しているレイヤオブジェクト以外のレイヤオブジェクトを指定する場合に、`g_open_layer` で作成したレイヤオブジェクト ID を指定します。

③ */layerNo*

レイヤ番号を指定します。

④ *color/ID*

`g_alloc_color` で作成した色 ID を指定します (`g_color_id` 参照)。

(3) **戻り値**

色 ID

(4) **関連項目**

`g_context`, `g_draw_obj`, `g_alloc_color`, `g_open_layer`, `g_layer_no`,
`g_layer_attrb`

(5) **使用例**

`gsamp056.psv`参照 (画面例は2.5.7の画面参照)

(6) **更新履歴**

なし

2.5.9 g_layer_name

レイヤ名の取得／設定、レイヤ番号の取得

(1) 形式

```
g_layer_name contextID 0 -n layerNo [-l layerID]
```

```
g_layer_name contextID 1 layerName -n layerNo [-l layerID]
```

```
g_layer_name contextID 2 layerName [-l layerID]
```

(2) 説明

g_layer_name マクロは、レイヤ名の取得／設定を行います。またレイヤ操作時は従来どおりレイヤ番号を使用するため、レイヤ名からレイヤ番号の逆引きも可能です。従って、マクロ自体の動作は 3 モードあり、本マクロの処理は第 2 引数のモード (0~2) で指定可能です。

① contextID

g_context で作成したコンテキスト ID を指定します。第 2 引数 (処理モード)

処理モードは 0~2 の整数値を指定します。

処理モード=0 の場合、指定したレイヤ番号のレイヤ名の取得を行います。

処理モード=1 の場合、指定したレイヤ番号へレイヤ名の設定を行います。

処理モード=2 の場合、指定したレイヤ名のレイヤ番号の取得を行います。

② -n layerNo

処理対象とするレイヤの番号を数字で指定します。本マクロで指定する前に、対象のレイヤ番号を使用しておいてください。

※ 図形作成時に対象レイヤ番号を指定することで対象レイヤ番号を使用したとみなします。

③ -l layerID

コンテキストオブジェクトに設定しているレイヤオブジェクト以外のレイヤオブジェクトを指定する場合、g_open_layer で作成したレイヤオブジェクト ID を指定します。

④ layerName

処理モード=1 の場合は、設定するレイヤ名を指定します。

処理モード=2 の場合は番号を取得するレイヤの名称を指定します。

(3) 戻り値

処理モード=0 の時：指定レイヤの名称。

エラー時はエラーメッセージを参照願います。

処理モード=1 の時：設定に成功したレイヤの名称。

エラー時はエラーメッセージを参照願います。

処理モード=2 の時：指定したレイヤの番号。

エラー時はエラーメッセージを参照願います。

(4) 関連項目

`g_context`, `g_open_layer`, `g_layer_no`

(5) 使用例

`gsamp120.psv` 参照 (以下画面例)

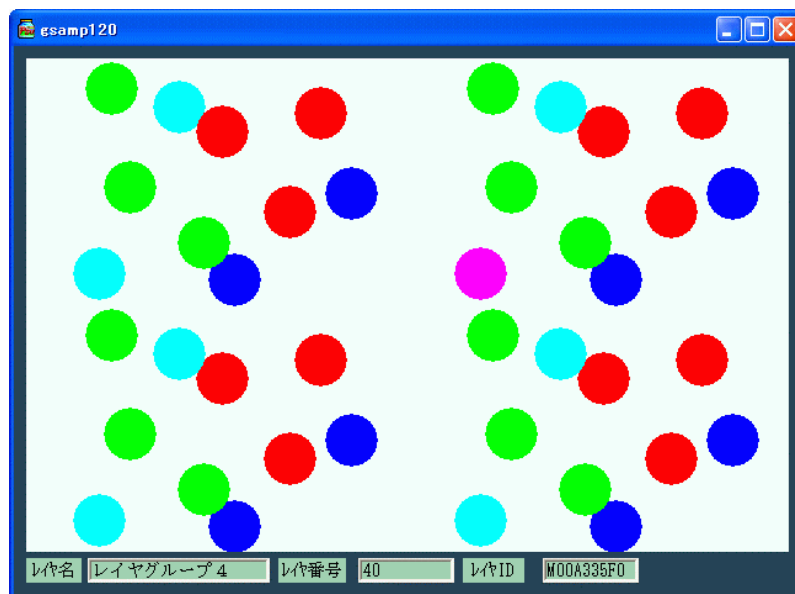


図 2-44 `gsamp120.psv` 実行画面

(6) 更新履歴

① WebTcl

WebTcl で新規に追加したマクロです。

2.5.10 g_layer_no

図形オブジェクトのレイヤ番号の取得／変更

(1) **形式**

`g_layer_no contextID [-l] objID [/layerNo]`

(2) **説明**

`g_layer_no` マクロは、図形オブジェクトのレイヤ番号を取得／変更します。引数にレイヤ番号を指定しない場合、レイヤ番号が取得されます。指定する場合はレイヤ番号が変更され、変更後の値が返ります。

なお、同一の図形に複数のレイヤ番号を設定することはできません。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-l layerID*

コンテキストオブジェクトに設定しているレイヤオブジェクト以外のレイヤオブジェクトを指定する場合、`g_open_layer` で作成したレイヤオブジェクト ID を指定します。

③ *objID*

作成した図形オブジェクトの ID を指定します。

④ */layerNo*

変更するレイヤ番号を数字で指定します。

(3) **戻り値**

レイヤオブジェクト ID レイヤ番号（リスト形式）

(4) **関連項目**

`g_context`, `g_open_layer`, `g_line`, `g_polyline`, `g_polygon`, `g_round_box`,
`g_regpoly`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`, `g_symbol`, `g_usr_def`

(5) **使用例**

`gsamp056.psv`参照 (画面例は2.5.7の画面参照)

(6) **更新履歴**

なし

2.5.11 g_layer_range

レイヤオブジェクトの表示縮尺範囲の取得／変更

(1) **形式**

`g_layer_range contextID [-1] layerNo [minVal maxVal]`

(2) **説明**

`g_layer_range` マクロは、レイヤオブジェクトのレイヤ番号ごとの表示の縮尺範囲を取得／変更します。引数に表示範囲の縮尺値を指定しない場合は、現在設定されている縮尺値が取得されます。値を指定する場合は、表示範囲の縮尺値が変更され、変更後の値が返ります。

表示範囲は縮尺値の分母を整数で指定します。グラフィックスウィンドウの縮尺値が指定範囲外の場合は自動的に非表示となります。

表示縮尺値の設定無しの場合は -1 となります。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-1 layerID*

コンテキストオブジェクトに設定しているレイヤオブジェクト以外のレイヤオブジェクトを指定する場合に、`g_open_layer` で作成したレイヤオブジェクト ID を指定します。

③ *layerNo*

レイヤ番号を指定します。

④ *minVal maxVal*

表示範囲の最低値と最大値。設定しない場合は -1 を指定します。

(3) **戻り値**

表示範囲の最低値と最大値。

(4) **関連項目**

`g_context`, `g_draw_obj`, `g_search_obj`, `g_attrib`, `g_open_layer`, `g_layer_no`, `g_layer_color_id`, `g_scale`

(5) 使用例

gsamp104.psv 参照 (以下画面例)

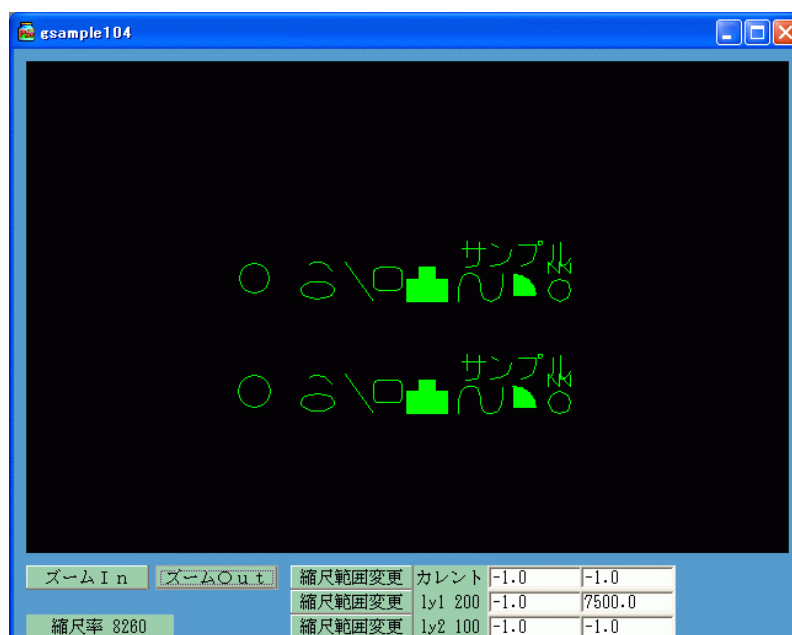


図 2-45 gsamp104.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.5.12 g_line

線オブジェクトの作成

(1) 形式

`g_line contextID [-dc] x1 y1 x2 y2 [[-t] [-c] [-a] [-g] [-l]] [-u]`

(2) 説明

`g_line` マクロは、線オブジェクトを作成します。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-dc`

座標を DC 座標で指定する時に指定します。

③ `x1 y1 x2 y2`

線の座標を指定します。始点 ($x1, y1$)、終点 ($x2, y2$) の順に指定します。

④ `-t tagID`

図形オブジェクトのタグ ID を数字で指定します。

⑤ `-c colorID`

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑥ `-a attribVal` (Ver. 4 以降削除 A版互換モードの場合)

図形オブジェクトの属性を数字で指定します (省略した場合には、コンテキストオブジェクトのカレント値を使用します。`g_attr` 参照)。

⑦ `-a objAttribVal drawAttribVal`

図形オブジェクトの属性値と描画属性値を数字で指定します (省略した場合には、コンテキストオブジェクトのカレント値を使用します。`g_obj_attr`、`g_draw_attr` 参照)。

⑧ `-g [groupID] groupNo`

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (`groupID`) 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グループオブジェクトが対象となります。

⑨ `-l [layerID] layerNo`

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID は、コンテキストオブジェクトのカレントのレイヤオブジェクトが対象となります。

⑩ `-u key body [key body ...]`

図形オブジェクトに格納するユーザ定義データのキー (`key`) とデータ (`body`) を指定します。

(3) 戻り値

線の図形オブジェクト ID

(4) 関連項目

`g_attr`, `g_context`, `g_draw_obj`, `g_mode`, `g_open_group`, `g_open_layer`

(5) 使用例

`gsamp059.psv` 参照 (以下画面例)

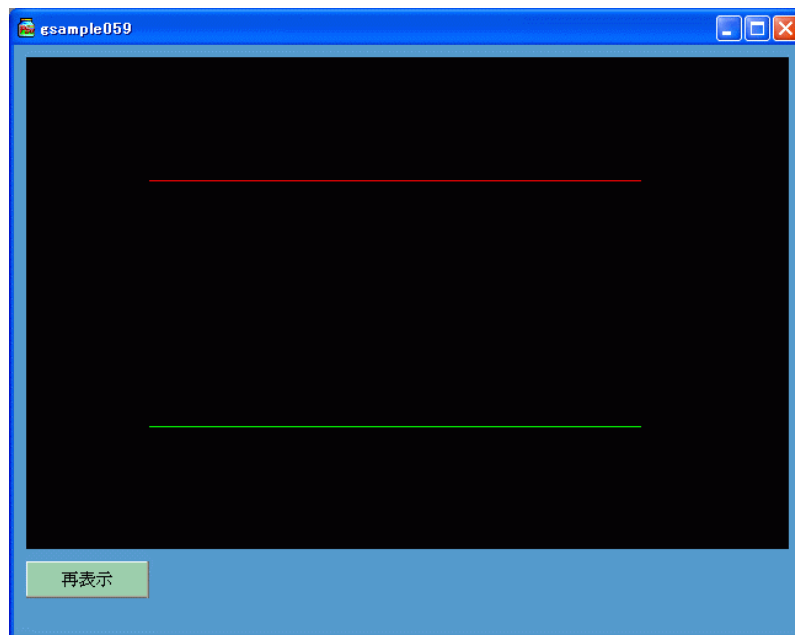


図 2-46 `gsamp059.psv` 実行画面

(6) 更新履歴

① Ver. 4.0

属性値のオプションの指定を変更しました (-a)。

2.5.13 g_linesymbol

ラインシンボルオブジェクトの作成

注) ラインシンボルオブジェクトは廃止予定の API です。新規のプログラムで使用しないでください。

(1) 形式

```
g_linesymbol contextID num [-dc] coords symNo tsymNo scale [[-t] [-c] [-a]
[-g] [-l]] [-u]
```

(2) 説明

g_linesymbol マクロはラインシンボルオブジェクトを作成します。

注) 一度作成したシンボルを再度作成する場合、初期に生成したシンボルでオブジェクトを作成します。そのため、ラインシンボル定義ファイル・シンボルサイズ（倍率）を変更しても、初期作成時のシンボル・サイズになります。

① contextID

g_context で作成したコンテキスト ID を指定します。

② num

折れ線の座標の点数を指定します。

③ -dc

座標を DC 座標で指定する時に指定します。

④ coords

x1 y1 x2 y2 ... の形式で、折れ線の座標を指定します。点 (x1、y1)、点 (x2、y2) ... の順に指定します。

⑤ symNo

繰り返し用シンボルの指定。g_linesymbol_file で指定したシンボルファイル中のシンボル番号を数字で指定します。

⑥ tsymNo

端点用シンボルの指定。g_linesymbol_file で指定したシンボルファイル中のシンボル番号を数字で指定します。

⑦ scale

倍率を指定します。

⑧ -t tagID

図形オブジェクトのタグ ID を数字で指定します。

⑨ -c colorID

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑩ `-a objAttribVal drawAttribVal`

図形オブジェクトの属性値と描画属性値を数字で指定します (`g_obj_attrib`、`g_draw_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑪ `-g [groupID] groupNo`

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクトID(*groupID*)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グループオブジェクトが対象となります。

⑫ `-l [layerID] layerNo`

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクトID(*layerID*)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ヤオブジェクトが対象となります。

⑬ `-u key body [key body ...]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) **戻り値**

シンボルの図形オブジェクト ID

(4) **関連項目**

`g_context`, `g_draw_obj`, `g_linesymbol_file`, `g_symbol`, `g_open_group`,
`g_open_layer`, `g_obj_attrib`, `g_draw_attrib`

(5) 使用例

gsamp100.psv 参照 (以下画面例)

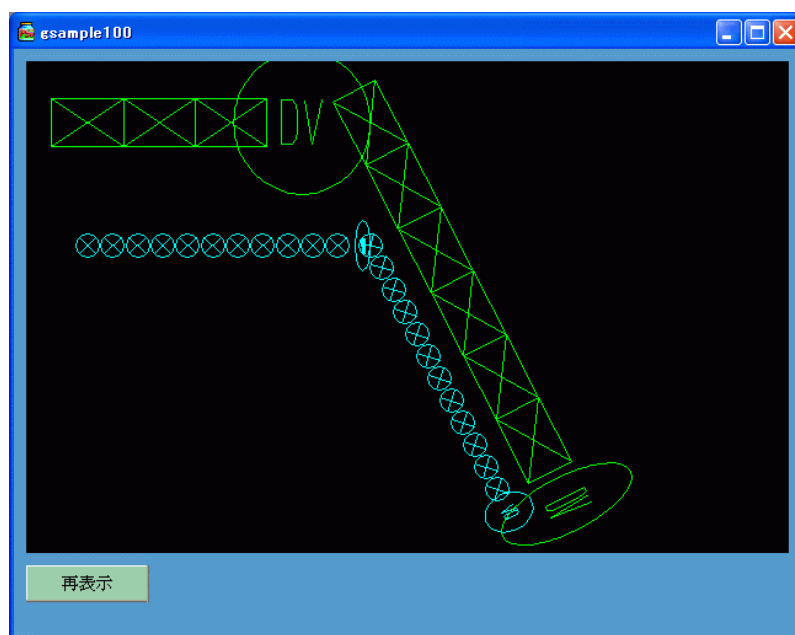


図 2-47 gsamp100.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.5.14 g_linesymbol_file

ラインシンボル定義ファイルの設定

注) ラインシンボル定義ファイルは廃止予定の API です。新規のプログラムで使用しないでください。

(1) 形式

`g_linesymbol_file [fileName] | [-s FileID]`

(2) 説明

`g_linesymbol_file` マクロは、ラインシンボル定義ファイルを指定／取得します。

① *fileName*

ラインシンボル定義ファイルを指定します。引数にファイル名を指定しない時はラインシンボル定義ファイルの現在の登録 ID が返ります。

② `-s FileID`

既に登録済みのシンボル定義ファイルを指定する場合は、登録時の戻り値である登録 ID を指定します。

(3) 戻り値

ラインシンボル定義ファイルの登録 ID

(4) 関連項目

`g_linesymbol`

(5) 使用例

gsamp100.psv参照 (画面例は2.5.13の画面参照)

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.6 M で始まるコマンド

2.6.1 g_make_group

図形オブジェクトのグループ化

(1) 形式

```
g_make_group contextID [-g] groupNo [-i] [-o] [a|x] objID [ [-o] [a|x]
objID ... ]
```

```
g_make_group contextID [-g] groupNo -r
```

(2) 説明

`g_make_group` マクロは、図形オブジェクトを指定のグループ番号でグループ化します。また、図形オブジェクトのグループ化を解除します。図形オブジェクトとグループ番号の対応は、次の 2 つのオブジェクトに設定されます。

- グループオブジェクト

グループ番号ごとに、グループに属する図形オブジェクトの ID を設定します。一つの図形オブジェクトが複数のグループに属する場合、それぞれのグループ番号に図形オブジェクト ID を設定します。

- 図形オブジェクト

図形オブジェクトの属するグループのグループオブジェクト ID とグループ番号を設定します。ただし、図形オブジェクトは 1 つのグループだけに登録可能で、最後に `g_make_group` で指定したグループにだけ属します。

このグループ番号を、次の場合に参照します

- `g_search_obj` での図形オブジェクト検索時、グループ番号に設定されている属性中の検知性の設定がチェックされます。
- `g_draw_obj` での図形オブジェクトの描画時、グループ番号に設定されている色 ID と、属性中の可視性の設定がチェックされます。

`g_make_group` マクロではグループ化の際、グループオブジェクトに図形オブジェクト ID を設定すると同時に、図形オブジェクトにグループ番号を設定するかどうかを設定できます（`-o` 指定）。上記を考慮し、設定要否を決定してください。

① contextID

`g_context` で作成したコンテキスト ID を指定します。

② -g groupID

コンテキストオブジェクトに設定しているグループオブジェクト以外のグループオブジェクトを指定する場合に、`g_open_group` で作成したグループオブジェクト ID を指定します。

③ groupNo

グループ番号を指定します。

④ -i

既に同一のグループ番号でグループ化している図形オブジェクトがあった時の処置を指定します。

-i を省略すると、今回指定した図形オブジェクトをグループに追加します。

-i を指定すると、既存の図形オブジェクトのグループ化を解除して、今回指定した図形オブジェクトを新規にグループ化します。

⑤ [-o] [a|x] *objID* [[-o] [a|x] *objID* ...]

グループ化する、またはグループ化を解除する図形オブジェクト ID を指定します。

a を指定すると、図形オブジェクトを指定グループ番号でグループ化します（既定値）。この時-o を指定すると、図形オブジェクトに対してもグループオブジェクト ID とグループ番号を設定します（g_group_no と同じ）。

x を指定すると、図形オブジェクトの指定グループ番号でのグループ化を解除します。

複数の図形オブジェクト ID を指定する場合、-o、a、x は ID ごとに指定します。

⑥ -r

指定のグループ番号でグループ化している図形オブジェクトに対し、まとめてグループ化の解除を行います。

(3) 戻り値

グループオブジェクト ID

(4) 関連項目

g_context, g_open_group, g_group_no, g_group_attrib, g_group_color_id

(5) 使用例

gsamp046.psv参照 (画面例は2.4.7の画面参照)

(6) 更新履歴

なし

2.6.2 g_map_attrb

マップオブジェクトの属性値の取得／変更

(1) 形式

```
g_map_attrb mapID [attVal]
```

(2) 説明

g_map_attrb マクロは、マップオブジェクトの属性値を取得／変更します。引数に属性値を指定しない時は、現在の属性値が取得されます。指定する場合は属性値が変更され、変更後の値が返ります。

① mapID

g_open_map で作成した マップオブジェクト ID を指定します。

② attVal

次の 32 ビットの整数で指定します。

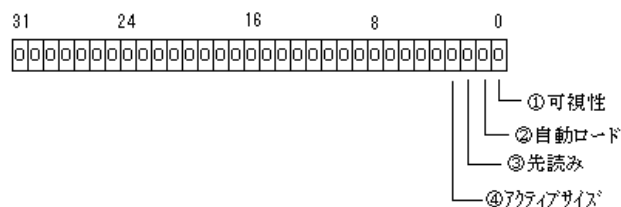


図 2-48 マップオブジェクト属性値ビット割り当て

(a) 可視性 (ビット 0)

マップオブジェクトの可視性を指定します。不可視とすると、そのマップオブジェクトで管理している図形オブジェクトを描画しません。

0 : 不可視

1 : 可視

(b) 自動ロード (ビット 1)

g_ortho や g_scroll、g_scale マクロで グラフィックス・ウィンドウの表示範囲を変更した時、地図データが未ロードであれば、自動的にロードします。

0 : 自動ロードしない

1 : 自動ロードする

(c) 先読み (ビット 2)

g_ortho や g_scroll、g_scale マクロでグラフィックス・ウィンドウの表示範囲を変更した時、表示範囲よりもある程度広い範囲の地図データを先読みします。

0 : 先読みを行わない

1 : 先読みを行う

(d) アクティブサイズ (ビット 3)

`g_ortho` や `g_scale` マクロでグラフィックス・ウィンドウの表示縮尺を変更した時、管理している図形オブジェクトの描画範囲 (`g_region_obj` 参照) を再計算します。固定サイズのビットマップシンボルや `g_obj_range` を用いており、描画範囲がグラフィックス・ウィンドウの表示縮尺によって異なる場合は、再計算を行わないと正しく表示できないことが有ります。

0 : 再計算を行わない

1 : 再計算を行う

(3) 戻り値

属性値

(4) 関連項目

`g_open_map`, `g_close_map`, `g_map_element`, `g_map_type`, `g_map_range`,
`g_map_margin`, `g_map_load`, `g_scale`, `g_ortho`

(5) 使用例

`gsamp102.psv` 参照 (画面例は 2. 2. 12 の画面参照)

(6) 更新履歴

① Ver. 4. 0

Ver. 4. 0 で新規に追加したマクロです。

2.6.3 g_map_element

マップオブジェクトの構成要素の取得／変更

(1) 形式

```
g_map_element mapID -r  
g_map_element mapID -d|-a dlistID [dlistID ...]  
g_map_element mapID -d|-a imageID [imageID ...]
```

(2) 説明

`g_map_element` マクロは、マップオブジェクトの構成要素を取得／変更します。構成要素は、マップオブジェクトがベクトルの場合ディスプレイリストメモリ、ラスタの場合はイメージオブジェクトとなります。引数を指定しない時は、現在の構成要素が取得されます。

① *mapID*

`g_open_map` で作成したマップオブジェクト ID を指定します。

② -r

指定のマップオブジェクトの構成要素数をゼロにします。

③ -d *dlistID* ...

ディスプレイリストメモリを削除する場合に、ディスプレイリストメモリ ID を指定します。

④ -a *dlistID* ...

ディスプレイリストメモリを追加する場合に、`g_open_dlist` で作成したディスプレイリストメモリ ID を指定します。

⑤ -d *imageID* ...

イメージオブジェクトを削除する場合に、イメージ ID を指定します。

⑥ -a *imageID* ...

イメージオブジェクトを追加する場合に、`g_open_image_region` で作成したイメージ ID を指定します。

(3) 戻り値

構成要素の ID (リスト形式)

(4) 関連項目

`g_open_map`, `g_close_map`, `g_map_attrib`, `g_map_type`, `g_map_range`,
`g_map_margin`, `g_map_load`

(5) 使用例

gsamp099.psv参照 (画面例は2.2.10の画面参照)

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.6.4 g_map_load

地図データの読み込み

(1) **形式**

`g_map_load mapID x1 y1 x2 y2`

(2) **説明**

`g_map_load` マクロは、読み込み範囲を指定して、地図データを読み込みます。本マクロはインターフェースのみの提供です。実処理は、各市販地図に対応したプラグイン製品側に実装されます。

マップオブジェクトのオブジェクト属性の自動読み込みを OFF に設定し、アプリケーションがロードする座標範囲を制御する場合に使用します。

① *mapID*

`g_open_map` で作成した マップオブジェクト ID を指定します。

② *x1 y1 x2 y2*

読み込む地図範囲の左下座標 (*x1, y1*)、右上座標 (*x2, y2*) を指定します。

(3) **戻り値**

無し

(4) **関連項目**

`g_open_map`, `g_close_map`, `g_map_element`, `g_map_type`, `g_map_range`,
`g_map_margin`

(5) **使用例**

なし

(6) **更新履歴**

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.6.5 g_map_margin

マップオブジェクト先読み範囲の取得／変更

(1) **形式**

`g_map_margin mapID xmargin ymargin`

(2) **説明**

`g_map_margin` マクロは、地図データの先読みを行う範囲を指定します。引数を指定しない場合は、現在のマージン幅が返ります。

本マクロはインターフェースのみの提供です。実際には、各市販地図に対応したプラグイン製品で実装されます。

① *mapID*

`g_open_map` で作成したマップオブジェクト ID を指定します。

② *xmargin ymargin*

マージンの X 軸方向の幅、および Y 軸方向の幅を指定します。

(3) **戻り値**

無し

(4) **関連項目**

`g_open_map`, `g_close_map`, `g_map_element`, `g_map_type`, `g_map_range`, `g_map_load`

(5) **使用例**

なし

(6) **更新履歴**

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.6.6 g_map_range

マップオブジェクトの表示縮尺範囲の取得／変更

(1) **形式**

`g_map_range mapID [minVal maxVal]`

(2) **説明**

`g_map_range` マクロは、マップオブジェクトの表示範囲を取得／変更します。引数に表示範囲の縮尺値を指定しない場合は、現在設定されている縮尺値が取得されます。値を指定する場合は、表示範囲の縮尺値が変更され、変更後の値が返ります。

表示範囲は縮尺値の分母を整数で指定します。グラフィックスウィンドウの縮尺値が指定範囲外の場合は自動的に非表示となります。

表示縮尺値の設定無しの場合は-1 となります。

① *mapID*

`g_open_map` で作成したマップオブジェクト ID を指定します。

② *minVal maxVal*

表示範囲の最低値と最大値。設定しない場合は-1 を指定します。

(3) **戻り値**

表示範囲の最低値と最大値。

(4) **関連項目**

`g_open_map`, `g_close_map`, `g_map_element`, `g_map_type`, `g_map_attr`,
`g_map_margin`, `g_map_load`, `g_scale`

(5) **使用例**

gsamp102.psv参照 (画面例は2.2.12の画面参照)

(6) **更新履歴**

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.6.7 g_map_type

マップオブジェクトの種別の取得

(1) **形式**

`g_map_type mapID`

(2) **説明**

`g_map_type` マクロは、マップオブジェクトの種別を取得します。

① *mapID*

`g_open_map` で作成したマップオブジェクト ID を指定します。

(3) **戻り値**

ベクトルマップの場合は“vector”、ラスタマップの場合は“raster”の文字列を返します。

(4) **関連項目**

`g_open_map`, `g_close_map`, `g_map_element`, `g_map_attrib`, `g_map_range`,
`g_map_margin`, `g_map_load`, `g_scale`

(5) **使用例**

gsamp102.psv参照 (画面例は2.2.12の画面参照)

(6) **更新履歴**

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.6.8 g_mode

図形オブジェクトのモードの取得／変更

注) Ver. 4 で図形属性を強化したため廃止した API です。

(1) 形式

`g_mode objID [mode]`

(2) 説明

g_mode マクロは、図形オブジェクトのモードを取得／変更します。引数にモードを指定しない時は現在のモードを取得します。指定する時は、モードを変更して、変更後の値が返ります。

① objID

作成した図形オブジェクトの ID を指定します。

② mode

次の 8 ビットの整数で指定します。

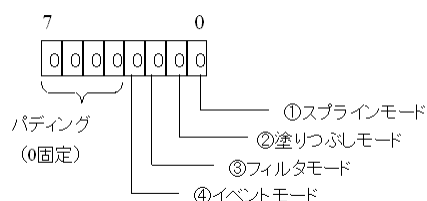


図 2-49 モードビット割り当て

(a) スプラインモード（ビット 0）

折れ線（g_polyline）にだけ有効です。折れ線か、スプライン曲線かを指定します。

0 : 折れ線

1 : スプライン曲線

(b) 塗りつぶしモード（ビット 1）

円（g_circle）、円弧（g_arc）、楕円（g_ellipse）、多角形（g_polygon）、扇型（g_pie）、角丸長方形（g_roundbox）、正多角形（g_regpoly）にだけ有効です。

図形オブジェクトを塗りつぶすか、塗りつぶさないかを指定します。

0 : 塗りつぶさない

1 : 塗りつぶす

塗りつぶす場合、塗りつぶしパターンは図形属性の指定に従います。また、線種の指定は無効となります。

(c) フィルタモード (ビット 2)

`g_draw_obj` でフィルタがかかった状態で描画を実行します。

※ PreSerV 内部で使用するビットのためユーザアプリケーション側で、このビットを変更しないでください。

(d) イベントモード (ビット 3)

現在未使用の予約ビットです。

※ PreSerV 内部で使用するビットのためユーザアプリケーション側で、このビットを変更しないでください。

(3) 戻り値

対象図形のモード値

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_round_box`, `g_regpoly`

(5) 使用例

`gsamp061.psv` 参照 (以下画面例)

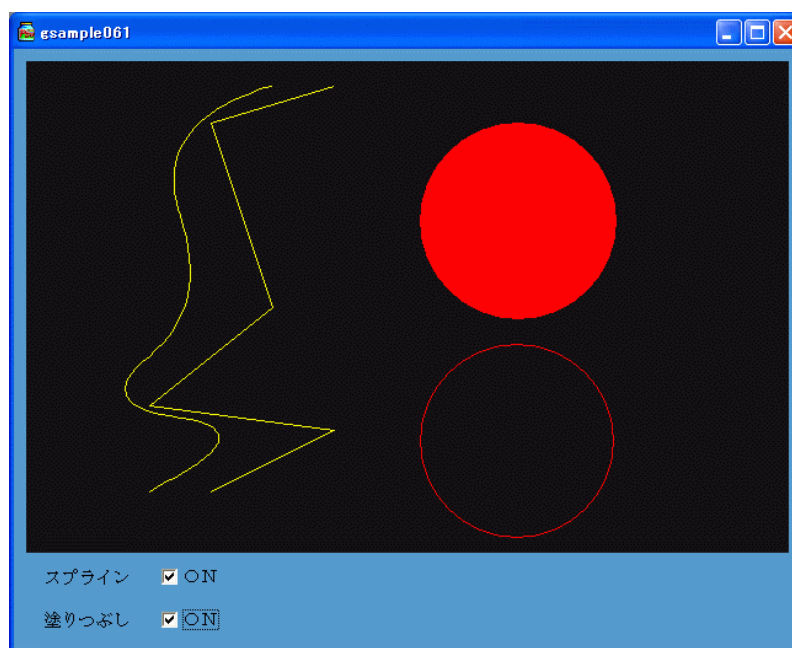


図 2-50 `gsamp061.psv` 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で削除したマクロです。互換性用マクロファイル `bcomp.psv` を参考に `g_obj_attrib`、`g_draw_attrib` マクロに置き換えてください。

2.6.9 mkbsym.exe (MS_DOSコマンド)

ビットマップシンボル定義ファイルの作成ツール

(1) 形式

`mkbsym bmpsymfile -f deffile`

`mkbsym bmpsymfile number bmpfilename [[number bmpfilename] ...]`

(2) 説明

MS_DOS コマンドプロンプトから、ツールを実行してビットマップシンボル定義ファイルを作成します。

① *bmpsymfile*

作成するビットマップシンボル定義ファイル名を指定します。
パス指定は必要です。動作時の相対パスを指定してください。

② `-f deffile`

テキスト形式の定義ファイル名を指定します。テキスト形式の 定義ファイルは、
1 行にビットマップのシンボル番号とファイル名を記述したものです。

シンボル番号	ファイル名

図 2-51 定義ファイル概念図

③ *number bmpfilename ...*

ビットマップの「シンボル番号」と「フォルダー・ファイル名」を指定します。

注) 「フォルダー・ファイル名」はシステム実行時の相対パス (例: ¥bmp¥bsym.bmp) 又は、絶対パス (例 c:¥bmp¥bsym.bmp) で入力してください。定義ファイル登録時に、ビットマップファイル存在チェックは、行いません。

(3) 戻り値

なし

(4) 関連項目

`g_bitmapsymbol_file`, `g_bitmapsymbol`

(5) 使用例

ball/sym : ビットマップシンボル定義ファイル名

bsym1.bmp : ビットマップファイル名 (ビットマップシンボルNO 1 に設定する)

bsym2.bmp : ビットマップファイル名 (ビットマップシンボルNO 2 に設定する)

bsymdef.txt : テキスト形式の定義ファイル名

bsymdef.txt の内容

1 <i>bsym1.bmp</i>

2 <i>bsym2.bmp</i>

テキスト形式の定義ファイル名より設定する

`mkbsym ball/sym -f bsymdef.txt`

ダイレクトにシンボル定義を設定する

`mkbsym ball/sym 1 bsym1.bmp 2 bsym2.bmp`

(6) 更新履歴

① Ver 4.0

Ver. 4.0 で新規に追加したツールです。

2.70 で始まるコマンド

2.7.1 g_obj_attrib

図形オブジェクトの属性値の取得／変更

(1) 形式

g_obj_attrib *objID* [*attribVal*]

(2) 説明

g_obj_attrib マクロは、図形オブジェクトの属性値を取得／変更します。引数に属性値を指定しない時は、現在の属性値が取得されます。指定する場合は属性値が変更され、変更後の値が返ります。

① *objID*

作成した図形オブジェクトの ID を指定します。

② *attribVal*

次の 32 ビットの整数で指定します。

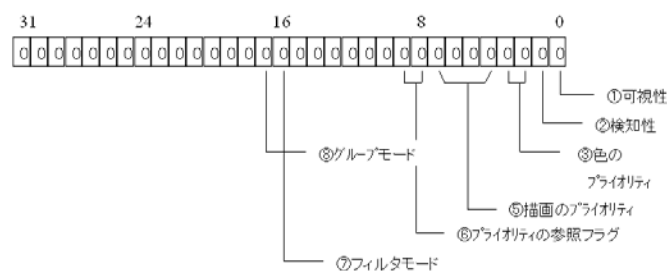


図 2-52 図形オブジェクト属性値ビット割り当て

(a) 可視性（ビット 0）

図形オブジェクトの可視性を指定します。不可視とすると、g_draw_obj で描画を実行してもその図形オブジェクトを描画しません。

0 : 不可視

1 : 可視

(b) マウスの検知性（ビット 1）

g_search_obj マクロで図形オブジェクトを検知可能とすることが指定します。

0 : 検知しない

1 : 検知する

(c) 色のプライオリティ (ビット 2~3)

図形オブジェクト描画時の、次の優先順位に従って、色の設定値を適用します。

a : 図形オブジェクトの設定色

b : 図形オブジェクトのグループ番号の設定色

c : 図形オブジェクトのレイヤ番号の設定色

優先順位にしたがって色の指定有無をチェックします。指定があればその色で描画して、指定がなければ次の指定の有無をチェックします。

0 : a、b、c の順 (既定値)

1 : b、a、c の順

3 : c、a、b の順

(d) 描画のプライオリティ (ビット 4~7)

複数の図形オブジェクトを一括して描画する際のプライオリティを指定します。

数値が大きいものほど後から描画するため、画面上では前面に表示します。

(e) プライオリティの参照フラグ (ビット 8~9)

描画プライオリティ決定の際に、優先順位を指定します。

0 : 図形のプライオリティを参照

1 : グループのプライオリティを参照

2 : レイヤのプライオリティを参照

(f) フィルタモード (ビット 16)

`g_draw_obj` でフィルタがかかった状態で描画を実行します。

※ PreSerV 内部で使用するビットのためユーザアプリケーション側で、このビットを変更しないでください。

(g) グループモード (ビット 17)

現在未使用の予約ビットです。

※ PreSerV 内部で使用するビットのためユーザアプリケーション側で、このビットを変更しないでください。

(3) 戻り値

属性値

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_round_box`, `g_regpoly`, `g_attrib`, `g_mode`

(5) 使用例

`gsamp097.psv` 参照 (画面例は 2. 2. 23 の画面参照)

(6) 更新履歴

① Ver. 4. 0

Ver. 4. 0 で新規に追加したマクロです。

2.7.2 g_obj_range

オブジェクト表示サイズの上限值、下限値の取得／変更

(1) 形式

`g_obj_range contextID [-l] layerNo [minVal maxVal]`

(2) 説明

`g_obj_range` マクロは、レイヤオブジェクトのレイヤ番号ごとのオブジェクトの表示サイズの上限值、下限値を取得／変更します。引数を指定しない場合は、現在設定されている上下限値が取得されます。値を指定する場合は、上下限値が変更され、変更後の値が返ります。

上下限値は DC 座標で指定します。

上下限値の設定無しの場合は 0 となります。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-l layerID*

コンテキストオブジェクトに設定しているレイヤオブジェクト以外のレイヤオブジェクトを指定する場合に、`g_open_layer` で作成したレイヤオブジェクト ID を指定します。

③ *layerNo*

レイヤ番号を指定します。

④ *minVal maxVal*

表示サイズの上限值と下限値。設定しない場合は 0 を指定します。

(3) 戻り値

表示サイズの上限值と下限値。

(4) 関連項目

`g_context`, `g_draw_obj`, `g_search_obj`, `g_attrib`, `g_open_layer`, `g_layer_no`,
`g_layer_color_id`, `g_scale`, `g_map_attrib`

(5) 使用例

gsamp106.psv 参照 (以下画面例)



図 2-53 gsamp106.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.7.3 g_open_dlist

ディスプレイリストメモリのオープン

(1) **形式**

`g_open_dlist contextID [x1 y1 x2 y2]`

(2) **説明**

ディスプレイリストメモリをオープンします。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *x1 y1 x2 y2*

ディスプレイリストメモリに範囲を設定する場合、左下座標 (*x1*、*y1*)、右上座標 (*x2*、*y2*) を指定します。

範囲指定を行った場合、`g_ins_obj` で格納する図形オブジェクトの範囲チェックが行われます。図形オブジェクトが範囲外の場合は、格納しません。

(3) **戻り値**

ディスプレイリストメモリ ID

(4) **関連項目**

`g_context`, `g_close_dlist`, `g_ins_obj`, `g_rm_obj`

(5) **使用例**

`gsamp002.psv`参照 (画面例は2.1.3の画面参照)

(6) **更新履歴**

なし

2.7.4 g_open_filter

フィルタオブジェクトの作成

(1) 形式

```
g_open_filter contextID [-t | -r | -s | -p]
```

(2) 説明

フィルタオブジェクトを作成します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② -t sx sy

移動用のフィルタを作成します。移動分は点 (sx, sy) から、g_filter_set_param で指定する座標の差になります。

③ -r rx ry

回転用のフィルタを作成します。回転軸は点 (rx, ry) となります。

④ -s rx ry

スケール変更用のフィルタを作成します。スケール変更の基点は、点 (rx, ry) となります。

⑤ -p mode [color radius paintflag]

ポリゴン／ポリライン編集用フィルタを作成します。

(a) mode

フィルタのモードを指定します。

-a : 点追加モード

-r : 点変更モード

-x : 点削除モード

(b) color

色替えの色を指定します。

(c) radius

点変更、点削除時の円の半径を指定します。

(d) paintflag

点変更、点削除時の円の塗りつぶしパターンを指定します。

(3) 戻り値

フィルタオブジェクト ID

(4) 関連項目

g_context, g_close_filter, g_filter_set_param, g_filter_add,
g_filter_remove

(5) **使用例**

gsamp063.psv参照 (画面例は2. 2. 5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。

2.7.5 g_open_group

グループオブジェクトの作成

(1) 形式

`g_open_group context/D`

(2) 説明

`g_open_group` マクロは、グループオブジェクトを作成します。コンテキストオブジェクトにカレントのグループオブジェクトが設定されていない場合は、自動的にカレントに設定されます。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

(3) 戻り値

グループオブジェクト ID

(4) 関連項目

`g_context`, `g_cur_group`, `g_make_group`, `g_group_no`, `g_group_attrib`,
`g_group_color_id`, `g_close_group`

(5) 使用例

`gsamp002.psv`参照 (画面例は2.1.3の画面参照)

(6) 更新履歴

なし

2.7.6 g_open_gwin

グラフィックスウィンドウのオープン

(1) 形式

```
g_open_gwin contextID windowID x1 y1 x2 y2
g_open_gwin contextID -p|-h paperSize filename x1 y1 x2 y2 [-s]
g_open_gwin contextID -w printername [-s paperSize]
               [-m leftMargin rightMargin topMargin bottomMargin] x1 y1 x2 y2
```

(2) 説明

g_open_gwin マクロは、グラフィックスウィンドウをオープンします。

① contextID

g_context で作成したコンテキスト ID を指定します。コンテキストオブジェクトにグラフィックスウィンドウが未設定の場合、自動的にカレント値として設定します。

② windowID

W マクロで作成したキャンバスのウィンドウ ID を指定します。

③ x1 y1 x2 y2

グラフィックスウィンドウの初期設定範囲を指定します。左下座標 (x1, y1)、右上座標 (x2, y2) で指定します。

④ -p|-h|-w

プリンタまたはプロッタに印刷する時に指定します。

-p: ポストスクリプト形式のファイルを作成します。

注) ポストスクリプト形式のファイル出力は現版では未サポートです。Windows プリンタ出力を利用してください。

-h: HP/GL2 形式のファイルを作成します。

注) ポストスクリプト形式のファイル出力は現版では未サポートです。Windows プリンタ出力を利用してください。

-w: Windows プリンタに印刷します (GDI 印刷)。

⑤ filename

印刷コードを出力するファイル名を指定します。

⑥ -s

ロール紙を使用しているプロッタの紙を節約します。A0 ロール紙に A1、または A1 ロール紙に A2 を印刷する時に指定します。HP/GL2 形式のファイル出力時だけ有効なオプションです。

⑦ printername

Windows プリンタの場合に、プリンタ名を指定します。

⑧ `-s paperSize`

印刷する紙のサイズと方向を指定します。有効なサイズは以下の通りです。

ポストスクリプト : A4, A3

HP/GL2 : A4, A3, A2, A1, A0

Windows : A4, A3

注) A2, A1, A0 に関しては指定可能ですが、このサイズの用紙は OS 標準サポートでないため、機種に依存したドライバ側の設定が必要な場合があります。

紙のサイズに印刷方向が横の場合は“l”, 縦の場合は“p”をつけて指定します。

例 : A4 横 → a4l, A1 縦 → a1p

デフォルト” a4p”

⑨ `-m leftMargin rightMargin topMargin bottomMargin`

Windows プリンタで印刷出力する領域の余白^{注1)}をmm単位で指定します。

leftMargin: プリンタの左方向の余白サイズ(mm単位)^{注2)}。デフォルト値 0mm。

rightMargin: プリンタの右方向の余白サイズ(mm単位)^{注2)}。デフォルト値 0mm。

topMargin: プリンタの上方向の余白サイズ(mm単位)^{注3)}。デフォルト値 0mm。

bottomMargin: プリンタの下方向の余白サイズ(mm単位)^{注3)}。デフォルト値 0mm。

注1) 各マージン値は、ハードウェアで定義される余白の最小サイズ以下の値を指定した場合は、ハードウェアの余白値に自動修正されます。また、半角数字以外の値を指定した場合もハードウェアの余白値に自動修正されます。

注2) *leftMargin*と*rightMargin*の最大値は用紙物理サイズ幅の 1/2 です。これを越えた場合は、用紙物理サイズ幅の 1/2 に自動修正されます。*leftMargin*と*rightMargin*を同時に最大値を越えるように値を設定すると、機種により動作が不定となります。同時に最大値を越えるように設定しないでください。

注3) *topMargin*と*bottomMargin*の最大値は用紙物理サイズ高さの 1/2 です。これを越えた場合は、用紙物理サイズ高さの 1/2 に自動修正されます。*topMargin*と*bottomMargin*を同時に最大値を越えるように値を設定すると、機種により動作が不定となります。同時に最大値を越えるように設定しないでください。

(3) 戻り値

グラフィックスウィンドウ ID

(4) 関連項目

`g_context`, `g_close_gwin`, `g_showpage`, `g_gwin_attrib`, `g_gwin_color`,
`g_gwin_tx_attrib`, `g_clear`, `g_scroll`, `g_rotate`, `g_viewport`, `g_ortho`,
`g_compute_size`, `g_wc2dc`, `g_dc2wc`, `g_raster`, `g_open_image`, `g_close_image`,
`g_add_image`, `g_image_dsp`

(5) 使用例

print.psv 参照 (以下画面例)

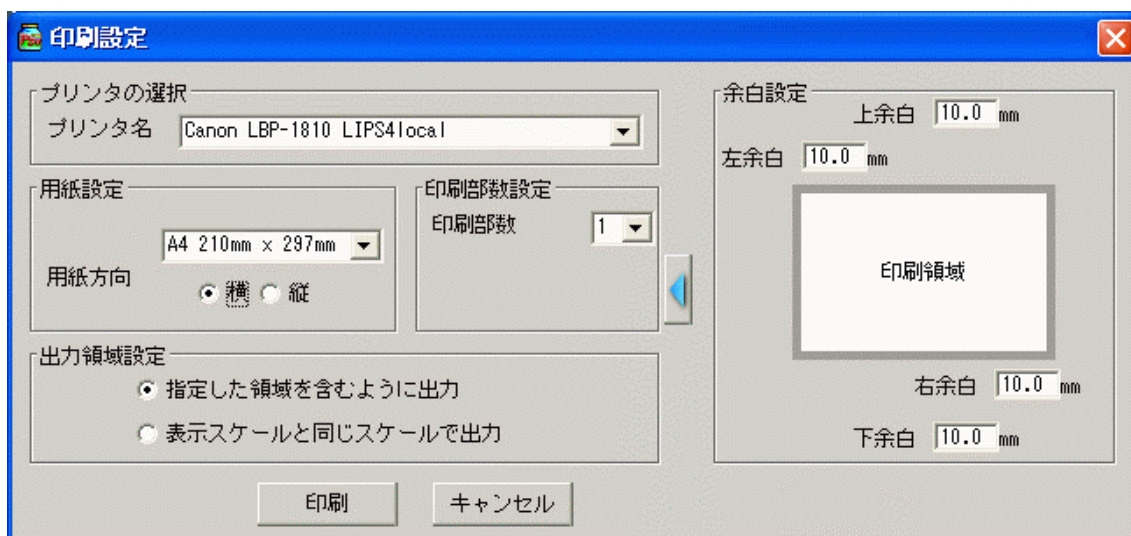


図 2-54 print.psv 実行画面

(6) 更新履歴

① Ver. 3.0

Windows プリンタに印刷可能になりました。

背景にイメージを指定可能になりました。

② Ver. 4.0

-i オプションは廃止しました。イメージの貼付はイメージリージョン (g_open_image_region) を使用してください。

③ Ver. 6.1

-m オプションを追加しました。-m オプションにより、印刷時の余白調整が mm 単位で可能です。

2.7.7 g_open_hash

ハッシュオブジェクトの作成

(1) **形式**

`g_open_hash [table_size]`

(2) **説明**

`g_open_hash` マクロは、ハッシュオブジェクトを作成します。

① *table_size*

テーブルサイズを指定します。省略した場合は、既定値の ‘1009’ を設定します。
テーブルサイズは大きいほどデータ検索の効率が向上しますが、それだけメモリの使用量が大きくなります。

また、約数の少ない数を指定した方が、検索効率が向上します（なるべく素数または約数の少ない奇数を指定します。偶数の指定は避けてください）。

(3) **戻り値**

ハッシュオブジェクト ID

(4) **関連項目**

`g_context`, `g_hash_fetch`, `g_hash_store`, `g_hash_del`, `g_close_hash`

(5) **使用例**

`gsamp066.psv` 参照 (画面例は 2.2.8 の画面参照)

(6) **更新履歴**

なし

2.7.8 g_open_image

イメージオブジェクトの作成

注) Ver. 4 でマップ概念導入のため廃止した API です。

(1) 形式

`g_open_image context/D`

(2) 説明

`g_open_image` マクロは、イメージオブジェクトを作成します。グラフィックスウィンドウの背景にイメージオブジェクトを指定している時に、`g_close_gwin` ではイメージオブジェクトは消去されません。`g_close_image` で明示的に消去してください。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

(3) 戻り値

イメージオブジェクト ID

(4) 関連項目

`g_add_image`, `g_open_gwin`, `g_close_image`, `g_image_dsp`

(5) 使用例

なし

(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 Ver. 4.0

Ver. 4.0 で削除したマクロです。互換性用マクロファイル `bcomp.psv` を参考に `g_open_image_region` マクロ関連の機能に置き換えてください。

2.7.9 g_open_image_region

イメージオブジェクトの作成

(1) 形式

`g_open_image_region filename x1 y1 x2 y2 [bgcolor fgcolor opaqueflag]`

(2) 説明

`g_open_image_region` マクロは、イメージオブジェクトを作成します。作成したイメージオブジェクトは、`g_open_map` マクロで作成したラスタマップに、`g_map_element` を用いて登録して使用します。

① *filename*

イメージデータを格納しているファイル名を指定します。Windows ビットマップ (***.bmp)、Tiff (***.tiff または ***.tif) が指定できます。

サポートする Tiff の圧縮フォーマットを以下に示します。

- (a) LZW(カラー)
- (b) Packbits(カラー)
- (c) Group 3 Fax Encoding(白黒画像)
- (d) Huffman RLE(白黒画像)

② *x1 y1 x2 y2*

イメージデータの設定範囲を指定します。左下座標 (*x1*, *y1*)、右上座標 (*x2*, *y2*) で指定します。

③ *bgcolor*

イメージデータの背景色を指定します。イメージデータが白黒 2 値データの時のみ有効です。背景色は以下の 2 通りのパターンで指定可能です。

- (a) RGB 値 (例: R=255, G=0, B=10 → &ff000a)
- (b) カラー名称(例: red)

④ *fgcolor*

イメージデータの前面色を指定します。イメージデータが白黒 2 値データの時のみ有効です。前面色は以下の 2 通りのパターンで指定可能です。

- (a) RGB 値 (例: R=255, G=0, B=10 → &ff000a)
- (b) カラー名称(例: red)

⑤ *opaqueflag*

イメージデータの透明・不透明を指定します。透明を指定した場合、イメージデータ内の白色ビットが透明となります。

true: 不透明

false: 透明

(3) 戻り値

イメージオブジェクトの ID

(4) **関連項目**

`g_close_image_region`、`g_map_element`

(5) **使用例**

gsamp099.psv参照 (画面例は2. 2. 10の画面参照)

(6) **更新履歴**

① Ver. 4. 0

Ver. 4. 0 で新規に追加したマクロです。

2.7.10 g_open_map

マップオブジェクトのオープン

(1) 形式

```
g_open_map [-a attVal -r minVal maxVal] v|r
```

(2) 説明

マップオブジェクトをオープンします。

① -a *attVal*

マップオブジェクトの属性を数字で指定します (g_map_attrib 参照)。省略した場合の既定値は 1 となります。

② -r *minVal* *maxVal*

マップオブジェクトの表示範囲を指定します (g_map_range 参照)。

③ v|r

マップオブジェクトのタイプを指定します。

v : ベクタマップ (構成要素はディスプレイリストメモリとなります)

r : ラスタマップ (構成要素はイメージリージョンとなります)

(3) 戻り値

マップオブジェクト ID

(4) 関連項目

g_close_map, g_map_element, g_map_type, g_map_attrib, g_map_range,
g_map_margin, g_map_load

(5) 使用例

gsamp102.psv参照 (画面例は2.2.12の画面参照)

gsamp099.psv参照 (画面例は2.2.10の画面参照)

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.7.11 g_open_layer

レイヤオブジェクトの作成

(1) **形式**

`g_open_layer contextID`

(2) **説明**

`g_open_layer` マクロは、レイヤオブジェクトを作成します。

コンテキストオブジェクトにカレントのレイヤオブジェクトが設定されていない場合は、自動的にカレントに設定されます。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

(3) **戻り値**

レイヤオブジェクトの ID

(4) **関連項目**

`g_context`, `g_cur_layer`, `g_layer_no`, `g_layer_attrib`, `g_layer_color_id`,
`g_close_layer`

(5) **使用例**

gsamp002.psv参照 (画面例は2.1.3の画面参照)

(6) **更新履歴**

なし

2.7.12 g_ortho

グラフィックスウィンドウの表示座標設定

(1) 形式

```
g_ortho context/D [-w] [-s|-r] [x1 y1 x2 y2]
```

(2) 説明

g_ortho マクロは、グラフィックスウィンドウの位置、サイズを変更／取得します。範囲を指定しない場合は、現在の表示範囲が返ります。指定範囲と現在のビューポート座標の縦横比が一致しない場合、同一縦横比で指定範囲が入りきるように、指定範囲の値が補正されます。

① context/D

g_context で作成したコンテキスト ID を指定します。

② -w gwin/D

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、g_open_gwin で作成したグラフィックスウィンドウ ID を指定します。

③ -s

現在のグラフィックウィンドウの範囲が指定した WC 座標の範囲に表示するように、グラフィックスウィンドウの範囲を設定します。

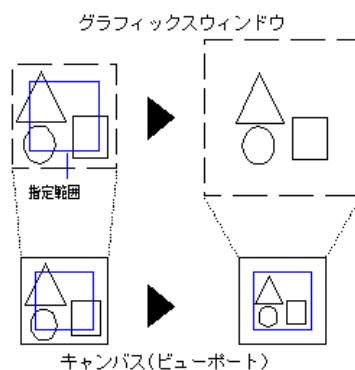


図 2-55 指定座標範囲とウィンドウの範囲

④ -r

ビューポートの縦横比との補正を行いません。

⑤ x1 y1 x2 y2

-s なしの時、グラフィックスウィンドウの左下座標 (x1, y1)、右上座標 (x2, y2) を指定します。

(3) 戻り値

グラフィックスウィンドウの新たな左下座標 (x1, y1) と右上座標 (x2, y2)

(4) 関連項目

g_context, g_open_gwin, g_viewport

(5) 使用例

gsamp068.psv 参照 (以下画面例)

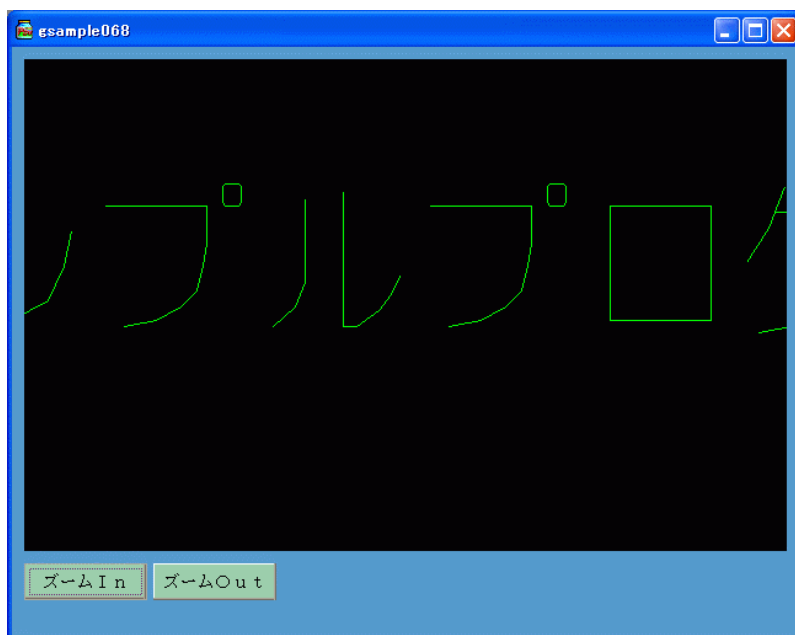


図 2-56 gsamp068.psv 実行画面

(6) 更新履歴

なし

2.8 P~R で始まるコマンド

2.8.1 g_paint_pattern_file

塗りつぶしビットマップ定義ファイルの設定

(1) 形式

`g_paint_pattern_file [fileName]`

(2) 説明

`g_paint_pattern_file` マクロは、塗りつぶしビットマップ定義ファイルを指定／取得します。

① *fileName*

塗りつぶしビットマップ定義ファイルを指定します。引数にファイル名を指定しない時は塗りつぶしビットマップ定義ファイル名が返ります。塗りつぶしビットマップ定義ファイル名は一度指定すると、変更はできません。

塗りつぶしビットマップ定義ファイルのデータフォーマットは、ビットマップシンボル定義ファイルと同じです。

mkbsym.exe コマンドを利用して定義ファイルを生成してください。

注) 0~7 はデフォルトの塗りつぶしタイプを優先するため無効です。

指定するビットマップは 8 x 8 のサイズで作成してください。8 x 8 のサイズ以外のビットマップでは、塗りつぶしできません。

(3) 戻り値

塗りつぶしビットマップ定義ファイルの登録 ID

(4) 関連項目

`g_draw_attrib`, `g_cur_draw_attrib`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`,
`g_pie`, `g_ellipse`, `g_round_box`, `g_regpoly`

(5) 使用例

gsamp098.psv 参照 (画面例は 2.2.19 の画面参照)

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.8.2 g_pie

扇型オブジェクトの作成

(1) 形式

```
g_pie contextID [-dc] cx cy radius sdeg edeg [[-t] [-c] [-a] [-g] [-l] [-m]]  
[-u]
```

(2) 説明

g_pie マクロは、扇型オブジェクトを作成します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② -dc

座標を DC 座標で指定する時に指定します。

③ cx cy radius sdeg edeg

扇型の中心座標 (cx, cy)、半径 (radius)、開始角 (sdeg)、終了角 (edeg) を指定します。角度は反時計回り、1/10 度単位で指定します。

④ -t tagID

図形オブジェクトのタグ ID を数字で指定します。

⑤ -c colorID

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値が使用されます。

⑥ -a attribVal (Ver. 4 以降削除 A 版互換モードの場合)

図形オブジェクトの属性を数字で指定します (g_attrib 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑦ -a objAttribVal drawAttribVal

図形オブジェクトの属性値と描画属性値を数字で指定します (g_obj_attrib、g_draw_attrib 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ -g [groupID] groupNo

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (groupID) 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グループオブジェクトが対象となります。

⑨ -l [layerID] layerNo

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID (layerID) 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ヤオブジェクトが対象となります。

⑩ -m mode (Ver. 4 以降削除 A 版互換モード時有効)

図形オブジェクトのモードを数字で指定します (g_mode 参照)。

⑪ `-u key body [key body]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

扇型の図形オブジェクト ID

(4) 関連項目

`g_arc`, `g_attrib`, `g_context`, `g_draw_obj`, `g_mode`, `g_open_group`, `g_open_layer`,
`g_obj_attrib`, `g_draw_attrib`

(5) 使用例

gsamp069.psv 参照 (以下画面例)



図 2-57 gsamp069.psv 実行画面

(6) 更新履歴

① Ver 4.0

属性値のオプションの指定を変更しました (-a)。

モード指定のオプションを削除しました (-m)。

2.8.3 g_polygon

多角形オブジェクトの作成

(1) 形式

```
g_polygon contextID num [-dc] x1 y1 x2 y2... [[-t] [-c] [-a] [-g] [-l] [-m]]  
[-u]
```

(2) 説明

g_polygon マクロは、多角形オブジェクトを作成します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② num

多角形の座標の点数を指定します。

③ -dc

座標を DC 座標で指定する時に指定します。

④ x1 y1 x2 y2...

多角形の頂点の座標を指定します。点 (x1, y1)、点 (x2, y2) ... の順に指定します。

⑤ -t tagID

図形オブジェクトのタグ ID を数字で指定します。

⑥ -c colorID

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑦ -a attribVal (Ver. 4 以降削除 A 版互換モードの場合)

図形オブジェクトの属性を数字で指定します (g_attrb 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ -a objAttribVal drawAttribVal

図形オブジェクトの属性値と描画属性値を数字で指定します (g_obj_attrb、g_draw_attrb 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑨ -g [groupID] groupNo

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (groupID) 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グループオブジェクトが対象となります。

⑩ -l [layerID] layerNo

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID (layerID) 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ヤオブジェクトが対象となります。

- ⑪ `-m mode` (Ver. 4 以降削除 A 版互換モード時有効)
図形オブジェクトのモードを数字で指定します (g_mode 参照)。
- ⑫ `-u key body [key body]`
図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。
- (3) 戻り値
多角形の図形オブジェクト ID
- (4) 関連項目
g_attrb, g_context, g_draw_obj, g_mode, g_open_group, g_open_layer,
g_obj_attrb, g_draw_attrb
- (5) 使用例
gsamp070.psv 参照 (以下画面例)

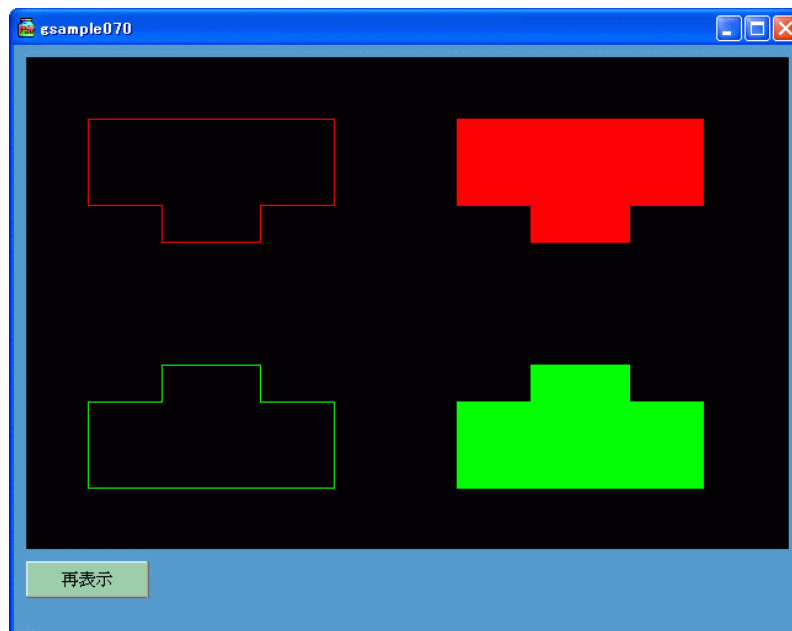


図 2-58 gsamp070.psv 実行画面

- (6) 更新履歴
 - ① Ver. 4.0
属性値のオプションの指定を変更しました (-a)。
モード指定のオプションを削除しました (-m)。

2.8.4 g_polyline

折れ線（曲線）オブジェクトの作成

(1) 形式

```
g_polyline contextID num [-dc] x1 y1 x2 y2... [[-t] [-c] [-a] [-g] [-l] [-m]]  
[-u]
```

(2) 説明

g_polyline マクロは、折れ線（曲線）オブジェクトを作成します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② num

折れ線の座標の点数を指定します。

③ -dc

座標を DC 座標で指定する時に指定します。

④ x1 y1 x2 y2...

折れ線の座標を指定します。点 (x1, y1)、点 (x2, y2) ... の順に指定します。

⑤ -t tagID

図形オブジェクトのタグ ID を数字で指定します。

⑥ -c colorID

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑦ -a attribVal (Ver. 4 以降削除 A 版互換モードの場合)

図形オブジェクトの属性を数字で指定します (g_attr 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ -a objAttribVal drawAttribVal

図形オブジェクトの属性値と描画属性値を数字で指定します (g_obj_attr、g_draw_attr 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑨ -g [groupID] groupNo

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (groupID) 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グループオブジェクトが対象となります。

⑩ -l [layerID] layerNo

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID (layerID) 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ヤオブジェクトが対象となります。

⑪ `-m mode` (Ver. 4 以降削除 A 版互換モード時有効)

図形オブジェクトのモードを数字で指定します (g_mode 参照)。

⑫ `-u key body [key body]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

折れ線 (曲線) の図形オブジェクト ID

(4) 関連項目

g_attrb, g_context, g_draw_obj, g_mode, g_open_group, g_open_layer,
g_obj_attrb, g_draw_attrb

(5) 使用例

gsamp071.psv 参照 (以下画面例)

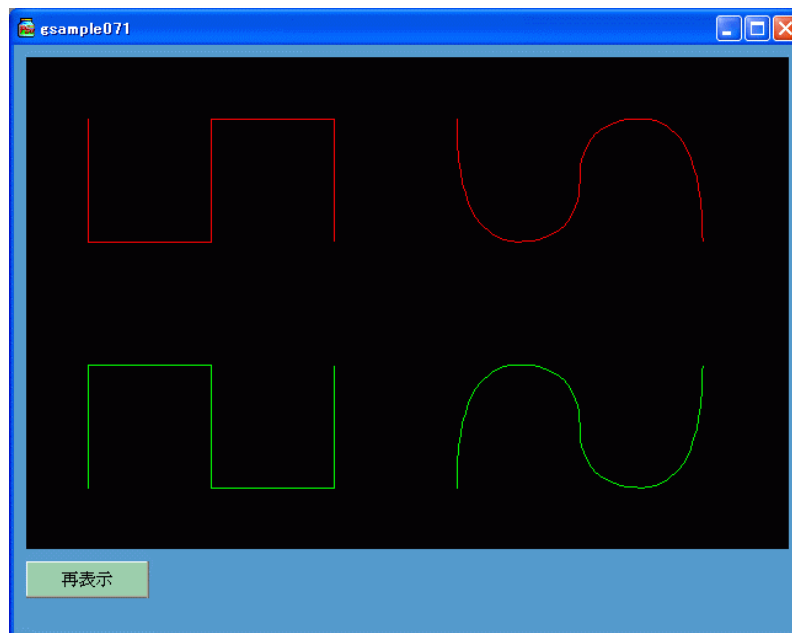


図 2-59 gsamp071.psv 実行画面

(6) 更新履歴

① Ver. 4.0

属性値のオプションの指定を変更しました (-a)

モード指定のオプションを削除しました (-m)

2.8.5 g_raster

ラスタオペレーションの操作

(1) **形式**

`g_raster context/D [-w] mode`

(2) **説明**

`g_raster` マクロは、グラフィックスウィンドウのラスタオペレーションの操作を行います。

① *context/D*

`g_context` で作成したコンテキスト ID を指定します。

② *-w gwin/D*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *mode*

ラスタオペレーションの内容を指定します。

`copy` : 描画色をそのまま反映します。

`xor` : 画面の色との排他的論理和で描画します。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_gwin`

(5) **使用例**

`gsamp063.psv`参照 (画面例は2. 2. 5の画面参照)

(6) **更新履歴**

なし

2.8.6 g_redraw

キャンバスの自動描画モードの設定

(1) 形式

```
g_redraw context/D canvas/D [-r]
g_redraw context/D canvas/D [-i] -p [-w [x|a] gwin/D] [-d [x|a] dlist/D
[dlist/D ...]] [-p ...]
g_redraw context/D canvas/D -m map/D [map/D ...]
```

(2) 説明

`g_redraw` マクロは、キャンバスの自動描画モードの設定を行いません。ウィンドウシステムからキャンバスへの再描画要求時に、図形オブジェクトを指定のディスプレイリストメモリから検索して指定のグラフィックスウィンドウへ描画を行いません。

① *context/D*

`g_context` で作成したコンテキスト ID を指定します。

② *canvas/D*

`w_canvas` で作成したウィンドウ ID を指定します。

③ `-r`

指定のキャンバスで自動描画モードとなっているグラフィックスウィンドウに対し、まとめてモードの解除を行います。

④ `-i`

既にキャンバスが自動描画モードとなっていた時の処置を指定します。

`-i` を省略すると、今回指定したグラフィックスウィンドウが自動描画モードとなります。

`-i` を指定すると、既存のグラフィックスウィンドウ全てを解除して、今回指定したグラフィックスウィンドウが新規に自動描画モードとなります。

⑤ `-w [x|a] gwin/D`

描画を行なうグラフィックスウィンドウを指定する場合に、`g_open_gwin` で成したグラフィックスウィンドウ ID を指定します。

`a` を指定すると、グラフィックスウィンドウの自動描画モードを設定します(既定値)。

`x` を指定すると、グラフィックスウィンドウの自動描画モードを解除します。

指定を省略するとコンテキストオブジェクトに設定している、カレントのグラフィックスウィンドウ ID が処理対象となります。

⑥ `-d [x|a] dlistID ...`

コンテキストオブジェクトに設定している、ディスプレイリストメモリ以外を指定する場合に、`g_open_dlist` で作成したディスプレイリストメモリ ID を指定します。

`a` で指定した `dlistID` は、自動描画モードになります（既定値）。

`x` で指定した `dlistID` は、自動描画モードが解除します。

指定を省略するとコンテキストオブジェクトに設定しているカレントのディスプレイリストメモリ ID が処理対象となります。複数のディスプレイリストメモリ ID を指定する場合、`a`、`x` は ID ごとに指定します。

⑦ `-m mapID [mapID ...]`

`g_open_map` で作成したマップオブジェクト ID を指定します。指定したマップオブジェクトの属性の自動ロードがオンとなっている場合には、再描画時に必要に応じて地図データを読み込みます。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_open_dlist`, `g_open_map`, `w_canvas`

(5) 使用例

`gsamp068.psv` 参照 (画面例は 2. 7. 12 の画面参照)

(6) 更新履歴

① Ver. 4. 0

マップオブジェクトを利用しての再描画が可能となりました (`-m` オプション)。

2.8.7 g_region_obj

図形オブジェクトの描画範囲取得

(1) **形式**

`g_region_obj [-s] objID [objID...]`

(2) **説明**

`g_region_obj` マクロは、図形オブジェクトの描画範囲を囲む矩形の左下点、中心点と右上点を取得します。

① **-s**

指定した複数の図形オブジェクトの描画範囲を囲む矩形の左下点、中心点と右上点を取得します。

単一図形オブジェクトの場合、指定は不要です。

② ***objID***

描画範囲を取得する図形オブジェクトの ID を指定します。

(3) **戻り値**

左下点、中心点と右上点（リスト形式）

(4) **関連項目**

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_round_box`, `g_regpoly`

(5) 使用例

gsamp090.psv 参照 (以下画面例)

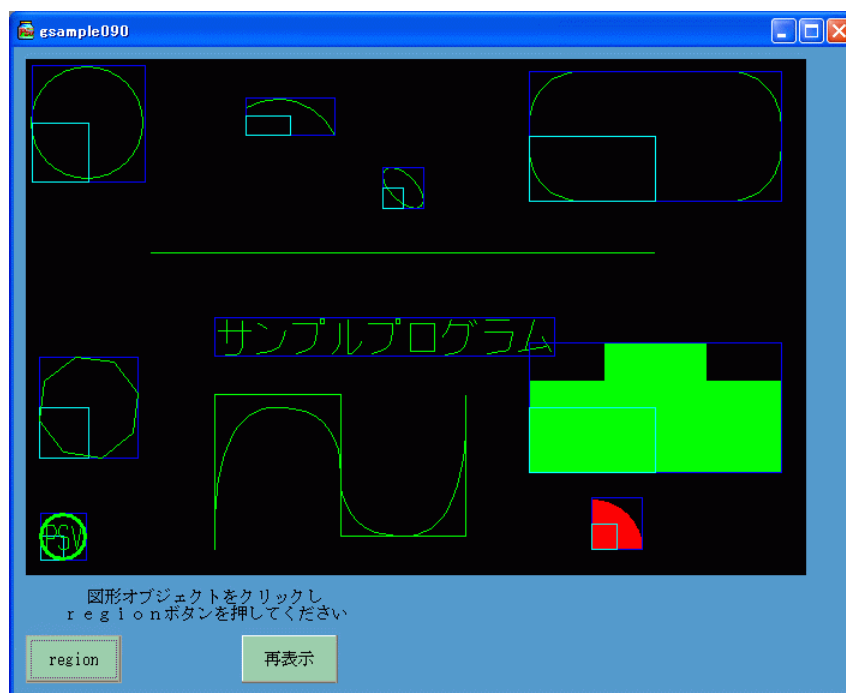


図 2-60 gsamp090.psv 実行画面

(6) 更新履歴

なし

2.8.8 g_regpoly

正多角形オブジェクトの作成

(1) 形式

```
g_regpoly contextID [-dc] cx cy sx sy num [[-t] [-c] [-a] [-g] [-l] [-m]]  
[-u]
```

(2) 説明

g_regpoly マクロは、正多角形オブジェクトを作成します。

① contextID

g_context で作成したコンテキスト ID を指定します。

② -dc

座標を DC 座標で指定する時に指定します。

③ cx cy sx sy

正多角形の中心座標 (cx, cy)、最初の頂点の開始座標 (sx, sy) を指定します。

④ num

角数を指定します。

⑤ -t tagID

図形オブジェクトのタグ ID を数字で指定します。

⑥ -c colorID

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑦ -a attribVal (Ver. 4 以降削除 A 版互換モードの場合)

図形オブジェクトの属性を数字で指定します (g_attr 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ -a objAttribVal drawAttribVal

図形オブジェクトの属性値と描画属性値を数字で指定します (g_obj_attr、g_draw_attr 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑨ -g [groupID] groupNo

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (groupID) 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グループオブジェクトが対象となります。

⑩ -l [layerID] layerNo

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID (layerID) 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ヤオブジェクトが対象となります。

- ⑪ `-m mode` (Ver. 4 以降削除 A 版互換モード時有効)
図形オブジェクトのモードを数字で指定します (g_mode 参照)。
- ⑫ `-u key body [key body]`
図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。
- (3) 戻り値
正多角形の図形オブジェクト ID
- (4) 関連項目
g_attrb, g_context, g_draw_obj, g_mode, g_open_group, g_open_layer, g_obj_attrb, g_draw_attrb
- (5) 使用例
gsamp074.psv 参照 (以下画面例)

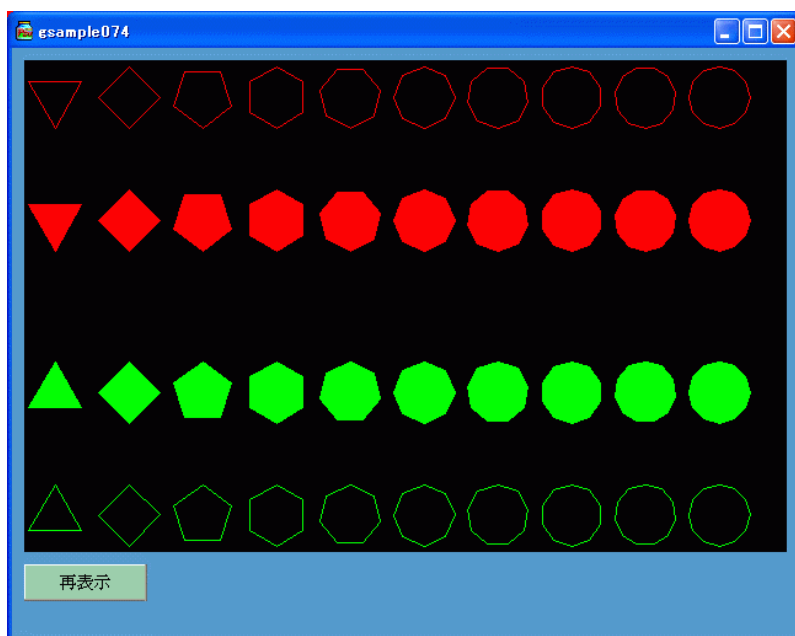


図 2-61 gsamp074.psv 実行画面

- (6) 更新履歴
 - ① Ver. 4.0
属性値のオプションの指定が変更しました (-a)
モード指定のオプションを削除しました (-m)

2.8.9 g_rm_obj

ディスプレイリストメモリからの図形オブジェクト削除

(1) **形式**

`g_rm_obj context/D [-d] obj/D [obj/D...]`

(2) **説明**

`g_rm_obj` マクロは、`g_ins_obj` マクロで格納した図形オブジェクトを、ディスプレイリストメモリから解放します。図形オブジェクトが、他のディスプレイリストに格納されていない場合は、自動的に削除されます。

① *context/D*

`g_context` で作成したコンテキスト ID を指定します。

② *-d dlist/D*

コンテキストオブジェクトに設定しているディスプレイリストメモリ以外のディスプレイリストメモリを指定する場合に、ディスプレイリストメモリ ID を指定します。`g_open_dlist` で作成したディスプレイリストメモリ ID を指定します。

③ *obj/D*

解放する図形オブジェクトの図形オブジェクト ID を指定します。

(3) **戻り値**

消された図形オブジェクトの ID (リスト形式)

(4) **関連項目**

`g_context`, `g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`, `g_symbol`, `g_usr_def`, `g_ins_obj`, `g_del_obj`, `g_round_box`, `g_regpoly`

(5) 使用例

gsamp075.psv 参照 (以下画面例)

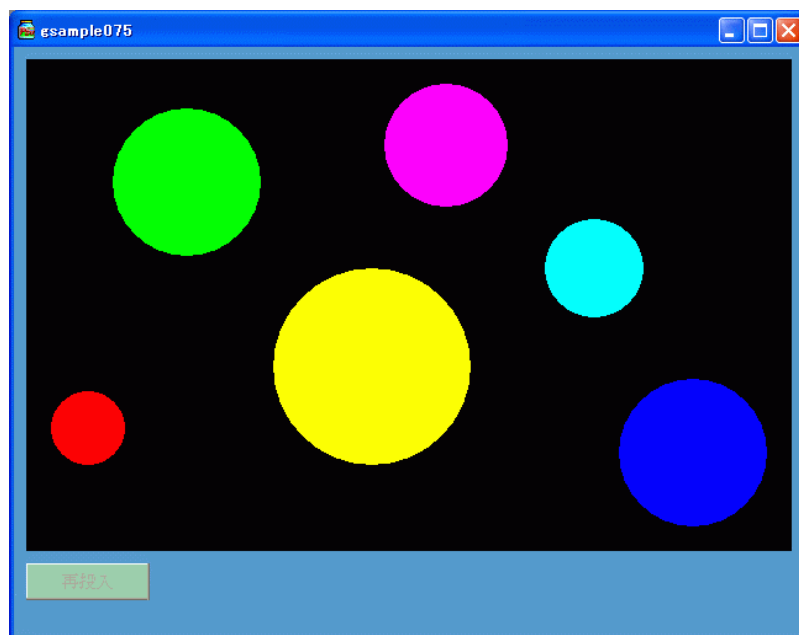


図 2-62 gsamp075.psv 実行画面

(6) 更新履歴

なし

2.8.10 g_rotate

グラフィックスウィンドウの回転

(1) 形式

`g_rotate context/D [-w] [-dc] cx cy deg`

(2) 説明

`g_rotate` マクロは、グラフィックスウィンドウの回転を行います。指定角度で、キャンバスの表示内容が反時計回りで回転します。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwin/D`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-dc`

中心座標 (`cx`, `cy`) を DC 座標で指定します。本引数を省略した時は、WC 座標で指定します。

④ `cx cy deg`

グラフィックスウィンドウの回転の中心座標 (`cx`, `cy`) と回転角度 (`deg`) を指定します。回転角度は反時計回り、1/10 度単位で指定します。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`

(5) 使用例

gsamp076.psv 参照 (以下画面例)

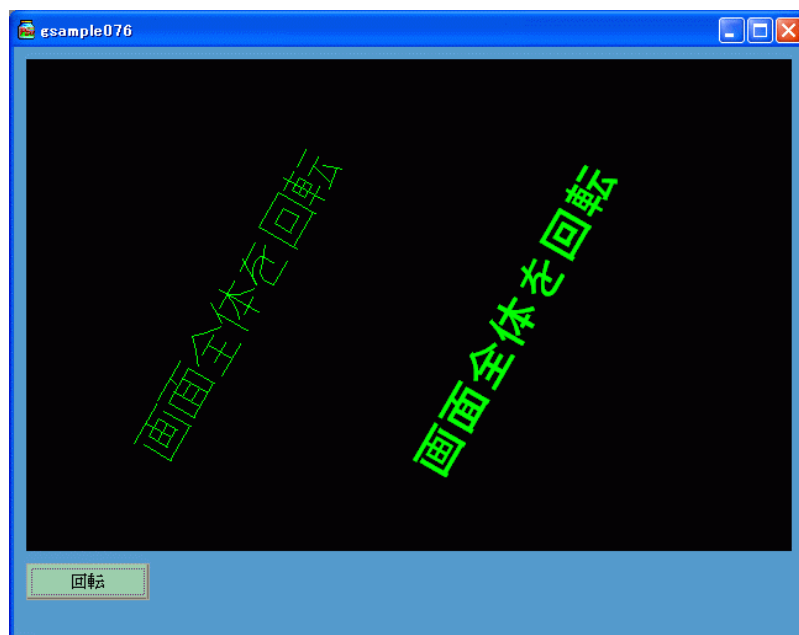


図 2-63 gsamp076.psv 実行画面

(6) 更新履歴

なし

2.8.11 g_rotate_obj

図形オブジェクトの回転移動

(1) 形式

`g_rotate_obj context/D [-d] [-dc] rx ry deg obj/D [obj/D...]`

(2) 説明

`g_rotate_obj` マクロは、図形オブジェクトの回転移動を行います。指定する図形は、指定したディスプレイリストメモリに格納されているか、まったくディスプレイリストメモリに格納されていない状態のみ有効です。複数のディスプレイリストメモリに格納されている場合は無効となります。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-d dlist/D`

コンテキストオブジェクトに設定しているディスプレイリストメモリ以外のディスプレイリストメモリを指定する場合に、ディスプレイリストメモリ ID を指定します。`g_open_dlist` で作成したディスプレイリストメモリ ID を指定します。

③ `-dc`

座標を DC 座標で指定する時に指定します。

④ `rx ry deg`

回転の中心座標 (rx , ry) と回転角度 (deg) を指定します。回転角度は反時計回り、1/10 度単位で指定します。

⑤ `obj/D [obj/D...]`

作成した図形オブジェクトの ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_translate_obj`, `g_scale_obj`, `g_transform_obj`,
`g_round_box`, `g_regpoly`

(5) 使用例

gsamp077.psv 参照 (以下画面例)

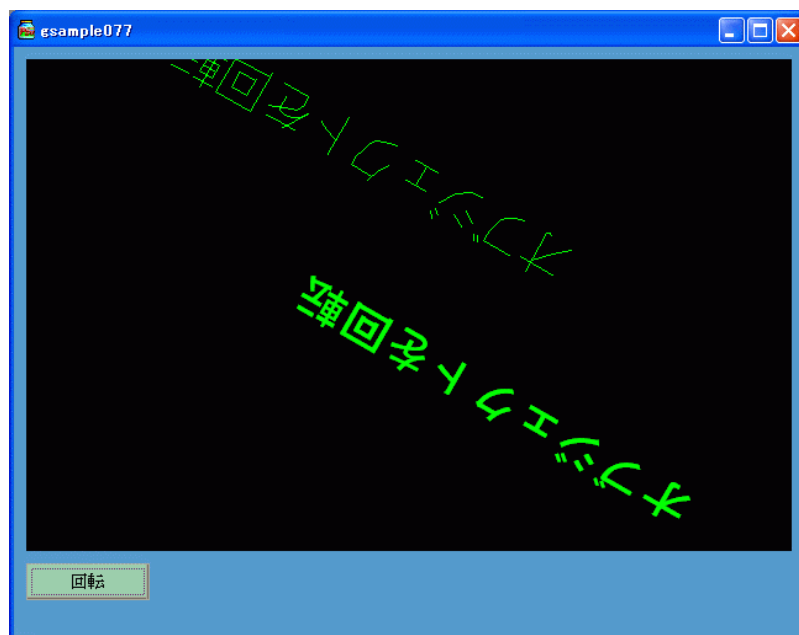


図 2-64 gsamp077.psv 実行画面

(6) 更新履歴

なし

2.8.12 g_round_box

角丸長方形オブジェクトの作成

(1) 形式

```
g_round_box contextID [-dc] x1 y1 x2 y2 rad degree [[-t] [-c] [-a] [-g] [-l]
[-m]] [-u]
```

(2) 説明

`g_round_box` マクロは、角丸長方形オブジェクトを作成します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-dc*

座標を DC 座標で指定する時に指定します。

③ *x1 y1 x2 y2*

角丸長方形の座標を指定します。左下点 (*x1*, *y1*) と右上点 (*x2*, *y2*) または、左上点 (*x1*, *y1*) と右下点 (*x2*, *y2*) を指定します。

④ *rad*

角の円弧の半径を指定します。

⑤ *degree*

角丸長方形の回転角度を 1/10 度単位で指定します。

⑥ *-t tagID*

図形オブジェクトのタグ ID を数字で指定します。

⑦ *-c colorID*

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ *-a attribVal* (Ver. 4 以降削除 A 版互換モードの場合)

図形オブジェクトの属性を数字で指定します (`g_attr` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑨ *-a objAttribVal drawAttribVal*

図形オブジェクトの属性値と描画属性値を数字で指定します (`g_obj_attr`、`g_draw_attr` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑩ *-g [groupID] groupNo*

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (*groupID*) 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グループオブジェクトが対象となります。

⑪ -l [*layerID*] *layerNo*

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID (*layerID*) 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ヤオブジェクトが対象となります。

⑫ -m *mode* (Ver. 4 以降削除 A 版互換モード時有効)

図形オブジェクトのモードを数字で指定します (g_mode 参照)。

⑬ -u *key body* [*key body*]

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

角丸長方形の図形オブジェクト ID

(4) 関連項目

g_attrb, g_context, g_draw_obj, g_mode, g_open_group, g_open_layer, g_obj_attrb, g_draw_attrb

(5) 使用例

gsamp078.psv 参照 (以下画面例)

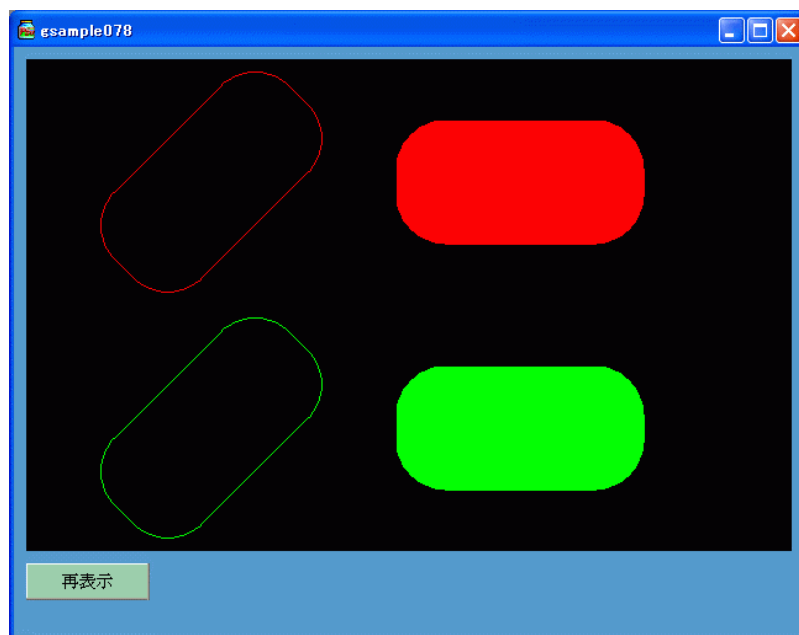


図 2-65 gsamp078.psv 実行画面

(6) 更新履歴

① Ver. 4.0

属性値のオプションの指定が変更しました (-a)

モード指定のオプションが削除しました (-m)

2.8.13 g_RRline

鉄道線オブジェクトの作成

(1) 形式

```
g_RRline contextID sno lwidth num [-dc] x1 y1 x2 y2... [[-t] [-c] [-a] [-g]
[-l] [-m]] [-u]
```

(2) 説明

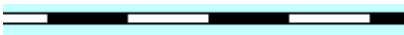
g_RRline マクロは、鉄道線オブジェクトを作成します。鉄道線オブジェクトは鉄道線の特殊な描画を行う、WC サイズの幅(lwidth)を持った特殊描画線です。鉄道線作成時に幅を考慮した描画領域を作成し、検索時の判定を正しい描画領域のポリゴンとして行うため、幅の広い線を描画しても正確に描画できます。

① contextID

g_context で作成したコンテキスト ID を指定します。

② sno

鉄道線の線種を指定します。線種と描画イメージの対応を以下に示します。

0 : 

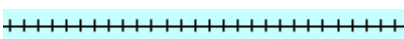
一般的な JR 線描画です。黒部分の色を線色で指定した色で描画します。白い部分の色は固定です。

1 : 

主に私鉄線の描画に使用できます。縁の黒部分を線色で指定した色で描画します。白い部分の色は固定です。

2 : 

線種 1 の白い部分を前景色で描画する線種です。道路地図レベルの表示で、国道・県道・高速道路を描画する時にも使用できます。

3 : 

私鉄線などを交差線で表現する時の描画です。線色で指定した色で描画します。突起部分の長さは、lwidth/2 になります。

③ lwidth

線を描画する幅を WC サイズで指定します。線種 0 ~ 2 の内側の線はこの WC サイズの 75% のサイズで描画します。

④ num

鉄道線の座標の点数を指定します。

⑤ -dc

座標を DC 座標で指定する時に指定します。

⑥ x1 y1 x2 y2...

鉄道線の座標を指定します。点 (x1, y1)、点 (x2, y2) ... の順に指定します。

⑦ `-t tagID`

図形オブジェクトのタグ ID を数字で指定します。

⑧ `-c colorID`

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑨ `-a objAttribVal drawAttribVal`

図形オブジェクトの属性値と描画属性値を数字で指定します (`g_obj_attrib`、`g_draw_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑩ `-g [groupID] groupNo`

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (`groupID`) 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グループオブジェクトが対象になります。

⑪ `-l [layerID] layerNo`

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID (`layerID`) 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ヤオブジェクトが対象になります。

⑫ `-m`

本オプションは現在未使用です。

⑬ `-u key body [key body]`

図形オブジェクトに格納するユーザ定義データのキー (`key`) とデータ (`body`) を指定します。

(3) **戻り値**

鉄道線の図形オブジェクト ID

(4) **関連項目**

`g_attrib`, `g_context`, `g_draw_obj`, `g_mode`, `g_open_group`, `g_open_layer`,
`g_obj_attrib`, `g_draw_attrib`

(5) 使用例

gsamp107.psv 参照 (以下画面例)

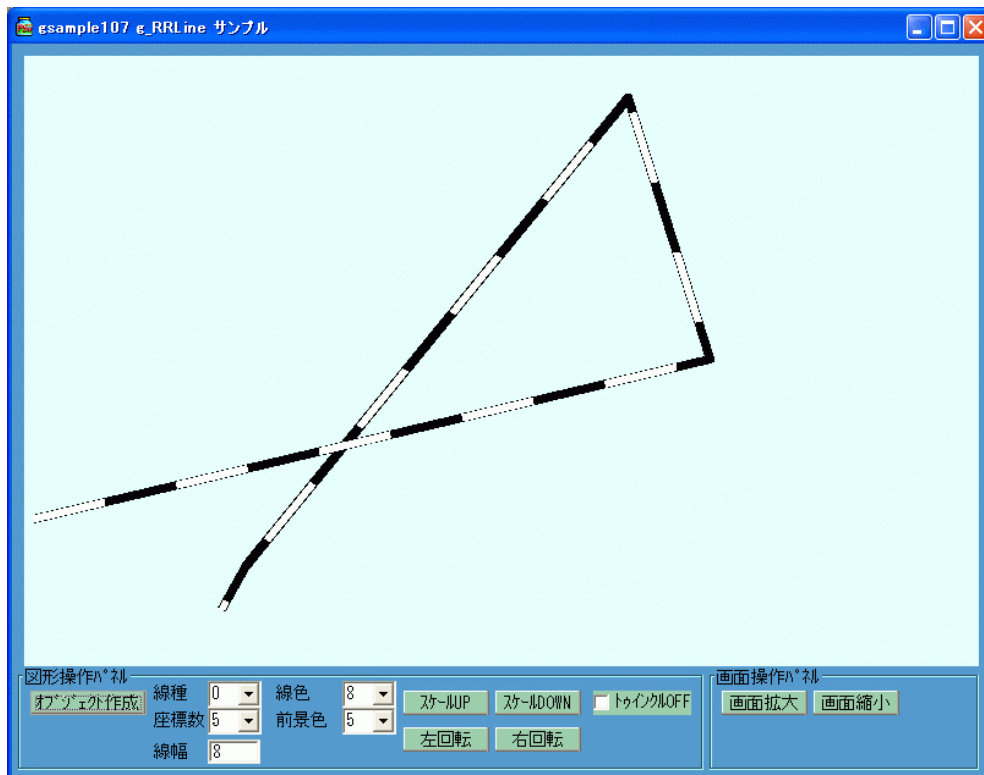


図 2-66 gsamp107.psv 実行画面

(6) 更新履歴

① Ver. 5.2

鉄道線オブジェクト作成のマクロを追加しました。

2.9 S で始まるコマンド

2.9.1 g_scale

グラフィックスウィンド表示縮尺値の取得／変更

(1) 形式

`g_scale contextID [-w] [scaleValue]`

(2) 説明

`g_scale` マクロは、グラフィックスウィンドウの表示縮尺値の取得／設定を行います。縮尺値は、縮尺値の分母を整数で指定します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwinID`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *scaleValue*

縮尺値の分母を整数で指定します。

(3) 戻り値

縮尺値の分母

(4) 関連項目

`g_context`, `g_open_gwin`, `g_ortho`, `g_layer_range`, `g_map_range`

(5) 使用例

gsamp103.psv 参照 (以下画面例)

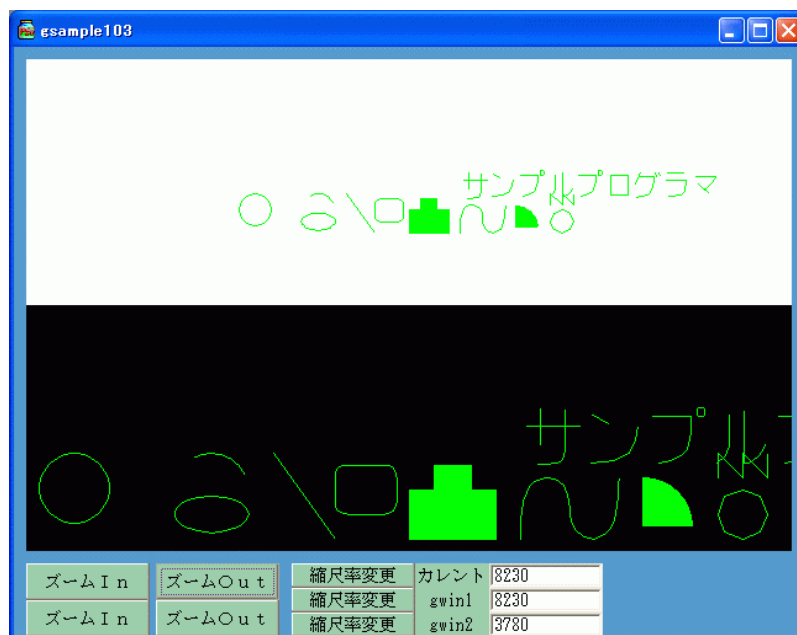


図 2-67 gsamp103.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。

2.9.2 g_scale_obj

図形オブジェクトの拡大／縮小

(1) 形式

`g_scale_obj context/D [-dc] rx ry sx sy obj/D [obj/D...]`

(2) 説明

`g_scale_obj` マクロは、図形オブジェクトの拡大／縮小を行います。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-dc`

座標を DC 座標で指定する時に指定します。

③ `rx ry sx sy`

拡大／縮小の中心座標 (rx , ry)、X 軸方向のスケール (sx)、Y 軸方向のスケール (sy) を指定します。

④ `obj/D [obj/D...]`

作成した図形オブジェクトの ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_translate_obj`, `g_transform_obj`, `g_rotate_obj`,
`g_round_box`, `g_regpoly`

(5) 使用例

gsamp079.psv 参照 (以下画面例)

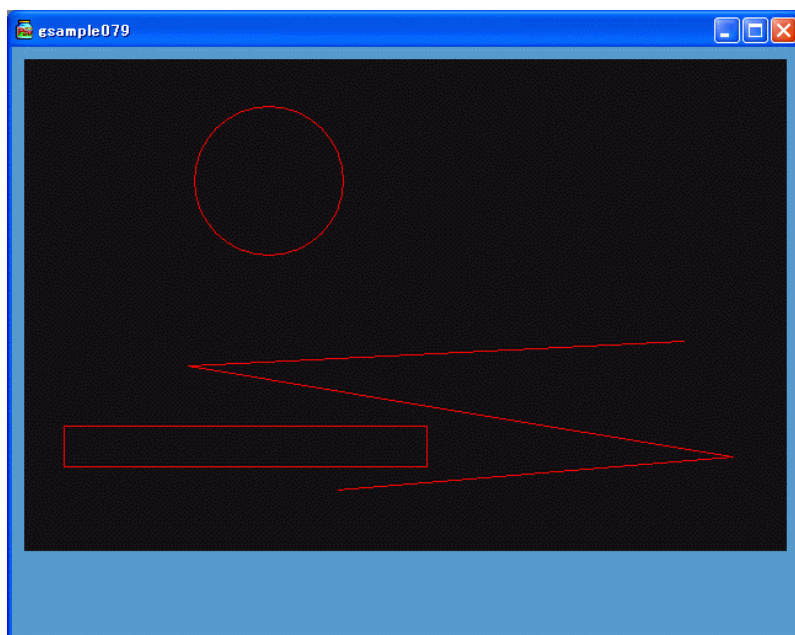


図 2-68 gsamp079.psv 実行画面

(6) 更新履歴

なし

2.9.3 g_scroll

グラフィックスウィンドウのスクロール

(1) 形式

`g_scroll context/D [-w] [-d dlist/D [dlist/D ...]] [-dc] dx dy`

(2) 説明

`g_scroll` マクロは、グラフィックスウィンドウのスクロールを行います。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwin/D`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `-d dlist/D ...`

コンテキストオブジェクトに設定している以外のディスプレイリストメモリを指定する場合に、`g_open_dlist` で作成したディスプレイリストメモリ ID を指定します。

④ `-dc`

スクロール移動量 (dx , dy) を DC 座標で指定します。本引数を省略した時は、WC 座標で指定します。

⑤ `dx dy`

グラフィックスウィンドウのスクロール移動量を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`, `g_open_dlist`

(5) 使用例

gsamp080.psv 参照 (以下画面例)

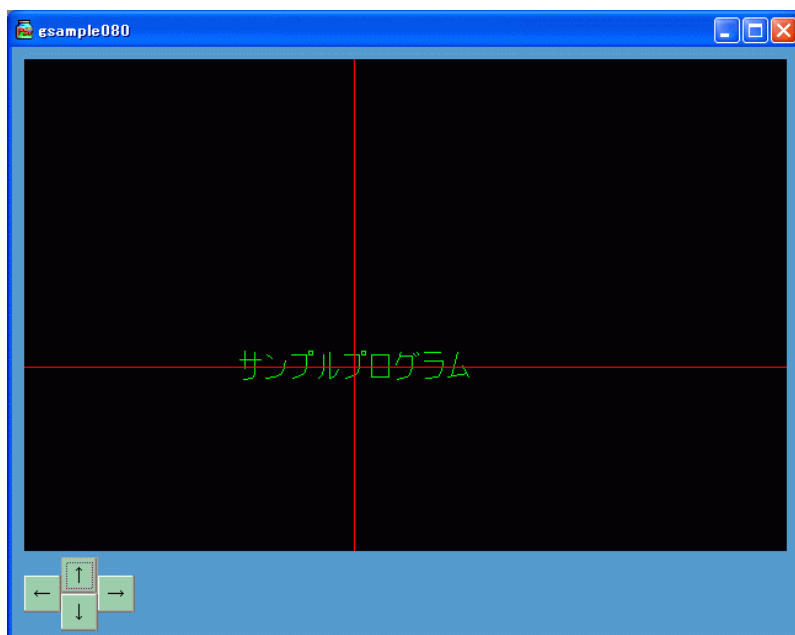


図 2-69 gsamp080.psv 実行画面

(6) 更新履歴

なし

2.9.4 g_search_obj

図形オブジェクトの検索

(1) 形式

```
g_search_obj context/D -g
g_search_obj context/D [-d dlist/D [dlist/D ...]] -l -n
g_search_obj context/D [-d dlist/D [dlist/D ...]] -l -r
g_search_obj context/D [-d dlist/D [dlist/D ...]] -l -p
```

(2) 説明

`g_search_obj` マクロは、ディスプレイリストメモリまたはグループオブジェクトから図形オブジェクトを検索します。該当件数が複数の場合、リスト形式でオブジェクトの ID を返します。

なお、図形オブジェクトを検索対象とするかどうかは、次の属性値での検知性の設定に従います。これらのすべてが「検知する」に設定されている場合だけ、検索対象になります。

- 図形オブジェクトに設定されている属性
- 図形オブジェクトに設定されているグループ番号に設定されている属性
- 図形オブジェクトに設定されているレイヤ番号に設定されている属性

ただし、例外として `-g` 指定による検索では、検知性はチェックされず、グループの全図形オブジェクトが検索されます。

① context/D

`g_context` で作成したコンテキスト ID を指定します。

② -g [group/D] groupNo...

グループ番号を指定して図形オブジェクトを検索します。グループオブジェクト ID (`group/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グループオブジェクトが対象となります。

③ -d dlist/D ...

`-n`、`-l`、`-r`、`-p` 指定の時、検索するディスプレイリストメモリを指定します。省略した場合は、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となります。

④ `-l [d/v/n] [layerID] 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 (*layerID*) 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ヤオブジェクトが対象となります。

`-n` または `-r` または `-p` と同時に指定することにより、レイヤと指定範囲が検索対象となります。

`d` を指定すると、マウスの検知性が可の図形オブジェクトが対象となります。

`v` を指定すると、可視性が可の図形オブジェクトが対象となります。

`n` を指定すると、すべての図形オブジェクトが対象となります。

省略するとマウスの検知性と可視性が可の図形オブジェクトが対象となります。

⑤ `-n [-dc] x y`

指定した座標 (*x*, *y*) にもっとも近い図形オブジェクトを検索します。`-l` と同時に指定することにより、レイヤと指定範囲が検索対象となります。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標で指定します。

⑥ `-r [d/v/n] x1 y1 x2 y2`

矩形の座標範囲を指定して、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。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標で指定します。

`-l` と同時に指定することにより、レイヤと指定範囲が検索対象となります。

`d` を指定すると、マウスの検知性が可の図形オブジェクトが対象となります。

`v` を指定すると、可視性が可の図形オブジェクトが対象となります。

`n` を指定すると、すべての図形オブジェクトが対象となります。

省略するとマウスの検知性と可視性が可の図形オブジェクトが対象となります。

⑦ `-p [d/v/n] num x1 y1 x2 y2`

多角形の座標範囲を指定して、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。検索範囲とする、多角形の頂点の座標を順に指定します。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標で指定します。

`-l` と同時に指定することにより、レイヤと指定範囲が検索対象となります。

`d` を指定すると、マウスの検知性が可の図形オブジェクトが対象となります。

`v` を指定すると、可視性が可の図形オブジェクトが対象となります。

`n` を指定すると、すべての図形オブジェクトが対象となります。

省略するとマウスの検知性と可視性が可の図形オブジェクトが対象となります。

(3) 戻り値

図形オブジェクト ID (リスト形式)

`-n` の場合は、もっとも近い図形オブジェクトの ID となります。

(4) 関連項目

`g_context`, `g_make_group`, `g_group_attrib`, `g_layer_no`, `g_layer_attrib`, `g_line`,
`g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`, `g_symbol`,
`g_usr_def`, `g_round_box`, `g_regpoly`, `g_seqsearch_obj`, `g_seqsearch_next`

(5) 使用例

`gsamp003.psv`参照 (画面例は2. 1. 4の画面参照)

(6) 更新履歴

① Ver. 3. 0

検索オプションの同時指定が可能になりました。

2.9.5 g_seqsearch_obj

図形オブジェクト順次検索の開始

(1) 形式

```
g_seqsearch_obj contextID [-d dlistID [dlistID ...]] -l|-r|-p  
g_seqsearch_obj contextID [-d dlistID [dlistID ...]] -l -r  
g_seqsearch_obj contextID [-d dlistID [dlistID ...]] -l -p
```

(2) 説明

g_seqsearch_obj マクロは、ディスプレイリストメモリまたはグループオブジェクト から図形オブジェクトを検索します。該当件数が複数の場合、リスト形式でオブジェクトの ID を返します。

なお、図形オブジェクトを検索対象とするかどうかは、次の属性値での検知性の設定に従います。これらのすべてが「検知する」に設定されている場合だけ、検索対象になります。

- 図形オブジェクトに設定されている属性
- 図形オブジェクトに設定されているグループ番号に設定されている属性
- 図形オブジェクトに設定されているレイヤ番号に設定されている属性

① contextID

g_context で作成したコンテキスト ID を指定します。

② -d dlistID ...

-l、-r、-p 指定の時、検索するディスプレイリストメモリを指定します。省略した場合は、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となります。

③ -l [d/v/n] [layerID] 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 (layerID) 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ヤオブジェクトが対象となります。

-r または -p と同時に指定することにより、レイヤと指定範囲が検索対象となります。

d を指定すると、マウスの検知性が可の図形オブジェクトが対象となります。

v を指定すると、可視性が可の図形オブジェクトが対象となります。

n を指定すると、すべての図形オブジェクトが対象となります。

省略するとマウスの検知性と可視性が可の図形オブジェクトが対象となります。

④ `-r [d/v/n] x1 y1 x2 y2`

矩形の座標範囲を指定して、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。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標で指定します。

`d` を指定すると、マウスの検知性が可の図形オブジェクトが対象となります。

`v` を指定すると、可視性が可の図形オブジェクトが対象となります。

`n` を指定すると、すべての図形オブジェクトが対象となります。

省略するとマウスの検知性と可視性が可の図形オブジェクトが対象となります。

⑤ `-p [d/v/n] num x1 y1 x2 y2`

多角形の座標範囲を指定して、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。検索範囲とする、多角形の頂点の座標を順に指定します。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標で指定します。

`d` を指定すると、マウスの検知性が可の図形オブジェクトが対象となります。

`v` を指定すると、可視性が可の図形オブジェクトが対象となります。

`n` を指定すると、すべての図形オブジェクトが対象となります。

省略するとマウスの検知性と可視性が可の図形オブジェクトが対象となります。

(3) 戻り値

一番最初に検索された図形オブジェクト ID

(4) 関連項目

`g_context`, `g_make_group`, `g_group_attrib`, `g_layer_no`, `g_layer_attrib`,
`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_round_box`, `g_regpoly`, `g_search_obj`, `g_seqsearch_next`

(5) 使用例

`gsamp095.psv` 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。

2.9.6 g_seqsearch_next

図形オブジェクト順次検索

(1) **形式**

`g_seqsearch_next objID`

(2) **説明**

`g_seqsearch_next` マクロは、図形オブジェクト順次検索の開始マクロで検索した結果より指定された図形オブジェクト ID の後に検索された図形オブジェクト ID を返します。

`g_seqsearch_next` マクロを呼び出す前に、必ず図形オブジェクト順次検索の開始マクロを実行してください。

① *objID*

図形オブジェクト順次検索の開始マクロまたは、図形オブジェクト順次検索マクロで取得した図形オブジェクト ID を指定します。

(3) **戻り値**

指定された図形オブジェクト ID の後に検索された図形オブジェクト ID

(4) **関連項目**

`g_search_obj`, `g_seqsearch_obj`

(5) **使用例**

`gsamp095.psv` 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。

2.9.7 g_set_clipboard

図形オブジェクトのクリップボードへのコピー

(1) 形式

```
g_set_clipboard contextID [-c] [-w] objID [objID...]
```

```
g_set_clipboard contextID [-c] [-w] [-d dlistID [dlistID...]] [-g|-l|-r|-p]
```

(2) 説明

g_set_clipboard マクロは、図形オブジェクトをクリップボードに描画します。図形オブジェクト ID を個々に指定する方法と、グループ番号やレイヤ番号、座標範囲を指定して一括描画する方法があります。

図形オブジェクトを描画するかは、次の属性値での可視性の設定にしています (g_attrib 参照)。これらのすべてが「可視」に設定されている場合だけ、描画されます。

- 図形オブジェクトに設定されている属性
- 図形オブジェクトに設定されているグループ番号に設定されている属性
- 図形オブジェクトに設定されているレイヤ番号に設定されている属性

① contextID

g_context で作成したコンテキスト ID を指定します。

② -c colorID

色を指定する場合に、g_alloc_color で作成した色 ID を指定します。

省略するか、-1 を指定すると、図形オブジェクトの属性で指定した優先順位にしたがって、次のいずれかの色で描画します (g_attrib 参照)。

- 図形オブジェクトに設定されている色
- 図形オブジェクトに設定されているグループ番号に設定されている色
- 図形オブジェクトに設定されているレイヤ番号に設定されている色

③ -w gwinID

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウの倍率で描画する場合に、g_open_gwin で作成したグラフィックスウィンドウ ID を指定します。

④ objID [objID...]

図形オブジェクトを個別に指定する場合に、図形オブジェクト ID を指定します。

⑤ -d dlistID ...

-l、-r、-p 指定の時、検索するディスプレイリストメモリを指定します。

省略すると、コンテキストオブジェクトに設定しているディスプレイリストメモリが対象となります。

⑥ `-g [groupID] groupNo ...`

グループ番号 (*groupNo*) を指定して図形オブジェクトを描画します。グループオブジェクト ID (*groupID*) の指定がない場合は、コンテキストオブジェクトに設定しているグループオブジェクトが対象となります。

⑦ `-l [layerID] layerNo`

レイヤ番号 (*layerNo*) を指定して、ディスプレイリストメモリ内の図形オブジェクトを描画します。レイヤオブジェクト ID (*layerID*) 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ヤオブジェクトが対象となります。

⑧ `-r x1 y1 x2 y2`

矩形の座標範囲を指定して、ディスプレイリストメモリ内の図形オブジェクトを描画します。範囲に部分的にかかる図形オブジェクトも対象となります。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標で指定します。

⑨ `-p num x1 y1 x2 y2`

多角形の座標範囲を指定して、ディスプレイリストメモリ内の図形オブジェクトを描画します。多角形の頂点の座標を (*x1, y1*)、(*x2, y2*) ... の順に指定します。範囲に部分的にかかる図形オブジェクトも対象となります。座標は、`-dc` 指定時は DC 座標、`-dc` 省略時は WC 座標で指定します。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_make_group`, `g_group_attrib`, `g_group_color_id`, `g_layer_no`,
`g_layer_attrib`, `g_layer_color_id`, `g_alloc_color`, `g_line`, `g_polyline`,
`g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`, `g_symbol`, `g_usr_def`,
`g_round_box`, `g_regpoly`

(5) 使用例

gsamp093.psv 参照 (以下画面例)

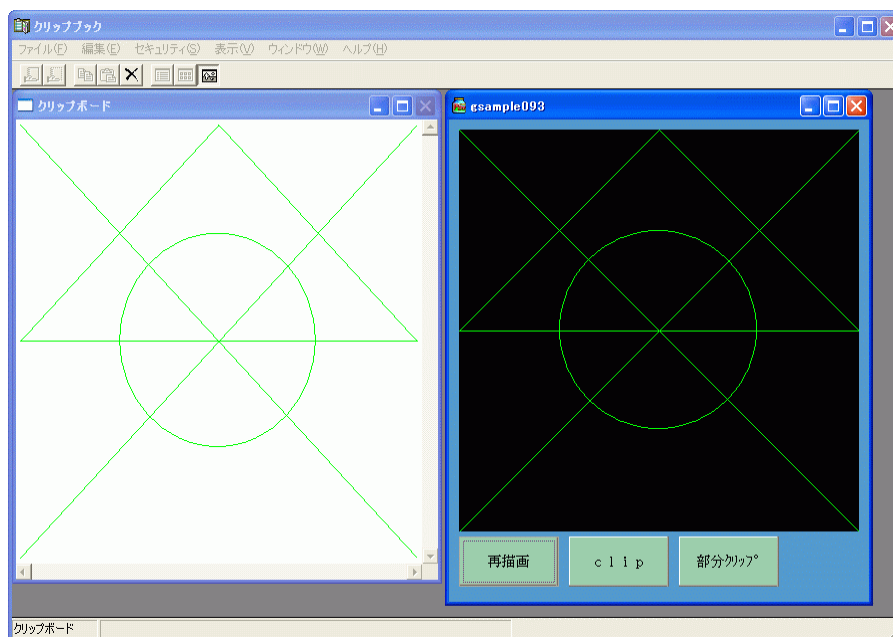


図 2-70 gsamp093.psv 実行画面

(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。

2.9.8 g_set_usrdata

ユーザ定義データの設定

(1) 形式

`g_set_usrdata objID [-r] key body [key body....]`

(2) 説明

`g_set_usrdata` マクロは、図形オブジェクトにユーザ定義データを格納します。

① *objID*

作成した図形オブジェクトの ID を指定します。

② *-r*

指定したキー (*key*) のデータを、指定データ (*body*) に変更します。*-r* 指定時に、指定したキーが存在しない時は追加になります。

-r を指定しない時、指定したキーが存在した場合は指定データ (*body*) には変更されません。

③ *key body [key body....]*

ユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_get_usrdata`, `g_round_box`, `g_regpoly`

(5) 使用例

`gsamp045.psv`参照 (画面例は2.4.6の画面参照)

(6) 更新履歴

① Ver. 3.0

ユーザ定義データを変更するオプションが追加になりました(*-r*)。

2.9.9 g_showpage

印刷の終了

(1) 形式

`g_showpage contextID flag [copies]`

(2) 説明

`g_showpage` マクロは、印刷用グラフィックスウィンドウの現在のページに対する出力を終了します。`g_open_gwin` で `-p` または `-h` オプション付きで作成したグラフィックスウィンドウに対して有効です。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *flag*

次ページの印刷を行うかを指定します。

0 : 現在のページで終了する。

1 : 次ページの印刷を行う。

③ *copies*

現在のページを何枚印刷するかを数字で指定します。省略すると 1 枚になります。

(3) 戻り値

なし

(4) 関連項目

`g_context`, `g_open_gwin`

(5) 使用例

`psamp001.psv` 参照 (画面例は 2.7.6 の画面参照)

(6) 更新履歴

なし

2.9.10 g_sort

図形オブジェクトの距離順ソート

(1) **形式**

`g_sort contextID [-dc] xVal/ yVal objID objID [objID...]`

(2) **説明**

`g_sort` マクロは、指定した点から近い順に図形オブジェクトをソートします。指定した複数の図形オブジェクト ID が、ソートされて戻り値に設定されます。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-dc`

座標を DC 座標で指定する時に指定します。

③ `xVal/ yVal`

座標を指定します。

④ `objID objID [objID ...]`

ソートする図形オブジェクトの ID を指定します。

(3) **戻り値**

指定点に近い順にソートされた図形オブジェクト ID (リスト形式)

(4) **関連項目**

`g_context`, `g_dist`

(5) **使用例**

`gsamp084.psv`参照 (画面例は2.3.4の画面参照)

(6) **更新履歴**

なし

2.9.11 g_symbol

シンボルオブジェクトの作成

(1) 形式

```
g_symbol contextID [-dc] cx cy symNo deg scale [[-t] [-c] [-a] [-g] [-l] [-m]]  
[-u]
```

(2) 説明

`g_symbol` マクロはシンボルオブジェクトを作成します。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `-dc`

座標を DC 座標で指定する時に指定します。

③ `cx cy`

シンボルの中心座標 (`cx`, `cy`) を指定します。

④ `symNo`

`g_symbol_file` で指定したシンボルファイル中のシンボル番号を数字で指定します。

⑤ `deg scale`

角度 (`deg`)、倍率 (`scale`) を指定します。角度は反時計回り、1/10 度単位で指定します。

⑥ `-t tagID`

図形オブジェクトのタグ ID を数字で指定します。

⑦ `-c colorID`

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します (カラーシンボル指定時は無効となります)。

⑧ `-a attribVal` (Ver. 4 以降削除 A 版互換モードの場合)

図形オブジェクトの属性を数字で指定します (`g_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑨ `-a objAttribVal drawAttribVal`

図形オブジェクトの属性値と描画属性値を数字で指定します (`g_obj_attrib`、`g_draw_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑩ `-g [groupID] groupNo`

図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID (`groupID`) 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グループオブジェクトが対象となります。

⑪ `-l [layerID] layerNo`

図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID (*layerID*) 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ヤオブジェクトが対象となります。

⑫ `-m mode` (Ver. 4 以降削除 A 版互換モード時有効)

図形オブジェクトのモードを数字で指定します (g_mode 参照)。

⑬ `-u key body [key body ...]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

シンボルの図形オブジェクト ID

(4) 関連項目

g_attrb, g_context, g_draw_obj, g_mode, g_symbol_file, g_open_group, g_open_layer, g_obj_attrb, g_draw_attrb

(5) 使用例

gsamp085.psv 参照 (以下画面例)

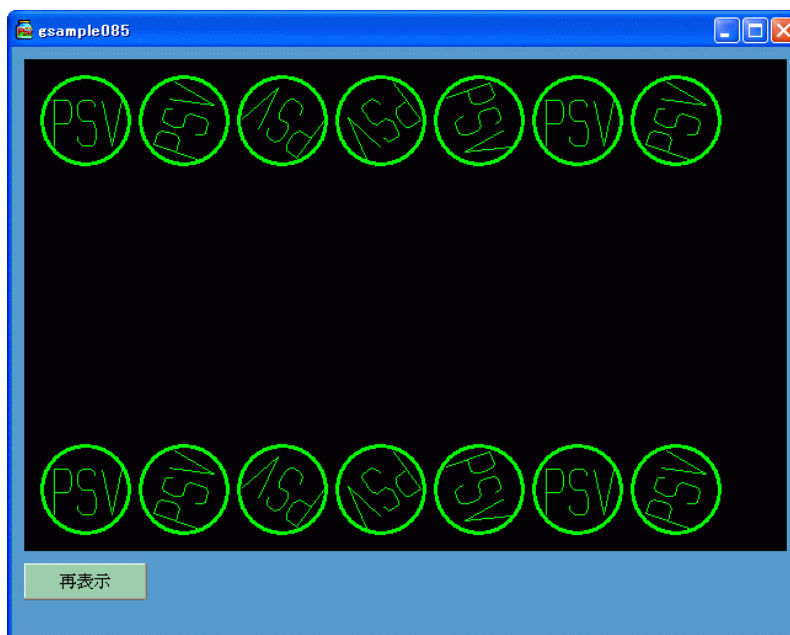


図 2-71 gsamp085.psv 実行画面

(6) 更新履歴

① Ver. 4.0

属性値のオプションの指定を変更しました (-a)

モード指定のオプションを削除しました (-m)

2.9.12 g_symbol_file

シンボル定義ファイルの設定

(1) **形式**

`g_symbol_file [fileName] | [-s FileID]`

(2) **説明**

`g_symbol_file` マクロは、シンボル定義ファイルを指定／取得します。

① *fileName*

シンボル定義ファイルを指定します。引数にファイル名を指定しない時はシンボル定義ファイルの現在の登録 ID が返ります。

② `-s FileID`

既に登録済みのシンボル定義ファイルを指定する場合は、登録時の戻り値である登録 ID を指定します。

(3) **戻り値**

シンボル定義ファイルの登録 ID

(4) **関連項目**

`g_symbol`, `g_draw_symbol`

(5) **使用例**

`gsamp037.psv`参照 (画面例は2.3.17の画面参照)

(6) **更新履歴**

① Ver. 4.0

指定が変更可能となりました。

2.10 T~Y で始まるコマンド

2.10.1 g_tag_id

図形オブジェクトのタグ ID の取得／変更

(1) 形式

`g_tag_id objID [tagID]`

(2) 説明

`g_tag_id` マクロは、図形オブジェクトのタグ ID を取得／変更します。引数にタグ ID を指定しない時は、現在のタグ ID が取得されます。指定する場合はタグ ID が変更され、変更後のタグ ID が返ります。

① *objID*

作成した図形オブジェクトの ID を指定します。

② *tagID*

変更するタグ ID を数字で指定します。

(3) 戻り値

タグ ID

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_round_box`, `g_regpoly`

(5) 使用例

`gsamp087.psv` 参照 (以下画面例)



図 2-72 `gsamp087.psv` 実行画面

(6) 更新履歴

なし

2.10.2 g_TBL2JBL

Tokyo97 (日本測地系) の緯度・経度座標を JGD2000 の緯度・経度に変換

(1) **形式**

`g_TBL2JBL contextID TB TL`

(2) **説明**

`g_TBL2JBL` マクロは旧日本測地系 (Tokyo97) の緯度経度座標から新日本測地系 (JGD2000) の緯度経度に変換します。本マクロは既存の緯度経度座標データを新測地系の緯度経度座標データに変換する時に使用してください。

変換後の緯度経度はリスト形式で出力されます。

① `contextID`

`g_context` で作成したコンテキスト ID を指定します。

② `TB`

旧日本測地系の緯度をラディアン値で指定します。指定する変数の精度は単数精度以上の精度で指定しないと、変換誤差の原因となります。

③ `TL`

旧日本測地系の経度をラディアン値で指定します。指定する変数の精度は単数精度以上の精度で指定しないと、変換誤差の原因となります。

(3) **戻り値**

新日本測地系の緯度・経度をラディアン値の値で、“緯度 経度”の並びのリスト構造で返します。

(4) **関連項目**

`g_TBL2TXY`, `g_JBL2JXY`

(5) **使用例**

`gsamp108.psv`参照 (画面例は2.5.6の画面参照)

(6) **更新履歴**

① Ver. 5.2

Ver. 5.2 で新規に追加したマクロです。

2.10.3 g_TBL2TXY

Tokyo97 の緯度・経度を平面直交座標系に変換

(1) **形式**

`g_TBL2TXY contextID TB TL`

(2) **説明**

`g_TBL2TXY` マクロは旧日本測地系(Tokyo97)の緯度経度座標を、ベッセル楕円体パラメータを使用して 19 系座標に変換します。変換後の XY 座標はリスト形式で出力されます。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *TB*

旧日本測地系の緯度をラディアン値で指定します。指定する変数の精度は単数精度以上の精度で指定しないと、変換誤差の原因となります。

③ *TL*

旧日本測地系の経度をラディアン値で指定します。指定する変数の精度は単数精度以上の精度で指定しないと、変換誤差の原因となります。

(3) **戻り値**

19 系座標の XY 座標を“X 座標 Y 座標”の並びのリスト構造で返します。数値の単位は“m”で、精度は倍数精度で返します。

(4) **関連項目**

`g_TBL2JBL`, `g_JBL2JXY`

(5) **使用例**

`gsamp108.psv`参照 (画面例は2.5.6の画面参照)

(6) **更新履歴**

① Ver. 5.2

Ver. 5.2 で新規に追加したマクロです。

2.10.4 g_TKY2JGD_ParaFile

Tokyo97 の緯度・経度を JGD2000 の緯度・経度に変換するパラメータファイルを指定

(1) **形式**

`g_TKY2JGD_ParaFile contextID ParaFileName`

(2) **説明**

国土地理院 TKY2JGD 準拠の座標変換 API を高精度計算で実行するため、座標変換パラメータファイルを読み込みます。本 API の実行は初回だけです。

本 API の未実行時と、パラメータファイルに含まれない地域を g_TBL2JBL で座標変換した場合は、低精度で変換を実行するため、住宅地図などの精度が必要な用途には適しません。

① *contextID*

g_context で作成したコンテキスト ID を指定します。

② *ParaFileName*

2007/03/30 現在、国土地理院様の HP からダウンロード可能な TKY2JGD S/W のページにある、座標変換パラメータファイルを指定してください。

※ 座標変換パラメータファイルは適宜更新しているので、製品には付属しません。

最新の座標変換パラメータファイル入手して適用してください。

日本全体だと容量が大きいため、適用するシステムの地域だけを使用することをお奨めします。

(3) **戻り値**

なし

(4) **関連項目**

g_TBL2JBL

(5) **使用例**

gsamp108.psv参照 (画面例は2.5.6の画面参照)

(6) **更新履歴**

① Ver 5.2

Ver 5.2 で新規に追加したマクロです。

2.10.5 g_text

文字列オブジェクトの作成

(1) 形式

```
g_text contextID [-dc] x y num string [[-t] [-c] [-a] [-h] [-w] [-o] [-r]
[-s] [-g] [-l] [-m] [-i] [-f]] [-u]
```

(2) 説明

`g_text` マクロは文字列オブジェクトを作成します。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-dc*

座標を DC 座標で指定する時に指定します。

③ *x y*

文字列の座標を指定します。横書きの時は文字列の左下、縦書きの時は文字列の左上になります。

④ *num*

文字数を指定します。

⑤ *string*

文字列を指定します。文字列はシフト JIS と EUC の 2 バイト文字だけ有効です。シフト JIS の時には、半角カタカナの指定が可能ですが、文字列先頭に半角カタカナがあると EUC として処理して、文字化けの原因となります。半角カタカナは文字列先頭以外に指定してください。

横書きの場合、複数行文字列が指定できます。改行文字は“`\n`”です。“`\n`”は文字数にはカウントしません。

※ 半角文字は、全角文字の半分の幅で描画します。

⑥ *-t tagID*

図形オブジェクトのタグ ID を数字で指定します。

⑦ *-c colorID*

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

⑧ *-a attribVal* (Ver. 4 以降削除 A 版互換モードの場合)

図形オブジェクトの属性を数字で指定します (`g_attr` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

- ⑨ `-a objAttribVal drawAttribVal`
図形オブジェクトの属性値と描画属性値を数字で指定します (`g_obj_attrib`、`g_draw_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。
- ⑩ `-h [hightVal]`
文字高さを数字で指定します (WC 座標単位)。
- ⑪ `-w [widthVal]`
文字幅を数字で指定します (WC 座標単位)。
- ⑫ `-o [h|v]`
文字列方向を指定します。
h : 横書き
v : 縦書き
- ⑬ `-r [deg]`
文字列の角度を反時計回、1/10 度単位で指定します。
- ⑭ `-s [spaceVal]`
文字間隔を数字で指定します (WC 座標単位)。
- ⑮ `-g [groupID] groupNo`
図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID(*groupID*)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グループオブジェクトが対象となります。
- ⑯ `-l [layerID] layerNo`
図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID(*layerID*)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ヤオブジェクトが対象となります。
- ⑰ `-m mode` (Ver. 4 以降削除 A 版互換モード時有効)
図形オブジェクトのモードを数字で指定します (`g_mode` 参照)。
- ⑱ `-i [pitchVal]`
行間を数字で指定します (WC 座標単位)。
- ⑲ `-f [fontType]`
フォント種別を数字で指定します。
0: ベクトルフォント
1: ゴシック (TrueType フォント)
2: 明朝 (TrueType フォント)
※ PreSerV ではどのフォント種別でも文字の高さ、幅が一定のサイズの固定ピッチテキストとして描画します。

⑳ `-u key body [key body]`

図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。

(3) 戻り値

文字列の図形オブジェクト ID

(4) 関連項目

`g_attrb`, `g_context`, `g_draw_obj`, `g_mode`, `g_open_group`, `g_open_layer`,
`g_obj_attrb`, `g_draw_attrb`

(5) 使用例

`gsamp088.psv` 参照 (以下画面例)

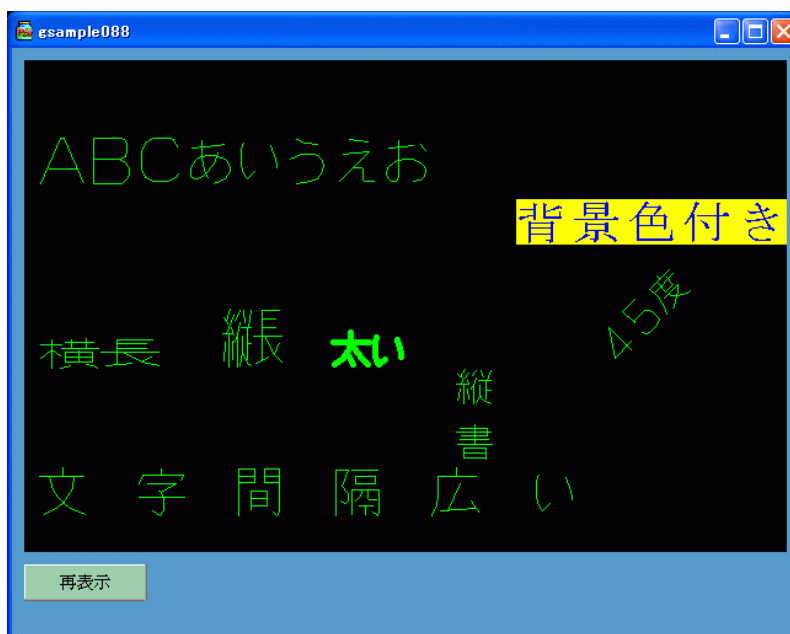


図 2-73 `gsamp088.psv` 実行画面

(6) 更新履歴

① Ver. 3.0

複数行文字 (シフト JIS の場合) の指定が可能になりました (`-i` オプションで行間の指定も可能です。)。

② Ver. 4.0

フォントタイプ指定用のオプションを追加しました (`-f`)。

属性値のオプションの指定を変更しました (`-a`)。

モード指定のオプションを削除しました (`-m`)。

半角記号の指定が可能になりました。

2.10.6 g_transform_obj

図形オブジェクトの変形

(1) 形式

```
g_transform_obj context/D [-d] -r|-a|-x index [-dc] x1 y1 obj/D [obj/D ...]
g_transform_obj context/D [-d] [-dc] num x1 y1 x2 y2 ... obj/D [ obj/D ...]
```

(2) 説明

g_transform_obj マクロは、図形オブジェクトの変形を行います。指定する図形は、指定したディスプレイリストメモリに格納されているか、まったくディスプレイリストメモリに格納されていない状態のみ有効です。

対象図形オブジェクトを複数のディスプレイリストメモリに格納している場合は、変形は無効とします。

① context/D

g_context で作成したコンテキスト ID を指定します。

② -d

コンテキストオブジェクトに設定しているディスプレイリストメモリ以外のディスプレイリストメモリを指定する場合に、ディスプレイリストメモリ ID を指定します。g_open_dlist で作成したディスプレイリストメモリ ID を指定します。

③ -r|-a|-x index

図形オブジェクトの座標の一部分を変更する場合に指定します。

-r を指定すると、index で指定した座標を変更します。

-a を指定すると、指定した座標を index の位置に追加します。

-x を指定すると、index で指定した座標を削除します。

index は 0 から始まります。

④ -dc

座標を DC 座標で指定する時に指定します。

⑤ x1 y1

変形で変更、追加する点の座標を指定します。削除時は座標値は使用しません。

⑥ obj/D [obj/D ...]

作成した図形オブジェクトの ID を指定します。

⑦ num

座標の点数を指定します。

⑧ $x1\ y1\ x2\ y2\ \dots$

変形後の座標を指定します。図形オブジェクトごとの指定可能な点数と座標は以下のとおりです。

(a) 線 (line)

点数 : 2

座標 : 始点 終点

(b) 多角形 (polygon)

点数 : 3 以上

座標 : 始点 頂点 1 頂点 2... 終点

(c) 折れ線 (polyline)

点数 : 2 以上

座標 : 始点 頂点 1 頂点 2... 終点

(d) 円 (circle)

点数 : 3

座標 : 円周上の点を反時計回りに指定

(e) 扇型 (pie)

点数 : 3

座標 : 円周上の開始点 円周上の終了点 中心点

(f) 円弧 (arc)

点数 : 3

座標 : 円周上の開始点 円周上の終了点 中心点

文字列 (text)、シンボル (symbol)、楕円 (ellipse)、正多角形 (regpoly)、角丸長方形 (round_box) は変形できません。

(3) 戻り値

なし

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_rotate_obj`,
`g_scale_obj`, `g_translate_obj`

(5) 使用例

gsamp089.psv 参照 (以下画面例)

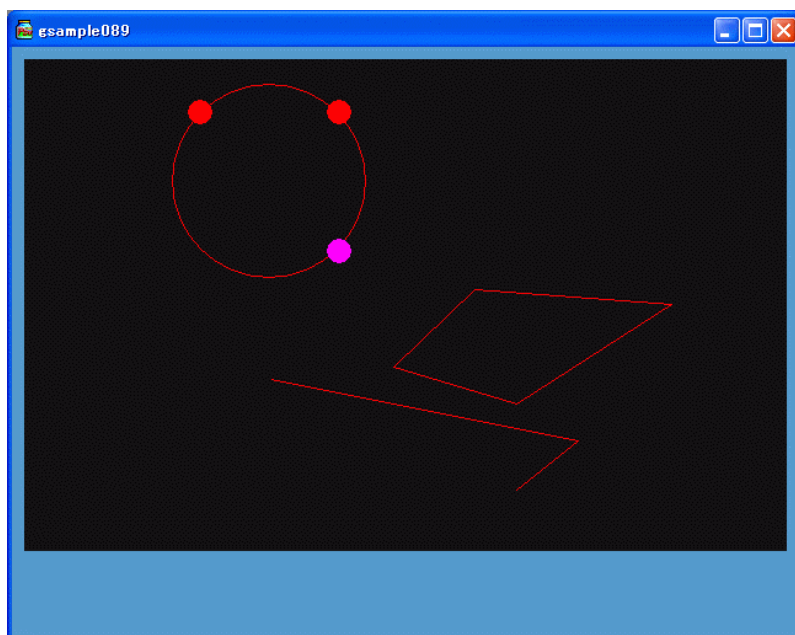


図 2-74 gsamp089.psv 実行画面

(6) 更新履歴

なし

2.10.7 g_translate_obj

図形オブジェクトの並行移動

(1) 形式

`g_translate_obj context/ID [-d] [-dc] dx dy obj/ID [obj/ID...]`

(2) 説明

`g_translate_obj` マクロは、図形オブジェクトの並行移動を行います。指定する図形は、指定したディスプレイリストメモリに格納されているか、まったくディスプレイリストメモリに格納されていない状態のみ有効です。複数のディスプレイリストメモリに格納されている場合は無効となります。

① `context/ID`

`g_context` で作成したコンテキスト ID を指定します。

② `-d`

コンテキストオブジェクトに設定しているディスプレイリストメモリ以外のディスプレイリストメモリを指定する場合に、ディスプレイリストメモリ ID を指定します。`g_open_dlist` で作成したディスプレイリストメモリ ID を指定します。

③ `-dc`

座標を DC 座標で指定する時に指定します。

④ `dx dy`

X 座標の移動量 (dx)、Y 座標の移動量 (dy) を指定します。

⑤ `obj/ID [obj/ID ...]`

作成した図形オブジェクトの ID を指定します。

(3) 戻り値

なし

(4) 関連項目

`g_line`, `g_polyline`, `g_polygon`, `g_circle`, `g_arc`, `g_pie`, `g_ellipse`, `g_text`,
`g_symbol`, `g_usr_def`, `g_rotate_obj`, `g_scale_obj`, `g_transform_obj`,
`g_round_box`, `g_regpoly`

(5) 使用例

`gsamp063.psv` 参照 (画面例は 2.2.5 の画面参照)

(6) 更新履歴

なし

2.10.8 g_usr_def

ユーザ定義オブジェクトの作成

(1) 形式

```
g_usr_def contextID [-dc] x1 y1 x2 y2 command [-t] [-c] [-a] [-g] [-l] [-m] [-u]
```

(2) 説明

g_usr_def マクロは、ユーザ定義オブジェクトを作成します。

ユーザ定義オブジェクトには、任意のマクロを設定しておくことができます。設定したマクロは、g_draw_obj でユーザ定義オブジェクトを描画する際に実行されます。例えば、図形描画用のマクロや図形オブジェクト作成用のマクロを実行できます。

① contextID

g_context で作成したコンテキスト ID を指定します。

② -dc

座標を DC 座標で指定する時に指定します。本座標は、座標範囲で図形オブジェクトを一括描画する際などに、ユーザ定義オブジェクトの位置情報として使用します。

③ x1 y1 x2 y2

ユーザ定義オブジェクトの描画範囲を囲む矩形の左下点 (x1, y1) と右上点 (x2, y2)、または左上点 (x1, y1) と右下点 (x2, y2) を指定します。

④ command

図形オブジェクトへ描画要求が来た時に実行するマクロを指定します。描画を要求したグラフィックスウインドウを、自動的に作成時のコンテキストオブジェクトのカレントグラフィックスウインドウに設定します。次の2種類の変数を使用できます。

変数 psv_usr_def(gc) は、作成時のコンテキストオブジェクトのコンテキスト ID をセットしています。

変数 psv_usr_def(self) には、ユーザ定義データの図形オブジェクト ID をセットしています。

⑤ -t tagID

図形オブジェクトのタグ ID を数字で指定します。

⑥ -c colorID

図形オブジェクトの色を色 ID で指定します。省略した場合には、コンテキストオブジェクトのカレント値を使用します。

- ⑦ `-a attribVal` (Ver. 4 以降削除 A 版互換モードの場合)
図形オブジェクトの属性を数字で指定します (`g_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。
- ⑧ `-a objAttribVal drawAttribVal`
図形オブジェクトの属性値と描画属性値を数字で指定します (`g_obj_attrib`、`g_draw_attrib` 参照)。省略した場合には、コンテキストオブジェクトのカレント値を使用します。
- ⑨ `-g groupNo`
図形オブジェクトをグループ化する時、グループ番号を数字で指定します。グループオブジェクト ID はコンテキストオブジェクトのカレント値を使用します。
- ⑩ `-l layerNo`
図形オブジェクトにレイヤを設定する時、レイヤ番号を数字で指定します。レイヤオブジェクト ID はコンテキストオブジェクトのカレント値を使用します。
- ⑪ `-m mode` (Ver. 4 以降削除 A 版互換モード時有効)
図形オブジェクトのモードを数字で指定します (`g_mode` 参照)。
- ⑫ `-u key body [key body]`
図形オブジェクトに格納するユーザ定義データのキー (*key*) とデータ (*body*) を指定します。
- (3) **戻り値**
ユーザ定義オブジェクトの図形オブジェクト ID
- (4) **関連項目**
`g_attrib`, `g_context`, `g_draw_obj`, `g_mode`, `g_open_group`, `g_open_layer`,
`g_obj_attrib`, `g_draw_attrib`

(5) 使用例

gsamp091.psv 参照 (以下画面例)

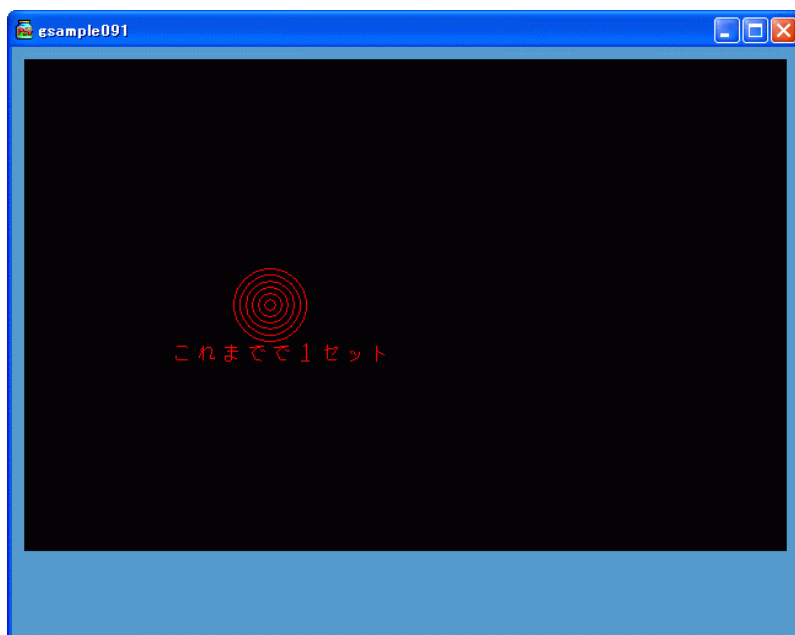


図 2-75 gsamp091.psv 実行画面

(6) 更新履歴

① Ver. 4.0

属性値のオプションの指定を変更しました (-a)

モード指定のオプションを削除しました (-m)

2.10.9 g_viewport

ビューポート座標の設定

(1) **形式**

`g_viewport contextID [-w] [x1 y1 x2 y2]`

(2) **説明**

`g_viewport` マクロは、グラフィックスウィンドウのビューポート座標を変更します。範囲を指定しない場合は、現在のビューポート座標の範囲が返ります。

なお、グラフィックスウィンドウ作成時のビューポートは、キャンバスいっぱいの範囲に設定されています。

① *contextID*

`g_context` で作成したコンテキスト ID を指定します。

② *-w gwinID*

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ *x1 y1 x2 y2*

ビューポート座標を左上座標 (*x1, y1*)、右下座標 (*x2, y2*) の順に指定します。

(3) **戻り値**

なし

(4) **関連項目**

`g_context`, `g_open_gwin`

(5) 使用例

gsamp092.psv 参照 (以下画面例)

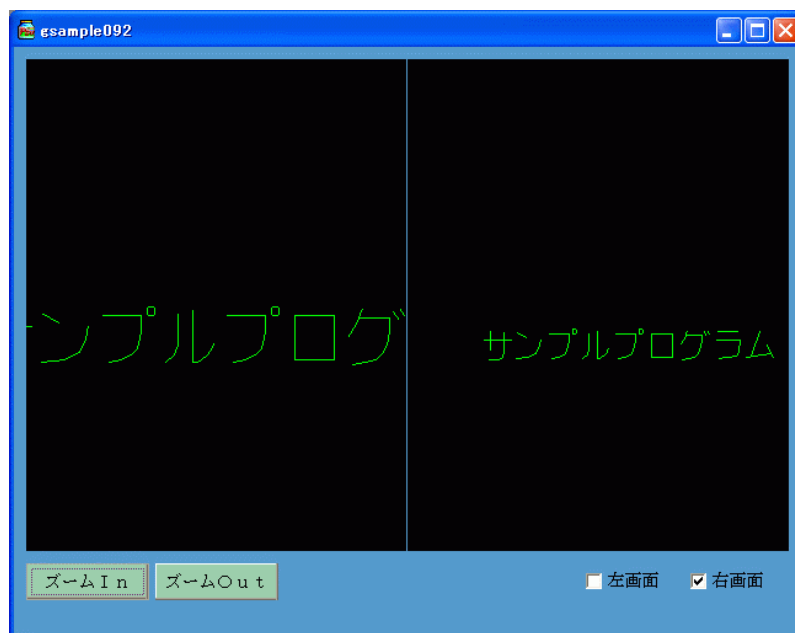


図 2-76 gsamp092.psv 実行画面

(6) 更新履歴

なし

2.10.10 g_wc2dc

WC 座標を DC 座標に変換

(1) 形式

`g_wc2dc context/D [-w] wx wy`

(2) 説明

`g_wc2dc` マクロは、グラフィックスウィンドウの WC 座標を、対応するキャンバス（ビューポート）の DC 座標に変換します。

① `context/D`

`g_context` で作成したコンテキスト ID を指定します。

② `-w gwin/D`

コンテキストオブジェクトに設定しているグラフィックスウィンドウ以外のグラフィックスウィンドウを指定する場合に、`g_open_gwin` で作成したグラフィックスウィンドウ ID を指定します。

③ `wx wy`

グラフィックスウィンドウの WC 座標の点 (`wx`、`wy`) を指定します。

(3) 戻り値

DC 座標での X 座標、Y 座標（リスト形式）

(4) 関連項目

`g_context`, `g_dc2wc`, `g_open_gwin`, `g_scroll`, `g_rotate`, `g_viewport`, `g_ortho`

(5) 使用例

`gsamp023.psv`参照 (画面例は2.3.1の画面参照)

(6) 更新履歴

なし

2.10.11 g_yoko_mercator

緯度経度座標と直交座標との変換

(1) **形式**

`g_yoko_mercator kei lDo ke lDo`

`g_yoko_mercator -r kei xval yval`

(2) **説明**

`g_yoko_mercator` マクロは緯度経度座標から平面直角座標への変換、平面直角座標から緯度経度座標への変換を行います。

① *kei*

平面直角座標系の系番号を指定します。

② *lDo*

緯度を指定します。

③ *ke lDo*

経度を指定します。

④ *-r*

平面直角座標から緯度経度座標を変換する時に指定します。

⑤ *xval*

平面直角座標系の X 座標値を指定します。

⑥ *yval*

平面直角座標系の Y 座標値を指定します。

(3) **戻り値**

平面直角座標系の X 及び Y 座標値、または緯度経度をリスト構造で返します。並び順は座標の場合、” X 座標 Y 座標”、緯度経度の場合、” 緯度 経度” となります。

(4) **関連項目**

なし

(5) 使用例

gsamp101.psv 参照 (以下画面例)

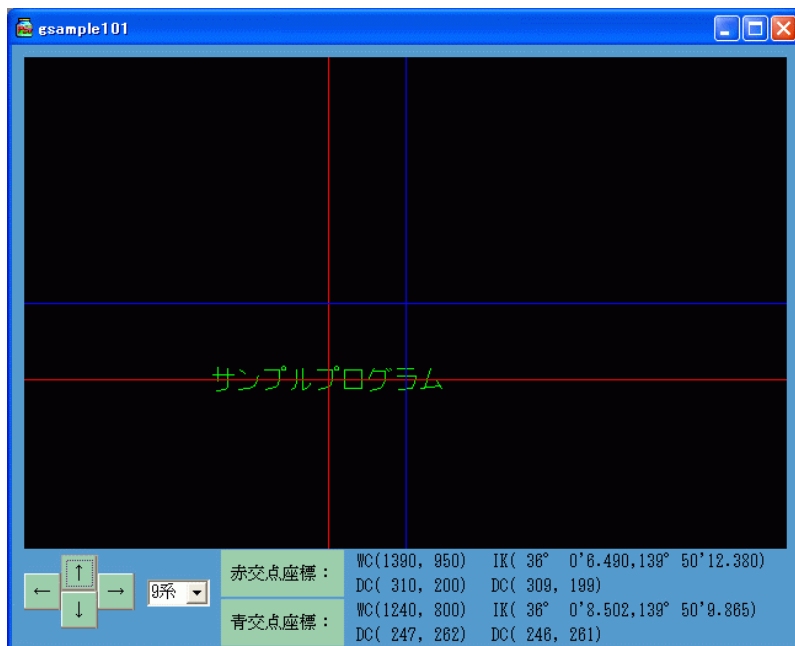


図 2-77 gsamp101.psv 実行画面

(6) 更新履歴

① Ver. 4.0

Ver. 4.0 で新規に追加したマクロです。