

# 日本語 OpenVMS

---

## 日本語画面管理ライブラリ 利用者の手引き

AA-PU8RF-TE

2004年2月

本書は、日本語 OpenVMS 上で実行される、画面管理実行時ライブラリ・ルーチン (SMG ルーチン) についての解説書です。

改訂 / 更新情報:

日本語 OpenVMS V7.3 『日本語画面管理ライブラリ 利用者の手引き』の改訂版です。

ソフトウェア・バージョン:

日本語 OpenVMS Alpha V7.3-2

日本語 OpenVMS VAX V7.3

日本ヒューレット・パッカート株式会社

---

© 2004 Hewlett-Packard Development Company, L.P.

本書の著作権は日本ヒューレット・パッカート株式会社が保有しており、本書中の解説および図、表は日本ヒューレット・パッカートの文書による許可なしに、その全体または一部を、いかなる場合にも再版あるいは複製することを禁じます。

また、本書に記載されている事項は、予告なく変更されることがありますので、あらかじめご承知おきください。万一、本書の記述に誤りがあった場合でも、日本ヒューレット・パッカートは一切その責任を負いかねます。

本書で解説するソフトウェア (対象ソフトウェア) は、所定のライセンス契約が締結された場合に限り、その使用あるいは複製が許可されます。

日本ヒューレット・パッカートは、日本ヒューレット・パッカートまたは日本ヒューレット・パッカートの指定する会社から納入された機器以外の機器で対象ソフトウェアを使用した場合、その性能あるいは信頼性について一切責任を負いかねます。

以下は、他社の商標です。

Adobe, Adobe Illustrator, POSTSCRIPT は米国 Adobe Systems 社の商標です。

BITSTREAM は米国 Bitstream 社の商標です。

Microsoft, MS および MS-DOS は米国 Microsoft 社の商標です。

Motif, OSF, OSF/1, OSF/Motif および Open Software Foundation は米国 Open Software Foundation 社の商標です。

その他のすべての商標および登録商標は、それぞれの所有者が保有しています。

本書は、日本語 VAX DOCUMENT V 2.1を用いて作成しています。

---

# 目次

まえがき	xi
第1部 概論	
1 日本語 SMG (SMG\$) の概要	
1.1	ペーストボード 1-5
1.2	仮想ディスプレイ 1-7
1.3	ビューポート 1-8
1.4	仮想キーボード 1-9
1.5	日本語画面管理ライブラリの使用方法 1-10
2 日本語 SMG の出力操作	
2.1	構成操作 2-1
2.1.1	ペースト操作 2-2
2.1.2	アンペースト操作 2-3
2.1.3	再ペースト操作 2-4
2.1.4	移動操作 2-5
2.1.5	削除操作とポップ操作 2-5
2.1.6	重なり合いの確認操作 2-6
2.2	仮想ディスプレイによる出力 2-7
2.2.1	カーソルの位置 2-8
2.2.2	削除操作 2-9
2.2.3	消去操作 2-10
2.2.4	挿入操作 2-10
2.2.5	書き込み操作 2-11
2.2.5.1	文字単位の出力 2-11
2.2.5.2	行単位の出力 2-11
2.2.6	仮想ディスプレイの属性の変更 2-12
2.2.7	ラインとライン描画文字を描く操作と削除する操作 2-13
2.2.8	外部テキストの表示 2-14
2.2.9	仮想ディスプレイからの読み込み 2-15
2.2.10	ビューポート 2-15
2.2.10.1	ビューポートの作成 2-15
2.2.10.2	ビューポートの削除 2-16
2.2.10.3	ビューポートのペーストとアンペースト 2-16
2.2.10.4	ビューポートのスクロールと移動 2-16
2.2.10.5	ビューポート属性の変更 2-17

2.2.11	メニュー	2-17
2.2.11.1	メニューの作成	2-17
2.2.11.2	メニューの削除	2-18
2.2.11.3	メニューからの選択	2-18
2.2.11.3.1	省略時のモード	2-18
2.2.11.3.2	RETURN_IMMED モード	2-18
2.2.11.3.3	REMOVE_ITEM モード	2-18
2.2.12	仮想ディスプレイの保存	2-19
2.2.13	ターミナル属性の変更	2-19
2.2.14	仮想ディスプレイに対する文字集合の設定	2-19
2.3	操作制御	2-20
2.3.1	最小限の更新	2-20
2.3.2	バッファ操作	2-20
2.3.3	タブ	2-21
2.4	出力のバッチ処理	2-21
2.4.1	ペーストボード更新のバッチ処理	2-21
<b>3</b>	<b>日本語 SMG 入力操作</b>	
3.1	仮想キーボードからのデータの取り込み	3-1
3.2	仮想キーボード属性の設定と検索	3-2
3.3	キーパッド・キーの使用による行の作成	3-3
3.4	状態	3-4
3.5	終了文字	3-5
3.6	行の再呼び出しと再呼び出しバッファ	3-8
3.7	入力と出力の相互作用	3-8
<b>4</b>	<b>日本語 SMG の高度な機能</b>	
4.1	非同期イベント	4-1
4.1.1	ブロードキャスト・メッセージ	4-2
4.1.2	要求されていない入力	4-2
4.1.3	アウト・オブ・バンド AST	4-2
4.2	サブプロセス	4-3
4.2.1	サブプロセスの作成	4-3
4.2.2	サブプロセスの削除	4-3
4.2.3	サブプロセス内でのコマンドの実行	4-3
4.3	カーソルの移動	4-4
4.4	終了ハンドラ	4-4
<b>5</b>	<b>フォーリン・ターミナルのサポート</b>	
5.1	TERMTABLE インターフェイス・ルーチン	5-2
5.2	機能フィールド	5-4
5.2.1	論理機能フィールド	5-5
5.2.2	数値機能フィールド	5-7
5.2.3	文字列機能フィールド	5-8
5.2.4	引数の置き換え	5-18

5.2.5	算術演算 .....	5-18
5.3	OpenVMS ターミナル機能ファイルの作成 .....	5-19
5.4	例 .....	5-20
5.5	TERMTABLE.EXE の作成 .....	5-22
5.6	画面管理が使用する機能フィールド .....	5-23
5.7	フォーリン・ターミナルに対する入力サポート .....	5-23
5.8	SET TERMINAL コマンドと SHOW TERMINAL コマンドのサポート .....	5-25
6	日本語 SMG を使用したプログラム開発	
6.1	日本語 SMG を使用しないルーチンの呼び出し .....	6-3
7	日本語 SMG ルーチンの呼び出しの例	

## 第 2 部 リファレンス・セクション

### 8 日本語 SMG リファレンス

SMG\$ADD_KEY_DEF .....	SMG-2
SMG\$BEGIN_DISPLAY_UPDATE .....	SMG-6
SMG\$BEGIN_PASTEBOARD_UPDATE .....	SMG-8
SMG\$CANCEL_INPUT .....	SMG-10
SMG\$CHANGE_PBD_CHARACTERISTICS .....	SMG-12
SMG\$CHANGE_RENDITION .....	SMG-16
SMG\$CHANGE_VIEWPORT .....	SMG-21
SMG\$CHANGE_VIRTUAL_DISPLAY .....	SMG-28
SMG\$CHECK_FOR_OCCLUSION .....	SMG-32
SMG\$CONTROL_MODE .....	SMG-37
SMG\$COPY_VIRTUAL_DISPLAY .....	SMG-41
SMG\$CREATE_KEY_TABLE .....	SMG-48
SMG\$CREATE_MENU .....	SMG-50
SMG\$CREATE_PASTEBOARD .....	SMG-55
SMG\$CREATE_SUBPROCESS .....	SMG-58
SMG\$CREATE_VIEWPORT .....	SMG-61
SMG\$CREATE_VIRTUAL_DISPLAY .....	SMG-67
SMG\$CREATE_VIRTUAL_KEYBOARD .....	SMG-73
SMG\$CURSOR_COLUMN .....	SMG-78
SMG\$CURSOR_ROW .....	SMG-80
SMG\$DEFINE_KEY .....	SMG-82
SMG\$DEL_TERM_TABLE .....	SMG-84
SMG\$DELETE_CHARS .....	SMG-85
SMG\$DELETE_KEY_DEF .....	SMG-90
SMG\$DELETE_LINE .....	SMG-92
SMG\$DELETE_MENU .....	SMG-97
SMG\$DELETE_PASTEBOARD .....	SMG-99

SMG\$DELETE_SUBPROCESS .....	SMG-101
SMG\$DELETE_VIEWPORT .....	SMG-103
SMG\$DELETE_VIRTUAL_DISPLAY .....	SMG-105
SMG\$DELETE_VIRTUAL_KEYBOARD .....	SMG-107
SMG\$DISABLE_BROADCAST_TRAPPING .....	SMG-109
SMG\$DISABLE_UNSOLICITED_INPUT .....	SMG-111
SMG\$DRAW_CHAR .....	SMG-113
SMG\$DRAW_LINE .....	SMG-118
SMG\$DRAW_RECTANGLE .....	SMG-124
SMG\$ENABLE_UNSOLICITED_INPUT .....	SMG-129
SMG\$END_DISPLAY_UPDATE .....	SMG-132
SMG\$END_PASTEBOARD_UPDATE .....	SMG-134
SMG\$ERASE_CHARS .....	SMG-136
SMG\$ERASE_COLUMN .....	SMG-141
SMG\$ERASE_DISPLAY .....	SMG-143
SMG\$ERASE_LINE .....	SMG-149
SMG\$ERASE_PASTEBOARD .....	SMG-154
SMG\$EXECUTE_COMMAND .....	SMG-157
SMG\$FIND_CURSOR_DISPLAY .....	SMG-160
SMG\$FLUSH_BUFFER .....	SMG-162
SMG\$GET_BROADCAST_MESSAGE .....	SMG-164
SMG\$GET_DISPLAY_ATTR .....	SMG-167
SMG\$GET_KEY_DEF .....	SMG-171
SMG\$GET_KEYBOARD_ATTRIBUTES .....	SMG-174
SMG\$GET_NUMERIC_DATA .....	SMG-176
SMG\$GET_PASTEBOARD_ATTRIBUTES .....	SMG-178
SMG\$GET_PASTING_INFO .....	SMG-181
SMG\$GET_TERM_DATA .....	SMG-184
SMG\$GET_VIEWPORT_CHAR .....	SMG-187
SMG\$HOME_CURSOR .....	SMG-191
SMG\$INIT_TERM_TABLE .....	SMG-193
SMG\$INIT_TERM_TABLE_BY_TYPE .....	SMG-195
SMG\$INSERT_CHARS .....	SMG-198
SMG\$INSERT_LINE .....	SMG-205
SMG\$INVALIDATE_DISPLAY .....	SMG-213
SMG\$KEYCODE_TO_NAME .....	SMG-215
SMG\$LABEL_BORDER .....	SMG-217
SMG\$LIST_KEY_DEFS .....	SMG-224
SMG\$LIST_PASTING_ORDER .....	SMG-227
SMG\$LOAD_KEY_DEFS .....	SMG-230
SMG\$LOAD_VIRTUAL_DISPLAY .....	SMG-232
SMG\$MOVE_TEXT .....	SMG-234
SMG\$MOVE_VIRTUAL_DISPLAY .....	SMG-238
SMG\$NAME_TO_KEYCODE .....	SMG-242
SMG\$PASTE_VIRTUAL_DISPLAY .....	SMG-244
SMG\$POP_VIRTUAL_DISPLAY .....	SMG-247
SMG\$PUT_CHARS .....	SMG-249

SMG\$PUT_CHARS_HIGHWIDE .....	SMG-255
SMG\$PUT_CHARS_MULTI .....	SMG-259
SMG\$PUT_CHARS_WIDE .....	SMG-264
SMG\$PUT_HELP_TEXT .....	SMG-269
SMG\$PUT_LINE .....	SMG-274
SMG\$PUT_LINE_HIGHWIDE .....	SMG-282
SMG\$PUT_LINE_MULTI .....	SMG-286
SMG\$PUT_LINE_WIDE .....	SMG-290
SMG\$PUT_STATUS_LINE .....	SMG-296
SMG\$READ_COMPOSED_LINE .....	SMG-298
SMG\$READ_FROM_DISPLAY .....	SMG-305
SMG\$READ_KEYSTROKE .....	SMG-311
SMG\$READ_STRING .....	SMG-318
SMG\$READ_VERIFY .....	SMG-329
SMG\$REMOVE_LINE .....	SMG-337
SMG\$REPAINT_SCREEN .....	SMG-340
SMG\$REPASTE_VIRTUAL_DISPLAY .....	SMG-344
SMG\$REPLACE_INPUT_LINE .....	SMG-351
SMG\$RESTORE_PHYSICAL_SCREEN .....	SMG-355
SMG\$RETURN_CURSOR_POS .....	SMG-357
SMG\$RETURN_INPUT_LINE .....	SMG-359
SMG\$RING_BELL .....	SMG-364
SMG\$SAVE_PHYSICAL_SCREEN .....	SMG-366
SMG\$SAVE_VIRTUAL_DISPLAY .....	SMG-369
SMG\$SCROLL_DISPLAY_AREA .....	SMG-371
SMG\$SCROLL_VIEWPORT .....	SMG-374
SMG\$SELECT_FROM_MENU .....	SMG-381
SMG\$SET_BROADCAST_TRAPPING .....	SMG-391
SMG\$SET_CURSOR_ABS .....	SMG-394
SMG\$SET_CURSOR_MODE .....	SMG-396
SMG\$SET_CURSOR_REL .....	SMG-398
SMG\$SET_DEFAULT_STATE .....	SMG-400
SMG\$SET_DISPLAY_SCROLL_REGION .....	SMG-402
SMG\$SET_KEYPAD_MODE .....	SMG-404
SMG\$SET_OUT_OF_BAND_ASTS .....	SMG-406
SMG\$SET_PHYSICAL_CURSOR .....	SMG-409
SMG\$SET_TERM_CHARACTERISTICS .....	SMG-411
SMG\$UNPASTE_VIRTUAL_DISPLAY .....	SMG-414

## 索引

例

7-1	FORTRAN での SMG\$ルーチンの使用 .....	7-1
-----	-------------------------------	-----

図

1-1	ペーストボード座標系 .....	1-6
2-1	ペースト操作 .....	2-2
2-2	アンペースト操作 .....	2-3
2-3	再ペースト操作 .....	2-4
2-4	移動操作 .....	2-5
2-5	ポップ操作 .....	2-6
2-6	重なり合いの確認 .....	2-7
2-7	SMG\$DRAW_LINE ルーチンを使用せずに描いたライン .....	2-14
2-8	SMG\$DRAW_LINE ルーチンを使用して描いたライン .....	2-14
8-1	仮想ディスプレイがペーストされた後に生成される出力 .....	SMG-25
8-2	ビューポートを作成した後に生成される出力 .....	SMG-26
8-3	SMG\$CHANGE_VIEWPORT ルーチンを呼び出した後に生成される出力 .....	SMG-27
8-4	SMG\$CHECK_FOR_OCCLUSION ルーチンを呼び出す FORTRAN プログラムが生成する出力 .....	SMG-36
8-5	この例で生成される最初の仮想ディスプレイ .....	SMG-45
8-6	この例で生成される 2 番目の仮想ディスプレイ .....	SMG-46
8-7	SMG\$COPY_VIRTUAL_DISPLAY ルーチンを呼び出した後で生成される出力 .....	SMG-47
8-8	ビューポートを作成することにより生成される出力 .....	SMG-66
8-9	SMG\$CREATE_VIRTUAL_DISPLAY ルーチンを呼び出す FORTRAN のプログラムが生成する出力 .....	SMG-72
8-10	SMG\$DELETE_CHARS ルーチンを呼び出す前に生成される出力 .....	SMG-88
8-11	SMG\$DELETE_CHARS ルーチンを呼び出した後で生成される出力 .....	SMG-89
8-12	SMG\$DELETE_LINE ルーチンを呼び出す前に FORTRAN プログラムが生成する出力 .....	SMG-95
8-13	SMG\$DELETE_LINE ルーチンを呼び出した後に生成される出力 .....	SMG-96
8-14	SMG\$DRAW_LINE ルーチンを呼び出す FORTRAN プログラムによって生成される出力 .....	SMG-123
8-15	SMG\$DRAW_RECTANGLE ルーチンを呼び出す FORTRAN プログラムによって生成される出力 .....	SMG-128
8-16	AST ルーチン引数 .....	SMG-130
8-17	SMG\$ERASE_CHARS ルーチンを呼び出す前に生成される出力 .....	SMG-139
8-18	SMG\$ERASE_CHARS ルーチンを呼び出した後に生成される出力 .....	SMG-140
8-19	SMG\$ERASE_DISPLAY ルーチンを呼び出す FORTRAN プログラムによって生成される最初の出力 .....	SMG-147
8-20	SMG\$ERASE_DISPLAY ルーチンを呼び出した後に生成される出力 .....	SMG-148
8-21	SMG\$ERASE_LINE ルーチンを呼び出す FORTRAN プログラムによって生成される最初の出力 .....	SMG-152
8-22	SMG\$ERASE_LINE ルーチンを呼び出した後で生成される出力 .....	SMG-153
8-23	SMG\$GET_VIEWPORT_CHAR ルーチンによって生成される出力 .....	SMG-190
8-24	SMG\$INSERT_CHARS ルーチンを呼び出す前に生成される出力 .....	SMG-203

8-25	SMG\$INSERT_CHARS ルーチンを実行後に生成される出力	SMG-204
8-26	SMG\$INSERT_LINE ルーチンを実行前に生成される出力	SMG-211
8-27	SMG\$INSERT_LINE ルーチンを実行後に生成される出力	SMG-212
8-28	SMG\$LABEL_BORDER ルーチンを実行するプログラムが生成する出力	SMG-223
8-29	テキスト移動のための引数	SMG-236
8-30	SMG\$PUT_CHARS ルーチンを実行する FORTRAN プログラムによって生成される出力	SMG-254
8-31	SMG\$PUT_LINE ルーチンを実行する FORTRAN プログラムによって生成される出力	SMG-279
8-32	SMG\$PUT_LINE_WIDE を実行する FORTRAN プログラムによって生成される出力	SMG-295
8-33	SMG\$READ_FROM_DISPLAY ルーチンを実行する出力	SMG-310
8-34	SMG\$READ_KEYSTROKE ルーチンを実行する FORTRAN プログラムが生成する出力	SMG-317
8-35	SMG\$READ_STRING ルーチンを実行前に生成される出力	SMG-327
8-36	SMG\$READ_STRING ルーチンを実行後に生成される出力	SMG-328
8-37	SMG\$REPAINT_SCREEN ルーチンを実行する FORTRAN プログラムが作成する出力	SMG-342
8-38	SMG\$REPAINT_SCREEN ルーチンを実行する FORTRAN プログラムが生成する出力	SMG-343
8-39	SMG\$MOVE_VIRTUAL_DISPLAY ルーチンを実行する前の出力	SMG-348
8-40	SMG\$MOVE_VIRTUAL_DISPLAY ルーチンを実行後に表示される出力	SMG-349
8-41	SMG\$REPASTE_VIRTUAL_DISPLAY ルーチンを実行後に表示される出力	SMG-350
8-42	仮想ディスプレイをペーストすることにより生成される出力	SMG-377
8-43	SMG\$SCROLL_VIEWPORT ルーチンを最初に実行後に生成される出力	SMG-378
8-44	SMG\$SCROLL_VIEWPORT ルーチンを 2 回目に実行後に生成される出力	SMG-379
8-45	SMG\$SCROLL_VIEWPORT ルーチンを最後に実行後に生成される出力	SMG-380
8-46	メニュー項目が選択される前に生成される出力	SMG-388
8-47	3 つのメニュー選択をした後に生成された出力	SMG-389
8-48	プログラムが終了した後に生成される出力	SMG-390

## 表

1	漢字ターミナル	xiv
1-1	弊社以外の漢字ターミナルをサポートするルーチン	1-1
1-2	日本語 SMG 出力ルーチン	1-2
1-3	日本語 SMG 入力ルーチン	1-4
2-1	ビットの設定	2-12
2-2	省略時文字集合名	2-19
3-1	終了文字の値	3-6
5-1	論理値機能	5-5
5-2	数値機能	5-8

5-3	文字列機能.....	5-9
5-4	String_2 機能.....	5-14

## 本書の目的

本書は、標準版 OpenVMS の画面管理ライブラリを日本語のために拡張した、日本語画面管理実行時ライブラリ・ルーチンについて説明します。これらのルーチンは、日本語 OpenVMS のもとで実行されます。本書では、日本語画面管理ルーチン (以下、日本語 SMG ルーチンと呼ぶ) の使い方を説明し、さらに参照情報を示します。

## 対象読者

本書は、実行時ライブラリ・ルーチンを使用するシステム・プログラマとアプリケーション・プログラマを対象にしています。

## 本書の構成

本書は 2 部で構成されています。

第 1 部は 7 つの章に分かれています。

- |       |   |
|-------|---|
| 第 1 章 | 日本語 SMG ルーチンを示し、おもな SMG\$ の構成要素の概要を示します。              |
| 第 2 章 | 日本語 SMG が提供する出力操作について説明します。                           |
| 第 3 章 | 仮想キーボードからの入力を実行するために使用する日本語 SMG ルーチンについて説明します。        |
| 第 4 章 | 日本語 SMG の高度な機能について説明します。                              |
| 第 5 章 | フォーリン・ターミナルをサポートする方法について説明します。                        |
| 第 6 章 | 新しいプログラムを開発するために日本語 SMG を使用する時の望ましい方法について説明します。       |
| 第 7 章 | VMS Calling Standard に準拠した言語から SMG\$ ルーチンを呼び出す例を示します。 |

第 2 部は日本語 SMG リファレンスです。

日本語 SMG リファレンス	実行時ライブラリの日本語 SMG に含まれている各ルーチンの詳細な参照情報です。この情報は、『OpenVMS Programming Interfaces: Calling a System Routine』に説明されている形式を使用して示されています。ルーチンの説明はルーチン名のアルファベット順に並べられています。
----------------	---

## 関連資料

実行時ライブラリ・ルーチンは複数の参照マニュアルに説明されています。実行時ライブラリの概要と実行時ライブラリ・ルーチンのアクセス方法の説明は、『OpenVMS Programming Interfaces: Calling a System Routine』にまとめられています。他のRTLの機能の説明と、各機能に対応するルーチンおよびその使い方は、次のマニュアルに説明されています。

- 『IMLIB/OpenVMS ライブラリ リファレンス・マニュアル』
- 『日本語ライブラリ 利用者の手引き』
- 『OpenVMS I/O User's Reference Manual』
- 『OpenVMS RTL Screen Management (SMGS) Manual』
- 『OpenVMS RTL Library (LIBS) Manual』
- 『OpenVMS RTL General Purpose (OTS\$) Manual』
- 『OpenVMS RTL Parallel Processing (PPLS) Manual』
- 『OpenVMS RTL String Manipulation (STRS) Manual』
- 『VMS RTL Mathematics (MTH\$) Manual』
- 『VMS RTL DECtalk (DTKS) Manual』

OpenVMS プロシージャ呼び出し/条件処理標準規則は『OpenVMS Programming Interfaces: Calling a System Routine』 および 『OpenVMS Calling Standard』に説明されており、実行時ライブラリ・ルーチンを使用する人にとって役立つ情報がまとめられています。

どの言語を使用するアプリケーション・プログラマも、モジュール方式のプログラミング標準規則と他のガイドラインについて、『OpenVMS ユーザーズ・マニュアル』を参照してください。

高級言語プログラマは、各言語の参照マニュアルに示されている実行時ライブラリ・ルーチンの呼び出しに関する補足情報を参照してください。補足情報はまた、VMS Calling Standard に準拠した言語に添付されている言語のユーザ・ガイドにも示されています。

かな漢字変換キーの設定については『ユーザ・キー定義 利用者の手引き』を参照してください。

## 表記法

本書では以下の表記法を使用します。

表記法	意味
Ctrl/x	Ctrl/xという表記は、Ctrl キーを押しながら別のキーまたはポインティング・デバイス・ボタンを押すことを示します。
PF1 x	PF1 xという表記は、PF1 に定義されたキーを押してから、別のキーまたはポインティング・デバイス・ボタンを押すことを示します。
<span style="border: 1px solid black; padding: 2px;">Return</span>	例の中で、キー名が四角で囲まれている場合には、キーボード上でそのキーを押すことを示します。テキストの中では、キー名は四角で囲まれていません。
...	HTML 形式のドキュメントでは、キー名は四角ではなく、括弧で囲まれています。
...	例の中の水平方向の反復記号は、次のいずれかを示します。 <ul style="list-style-type: none"> <li>• 文中のオプションの引数が省略されている。</li> <li>• 前出の 1 つまたは複数の項目を繰り返すことができる。</li> <li>• パラメータや値などの情報をさらに入力できる。</li> </ul>
.	垂直方向の反復記号は、コードの例やコマンド形式の中の項目が省略されていることを示します。このように項目が省略されるのは、その項目が説明している内容にとって重要ではないからです。
( )	コマンドの形式の説明において、括弧は、複数のオプションを選択した場合に、選択したオプションを括弧で囲まなければならないことを示しています。
[ ]	コマンドの形式の説明において、大括弧で囲まれた要素は任意のオプションです。オプションをすべて選択しても、いずれか 1 つを選択しても、あるいは 1 つも選択しなくても構いません。ただし、OpenVMS ファイル指定のディレクトリ名の構文や、割り当て文の部分文字列指定の構文の中では、大括弧に囲まれた要素は省略できません。
[   ]	コマンド形式の説明では、括弧内の要素を分けている垂直棒線はオプションを 1 つまたは複数選択するか、または何も選択しないことを意味します。
{ }	コマンドの形式の説明において、中括弧で囲まれた要素は必須オプションです。いずれか 1 のオプションを指定しなければなりません。
太字	太字のテキストは、新しい用語、引数、属性、条件を示しています。
<i>italic text</i>	イタリック体のテキストは、重要な情報を示します。また、システム・メッセージ (たとえば内部エラー <i>number</i> )、コマンド・ライン (たとえば <i>/PRODUCER=name</i> )、コマンド・パラメータ (たとえば <i>device-name</i> ) などの変数を示す場合にも使用されます。
UPPERCASE TEXT	英大文字のテキストは、コマンド、ルーチン名、ファイル名、ファイル保護コード名、システム特権の短縮形を示します。
Monospace type	モノスペース・タイプの文字は、コード例および会話型の画面表示を示します。 C プログラミング言語では、テキスト中のモノスペース・タイプの文字は、キーワード、別々にコンパイルされた外部関数およびファイルの名前、構文の要約、または例に示される変数または識別子への参照などを示します。
-	コマンド形式の記述の最後、コマンド・ライン、コード・ラインにおいて、ハイフンは、要求に対する引数がその後の行に続くことを示します。
数字	特に明記しない限り、本文中の数字はすべて 10 進数です。10 進数以外 (2 進数, 8 進数, 16 進数) は、その旨を明記してあります。

実行時ライブラリ・ルーチンの説明で使用される他の表記法については、  
『OpenVMS Programming Interfaces: Calling a System Routine』を参照し  
てください。

## サポートされるハードウェア

日本語 SMG では次の漢字ターミナルをサポートします。

- VT80 (限定サポート)
- VT280 シリーズ
- VT382F

表 1 は、各漢字ターミナルがサポートする文字集合を表しています。

表 1 漢字ターミナル

文字集合	VT80	VT280 シリーズ	VT382F
ASCII 文字集合			
DEC 特殊文字集合			
DEC 補助文字集合			
ローマ字文字集合			
JIS カタカナ文字集合			
DEC 漢字文字集合			
ISO Latin-1 補助文字集合			

## 用語

- 仮想ディスプレイ (Virtual Display)  
プログラムが日本語 SMG ルーチン呼び出してデータを書き込む四角い領域です。
- カラム数 (Number of Columns)  
カラム数は、SMG\$PUT\_CHARS ルーチンがターミナル画面に書き込む ASCII 文字単位によってカウントされます。
- 行数 (Number of Lines)  
行数は、SMG\$PUT\_CHARS ルーチンがターミナル画面に書き込む ASCII 文字単位によってカウントされます。
- ターミナル画面 (Terminal Screen)  
このバージョンでは、ビデオ・ターミナルの表示可能な部分を示します。
- 物理画面 (Physical Screen)  
日本語 SMG のこのバージョンでは、ターミナル画面とほぼ同義です。
- 未定義文字 (Undefine Character)  
値が未定義で、文字集合が認識されない文字ですが、他の属性は ASCII 文字と同じです。

# 第1部

---

## 概論

第1部では、日本語画面管理ライブラリについて説明します。ライブラリ・ルーチンの詳細については、第2部「日本語 SMG リファレンス」をご覧ください。

第1部は次の7つの章で構成されています。

- 第1章, 日本語 SMG (SMG\$) の概要
- 第2章, 日本語 SMG の出力操作
- 第3章, 日本語 SMG 入力操作
- 第4章, 日本語 SMG の高度な機能
- 第5章, フォーリン・ターミナルのサポート
- 第6章, 日本語 SMG を使用したプログラム開発
- 第7章, 日本語 SMG ルーチンの呼び出しの例

---

## 日本語 SMG (SMG\$) の概要

この章では、漢字ターミナルから独立した機能を実行する、実行時ライブラリ・ルーチンについて説明します。日本語 SMG のもっとも重要な要素は、入出力を実際に行う物理装置とユーザ・プログラムが完全に分離されるということです。ユーザ・プログラムは物理画面に直接書き込みを実行するのではなく、仮想ディスプレイに書き込みます。同様に、ユーザ・プログラムは物理キーボードから直接入力を実行するのではなく、仮想キーボードから入力を実行します(仮想ディスプレイと仮想キーボードは論理的なものであり、その使い方についてはこの後の節で詳しく説明します)。このように、仮想操作を物理操作から分離することにより、入出力をターミナルから独立させることが可能になります。

この後に示す日本語 SMG ルーチンは、ビデオ画面上で複雑なイメージを設計し、構成し、管理するのに役立ちます。これらのルーチンは、通常は、VT100 クラスの漢字ターミナルで実行される操作タイプに対応しており、また、日本語 SMG がハードウェアで実現されていない漢字ターミナルで、その機能をソフトウェアでエミュレーションできるようにもします。各ルーチンはその機能に従って次の3つに分類されています。

- 弊社製品以外の漢字ターミナルをサポートするルーチン
- 日本語 SMG 入力ルーチン
- 日本語 SMG 出力ルーチン

表 1-1 は、弊社製品以外の漢字ターミナルをサポートするルーチンです。これらのルーチンについては、第 5 章を参照してください。

表 1-1 弊社以外の漢字ターミナルをサポートするルーチン

---

弊社以外の漢字ターミナルをサポートするルーチン

---

SMG\$DEL\_TERM\_TABLE  
SMG\$GET\_NUMERIC\_DATA  
SMG\$GET\_TERM\_DATA  
SMG\$INIT\_TERM\_TABLE  
SMG\$INIT\_TERM\_TABLE\_BY\_TYPE

---

表 1-2 は、日本語 SMG 出力ルーチンを示しています。これらのルーチンについては、第 2 章を参照してください。

表 1-2 日本語 SMG 出力ルーチン

---

出力ルーチン

---

SMG\$BEGIN\_PASTEBOARD\_UPDATE  
SMG\$CHANGE\_PBD\_CHARACTERISTICS  
SMG\$CHANGE\_RENDITION  
SMG\$CHANGE\_VIEWPORT  
SMG\$CHANGE\_VIRTUAL\_DISPLAY  
SMG\$CHECK\_FOR\_OCCLUSION  
SMG\$CONTROL\_MODE  
SMG\$COPY\_VIRTUAL\_DISPLAY  
SMG\$CREATE\_MENU  
SMG\$CREATE\_PASTEBOARD  
SMG\$CREATE\_SUBPROCESS  
SMG\$CREATE\_VIEWPORT  
SMG\$CREATE\_VIRTUAL\_DISPLAY  
SMG\$CURSOR\_COLUMN  
SMG\$CURSOR\_ROW  
SMG\$DELETE\_CHARS  
SMG\$DELETE\_LINE  
SMG\$DELETE\_MENU  
SMG\$DELETE\_PASTEBOARD  
SMG\$DELETE\_SUBPROCESS  
SMG\$DELETE\_VIEWPORT  
SMG\$DELETE\_VIRTUAL\_DISPLAY  
SMG\$DISABLE\_BROADCAST\_TRAPPING  
SMG\$DISABLE\_UNSOLICITED\_INPUT  
SMG\$DRAW\_CHAR  
SMG\$DRAW\_LINE  
SMG\$DRAW\_RECTANGLE  
SMG\$ENABLE\_UNSOLICITED\_INPUT  
SMG\$END\_PASTEBOARD\_UPDATE  
SMG\$ERASE\_CHARS  
SMG\$ERASE\_COLUMN  
SMG\$ERASE\_DISPLAY  
SMG\$ERASE\_LINE  
SMG\$ERASE\_PASTEBOARD  
SMG\$EXECUTE\_COMMAND  
SMG\$FIND\_CURSOR\_DISPLAY  
SMG\$FLUSH\_BUFFER

---

(次ページに続く)

表 1-2 (続き) 日本語 SMG 出力ルーチン

---

出力ルーチン

---

SMG\$GET\_BROADCAST\_MESSAGE  
SMG\$GET\_DISPLAY\_ATTR  
SMG\$GET\_PASTEBOARD\_ATTRIBUTES  
SMG\$GET\_PASTING\_INFO  
SMG\$GET\_VIEWPORT\_CHAR  
SMG\$HOME\_CURSOR  
SMG\$INSERT\_CHARS  
SMG\$INSERT\_LINE  
SMG\$LABEL\_BORDER  
SMG\$LIST\_PASTING\_ORDER  
SMG\$LOAD\_VIRTUAL\_DISPLAY  
SMG\$MOVE\_TEXT  
SMG\$MOVE\_VIRTUAL\_DISPLAY  
SMG\$PASTE\_VIRTUAL\_DISPLAY  
SMG\$POP\_VIRTUAL\_DISPLAY  
SMG\$PUT\_CHARS  
SMG\$PUT\_CHARS\_HIGHWIDE  
SMG\$PUT\_CHARS\_MULTI  
SMG\$PUT\_CHARS\_WIDE  
SMG\$PUT\_HELP\_TEXT  
SMG\$PUT\_LINE  
SMG\$PUT\_LINE\_HIGHWIDE  
SMG\$PUT\_LINE\_MULTI  
SMG\$PUT\_LINE\_WIDE  
SMG\$PUT\_STATUS\_LINE  
SMG\$READ\_FROM\_DISPLAY  
SMG\$REMOVE\_LINE  
SMG\$REPAINT\_SCREEN  
SMG\$REPASTE\_VIRTUAL\_DISPLAY  
SMG\$RESTORE\_PHYSICAL\_SCREEN  
SMG\$RETURN\_CURSOR\_POS  
SMG\$RING\_BELL  
SMG\$SAVE\_PHYSICAL\_SCREEN  
SMG\$SAVE\_VIRTUAL\_DISPLAY  
SMG\$SCROLL\_DISPLAY\_AREA  
SMG\$SCROLL\_VIEWPORT  
SMG\$SELECT\_FROM\_MENU

---

(次ページに続く)

表 1-2 (続き) 日本語 SMG 出力ルーチン

---

出力ルーチン

---

SMG\$SET\_BROADCAST\_TRAPPING  
SMG\$SET\_CURSOR\_ABS  
SMG\$SET\_CURSOR\_MODE  
SMG\$SET\_CURSOR\_REL  
SMG\$SET\_DISPLAY\_SCROLL\_REGION  
SMG\$SET\_OUT\_OF\_BAND\_ASTS  
SMG\$SET\_PHYSICAL\_CURSOR  
SMG\$SET\_TERM\_CHARACTERISTICS  
SMG\$UNPASTE\_VIRTUAL\_DISPLAY

---

表 1-3 は、日本語 SMG 入力ルーチンをまとめています。これらのルーチンについては、第 3 章を参照してください。

表 1-3 日本語 SMG 入力ルーチン

---

入力ルーチン

---

SMG\$ADD\_KEY\_DEF  
SMG\$CANCEL\_INPUT  
SMG\$CREATE\_KEY\_TABLE  
SMG\$CREATE\_VIRTUAL\_KEYBOARD  
SMG\$DEFINE\_KEY  
SMG\$DELETE\_KEY\_DEF  
SMG\$DELETE\_VIRTUAL\_KEYBOARD  
SMG\$GET\_KEY\_DEF  
SMG\$GET\_KEYBOARD\_ATTRIBUTES  
SMG\$KEYCODE\_TO\_NAME  
SMG\$LIST\_KEY\_DEFS  
SMG\$LOAD\_KEY\_DEFS  
SMG\$NAME\_TO\_KEYCODE  
SMG\$READ\_COMPOSED\_LINE  
SMG\$READ\_KEYSTROKE  
SMG\$READ\_STRING  
SMG\$READ\_VERIFY  
SMG\$REPLACE\_INPUT\_LINE  
SMG\$RETURN\_INPUT\_LINE  
SMG\$SET\_DEFAULT\_STATE  
SMG\$SET\_KEYPAD\_MODE

---

日本語 SMG は、次に示す 2 つの重要なサービスを提供します。

- 漢字ターミナルからの独立性

日本語 SMG ルーチンは、使用している漢字ターミナルのタイプを考慮せずに、一般に必要とされる画面機能を実行できるようにすることにより、漢字ターミナルからの独立性を実現しています。入出力も含めてすべての操作は、呼び出し側の漢字ターミナルから独立した要求を、その動作を実行するのに必要な、一連のコードに変換するルーチンを呼び出すことにより実行されます。

使用している漢字ターミナルが、要求された操作をハードウェアでサポートしない場合には、ほとんどの場合、日本語 SMG は、その機能をソフトウェアでエミュレートすることにより、要求された動作を実行します。同様に、日本語 SMG ルーチンは、使用しているキーボードのタイプを考慮せずに、キーボードから入力を実行できるように、漢字ターミナルから独立した入力方法も提供します。

---

### 注意

---

日本語 SMG は、漢字ターミナルが完全に日本語 SMG の管理下にあることを仮定しています。したがって、アプリケーションは、日本語 SMG ルーチンと日本語 DIGITAL GKS や日本語 FMS のような画面管理用製品を同時に呼ぶべきではありません。

---

- イメージ構成の容易性

日本語 SMG ルーチンは、画面上で複雑なイメージを構成するのに役立ちます。たとえば、画面の一部からユーザ入力を要求し、結果を画面の別の部分に表示し、さらに画面の別の部分に状態情報を表示できます。通常、これらのいずれかの領域から読み込むルーチンやこれらの領域に書き込むルーチンは、他の領域が存在するかどうかを確認しなければなりません。これらの情報を知らなければ、行情報とカラム情報を使用して、画面の適切な部分に情報を書き込んだり、画面の適切な部分から情報を読み込むことができません。日本語 SMG ルーチンを使用すれば、各ルーチンは領域の位置とは無関係に、画面上の特定の領域に書き込むことができます。行とカラムに対する参照は、ルーチンが指定する画面の領域にだけ関係します。

この後の節では、画面管理の基本要素について説明します。これらの要素はペーストボード、仮想ディスプレイ、ビューポート、および仮想キーボードです。

---

## 1.1 ペーストボード

ペーストボードは、漢字ターミナルの画面に対して出力操作を実行するための論理構造です。ペーストボードは、仮想ディスプレイを配置し、操作するための 2 次元の領域として考えることができます。ペーストボードは常に物理装置に対応しますが、ペーストボードのサイズは物理画面より大きくても、小さくてもかまいません。各出力装置に対し、ペーストボードは 1 つだけ設定できます。

---

注意

---

日本語 SMG は、RMS ファイルとプリンタを出力装置としてサポートしません。

---

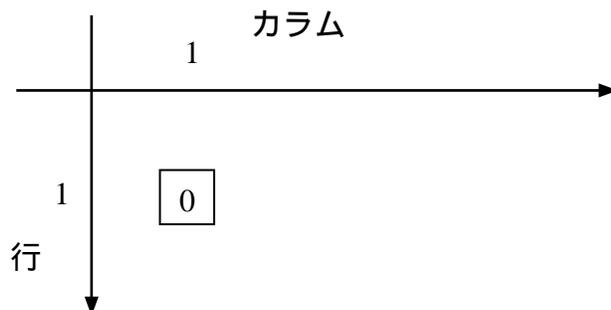
ペーストボードを作成するには、SMG\$CREATE\_PASTEBOARD ルーチンを呼び出し、引数としてペーストボードに対応づける物理装置を指定します。SMG\$CREATE\_PASTEBOARD ルーチンは、固有のペーストボード識別子 (pasteboard-id) を返します。この情報は、後続のルーチン呼び出しでペーストボード識別子が必要なときに使用できます。

たとえば、仮想ディスプレイをペーストする物理ターミナル画面を指定する場合には、pasteboard-id 引数を使用します。SMG\$CREATE\_PASTEBOARD ルーチンは出力引数として、対応する装置で使用できる行数とカラム数も指定します。この情報を使用すれば、物理画面のサイズで仮想ディスプレイを作成できます (仮想ディスプレイについては、次の節で説明します)。

ペーストボードは、1 つ以上の仮想ディスプレイの相対的な位置を指定するための論理座標系である、と考えると便利です (ペーストボード自体には物理的な境界はありませんが、物理画面には物理的な境界があります)。図 1-1 は、ペーストボード座標系を示しています。

原点 (セル位置 1, 1) は物理画面の左上の角に対応します。行番号とカラム番号はこの原点から始まります。たとえば、VT200 シリーズ・ターミナルの場合には、24 行 × 80 カラムであり、ペーストボード座標系の最初の 24 行と最初の 80 カラムが物理画面にマップされます。仮想ディスプレイはこの座標系のどこにでも配置でき、物理画面に対応する象限に制限されません。したがって、ペーストするとき (つまり、ペーストボードに配置するとき)、仮想ディスプレイが物理画面上に表示されなかったり、あるいは部分的にしか表示されないことがあります。

図 1-1 ペーストボード座標系



ペーストボードを削除する(つまり、特定の装置から切り離す)には、SMG\$DELETE\_PASTEBOARD ルーチンを使用します。ペーストボードを削除すると、このペーストボードにペーストされたすべての仮想ディスプレイはアンペーストされます。

ペーストボードを作成した後、SMG\$GET\_PASTEBOARD\_ATTRIBUTES ルーチン呼び出しにより、そのペーストボードの属性(特にそのサイズ)を調べることができます。ペーストボードの属性を変更するには、SMG\$CHANGE\_PBD\_CHARACTERISTICS ルーチンを使用します。ただし、属性を変更できるのは、対応する物理装置がその属性の変更を認めている場合に限りです。たとえば、装置が VT100 の場合には、ペーストボードの幅を 80 カラムから 132 カラムに変更できます。

ペーストボードを作成するときに、日本語 SMG は、画面をクリアします。しかし、画面をそのまま保存することを要求することもできます。さらに、SMG\$ERASE\_PASTEBOARD ルーチンを使用すれば、画面を消去することを要求できます。

---

## 1.2 仮想ディスプレイ

仮想ディスプレイはターミナル画面の中の長方形の部分であり、プログラムはルーチン呼び出しを使用して、その領域にデータを書き込みます。仮想ディスプレイは日本語 SMG の中心的部分です。画面に配置するイメージを作成する場合には、物理画面ではなく、仮想ディスプレイを基準にしてイメージを考慮しなければなりません。このように仮想ディスプレイを物理画面と切り離すことにより、メイン・プログラムは仮想ディスプレイの位置を変更でき、仮想ディスプレイに書き込むサブルーチンが物理画面でのディスプレイの位置を考慮する必要がないようにしています。

仮想ディスプレイがペーストボードに対応づけられている場合には、仮想ディスプレイはペーストボードにペーストされていると言います。ディスプレイをペーストボードから削除することを、アンペーストと言います。仮想ディスプレイがペーストボードにペーストされていない限り、その仮想ディスプレイは表示されません(仮想ディスプレイのペーストについての詳しい説明は、第 2.1.1 項を参照してください)。

プログラムは仮想ディスプレイをいくつでも作成し、管理できます(仮想ディスプレイの数は、使用できる仮想アドレス空間によってのみ制限されます)。1 つの仮想ディスプレイを一度に複数のペーストボードにペーストできます。したがって、プログラムは仮想ディスプレイだけを管理すれば十分です。仮想ディスプレイが変更された場合には、その変更結果は、ディスプレイがペーストされている各ペーストボード(およびそれに対応するターミナル画面)で自動的に反映されます。

仮想ディスプレイを作成するには、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンを使用します。このルーチン呼び出し場合には、仮想ディスプレイの行数とカラム数も指定しなければなりません。プログラムはまた、ディスプレイに適用される特定のディスプレイ属性とビデオ属性も指定できます。

SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンは固有の仮想ディスプレイ識別子 (display-id) を返します。ディスプレイを変更する後続のルーチン呼び出しで仮想ディスプレイを識別する場合には、この display-id 引数を使用します。

プログラムやサブルーチンは SMG\$GET\_DISPLAY\_ATTR ルーチンを呼び出すことにより、仮想ディスプレイの属性とサイズを判断できます。複数の仮想ディスプレイをペーストボードにペーストしている場合には、SMG\$LIST\_PASTING\_ORDER ルーチンを使用することにより、仮想ディスプレイをペーストした順序を判断できます。

他の属性が指定されていない場合には、省略時のビデオ属性が出力に適用されます。レンディション (属性) は、オンまたはオフに設定できるビデオ属性です。このようなビデオ属性としては、点滅表示、高輝度表示、反転表示、下線付きテキストがあります。ディスプレイ属性は、ディスプレイが次の操作を実行するかどうかを指定する属性です。

- 境界を表示するかどうか (境界にはラベルを付けることができる)
- キャリッジ制御文字をエコー表示することができるかどうか (フォーム・フィード、垂直タブなど)
- テキストがディスプレイの右端の位置をこえたときに、ユーザに対してダイヤ形のアイコンを表示するかどうか

仮想ディスプレイを指定するときに指定したビデオ属性とディスプレイ属性は、後で変更できます。また、SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンを使用すれば、ビデオ属性とディスプレイ属性をどちらも変更できます。

たとえば、SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンを使用すれば、仮想ディスプレイのサイズを変更できます。仮想ディスプレイのサイズを変更する場合には、そのディスプレイ内のデータは、サイズを変更した後のディスプレイにコピーされます。つまり、現在の内容は (1 行目、1 カラム目から順に) サイズを変更した新しいディスプレイに格納できるだけ保存されます。

仮想ディスプレイを削除するには、SMG\$DELETE\_VIRTUAL\_DISPLAY ルーチンを使用します。削除操作についての詳しい説明は、第 2.1.5 項を参照してください。

---

## 1.3 ビューポート

仮想ディスプレイは非常に大きいものである可能性があるため、ディスプレイ全体を一度に画面に表示できないことがあります。このような場合には、仮想ディスプレイの各部分を表示するために、大きい仮想ディスプレイを再度ペーストしなければなりません。仮想ディスプレイに対応するビューポートはこの操作を容易にします。

ビューポート操作とは、仮想ディスプレイの各部分を表示するために、仮想ディスプレイを囲む長方形の表示領域を移動する操作です。ビューポートは仮想ディスプレイに対応づけられます。したがって、仮想ディスプレイに対して実行される出力操作はすべて、ビューポートに対して実行されます。ビューポートは作成、削除、ペースト、アンペースト、スクロール、および移動できます。ビューポートについての詳しい説明は、第 2.2.10 項を参照してください。

---

## 1.4 仮想キーボード

ペーストボードが出力操作のための論理構造であるのと同様に、仮想キーボードは入力操作のための論理構造です。仮想キーボードを使用すれば、装置からの独立性を実現できます。日本語 SMG 入力ルーチンを使用する場合には、使用している漢字ターミナルのタイプを考慮する必要がありません。たとえば、各プログラムは特定の漢字ターミナルでどの行終了文字が使用されているかを知る必要がありません。日本語 SMG は異なる行終了文字を統一された機能コードに変換します (行終了文字コードについての詳しい説明は、第 3 章を参照してください)。

仮想キーボードは通常、漢字ターミナルの物理キーボードに対応づけられます。複数の仮想キーボードを 1 つの入力装置に対して対応づけることができます。

---

### 注意

---

日本語 SMG は、RMS ファイルを入力装置としてサポートしません。

---

入力ソース (仮想キーボード) を設定するには、SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンを使用します。仮想キーボードを削除するには、SMG\$DELETE\_VIRTUAL\_KEYBOARD ルーチンを使用します。

仮想キーボードを作成した後、SMG\$READ\_COMPOSED\_LINE、SMG\$READ\_KEYSTROKE、SMG\$READ\_STRING、SMG\$READ\_VERIFY ルーチンのいずれかを使用することにより、仮想キーボードからデータを受け付けることができます。

SMG\$READ\_COMPOSED\_LINE ルーチンは、通常のキーストロークと、キーボードおよび制御キーに対応する定義済み文字列で構成される行を読み込みます。このルーチンは、単一キー・コマンド機能を提供することにより、コマンド向きユーティリティのためのインタフェースを容易にコーディングできるようにします。SMG\$READ\_KEYSTROKE ルーチンはキーボードから入力された 1 つのキーストロークを読み込むために使用します。SMG\$READ\_STRING ルーチンは文字と終了文字で構成される文字列を読み込みます。このルーチンは汎用性と柔軟性が高く、OpenVMS ターミナル・ドライバの多くの機能をアクセスできます。SMG\$READ\_VERIFY ルーチンは書式化された入力を読み込むために使用します。

どのタイプの読み込み操作も、SMG\$CANCEL\_INPUT ルーチンを呼び出すことにより強制終了できます。

## 1.5 日本語画面管理ライブラリの使用方法

日本語画面管理ライブラリ (JSY\$SMGSHR.EXE) をご使用の際には、リンク時に明示的に指定する必要があります。自動的に JSY\$SMGSHR.EXE とリンクされることはありません。JSY\$SMGSHR.EXE とリンクするには以下のようにします。

```
$ link foo.obj, sys$input/option  
sys$share:jsy$smgshr.exe/share  
^Z
```

---

## 日本語 SMG の出力操作

この章では、日本語 SMG が提供する出力操作について説明します。これらの出力操作について、構成操作 (漢字ターミナルの画面にイメージを作成する操作) と仮想ディスプレイを通じて実行される出力操作に分けて説明します。

---

### 2.1 構成操作

構成操作は、ペーストボードで仮想ディスプレイを操作したり、ターミナル画面にイメージを作成するために使用されるルーチンです。このような操作としては、仮想ディスプレイのペースト、仮想ディスプレイのアンペースト、仮想ディスプレイの再ペースト、仮想ディスプレイの移動、仮想ディスプレイのポップ、仮想ディスプレイが隠されているかどうかの確認、ペースト順序の一覧の作成などがあります。

### 2.1.1 ペースト操作

仮想ディスプレイは、ペーストボードにペーストされている間だけ、物理装置に表示されます。ディスプレイをペーストボードにペーストするには、SMG\$PASTE\_VIRTUAL\_DISPLAY ルーチンを呼び出し、仮想ディスプレイの原点として使用するペーストボード座標を指定します (原点は左上の角です)。ペーストボード自体には境界はありませんが、もちろん、物理画面には境界があります。したがって、ディスプレイ全体またはその一部がターミナル画面に表示されないような形式でも、ディスプレイをペーストボードにペーストできます。

仮想ディスプレイをペーストボードにペーストする操作は、仮想ディスプレイの1行目および1カラム目と一致するペーストボードの行とカラムを指定することにより、仮想ディスプレイの内容を画面上の適切な位置に変換する論理操作です。たとえば、6行の仮想ディスプレイ“A”をペーストボードの1～6行目にペーストし、6行の仮想ディスプレイ“B”をペーストボードの7～12行目にペーストすると、仮想ディスプレイ“B”は画面上で仮想ディスプレイ“A”のすぐ下に表示されます。図2-1を参照してください。

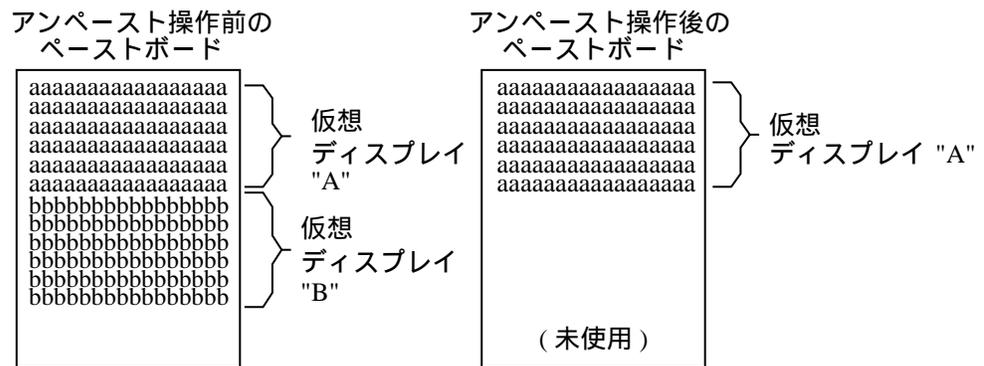
図 2-1 ペースト操作



## 2.1.2 アンペースト操作

SMG\$UNPASTE\_VIRTUAL\_DISPLAY ルーチンを使用すれば、仮想ディスプレイを物理画面から消去できます。第 2.1.1 項で示した例で仮想ディスプレイ“B”をアンペーストすると、結果は図 2-2 に示すとおりになります。

図 2-2 アンペースト操作



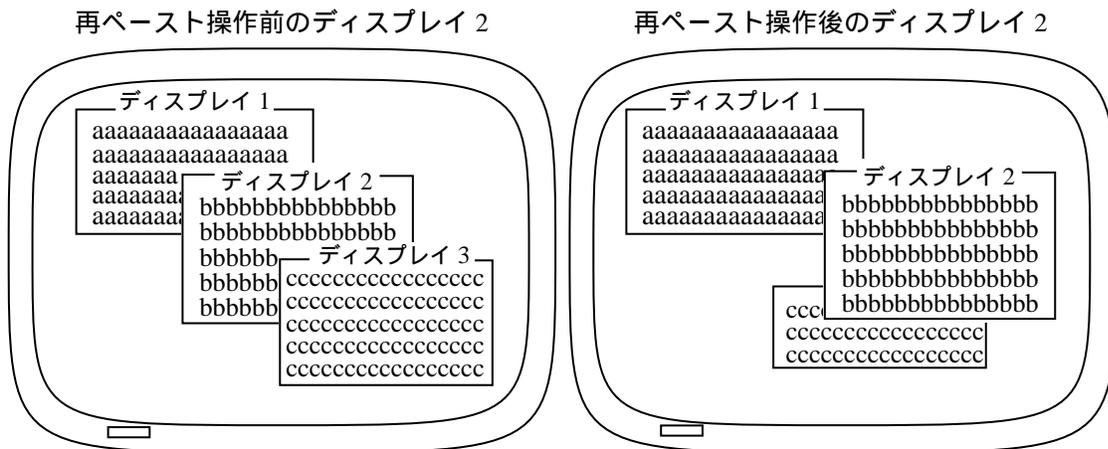
仮想ディスプレイをアンペーストしても、仮想ディスプレイが破壊されたり、その内容が破壊されるわけではありません。アンペーストは、単にディスプレイをペーストボードから削除するだけです。

ディスプレイは、そのサイズやペーストする位置、あるいはペーストする順序に応じて、部分的あるいは完全に重なり合うことがあります。一番上のディスプレイをアンペーストすると、その下に隠されていたディスプレイが表示されるようになります。

### 2.1.3 再ペースト操作

SMG\$REPASTE\_VIRTUAL\_DISPLAY ルーチンを使用すれば、ペーストボードで仮想ディスプレイを別の位置に移動できます。このルーチンを使用すれば、アンペースト操作と再ペースト操作の間に画面が空白になるのを防止できます。図 2-3 は 2 番目のディスプレイをさらに右に再ペーストした結果を示しています。ディスプレイ 2 は再ペーストされたため、ペースト順序が変化し、一番上に表示されています。この結果、再ペースト操作を実行する前は一番上に表示されていたディスプレイ 3 は、その一部が隠されるようになります。

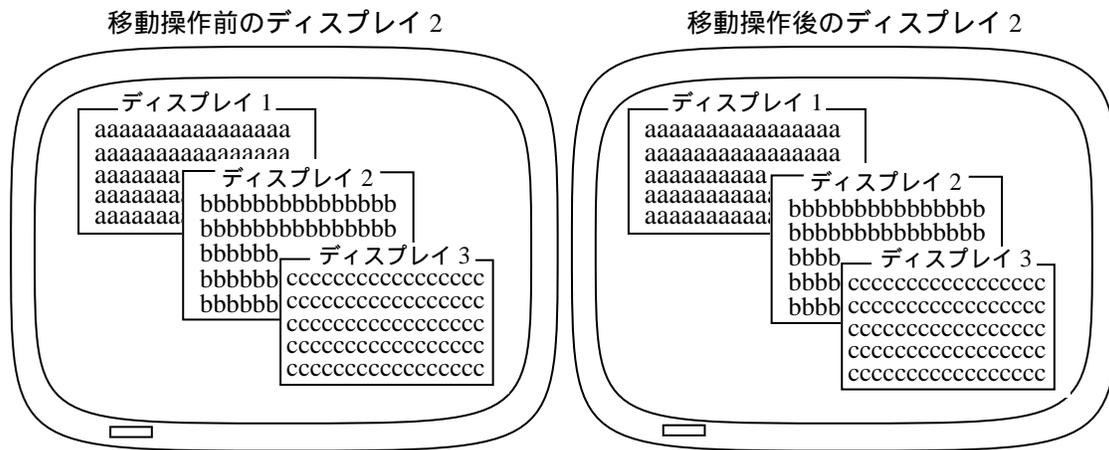
図 2-3 再ペースト操作



### 2.1.4 移動操作

SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチンを使用すれば、ペースト順序を変更せずに、ペーストボードで仮想ディスプレイを移動できます。図 2-4 は 2 番目のディスプレイを右に移動した結果を示しています。アンペースト操作と移動操作の相違点に注意してください。移動操作では、ペースト順序は変化しません。したがって、ディスプレイ 2 はディスプレイ 3 に部分的に隠された状態のままです。

図 2-4 移動操作



SMG\$MOVE\_TEXT ルーチンを使用すれば、1 つの仮想ディスプレイから別の仮想ディスプレイにテキストを移動できます。長方形の対角線上の 2 つの点が与えられると、SMG\$MOVE\_TEXT ルーチンは適切な幅と高さを判断します。最初の仮想ディスプレイの属性はそのまま移動されます。長方形領域のテキストが移動した後で、その領域は最初の仮想ディスプレイから消去されます。

また、この操作によって、文字が分割されてしまうような場合 (例えば、全角文字が 1 バイトずつに分割した場合など) には、分割された文字は未定義文字となります。

### 2.1.5 削除操作とポップ操作

これまで説明したアンペースト操作、再ペースト操作、および移動操作は仮想ディスプレイを消失させませんでした。SMG\$DELETE\_VIRTUAL\_DISPLAY ルーチンを使用すれば、仮想ディスプレイを削除できます。また SMG\$POP\_VIRTUAL\_DISPLAY ルーチンを使用すれば、1 回の操作で 1 つのペーストボードから多くの仮想ディスプレイを削除できます。このルーチンは指定された仮想ディスプレイと、そ

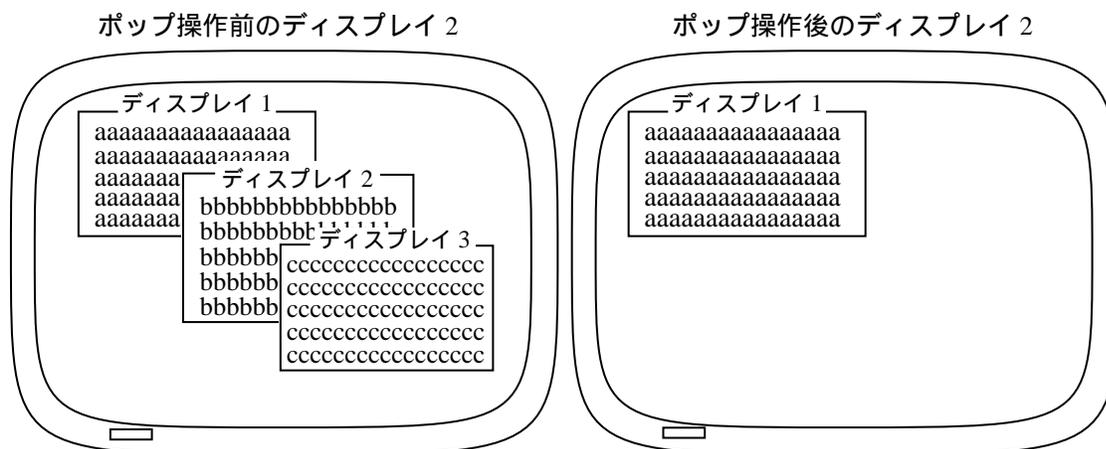
## 日本語 SMG の出力操作

### 2.1 構成操作

の仮想ディスプレイより後でペーストされた他のすべての仮想ディスプレイをアンペーストし、削除します。

ポップ操作はモジュール環境で役立ちます。たとえば、サブルーチン呼び出し、出力を作成する対象となるペーストボードの `pasteboard-id` 引数だけを渡すことができます。その後、そのサブルーチンはさらに仮想ディスプレイを作成し、それらの仮想ディスプレイを指定されたペーストボードにペーストできます。サブルーチンは制御を呼び出し側に戻すときに、ペーストした最初の仮想ディスプレイの `display-id` 引数を戻します。その後、呼び出しプログラムは `SMG$POP_VIRTUAL_DISPLAY` ルーチン呼び出し、サブルーチンから戻された仮想ディスプレイの `display-id` 引数を指定することにより、サブルーチンの処理を取り消すことができます。この方法を使用すれば、呼び出しプログラムとサブルーチンの間で渡さなければならない情報の量を最低限に抑えることができます。図 2-5 はディスプレイ 2 をポップした結果を示しています。

図 2-5 ポップ操作

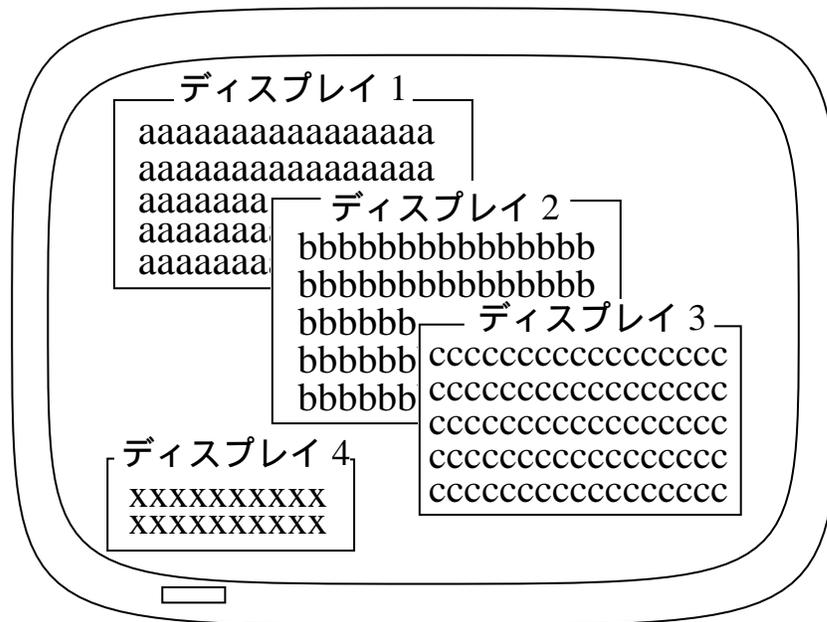


#### 2.1.6 重なり合いの確認操作

特定のペーストボードにペーストされたディスプレイが重なっているかどうかを判断しなければならないことがあります。このような場合には、`SMG$CHECK_FOR_OCCLUSION` ルーチンを使用します。たとえば、図 2-6 に示した構成では、ディスプレイ 1 とディスプレイ 2 は重なっていると報告され、ディスプレイ 3 とディスプレイ 4 は重なっていないと報告されます。しかし、この方法を使用して、どのディスプレイがペーストボードの一番上にペーストされているかを判断することはできません。この判定では、ペーストされているディスプレイが重なっているかどうかだけしか判断できません。

複数の仮想ディスプレイをペーストボードにペーストした場合には、SMG\$LIST\_PASTING\_ORDER ルーチンを使用することにより、仮想ディスプレイをペーストした順序を判断できます。このルーチンは、最初にペーストされた（一番下の）仮想ディスプレイの識別子を戻します。その後、ペーストされたすべての仮想ディスプレイの識別子が戻されるまで、SMG\$LIST\_PASTING\_ORDER ルーチンを繰り返し呼び出すことができます。

図 2-6 重なり合いの確認



## 2.2 仮想ディスプレイによる出力

この節では、仮想ディスプレイを通じて出力を実行するために使用する日本語 SMG ルーチンについて説明します。

仮想ディスプレイに書き込む操作は、漢字ターミナルに直接書き込む操作によく似ています。しかし、仮想ディスプレイに書き込む操作は、すべて日本語 SMG ルーチンを呼び出すことにより実行されます。画面の消去、カーソルの位置の設定、出力テキストのスクロールは、日本語 SMG が提供する典型的な機能です。テキストは仮想ディスプレイのバッファに並べられます。したがって、出力を受け付ける前に、ディスプレイをペーストしておく必要はありません。物理画面に書き込む場合には、画面の物理境界に制限されます。同様に、日本語 SMG 出力操作は仮想ディスプレイの境界に制限されます。仮想ディスプレイの最後のカラムをこえてテキストを書き込むことはできません。

## 日本語 SMG の出力操作 2.2 仮想ディスプレイによる出力

ペーストボードの中で画面に表示されている部分に仮想ディスプレイがペーストされていない限り、仮想ディスプレイを変更しても、その変更結果は画面に表示されません。仮想ディスプレイがペーストされていない場合や、表示されていない位置にペーストされている場合には、このような変更結果は仮想ディスプレイを表現する内部データベースにのみ反映されます。

### 2.2.1 カーソルの位置

仮想ディスプレイを最初に作成した状態では、仮想カーソルは仮想ディスプレイの1行目、1カラム目に配置されます。仮想ディスプレイに対する様々な出力操作は、通常の出力操作が物理ターミナルでカーソルを移動するのと同様に、仮想カーソルを移動します。

仮想ディスプレイ内での仮想カーソルの位置と、画面上の物理カーソルの位置を混同しないでください。1つのペーストボードに複数の仮想ディスプレイをペーストできるため、物理画面には同時に複数の仮想ディスプレイが表示される可能性があります。各仮想ディスプレイにはそれぞれ対応する仮想カーソルの位置がありますが、これらのすべてのディスプレイの仮想カーソルのうち、1つの仮想カーソルの位置だけが物理カーソルに対応します。これは通常、最後に変更された仮想ディスプレイのカーソルの位置です。

仮想ディスプレイ内での仮想カーソルの現在の位置を判断するには、SMG\$RETURN\_CURSOR\_POS ルーチンを使用します。このルーチンは現在の仮想カーソルの行とカラムを戻します。

プログラミングを容易にするために、この情報は SMG\$CURSOR\_ROW ルーチンと SMG\$CURSOR\_COLUMN ルーチンを使用して別々に検索することもできます。これらのルーチンは関数として動作します。これらの2つのルーチンを使用すれば、次のような構造を簡単にコーディングできます。

```
IF SMG$CURSOR_ROW ( Display-id ) > Max-row
THEN
  BEGIN
  .
  .
  .
  END
```

SMG\$RETURN\_CURSOR\_POS ルーチンを使用してこの情報を検索する場合には、次のように指定しなければなりません。

```
CALL SMG$RETURN_CURSOR_POS ( Display-id, Cursor-row, Cursor-column )
IF Cursor-row > Max-row
THEN
  BEGIN
```

・  
・  
・

END

SMG\$RETURN\_CURSOR\_POS ルーチンを使用する場合には、*cursor-row* と *cursor-column* という 2 つのローカル変数を定義しなければなりません。これらのローカル変数は、このテストを実行する場合にだけしか必要ありませんが、このルーチンを使用すれば、1 回のルーチン呼び出しで行とカラムをどちらも検索できます。

次の 3 つのルーチンを使用すれば、仮想ディスプレイ内での仮想カーソルの位置を設定できます。

- SMG\$SET\_CURSOR\_ABS  
仮想ディスプレイ内の指定された位置に仮想カーソルを設定する。
- SMG\$SET\_CURSOR\_REL  
現在のディスプレイ・カーソルの位置から指定された距離に、仮想カーソルの位置を設定する。
- SMG\$HOME\_CURSOR  
仮想ディスプレイの指定された角に仮想カーソルを設定する。

## 2.2.2 削除操作

仮想ディスプレイの一部を削除するために、次に示す 2 つのルーチンが準備されています。

- SMG\$DELETE\_CHARS  
1 行内の文字を、1 文字以上削除します。このルーチンによって削除された文字があった位置は、同じ行で、削除された文字の右側にあった文字によって置換されます。行の最後の部分の空になった文字位置には空白が挿入されます。  
DEC 漢字または Super DEC 漢字文字集合を使用している際に、全角文字の右半分から削除操作が始まる場合は、その文字の左半分は未定義文字になります。また、全角文字の左半分で削除操作が終了する場合には、その文字の右半分は未定義文字になります。
- SMG\$DELETE\_LINE  
1 行以上を行単位で削除します。このルーチンによって削除された行は、削除された行のすぐ下の行によって置換されます。仮想ディスプレイの一番下の空になった行は空白になります。

### 2.2.3 消去操作

消去操作では、仮想ディスプレイ内の消去された部分は空白になります。仮想ディスプレイの他の部分は並べ換えられません。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、消去操作が全角文字の右半分から始まる場合は、その文字の右半分は消去されて、左半分は未定義文字になります。また消去操作が全角文字の左半分で終了する場合には、その文字の右半分は未定義文字になります。

仮想ディスプレイの一部を消去するために、次の 4 つのルーチンが提供されます。

- SMG\$ERASE\_CHARS  
指定されたカラム数だけ指定された行の中から消去します。
- SMG\$ERASE\_COLUMN  
1 列内の文字を、指定された開始位置から列の最後まで、または指定された終了位置まで消去します。
- SMG\$ERASE\_LINE  
1 行内の文字を、指定された開始位置から行の最後まで消去します。
- SMG\$ERASE\_DISPLAY  
仮想ディスプレイの一部または全部を消去します。

### 2.2.4 挿入操作

仮想ディスプレイにテキストを挿入するために、2 つのルーチンが準備されています。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、全角文字の右半分から挿入操作が始まり、その文字を分割してしまう場合には、分割された文字は、2 つの未定義文字になります。全角文字の右半分から切り捨て操作が始まる場合には、その文字の左半分が未定義文字になります。

- SMG\$INSERT\_CHARS  
指定された文字列を指定された開始位置に挿入します。これらの位置の既存の文字は右に移動し、各文字を挿入するための空間が確保されます。右端のカラムをこえて移動された文字は切り捨てられます。
- SMG\$INSERT\_LINE  
指定されたテキスト行を指定された位置に挿入し、行を挿入するための空間を確保するために、仮想ディスプレイ内の既存の行を上または下にスクロールします。仮想ディスプレイの上端より上、または下端より下にスクロールされた行は切り捨てられます。自動改行が設定されていない場合で、挿入行が仮想ディスプレイの 1 行以内に収まらない場合には、あふれた文字は切り捨てられます。

## 2.2.5 書き込み操作

日本語 SMG は、仮想ディスプレイにテキストを書き込むために 2 種類のルーチンを提供します。1 つは文字単位の出カルーチンで、もう 1 つは行単位の出カルーチンです。この後の節では、これらの各ルーチンについて説明します。

### 2.2.5.1 文字単位の出力

仮想ディスプレイを直接アクセス装置として使用する場合には、通常、文字単位の出カルーチンを使用します。この操作モードでは、プログラムは仮想ディスプレイ内でカーソルを明示的に設定し、そこにテキストを置きます。次の出力操作は通常、前の出力操作と空間的な関係がないため、カーソルの位置とディスプレイのスクロールを明示的に制御する必要があります。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、全角文字の右半分から出力が始まる場合は、その文字の左半分は未定義文字になります。また、全角文字の左半分で出力が終了する場合は、その文字の右半分は未定義文字になります。

文字単位の出カルーチンとしては、次に示すルーチンがあります。

- SMG\$PUT\_CHARS  
通常の文字を仮想ディスプレイに書き込みます。
- SMG\$PUT\_CHARS\_WIDE  
倍角の文字を仮想ディスプレイに書き込みます。
- SMG\$PUT\_CHARS\_HIGHWIDE  
4 倍角の文字を仮想ディスプレイに書き込みます。
- SMG\$PUT\_CHARS\_MULTI  
複数の属性で、文字を仮想ディスプレイに書き込みます。

### 2.2.5.2 行単位の出力

文字単位の出カルーチンと異なり、行単位のルーチンは漢字ターミナルを順次アクセス装置として取り扱います。この操作モードでは、プログラムは通常、情報を 1 行ずつ順に書き込みます。概念的には、この動作は情報ストリーム (たとえばファイル) を仮想ディスプレイにコピーする操作に対応します。各ルーチン呼び出しは、操作が正常終了した後、カーソルを次の行の 1 カラム目に設定します。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、出力が全角文字の右半分から始まる場合には、その文字の左半分は未定義文字になります。

行単位の出カルーチンとしては、次に示すルーチンがあります。

- SMG\$PUT\_LINE  
テキスト行を仮想ディスプレイに書き込みます。
- SMG\$PUT\_LINE\_WIDE

倍角のテキスト行を仮想ディスプレイに書き込みます。

- SMG\$PUT\_LINE\_HIGHWIDE

4 倍角のテキスト行を仮想ディスプレイに書き込みます。

- SMG\$PUT\_LINE\_MULTI

複数の属性でテキスト行を仮想ディスプレイに書き込みます。

## 2.2.6 仮想ディスプレイの属性の変更

SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンを使用して仮想ディスプレイを作成する場合には、仮想ディスプレイ内に表示されるすべてのテキストに対して省略時の属性を指定します。既存の仮想ディスプレイの属性は、SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンを呼び出すことにより変更できます。

SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンを使用すれば、ディスプレイ全体のビデオ属性とディスプレイ属性の両方を変更できます。

仮想ディスプレイに対して省略時の属性を指定するには、ビット・マスクを使用してディスプレイ属性引数にビットを設定します。次の各ビットを設定できます。

表 2-1 ビットの設定

SMG\$M_BLINK	点滅文字を指定します。
SMG\$M_BOLD	通常より明るい輝度の文字を指定します。
SMG\$M_REVERSE	文字を反転表示することを指定します。つまり、仮想ディスプレイの現在の属性の反対の状態では、文字を表示しません。
SMG\$M_UNDERLINE	下線付き文字を指定します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、文字は仮想ディスプレイ内に存在しますが、ペーストボードには表示されません。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を指定します。

SMG\$M\_USER1 から SMG\$M\_USER8 のユーザ定義属性を使用する場合には、STRING\_2 機能を使用して、TERMTABLE.TXT ファイルに適切な定義を指定しなければなりません。TERMTABLE 定義と STRING\_2 機能については第 5 章を参照してください。

これまで示した属性の一部またはすべてを、仮想ディスプレイの属性として指定できます。複数のビデオ属性を指定する場合には、これらの属性の論理和を使用します。たとえば、仮想ディスプレイの省略時の設定として下線付き文字を反転表示することを指定する場合には、適切なビット・マスクの論理和を display-attributes 引数に割り当てます。

```
Display_attributes = ( SMG$M_REVERSE OR SMG$M_UNDERLINE )
```

この後、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチン呼び出すとき、この display-attributes 引数を指定します。

日本語 SMG 出力ルーチンを使用すれば、省略時の属性を変更できます。このようにすれば、他の属性でテキストを書き込むときに、そのたびに省略時の設定を変更する必要がなくなります。省略時の属性を変更するには、2つの引数を使用します。それは rendition-set 引数と rendition-complement 引数です。これらの引数にビデオ属性を設定する方法は、仮想ディスプレイを作成するときにビデオ属性を設定する方法と同じです。

省略時のビデオ属性、rendition-set 引数、および rendition-complement 引数を組み合わせることにより、次の方法に従って、出力属性を指定できます。

1. 省略時のビデオ属性と rendition-set 引数を格納したマスクに対して、論理和またはビット単位の論理和演算が実行されます。
2. 前の論理和演算の結果と rendition-complement 引数に対して、排他的論理和演算またはビット単位の排他的論理和演算が実行されます。

この方法の結果は次の表に示すとおりです。

set	complement	動作
0	0	属性を省略時の値に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の値の補数に設定する。
1	1	属性をオフに設定する。

この方法の結果は、仮想ディスプレイの現在の属性ではなく、省略時の属性設定に対して決まります。したがって、属性を明示的に指定する日本語 SMG 出力ルーチンをすでに使用している場合には、現在の属性はその仮想ディスプレイの省略時の属性と一致しない可能性があります。

### 2.2.7 ラインとライン描画文字を描く操作と削除する操作

水平線と垂直線を作成するための簡単な方法として、3つのルーチンが準備されています。

- SMG\$DRAW\_LINE ルーチンは指定されたラインの端点をもとに、水平線または垂直線を作成します。
- SMG\$DRAW\_RECTANGLE ルーチンは、指定された左上の角と右下の角の位置をもとに、長方形を描きます。
- SMG\$DRAW\_CHAR ルーチンは1文字のライン描画文字を描きます。

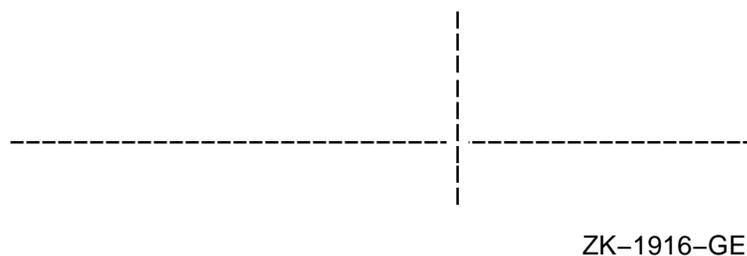
## 日本語 SMG の出力操作 2.2 仮想ディスプレイによる出力

SMG\$DRAW\_LINE ルーチンまたは SMG\$DRAW\_RECTANGLE ルーチンを使用して描いたラインを消去する場合には、SMG\$REMOVE\_LINE ルーチンを使用します。このルーチンはラインを削除しますが、ラインの交点にあるライン描画文字は保存されます。

他のすべての日本語 SMG ルーチンと同様に、これらのルーチンも装置から独立しています。ラインを VT 100 で描く場合には、VT 100 の曲線描画文字設定が使用されます。同じラインを VT 52 で描く場合には (VT 52 はこのハードウェア機能を備えていません)、ラインはプラス記号(+), 縦線(|), ダッシュ(-)を使用することにより自動的に近似されます。ユーザ・プログラムが各ターミナル・タイプに対して異なる文字コードを提供する必要はありません。

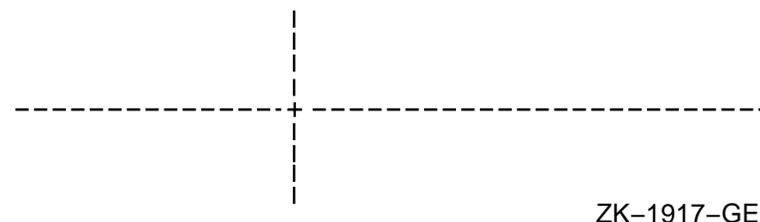
さらに、これらのルーチンは2本のラインの交点に、適切な文字を自動的に描きます。たとえば、プログラムが画面に直接水平線を書き込み、その後、その水平線と交差する垂直線を書き込む場合には、通常、図 2-7 のようになります。

図 2-7 SMG\$DRAW\_LINE ルーチンを使用せずに描いたライン



SMG\$DRAW\_LINE ルーチンを使用して、これらの同じラインを描いた場合には、画面は図 2-8 に示すようになります。

図 2-8 SMG\$DRAW\_LINE ルーチンを使用して描いたライン



### 2.2.8 外部テキストの表示

外部テキストを仮想ディスプレイまたはターミナルに表示する、2つのルーチンがあります。

- SMG\$PUT\_HELP\_TEXT ルーチンは、仮想ディスプレイに指定されたトピックのヘルプ・テキストを出力します。
- SMG\$PUT\_STATUS\_LINE ルーチンは、テキスト行をターミナルのハードウェア・ステータス行に出力します。一部のターミナルでは、ハードウェア・ステータス行がスクリーンの一番下 (25 行目) にあります。この行が「ホスト制御」に設定されていれば、このルーチンを使って 1 行のテキストを出力することができます。

## 2.2.9 仮想ディスプレイからの読み込み

SMG\$READ\_FROM\_DISPLAY ルーチンを使用すれば、仮想ディスプレイからテキストを簡単に読み込むことができます。このルーチンは、仮想ディスプレイを使用することにより、画面にメニュー項目を表示するアプリケーションで使用できます。このようなアプリケーションでは、ユーザはメニュー項目間でカーソルを移動し、その後、適切な項目を選択できます (たとえば、`[Return]`を押すことにより)。この時点で、プログラムはディスプレイの現在のカーソルの位置から文字を読み込み、どのメニュー項目が選択されたかを判断できます。

このルーチンはまた、SMG\$M\_INVISIBLE 属性を使用して書き込まれた文字を読み込むための方法としても使用できます。

## 2.2.10 ビューポート

仮想ディスプレイは非常に大きくなる可能性があるため、ディスプレイ全体を一度に画面に表示できるとは限りません。ユーザは大きな仮想ディスプレイの各部分を表示するために、仮想ディスプレイを再ペーストしなければなりません。仮想ディスプレイに対応づけられたビューポートはこの操作を容易にします。

ビューポート操作とは、仮想ディスプレイの各部分を表示するために、仮想ディスプレイ上で長方形の表示領域を移動する処理のことです。ビューポートは仮想ディスプレイに対応づけられるため、仮想ディスプレイに対して実行され、出力操作はすべて、ビューポート上で反映されます。

### 2.2.10.1 ビューポートの作成

SMG\$CREATE\_VIEWPORT ルーチンは、特定の仮想ディスプレイに対応するビューポートを作成します。仮想ディスプレイは、ビューポートを作成する前に作成しておかなければならず、各仮想ディスプレイに対してビューポートを 1 つだけ作成できます。

ビューポートを表示するには、SMG\$PASTE\_VIRTUAL\_DISPLAY ルーチンを呼び出すことにより、仮想ディスプレイをペーストしなければなりません。仮想ディスプレイの中で、ビューポートの内部に存在する部分だけが表示されます。

### 2.2.10.2 ビューポートの削除

ビューポートは `SMG$DELETE_VIEWPORT` ルーチンを使用して削除できます。このルーチン呼び出すと、ビューポートはペーストされているペーストボードから自動的にアンペーストされます。しかし、ビューポートに対応づけられている仮想ディスプレイは削除されません。したがって、`SMG$PASTE_VIRTUAL_DISPLAY` ルーチン呼び出すことにより、その仮想ディスプレイを表示できます。

### 2.2.10.3 ビューポートのペーストとアンペースト

`SMG$PASTE_VIRTUAL_DISPLAY` ルーチンは、ビューポートまたは仮想ディスプレイをペーストボードにペーストします。ビューポートを仮想ディスプレイに対応づけた後、`SMG$PASTE_VIRTUAL_DISPLAY` ルーチン呼び出す場合には、仮想ディスプレイではなく、ビューポートが使用されます。つまり、仮想ディスプレイに対するビューポートを作成した後、その仮想ディスプレイの中で表示できる部分は、ビューポートの内部に含まれる長方形の領域だけです。ビューポートを削除せずに、そのビューポートをアンペーストする場合には、`SMG$UNPASTE_VIRTUAL_DISPLAY` ルーチンを使用します。

対応する仮想ディスプレイがすでにペーストされているときに、ビューポートを作成する場合には、そのビューポートは表示されません。`SMG$PASTE_VIRTUAL_DISPLAY` ルーチン呼び出すと、仮想ディスプレイはアンペーストされ、その場所にビューポートがペーストされます。

### 2.2.10.4 ビューポートのスクロールと移動

仮想ディスプレイに対応するビューポートは、ペーストボードに全体、またはある一部分をペーストできます。あるいはペーストボードの完全な外に置くこともできます。ただし、ビューポートは対応づけられた仮想ディスプレイを越えて拡大することはできません。仮想ディスプレイの境界をこえてビューポートを拡張しようとした場合には、日本語 SMG は仮想ディスプレイの内部に収まるように、自動的にビューポートを切り詰めます。

ビューポートを“スクロール”するには、そのビューポートに対応する仮想ディスプレイをスクロールします。この場合には、`SMG$SCROLL_VIEWPORT` ルーチン呼び出します。実際には、スクロールをシミュレートするために、ビューポートが仮想ディスプレイ上で移動するたびに、ビューポートの座標が変化します。しかし、画面上でのビューポートの位置は変化しません。`SMG$SCROLL_VIEWPORT` ルーチンを使用する場合には、スクロールの方向(上, 下, 左, 右)を指定できます。

ビューポートを移動する場合には、`SMG$CHANGE_VIEWPORT` ルーチン呼び出します。このルーチンを使用する場合には、ビューポートの新しい開始位置とサイズを指定できます。ビューポートの開始位置とサイズを変更すれば、仮想ディスプレイ上でウィンドウを移動できます。

#### 2.2.10.5 ビューポート属性の変更

SMG\$GET\_VIEWPORT\_CHAR ルーチンを使用すれば、ビューポートの現在の属性を検索できます。ビューポートの属性は、ビューポートの開始行と終了行、および開始カラムと終了カラムで構成されます。このルーチンは SMG\$CHANGE\_VIEWPORT ルーチンと組み合わせて使用でき、既存のビューポートの開始位置と終了位置を変更できます。

開始位置または終了位置以外のビューポート属性を変更する場合には、SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンを使用しなければなりません。仮想ディスプレイへのすべての変更は、対応するビューポートで反映されます。

たとえば、仮想ディスプレイに境界がある場合には、対応するビューポートにも境界があります。仮想ディスプレイに境界がない場合には、ビューポートにも境界がありません。ビューポートに境界を追加したり、削除する場合には、SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンを使用して仮想ディスプレイに境界を追加するか、または仮想ディスプレイから境界を削除します。この変更操作はビューポートで自動的に反映されます。

#### 2.2.11 メニュー

日本語 SMG には、メニューに選択項目を作成する機能とメニューから項目を選択する機能があります。メニューはブロック・メニュー、垂直メニュー、水平メニューのいずれでもかまいません。ブロック・メニューは項目を 2 次元に並べた配列であり、おもに使用されるメニュー・タイプです。垂直メニューはメニュー項目を 1 列に縦に並べて表示し、水平メニューはメニュー項目を 1 行に横に並べて表示します。ビューポートの境界内に収まらないメニュー項目は、表示領域にスクロールされるまで表示されません。

##### 2.2.11.1 メニューの作成

SMG\$CREATE\_MENU ルーチンは、指定された仮想ディスプレイのスクロール領域にメニューを作成します (省略時の設定では、スクロール領域は仮想ディスプレイ全体です。SMG\$SET\_DISPLAY\_SCROLL\_REGION ルーチンを使用すれば、スクロール領域を変更できます)。メニューを作成するときに、メニューの形式 (ブロック、垂直、水平のいずれか) を指定します。

ブロック・メニューはメニューの省略時の形式です。メニュー内の項目は一定の文字列配列の形式でルーチンに渡されます。省略時の設定では、各メニュー項目はシングル・スペースで配置されますが、ダブル・スペースも要求できます。水平方向には、各メニュー項目は 4 つのスペースで区切られます。さらに、メニュー項目を固定形式カラムに表示することも要求できます。この場合には、カラムの幅は渡される固定長文字列のサイズに等しくなります。

各仮想ディスプレイには、1つのメニューだけを格納できます。また、SMG\$CREATE\_MENU ルーチン呼び出し後、ディスプレイ内のメニューを格納した領域に、他の文字を出力することはできません。このような操作を実行すると、予測できない結果になります。メニューは仮想ディスプレイのスクロール領域に出力されます。

#### 2.2.11.2 メニューの削除

メニューを削除するには、SMG\$DELETE\_MENU ルーチンを使用します。このルーチンは、指定された仮想ディスプレイ内のメニュー項目に対するアクセスを禁止します。さらに、SMG\$DELETE\_MENU ルーチンがメニューを削除するときに、すべてのメニュー項目をディスプレイから削除することも要求できます。

#### 2.2.11.3 メニューからの選択

メニューを作成した後、SMG\$SELECT\_FROM\_MENU ルーチンを使用してそのメニューから項目を選択できます。1つのメニュー項目から別のメニュー項目に移動すると、省略時の設定では、現在選択されている項目が反転表示されます。SMG\$SELECT\_FROM\_MENU ルーチン呼び出し時に、省略時の選択項目を指定できます。省略時の選択項目を指定しなかった場合には、前に選択した項目が強調表示されたままになります。

SMG\$SELECT\_FROM\_MENU ルーチンには、3種類の操作モードがあります。これらのモードは flags 引数を使用して切り換えることができます。各モードについては、この後の項で説明します。

##### 2.2.11.3.1 省略時のモード

SMG\$SELECT\_FROM\_MENU ルーチンの省略時の操作モードは、flags 引数を省略することにより有効になります。このモードでは、矢印キーを使用して各メニュー項目間を移動でき、項目を選択した後、別の項目を追加選択できます。また、省略時のモードでは、すでに選択されている項目を“再選択”できます。

##### 2.2.11.3.2 RETURN\_IMMED モード

SMG\$SELECT\_FROM\_MENU ルーチンの flags 引数の値として、SMG\$M\_RETURN\_IMMED を指定した場合には、矢印キーを使用してメニュー項目間を移動できます。しかし、他のキーを押すと、制御はユーザに戻されます。Ctrl/Zは現在の項目を選択し、SMG\$\_EOF を戻します。他のキーを押した場合には、現在の項目が選択されます。

省略時のモードが提供するキー定義以外のキー定義が必要な場合には、SMG\$M\_RETURN\_IMMED モードを使用します。

##### 2.2.11.3.3 REMOVE\_ITEM モード

SMG\$SELECT\_FROM\_MENU の flags 引数の値として、SMG\$M\_REMOVE\_ITEM を指定した場合には、メニュー内の項目を“再選択”することはできません。ただし、項目はメニュー内に残されます。また、その項目はメニューを格納した仮想ディスプレイの省略時の属性で表示されます。

このモードで省略時の選択項目を指定し、その項目がすでに選択されている場合には、メニュー内の最初の“選択可能な”項目が強調表示されます。どの項目も選択可能でない場合には、エラーが戻されます。

### 2.2.12 仮想ディスプレイの保存

SMG\$SAVE\_VIRTUAL\_DISPLAY ルーチンは、仮想ディスプレイの内容をファイルに保存します。仮想ディスプレイを再作成するのに必要なテキスト、ビデオ属性、およびすべてのディスプレイ属性が保存されますが、メニュー、ビューポート、サブプロセスの内容は保存されません。

SMG\$SAVE\_VIRTUAL\_DISPLAY ルーチンによって保存したファイルをプリントすることはできません。仮想ディスプレイを復元するには、SMG\$LOAD\_VIRTUAL\_DISPLAY ルーチンを使用します。このルーチンは新しい仮想ディスプレイを作成し、その仮想ディスプレイに保存されているディスプレイの内容をロードします。新しい仮想ディスプレイはどのペーストボードにもペーストされない状態です。

### 2.2.13 ターミナル属性の変更

SMG\$SET\_TERM\_CHARACTERISTICS ルーチンは、指定されたペーストボードのターミナル属性を変更または検索します。このルーチンを使用すれば、1回のルーチン呼び出しで複数のターミナル属性を制御できます。

### 2.2.14 仮想ディスプレイに対する文字集合の設定

SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンは、新規に作成された仮想ディスプレイに文字集合を設定するために、省略可能な引数 (character-set) を持っています。この引数を省略した場合は、SMG\$DEFAULT\_CHARACTER\_SET 論理名が用いられます。

表 2-2 は、この論理名に定義できる等価名を示しています。

表 2-2 省略時文字集合名

等価名	文字集合 (コードセット)
ASCII	ASCII 文字集合
SPEC_GRAPHICS	DEC 特殊文字集合
SUPPLEMENTAL	補助文字集合 <sup>1</sup>
KANJI	DEC 漢字文字集合
JIS_KANA	JIS カタカナ文字集合
SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

この論理名が無効な文字集合を指していたり、定義されていなかった場合には、文字集合は未定義になります。

---

## 2.3 操作制御

この節では、特殊な操作モードを制御する日本語 SMG ルーチン (更新処理におけるタブの使用の制御、最低限の更新、およびバッファ処理を行なうルーチン) について説明します。これらのモードを使用すれば、情報を実際に画面に書き込む方法を最適化できます。これらのモードを起動するには、`SMG$CONTROL_MODE` ルーチンを使用します。

通常、ユーザがこれらのモードを考慮する必要はありません。日本語 SMG は、文字が画面にただちに表示されるように出力を最適化します。しかし、一部のアプリケーションでは、これらのモード設定を利用しなければならないことがあります。この後の節では、これらの操作モードについて説明します。

### 2.3.1 最小限の更新

省略時の設定では、日本語 SMG は画面の中で変更された部分だけを再度書き直すことにより、画面に書き込む文字数をできるだけ少なくしようとします。しかし、日本語 SMG はこの方法とは異なる更新操作もサポートし、その操作では、変更によって影響を受けるすべての行が、最初に変更された文字から行の最後まで、すべて書き直されます。

### 2.3.2 バッファ操作

省略時の設定では、出力操作を実行すると、バッファが満杯になったときに画面を更新するのではなく、部分的に格納された多くの小さいバッファを漢字ターミナルに送信することにより、画面がただちに更新されます。これらの入出力トランザクションの数は、バッファ処理モードを有効に設定することにより最低限に抑えることができ、その結果、プログラムの実行速度を向上することができます。

バッファ処理モードでは、日本語 SMG はバッファが満杯になった場合にだけ、ターミナル・バッファを画面に書き込みます。したがって、結果が画面に表示されるまでに、複数の出力操作が実行される可能性があります。多くのアプリケーションではこのような遅延時間が認められないため、バッファ処理と組み合わせて使用するようになり、`SMG$FLUSH_BUFFER` という特殊なルーチンが準備されています。

`SMG$FLUSH_BUFFER` ルーチンは、そのバッファが満杯になったかどうかとは無関係に、バッファを漢字ターミナルに強制的に書き込みます。このルーチンは、通常はこのような出力の遅延を認めることができるものの、場合によっては画面をただちに更新しなければならないようなアプリケーションにとって役立ちます。常に画面を

即時変更することが必要なアプリケーションの場合には、バッファ処理を無効に設定しておかなければなりません。

### 2.3.3 タブ

タブは更新操作をできるだけ少なくするために使用します。タブを使用する場合には、タブ・ストップが HP 指定の位置に設定されているかどうかを確認しなければなりません。ユーザが漢字ターミナルにどのようにタブを設定したかということとは無関係に、アプリケーションを実行したい場合には、タブを使用すべきではありません。

画面に出力されるタブはすべて、実際に画面に出力される前に、日本語 SMG ルーチンによって 8 つのスペースに変換されます。この規則には 1 つだけ例外があります。それは、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンを使用するとき、display-attributes 引数を SMG\$M\_DISPLAY\_CONTROLS ルーチンに設定した場合です。この場合には、タブ文字を 8 つのスペースとして解釈するのではなく、タブ文字がそのまま出力されます。

---

## 2.4 出力のバッチ処理

複雑なペーストボード・イメージを作成するときに、イメージ全体が完成した後でそのイメージを画面に表示したいことがあります。このような場合には、イメージを作成する途中では画面を空白のままにしておき、一連の作成操作をバッチ処理し、最終結果だけを画面に表示できます。

日本語 SMG には、ペーストボード・レベルで一連の操作をバッチ処理するための方法があります。この方法については、この後の節で説明します。

### 2.4.1 ペーストボード更新のバッチ処理

ペーストボード更新のバッチ処理は、SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチンを呼び出し、複数の作成操作を実行し、最後に SMG\$END\_PASTEBOARD\_UPDATE ルーチンを呼び出すことにより実行できます。SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチンは、出力操作を物理画面上ではなく、ペーストボード・バッファ内だけで反映します。SMG\$END\_PASTEBOARD\_UPDATE ルーチンを呼び出すと、ペーストボード・バッファは物理画面に書き込まれます。

SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチンと SMG\$END\_PASTEBOARD\_UPDATE ルーチンは、カウンタを増分および減分します。このカウンタの値が 0 になると、ペーストボードに対する出力はただちに物理画面に送信されます。カウンタの値が 0 以外の場合には、出力操作はバッチ処理されます。ペーストボード・バッチ処理レベルはカウンタの値に等しくなります。このカウンタ方式を使用すれば、サブ

## 日本語 SMG の出力操作

### 2.4 出力のバッチ処理

ルーチンは呼び出しプログラムのバッチ処理レベルを妨害せずに、バッチ処理を要求したり、バッチ処理をオフにすることができます。

---

## 日本語 SMG 入力操作

この章では、仮想キーボードからの入力を実行するために使用する日本語 SMG ルーチンについて説明します。仮想キーボードは通常、漢字ターミナルに対応づけられます。

日本語 SMG には、漢字ターミナルから入力を実行するための柔軟なルーチン群があります。入力ルーチンは出力ルーチンと組み合わせて使用でき、また、単独で使用することも可能です。仮想キーボードと呼ぶ入力ソースを設定するには、SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンを使用します。仮想キーボードを削除する場合には、SMG\$DELETE\_VIRTUAL\_KEYBOARD ルーチンを使用します。なおペーストボードを併用するときには、SMG\$CREATE\_VIRTUAL\_KEYBOARD の前に必ず SMG\$CREATE\_PASTE\_BOARD を呼び出さなければなりません。

---

### 注意

---

日本語 SMG は RMS ファイルを入力装置としてサポートしません。

---

ペーストボードが出力操作のための論理構造であるのと同様に、仮想キーボードは入力操作のための論理構造です。仮想キーボードを使用すれば、装置からの独立性を実現できます。日本語 SMG 入力ルーチンを使用する場合には、使用する漢字ターミナルのタイプを考慮する必要はありません。

たとえば、ユーザ・プログラムは、各ターミナルがどの行終了文字を使用するかを考慮する必要がありません。日本語 SMG ルーチンは異なる終了文字を統一された機能コードに変換します (終了文字コードについての詳しい説明は、第 3.5 節を参照してください)。また、ファンクション・キー/キーパッド・キーを処理する場合にも、仮想キーボードは重要です。

---

### 3.1 仮想キーボードからのデータの取り込み

データは以下の 4 種類の方法で仮想キーボードから取り込むことができます。

- SMG\$READ\_COMPOSED\_LINE

通常のキーストロークの他、キーパッド・キーとファンクション・キーに対応する定義済み文字列で構成される行を読み込みます。このルーチンは、単一キー・コマンド機能を提供することにより、コマンド向きユーティリティのためのインターフェイスを容易にコーディングする機能を備えています。

- SMG\$READ\_KEYSTROKE

キーボードから入力された1つのキーストロークを読み込みます。このルーチンは、英数字キーの他、ファンクション・キーとキーパッド・キーも読み込みます。

- SMG\$READ\_STRING

文字と区切り文字で構成される文字列を読み込みます。この柔軟なルーチンは OpenVMS ターミナル・ドライバの多くの機能をアクセスできます。

- SMG\$READ\_VERIFY

文字列を読み込み、その文字列をユーザが指定したピクチャ文字列と比較します。

どの読み込み操作も、SMG\$CANCEL\_INPUT ルーチン呼び出すことにより強制終了できます。

SMG\$READ\_COMPOSED\_LINE, SMG\$READ\_STRING, SMG\$READ\_VERIFY ルーチンは、入力される文章に対する文字集合を設定するために、省略可能な引数 ( character-set ) を持っています。

この引数を省略した場合は、SMG\$DEFAULT\_CHARACTER\_SET 論理名が用いられます。SMG\$DEFAULT\_CHARACTER\_SET 論理名の有効な定義については第 2.2.14 項の表 2-2 を参照してください。この論理名が無効な文字集合を指していたり、定義されていなかった場合には、文字集合は未定義になります。

日本語 SMG の入力ルーチンはそのルーチン自体でかな漢字変換を行えます。ユーザはアプリケーションの中から日本語 OpenVMS の日本語ライブラリ中に入っているかな漢字変換ルーチン呼び出す必要はありません。かな漢字変換キーの設定方法については『ユーザ・キー定義 利用者の手引き』を参照してください。

---

## 3.2 仮想キーボード属性の設定と検索

ペーストボード属性に関する情報を検索したり、ペーストボード属性を設定する場合と同様に、日本語 SMG には仮想キーボードの属性を検索および設定するためのルーチンがあります。

SMG\$GET\_KEYBOARD\_ATTRIBUTES ルーチンを使用すれば、仮想キーボードに関して指定した情報を検索できます。この情報はキーボード情報テーブル (KIT) と呼ぶユーザ指定領域に格納されます。戻される情報は次のとおりです。

- 現在の装置属性
- 装置クラス
- 再呼び出しバッファのサイズ
- 物理装置タイプ

- 先打ちバッファの最初の文字
- 漢字ターミナルの幅
- 先打ちバッファ内の文字数

SMG\$SET\_KEYPAD\_MODE ルーチンを使用すれば、漢字ターミナルの数値キーパッドを数値モードまたはアプリケーション・モードに設定できます。アプリケーション・モードでは、数値キーパッド・キーはファンクション・キーとして解釈され、終了文字として使用できます。数値モードでは、これらのキーはメイン・キーボードの対応するキーと等しく取り扱われます。ターミナルはアプリケーション・モードをサポートしていなければなりません。サポートしていない場合には、SMG\$SET\_KEYPAD\_MODE ルーチン呼び出しでも、その操作は異常終了します。

---

### 3.3 キーパッド・キーの使用による行の作成

SMG\$READ\_STRING ルーチンが提供する機能の他に、キーパッド・キーによる行作成機能を使用すれば、行単位のコマンド (たとえば、Command Definition コーティリティを使用するユーティリティなど) を含むアプリケーションにとって強力な柔軟なツールになります (詳しい説明は、『OpenVMS Command Definition, Librarian, and Message Utilities Manual』を参照してください)。

行作成機能を使用すれば、特定のキー (この後の説明を参照) を文字列に等しいものとして定義できます。1 行を入力した後、これらのキーのいずれかを押すと、そのキーに対応する文字列 (同値文字列) が、戻されるコマンド文字列に挿入されます。たとえば、アプリケーションでPF2 キーを“HELP” という文字列と等しく定義した場合には、PF2 キーを押すと、そのコマンドはアプリケーションに戻されます。

また、同値文字列をエコー表示することも指定できます。この場合には、“HELP” という文字列がエコー表示されます。キーパッド・キーを認識する操作と同値文字列を挿入する操作は、SMG\$READ\_COMPOSED\_LINE ルーチンが自動的に処理します。アプリケーションは、戻された行をユーザが行全体を入力したかのように取り扱います。

キー定義はキー定義テーブルに登録されます。このテーブルは SMG\$CREATE\_KEY\_TABLE ルーチン呼び出すことにより作成されます。

キー定義は SMG\$ADD\_KEY\_DEF ルーチン呼び出すことによりテーブルに追加でき、SMG\$DELETE\_KEY\_DEF ルーチン呼び出すことによりテーブルから削除できます。また、SMG\$DEFINE\_KEY ルーチンと SMG\$LOAD\_KEY\_DEFS ルーチン呼び出すことにより、キー定義を追加することも可能です。これらのルーチンは DCL の DEFINE/KEY コマンド (またはこれらのコマンドを登録したファイル) を受け付けます。詳しい説明は、これらのルーチンの説明を参照してください。また、

DEFINE/KEY コマンドについては、『OpenVMS DCL デイクショナリ』を参照してください。

定義可能なキーは、表 3-1 に示したファンクション・キーとキーパッド・キー、制御キー・シーケンス (Ctrl/A ~ Ctrl/Z) ただし、Ctrl/M (RETURN) を除く) であり、行編集が許可されていない場合は行編集キーも含まれます。

キー定義には複数の属性が割り当てられます。

- TERMINATE 属性は、このキーが押されたときに、入力行を終了するかどうかを指定します。
- NOTERMINATE 属性は、追加文字と追加キーストロークを入力できることを指定します。省略時の設定は TERMINATE です。
- ECHO 属性は、キーが押されたときに、同値文字列をエコー表示するかどうかを指定します。省略時の設定は ECHO です。
- PROTECT 属性は、キーを定義した後、そのキー定義を変更または削除できるかどうかを指定します。省略時の設定は NOPROTECT です。

この他に、LOCK\_STATE 属性、IF\_STATE 属性、および STATE 属性があります。これらの属性については、この後の節で説明します。

---

## 3.4 状態

各キーには、現在の状態値に応じて多くの定義を割り当てることができます。状態はキーの意味を判断するために使用されます。たとえば、状態を“GOLD”に設定するために PF1 を定義し、IF\_STATE=“GOLD”状態の PF2 を“HELP \*”として定義した場合には、PF1 と PF2 を押した結果、“HELP \*”がコマンド行として戻されます。この場合、PF1 の定義には同値文字列が存在せず、NOTERMINATE 属性を指定しなければなりません。

状態名は最大 31 文字の英数字で構成される文字列であり、ドル記号 (\$) とアンダースコア (\_) も使用できます。1 行が通常のキーストロークと同値文字列で構成される場合には、SMG\$READ\_COMPOSED\_LINE ルーチンは現在の状態名と呼ぶ文字列を管理します。最初のキーが押される前の現在の状態は“DEFAULT”です。STATE 属性に対して値が指定されている定義を持つキーを押した場合には、現在の状態は指定された状態に変更されます。LOCK\_STATE 属性を指定しない限り、次の定義済みキーを押した後、状態名は“DEFAULT”に戻ります。

## 3.5 終了文字

終了文字は、仮想キーボードからデータの転送を終了します。終了文字はキャリッジ・リターンや`Ctrl/Z`などの1文字、あるいはキーボードのキーを押すことにより生成される文字シーケンス (エスケープ・シーケンス)、あるいは時間切れやバッファの満杯などの条件のいずれでもかまいません。

終了文字は仮想キーボードから読み込まれるデータの一部ではありません。終了文字は独立した引数を使用して、整数値 (符号なしワード) として呼び出しプログラムに戻されます。単一文字終了文字の場合には、値は終了文字の8ビット・コードです。単一文字終了文字コードは0 ~ 255の範囲です。

文字シーケンス終了文字は装置から独立した形式で戻されます。これらのコードは `SMG$K_TRM_keyname` の形式です (たとえば, `SMG$K_TRM_DELETE`)。VT220 (および VT200 と互換性のある) 漢字ターミナルの可能な各ファンクション・キーに対して、固有のコードが割り当てられています。他の漢字ターミナルのキー・コードは、VT220 の対応するキーのコードを使用して戻されます。したがって、アプリケーション・プログラムは使用する漢字ターミナルのタイプを認識する必要がありません。日本語 SMG ルーチンは異なる終了文字シーケンスを統一された機能コードに自動的に変換します。

条件で終了する入力操作は、`SMG$K_TRM_CANCELLED`、`SMG$K_TRM_TIMEOUT`、`SMG$K_TRM_BUFFER_FULL`、`SMG$K_TRM_UNKNOWN` という終了文字コードによって示されます。

`SMG$READ_STRING` ルーチンと `SMG$READ_VERIFY` ルーチンを呼び出す場合には、省略時の単一文字終了文字は0 ~ 31の範囲のすべての文字ですが、バックスペース (8)、水平タブ (9)、ライン・フィード (10)、垂直タブ (11)、およびフォーム・フィード (12) を除きます。これらの文字は OpenVMS ターミナル・ドライバに対して設定されている省略時の終了文字です。しかし、8ビット文字コードはすべて終了文字として使用できます。終了文字集合は `SMG$READ_STRING` ルーチンまたは `SMG$READ_VERIFY` ルーチンのそれぞれの呼び出しごとに変更できます。`SMG$READ_COMPOSED_LINE` ルーチンを呼び出す場合には、省略時の単一文字終了文字はキャリッジ・リターン (13) と `CTRL/Z` だけです。`SMG$READ_COMPOSED_LINE` ルーチンに対して終了文字セットを変更する場合には、キー定義機能を使用します。詳しい説明は、第3.3節の行作成の説明を参照してください。

`SMG$NAME_TO_KEYCODE` ルーチンは、キーボードのキーの名前を対応する終了文字コードに変換します。一方、`SMG$KEYCODE_TO_NAME` ルーチンは終了文字コードをキーボードのキーの対応する名前に変換します。

表 3-1 は、単一文字ではない各終了文字の終了文字名または状態を示しています。この表には、日本語 SMG 入力ルーチンがサポートする各ターミナル・タイプの各終了文字に対して、そのコードとキーも示されています。

日本語 SMG 入力操作  
3.5 終了文字

表 3-1 終了文字の値

キー名	値	VT200 およ び VT300 シリーズ	VT100	VT52
キーパッド・キー				
DELETE	SMG\$K_TRM_DELETE	<X>	DELETE	DEL
PF1	SMG\$K_TRM_PF1	PF1	PF1	Blue
PF2	SMG\$K_TRM_PF2	PF2	PF2	Red
PF3	SMG\$K_TRM_PF3	PF3	PF3	Black
PF4	SMG\$K_TRM_PF4	PF4	PF4	
KP0	SMG\$K_TRM_KP0 <sup>1</sup>	0	0	0
KP1	SMG\$K_TRM_KP1 <sup>1</sup>	1	1	1
KP2	SMG\$K_TRM_KP2 <sup>1</sup>	2	2	2
KP3	SMG\$K_TRM_KP3 <sup>1</sup>	3	3	3
KP4	SMG\$K_TRM_KP4 <sup>1</sup>	4	4	4
KP5	SMG\$K_TRM_KP5 <sup>1</sup>	5	5	5
KP6	SMG\$K_TRM_KP6 <sup>1</sup>	6	6	6
KP7	SMG\$K_TRM_KP7 <sup>1</sup>	7	7	7
KP8	SMG\$K_TRM_KP8 <sup>1</sup>	8	8	8
KP9	SMG\$K_TRM_KP9 <sup>1</sup>	9	9	9
ENTER	SMG\$K_TRM_ENTER <sup>2</sup>	ENTER	ENTER	ENTER
MINUS	SMG\$K_TRM_MINUS <sup>1</sup>	-	-	
COMMA	SMG\$K_TRM_COMMA <sup>1</sup>	,	,	
PERIOD	SMG\$K_TRM_PERIOD <sup>1</sup>	.	.	
カーソル移動キー				
UP	SMG\$K_TRM_UP	上向き矢印	上向き矢印	上向き矢印
DOWN	SMG\$K_TRM_DOWN	下向き矢印	下向き矢印	下向き矢印
LEFT	SMG\$K_TRM_LEFT	左向き矢印	左向き矢印	左向き矢印
RIGHT	SMG\$K_TRM_RIGHT	右向き矢印	右向き矢印	右向き矢印
ファンクション・キー				
F6	SMG\$K_TRM_F6	F6		
F7	SMG\$K_TRM_F7	F7		
F8	SMG\$K_TRM_F8	F8		
F9	SMG\$K_TRM_F9	F9		
F10	SMG\$K_TRM_F10	F10		
F11	SMG\$K_TRM_F11	F11		
F12	SMG\$K_TRM_F12	F12		

<sup>1</sup>これらはメイン・キーボードではなく、数値キーパッドのキーです。これらの値は、ターミナル・キーパッドがアプリケーション・モードの場合にだけ使用されます。キーパッドが数値モードの場合には、キーはメイン・キーボードの対応するキーと同じです。詳しくは、SMG\$SET\_KEYPAD\_MODE ルーチンの説明を参照してください。

<sup>2</sup>キーパッドが数値モードの場合には、ENTER はキャリッジ・リターンと同じです。詳しくは、SMG\$SET\_KEYPAD\_MODE ルーチンの説明を参照してください。

(次ページに続く)

表 3-1 (続き) 終了文字の値

キー名	値	VT200 およ び VT300 シリーズ	VT100	VT52
F13	SMG\$K_TRM_F13	F13		
F14	SMG\$K_TRM_F14	F14		
HELP	SMG\$K_TRM_HELP <sup>3</sup>	HELP		
DO	SMG\$K_TRM_DO <sup>3</sup>	DO		
F17	SMG\$K_TRM_F17	F17		
F18	SMG\$K_TRM_F18	F18		
F19	SMG\$K_TRM_F19	F19		
F20	SMG\$K_TRM_F20	F20		
編集キー				
FIND	SMG\$K_TRM_FIND	Find		
INSERT_HERE	SMG\$K_TRM_INSERT_HERE	Insert Here		
REMOVE	SMG\$K_TRM_REMOVE	Remove		
SELECT	SMG\$K_TRM_SELECT	Select		
PREV_SCREEN	SMG\$K_TRM_PREV_SCREEN	Prev Screen		
編集キー				
NEXT_SCREEN	SMG\$K_TRM_NEXT_SCREEN	Next Screen		
条件				
CANCELLED	SMG\$K_TRM_CANCELLED			
TIMEOUT	SMG\$K_TRM_TIMEOUT			
BUFFER_FULL	SMG\$K_TRM_BUFFER_FULL			
UNKNOWN	SMG\$K_TRM_UNKNOWN <sup>4</sup>			

<sup>3</sup>HELP と DO は VT220 キーボードの F15 および F16 と同じです。

<sup>4</sup>VT200 および認識されない終了文字を受け付けた場合には、値は SMG\$K\_TRM\_UNKNOWN になります。

終了文字の値のシンボリック定義は、弊社が提供する \$SMGDEF というの名前のシンボル・ライブラリに登録されています (たとえば、MACRO プログラムでこれらの定義を検索する場合には、\$SMGDEF を呼び出します)。シンボル名は SMG\$K\_TRM\_keyname の形式です。ただし、keyname 引数は表 3-1 に示したキー名です。終了文字コードが、1 ~ 26 の場合には、それは **Ctrl/A** ~ **Ctrl/Z** の制御シーケンスに対応し、キー名は、**Ctrl/A** の場合は CTRLA であり、**Ctrl/B** の場合は CTRLB です。以下も同様です。次の同意語も定義されています。

---

シンボル	同意語
SMG\$K_TRM_BS	SMG\$K_TRM_CTRLH
SMG\$K_TRM_HT	SMG\$K_TRM_CTRLI
SMG\$K_TRM_LF	SMG\$K_TRM_CTRLJ
SMG\$K_TRM_CR	SMG\$K_TRM_CTRLM
SMG\$K_TRM_E1	SMG\$K_TRM_FIND
SMG\$K_TRM_E2	SMG\$K_TRM_INSERT_HERE
SMG\$K_TRM_E3	SMG\$K_TRM_REMOVE
SMG\$K_TRM_E4	SMG\$K_TRM_SELECT
SMG\$K_TRM_E5	SMG\$K_TRM_PREV_SCREEN
SMG\$K_TRM_E6	SMG\$K_TRM_NEXT_SCREEN
SMG\$K_TRM_F15	HELP
SMG\$K_TRM_F16	DO

---

---

### 3.6 行の再呼び出しと再呼び出しバッファ

日本語 SMG を使用すれば、アプリケーション再呼び出しバッファの内容をアクセスしたり、変更することができます。省略時の設定では、再呼び出しバッファには、前に入力した 20 個のコマンド、またはユーザがアプリケーションに対して入力したデータ行が格納されます。

SMG\$RETURN\_INPUT\_LINE ルーチンを使用すれば、再呼び出しバッファから特定の行を要求できます。呼び出す行に対して適切な行番号を指定するか、または照合文字列を指定できます。照合文字列を使用する場合には、SMG\$RETURN\_INPUT\_LINE ルーチンは指定された文字列と一致する行を検索し、その行を戻します。このルーチンは、DCL 形式の RECALL コマンドを実現するための補助機能として提供されます。

SMG\$REPLACE\_INPUT\_LINE ルーチンを使用すれば、再呼び出しバッファ内の指定した行 (1 行以上) を、指定した文字列と置き換えられます。再呼び出しバッファの他の行は削除されます。このルーチンは行の継続を処理するのに役立ちます。

---

### 3.7 入力と出力の相互作用

次の 4 つのルーチンは display-id 引数をオプションとして受け付けます。

- SMG\$READ\_COMPOSED\_LINE
- SMG\$READ\_KEYSTROKE
- SMG\$READ\_STRING
- SMG\$READ\_VERIFY

display-id 引数を指定した場合には、その引数は入力操作を実行する仮想ディスプレイを指定します。display-id 引数を指定することにより、文字のエコー表示による変化を、日本語 SMG が継続的に認識するように設定できます。display-id 引数を省略した場合には、日本語 SMG は、日本語 SMG の出力機能が使用されていないものと解釈します。

display-id 引数を、これまで説明した入力ルーチンに対して指定する場合には、プロンプト文字列と入力の長さの合計はディスプレイのカラム数に制限されるか、または読み込む文字の最大数を指定した場合には、その最大数に制限されます (SMG\$READ\_KEYSTROKE ルーチンの場合には、この制限はプロンプト文字列の長さだけに適用されます)。

---

## 日本語 SMG の高度な機能

日本語 SMG には、次のような高度な機能があります。

- 非同期イベントをトラップする。
- サブプロセス内でコマンドを生成し、実行する。
- 物理カーソルを移動する。
- 終了時に画面を整理する。

この後の節では、これらの高度な機能について説明します。

---

### 注意

---

日本語 SMG は、漢字ターミナルが完全に日本語 SMG の管理下にあることを仮定しています。したがって、アプリケーションは、日本語 SMG ルーチンと日本語 DIGITAL GKS や 日本語 FMS のような画面管理用製品を同時に呼ぶべきではありません。

日本語 SMG はリエントラントではありません。したがって複数のプロセスが、1つのターミナルで同時に操作を行うために、日本語 SMG ルーチンを呼び出そうとした場合、予想できない結果を招く恐れがあります。アプリケーションが複数のプロセスを持ち、画面修正が必要な場合は、一方のプロセスは日本語 SMG ルーチンを呼び出し、もう一方のプロセスはルーチンを呼び出しているプロセスに対して、メッセージを送る必要があります。それによって、プロセスは適切な日本語 SMG ルーチンを呼び出すことができます。

---

---

### 4.1 非同期イベント

次に示す 3 種類の非同期イベントは画面のイメージを破壊する可能性があります。

- ブロードキャスト・メッセージ
- 要求されていない入力
- アウト・オブ・バンド非同期システム・トラップ

この後の節では、これらの操作を制御する方法について説明します。

---

注意

---

日本語 SMG は AST リエントラントではありません。したがって、この章で説明する日本語 SMG ルーチン呼び出すプログラムは、必要な同期処理を実行しなければなりません。

---

#### 4.1.1 ブロードキャスト・メッセージ

通常、ブロードキャスト・メッセージ(たとえば、MAIL の通知やオペレータ・メッセージなど)は、いつでもターミナル画面に表示される可能性があり、その結果、画面イメージは破壊されます。SMG\$SET\_BROADCAST\_TRAPPING ルーチンを使用すれば、指定された漢字ターミナル(ペーストボード)にブロード・キャストされたメッセージを受け取ることができ、さらに、ブロードキャスト・メッセージが受け取られるときに必ず呼び出される AST ルーチンを指定できます。ユーザが指定する AST ルーチンは、SMG\$GET\_BROADCAST\_MESSAGE ルーチン呼び出すことにより、ブロードキャスト・メッセージをアクセスできます。

SMG\$SET\_BROADCAST\_TRAPPING ルーチン呼び出すときに AST ルーチンを指定したかどうかとは無関係に、ブロードキャスト・メッセージを受信したかどうかということはいつでも確認できます。ブロードキャスト・メッセージの受信を確認する場合には、SMG\$GET\_BROADCAST\_MESSAGE ルーチン呼び出します。

#### 4.1.2 要求されていない入力

SMG\$ENABLE\_UNSOLICITED\_INPUT ルーチンは、要求されていない入力の存在を検出します。このルーチンは入力文字を読み込むわけではなく、単に SMG\$READ\_COMPOSED\_LINE, SMG\$READ\_KEYSTROKE, SMG\$READ\_STRING, SMG\$READ\_VERIFY ルーチンを使用して読み込み操作を実行しなければならないことをアプリケーションに通知するために、AST ルーチン呼び出すだけです。要求されない入力を実際に読み込むのはユーザの役割です。

#### 4.1.3 アウト・オブ・バンド AST

SMG\$SET\_OUT\_OF\_BAND\_ASTS ルーチンは、`[Ctrl/Y]`や`[Ctrl/C]`, `[Ctrl/O]`などのアウト・オブ・バンド文字を受け取るための方法を提供します。このルーチンを使用すれば、どの文字をアウト・オブ・バンド文字として取り扱うかを指定でき、また、これらの文字が入力されたときに呼び出される AST ルーチンも指定できます。

## 4.2 サブプロセス

日本語 SMG では、サブプロセスを使って、アプリケーションから DCL コマンドを実行することができます。各仮想ディスプレイに対して、サブプロセスは 1 つだけ作成できます。

### 4.2.1 サブプロセスの作成

サブプロセスは SMG\$CREATE\_SUBPROCESS ルーチンを使用して作成されます。このルーチンは DCL サブプロセスを生成し、そのサブプロセスをユーザが指定した仮想ディスプレイに対応づけます。サブプロセスは SET NOVERIFY と SET NOON の DCL コマンドを使用して初期化されます。プロセスを生成するには、BYTLM の値が少なくとも 5000 で、PRCLM の値が少なくとも 1 でなければなりません。日本語 SMG は、サブプロセスを生成する前に、十分なリソースが残されているかどうかを確認します。

### 4.2.2 サブプロセスの削除

サブプロセス・コマンドを実行した後、SMG\$DELETE\_SUBPROCESS ルーチンを使用してサブプロセスを削除できます。このルーチンを呼び出さずに処理を終了した場合には、日本語 SMG ルーチンはサブプロセスを削除するために終了ハンドラを使用します。しかし、このような日本語 SMG が提供する終了ハンドラを実行できないような状況があります。その場合には、次のコマンドを使用してサブプロセスを削除しなければなりません。

```
$ SHOW PROCESS/SUB  
$ STOP/IDENT=xxxx
```

### 4.2.3 サブプロセス内でのコマンドの実行

SMG\$EXECUTE\_COMMAND ルーチンを使用すれば、作成したサブプロセス内で指定されたコマンドを実行できます。SMG\$CREATE\_SUBPROCESS ルーチンで AST ルーチンが指定され、コマンドがバッファに登録されている場合には、SMG\$EXECUTE\_COMMAND ルーチンは、指定されたコマンドをバッファに登録し、制御を戻します。コマンドの実行が終了すると、コマンドの状態によってユーザが指定した AST ルーチンが起動されます。SMG\$CREATE\_SUBPROCESS ルーチンで AST ルーチンが指定されずコマンドがバッファに登録されていない場合には、SMG\$EXECUTE\_COMMAND ルーチンはコマンドが実行がされるのを待ち、終了した後、コマンドの状態を戻します。

実行するコマンド文字列を指定する場合には、DCL コマンドの 1 文字目として "\$" を指定しなければなりません。日本語 SMG は、"\$" から始まらないコマンド文字列を前のコマンドの入力データとして解釈します。コマンドとその出力は、コマンドが実行されるときに指定された仮想ディスプレイに表示されます。SPAWN コマンド、

GOTO コマンド，および LOGOUT コマンドをコマンド文字列として使用することはできず，これらのコマンドを使用した場合には，予測できない結果を生じます。

また，入出力はターミナル・ドライバを通じてではなく，メールボックスを使用して実行されるため，`Ctrl/C`や`Ctrl/Y`，`Ctrl/Z`などの単一文字コマンドはまったく効果がありません。`Ctrl/Z`をサブプロセスに渡すには，flags 引数として `SMG$M_SEND_EOF` を使用しなければなりません。

---

### 4.3 カーソルの移動

日本語 SMG では，物理画面の指定された位置にカーソルを移動できます。この操作は `SMG$SET_PHYSICAL_CURSOR` ルーチンを使用して実行します。しかし，カーソルをペーストボードの画面境界の外部に移動しようとした場合には，エラーが返されます。

---

### 4.4 終了ハンドラ

日本語 SMG は，イメージ終了の前に起動される終了ハンドラを提供しています。このハンドラは，現在のイメージに関連するすべてのペーストボードと仮想キーボードを削除し，ターミナル属性を再設定します。日本語 SMG の終了ハンドラは，ユーザが指定した終了ハンドラの前に起動される場合もありますし，後で起動される場合もあります。したがって，ユーザ指定の終了ハンドラの内部からペーストボードや仮想ディスプレイを削除すべきではありません。ペーストボードや仮想ディスプレイは，日本語 SMG の終了ハンドラによってすでに削除され，その識別子の割り当てが取り消されている可能性があるからです。

---

## フォーリン・ターミナルのサポート

この章では、フォーリン・ターミナルに対する日本語 SMG ルーチンのサポートについて説明します。

フォーリン・ターミナルとは、装置タイプが OpenVMS で認識されない漢字ターミナル、あるいは ANSI\_CRT 属性が設定されていない漢字ターミナルです。このサポート機能はおもに、日本語 SMG が使用しますが、フォーリン・ターミナルに対して独自の入出力を実行する必要のあるアプリケーション・プログラムもこの機能を使用できます。したがって、日本語 SMG を使用する場合には、フォーリン・ターミナルの機能の定義だけを考慮すればよく、フォーリン・ターミナル・ルーチンを直接に呼び出すことを考慮する必要はありません。

さらに、日本語 SMG がターミナルの画面を効果的に制御できるようにするには、ターミナル機能の一部 (“set absolute cursor position”, “erase to end of display”, “erase to end of line”) だけを定義すれば十分です。しかし、この章では日本語 SMG が使用するルーチンを示すことにより、ユーザがフォーリン・ターミナルに対して独自の入出力を実行できるようにしています。

サポート機能はまず、TERMTABLE.TXT という名前のソース・ファイルから始まります。このファイルには、ターミナル名と関連する機能のリストが登録されています。このファイルは TERMTABLE.EXE というイメージ・ファイルを作成するために、SYSSSYSTEM:SMGBLDTRM.EXE プログラムによって処理されます。この後の節では、TERMTABLE データベースの作成と処理について説明します。

TERMTABLE サポート機能は、すべての漢字ターミナルに対して日本語 SMG が使用します。ターミナルに対する定義は SYSSSYSTEM:SMGTERMS.TXT という名前のファイルに登録されています。このファイルは日本語 SMG の一部として提供されます。この後の節の例では、フォーリン・ターミナル・パッケージを使用してターミナルを定義する方法が示されています。このような例を示したのは、大部分のユーザがこれらの漢字ターミナルを十分理解しているからです。しかし、ユーザ独自の定義を実際に作成する必要はなく、また SYSSSYSTEM:SMGTERMS.TXT 内の定義を変更すべきでもありません。

---

## 5.1 TERMTABLE インターフェイス・ルーチン

TERMTABLE.EXE は、複数の異なるタイプの漢字ターミナルに関する情報を登録したデータベースです。このデータベースから情報を検索する場合には、次の操作を実行します。

1. ターミナル名をデータベースに与えます。
2. そのターミナル・タイプについての情報を検索します (この操作は何回でも繰り返すことができます)。
3. データベースに対するアクセスを終了します。

日本語 SMG ルーチンを使用してフォーリン・ターミナルに対して入出力を実行する場合には、使用するフォーリン・ターミナルにとって適切な TERMTABLE エントリだけを作成しなければなりません。ここに示した操作が必要なのは、プログラムがフォーリン・ターミナルに対して直接に入出力を実行する場合だけです。

上記の 1 の操作は次の 2 種類の方法のいずれかを使用して実行できます。

- ターミナル名 (たとえば VT100) を登録した文字列を SMG\$INIT\_TERM\_TABLE ルーチンに渡すことができます。
- OpenVMS システム・サービス \$GETDVI が戻した値を SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチンに渡すこともできます。戻される値はシンボリック・ターミナル・タイプ (たとえば, TT\$\_VT100 や TT\$\_VT52) であるか、または特定のフォーリン・ターミナルを指定するために日本語 SMG フォーリン・ターミナル・ルーチンが割り当てた値です。

2 の操作では、SMG\$GET\_TERM\_DATA ルーチンを呼び出さなければなりません。このルーチンは、TERMTABLE からコマンド文字列 (たとえばエスケープ・シーケンス) を検索し、それをユーザが提供するバッファに入れます。その後、ユーザはコマンド文字列を漢字ターミナルに書き込まなければなりません。

SMG\$GET\_TERM\_DATA ルーチンは何度も呼び出さなければならない可能性があります。コマンド・シーケンスを受け取るたびに、異なる操作を実行できます。また、引数に対して置換操作や算術演算が必要な機能文字列を使用する場合には、そのたびに SMG\$GET\_TERM\_DATA ルーチンを呼び出さなければなりません。しかし、変更する必要のない機能文字列をプログラムのローカル記憶領域に保存しておくこともできます。これらの変更する必要のない機能文字列は 1 度検索すると、その後、何度でも使用できます。

3 の操作は省略可能です。この操作は、データベース内の情報をアクセスするために使用した仮想メモリの割り当てを、単に解放するだけです。

DCL の SET TERMINAL コマンドと SHOW TERMINAL コマンドは、OpenVMS ターミナルを認識すると同時に、TERMTABLE に定義されている名前を認識します。OpenVMS オペレーティング・システムが認識しない漢字ターミナルのターミナ

ル・タイプを指定するために、SET TERMINAL/DEVICE=name コマンドを使用した場合には、TERMTABLE データベースから指定された漢字ターミナルが検索されます。

特定のターミナル定義のアドレスを検索するために、2つのルーチンが提供されます。SMG\$INIT\_TERM\_TABLE ルーチンは入力としてターミナル名を受け付けます。SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチンは入力として装置タイプを受け付けます。ターミナル名と装置タイプはそれぞれ、TERMTABLE.EXE セクション内の特定のターミナル・エントリにマップされます。これらのルーチンは将来の呼び出しで使用するために、この識別子を呼び出しプログラムに戻します。

SMG\$GET\_TERM\_DATA ルーチンは、コンパイルされた TERMTABLE データベースの識別子と要求コードを受け付けます。要求コードは、適切なエスケープ・シーケンスを検索するためにデータに対するインデックスとして使用されます。一部のシーケンスは静的で変化しません。これらのシーケンスには可変情報は含まれず、単に呼び出し側のバッファにコピーされるだけです。

"! "ディレクティブと"%ディレクティブを含む可変シーケンスを使用すると、追加処理が実行されます。可変シーケンスの例として、VT300 シリーズのカーソル設定コマンドがあります。この場合、要求されたバイナリの行番号とカラム番号をカーソル設定のために ASCII に変換しなければなりません。SMG\$GET\_TERM\_DATA ルーチンは、入力された引数を変換し、シーケンスを呼び出し側のバッファにコピーします。

SMG\$GET\_TERM\_DATA ルーチンに入力引数を指定しなかった場合には、機能が必要とする引数に対して、省略時の値として 1 が使用されます。しかし、省略可能な引数の一部だけを指定し、他の引数に対しては省略時の値を使用することは認められません。このような場合には、すべての引数を指定するか、またはすべての引数の値として省略時の設定を使用しなければなりません。

SMG\$GET\_NUMERIC\_DATA ルーチンは、数値データまたは論理データだけを探索したいユーザのために、簡略化されたインターフェイスを提供します。

すべてのターミナル入出力が終了した後、使用した仮想メモリの割り当てを解放するために SMG\$DEL\_TERM\_TABLE ルーチンを呼び出すことができます。このルーチンは、プログラム途中で TERMTABLE を必要としない場合にだけ役立ちます。仮想メモリの割り当てを解放すると、プログラムはそのメモリを再利用できるようになります。

TERMTABLE.TXT というひな型は SYSS\$SYSTEM: に提供されています。OpenVMS が認識するターミナルを定義する SMGTERMS.TXT も提供されます。TERMTABLE.TXT というひな型は SMGTERMS.TXT という別のソース・ファイルを含むように REQUIRE ディレクティブを使用しています。したがって、実際にはフォーリン・ターミナルだけが TERMTABLE.TXT ソース・ファイルに定義されます。

ユーザは SMGTERMS.TXT を変更してはいけません。

---

## 5.2 機能フィールド

フォーリン・ターミナルを使用している場合には、日本語 SMG にとって漢字ターミナルが何を実行でき、何を実行できないか、つまり、漢字ターミナルがどのような機能を備えているかがわかりません。機能フィールドを使用することにより、フォーリン・ターミナルに対してどのような機能がサポートされるかを日本語 SMG に示すことができます。これらのフィールドを調べることにより、日本語 SMG ルーチンは共通のターミナル機能をエミュレートするのではなく、個々の漢字ターミナルの機能を使用できます。この結果、日本語 SMG ルーチンの性能を向上できます。

TERMTABLE エントリには次の 3 種類の機能フィールドを指定できます。

- 論理値
- 数値
- 文字列

この後の節では、これらの機能フィールドについて詳しく説明します。

ほとんどの漢字ターミナルに共通の機能は、使用可能なフィールドとして選択されています。すなわち、すべてのターミナル・タイプのすべての機能が表現されているわけではありません (特に、ブロック・モード、グラフィック、タイプセット機能などはサポートされません)。画面对応型アプリケーションは典型的なターミナル機能を中心に設計しなければならず、特定のモデルにだけしか存在しない機能に依存しないようにしなければなりません。

特殊な漢字ターミナルをサポートしなければならないアプリケーションの場合には、ユーザが定義できるように汎用機能名が確保されています。PRIVATE\_BOO\_n, PRIVATE\_NUM\_n, PRIVATE\_STR\_n (ただし、n は 1 ~ 10 の数値です) の形式の名前は、ユーザ定義ターミナル定義として登録でき、TERMTABLE インターフェイス・ルーチンから戻されます。これらの名前の意味はユーザが割り当てるため、各機能はアプリケーション間で異なるものになる可能性があります。

複数のアプリケーションを実行するシステムは、1 つのプライベート機能に対して複数の定義が設定されないように注意しなければなりません (たとえば、実行するアプリケーション・プログラムに応じて、PRIVATE\_STR\_1 が 2 種類の異なる意味を持つ場合には、その漢字ターミナルに対して別々のターミナル・エントリを設定しなければなりません)。一般に、このようなプライベート機能を使用する必要はありません。

次に示す文字は機能フィールド内で区切り文字として使用されます。

区切り文字	意味
!	コメントの先頭を示す
=	機能フィールド名と値を区切る
,	機能フィールドを区切る
"	文字列を区切る

### 5.2.1 論理機能フィールド

値が論理値である機能は、各漢字ターミナルに対して、存在する場合も存在しない場合もあります。

論理値機能フィールドの形式は次のとおりです。

```
BOOLEAN
  {boolean-capability = binary-digit} [,...]
```

各要素の意味は次のとおりです。

- ・ boolean-capability 表 5-1 に示されている機能フィールドのいずれか
- ・ binary-digit 1 または 0

表 5-1 はこれらの論理値機能フィールドを示しています。

表 5-1 論理値機能

VMS 名	SMG による 使用	説明
ADVANCED_VIDEO	×	設定されている場合には、漢字ターミナルは拡張ビデオ属性を備えており、132 カラム・モード操作が可能である。
ANSI_CRT	×	設定されている場合には、漢字ターミナルは ANSI CRT プログラミング標準規格に従う。
AUTO_MARGIN	×	設定されている場合には、漢字ターミナルは自動マージンを備えている。
BACKSPACE		設定されている場合には、漢字ターミナルは <b>Ctrl/H</b> によってバックスペースできる。
BLOCK_MODE	×	設定されている場合には、漢字ターミナルはブロック・モード送信、ローカル編集、フィールド保護を実行できる。
CURSOR_REPORT_ANSI	×	設定されている場合には漢字ターミナルは現在のカーソル位置を報告するために、ANSI シーケンスを使用する。

(次ページに続く)

表 5-1 (続き) 論理値機能

VMS 名	SMG による 使用	説明
DEC_CRT	×	設定されている場合には、漢字ターミナルは DEC VT100 ファミリ標準規格に従う。
DEC_CRT_2	×	設定されている場合には、漢字ターミナルは DEC VT200 ファミリ標準規格に従う。
DEC_CRT_3	×	設定されている場合には、漢字ターミナルは DEC VT300 ファミリ標準規格に従う。
EDIT	×	設定されている場合には、漢字ターミナルは ANSI で定義されている拡張編集機能を実行できる。
EIGHT_BIT	×	設定されている場合には、漢字ターミナルは 8 ビットの ASCII 文字コードを使用する。
FULLDUP	×	設定されている場合には、漢字ターミナルの操作モードは全二重である (設定されていない場合は、半二重である)。
IGNORE_NEWLINE	×	設定されている場合には、漢字ターミナルは自動改行 (ラッピング) の後の、復帰改行 (ニューライン) を無視する。
INSERT_MODE_NULLS	×	設定されている場合には、挿入モードはディスプレイで NULL を区別する。
LOWERCASE	×	設定されている場合には、漢字ターミナルは英字の大文字と小文字のどちらも備えている。
NO_ERASE	×	設定されている場合には、ボールドの文字は、その上に書き込んでも消去されない。
NO_SCROLL	×	設定されている場合には、漢字ターミナルはスクロールを実行できない。
OVERSTRIKE	×	設定されている場合には、漢字ターミナルは置換 (オーバーストライク) を実行できる。
PHYSICAL_FF	×	設定されている場合には、漢字ターミナルは改ページ (フォーム・フィード) を受け付けることができる。設定されていない場合には、ターミナル・ドライバは改ページ (フォーム・フィード) を複数の改行 (ライン・フィード) に変換しなければならない。
PHYSICAL_TABS	×	設定されている場合には、漢字ターミナルはハードウェア・タブを備えている (これらのタブは初期化文字列によって設定しなければならない可能性がある)。
PRINTER_PORT	×	設定されている場合には、漢字ターミナルはプリンタ・ポートを備えている。
PRIVATE_BOO_1 to 10	×	設定されている場合には、これらのフィールドは 1 ~ 10 のユーザ定義機能を示す。

(次ページに続く)

表 5-1 (続き) 論理値機能

VMS 名	SMG による 使用	説明
REGIS	×	設定されている場合には、漢字ターミナルは ReGIS グラフィック・コマンドを解釈する。
SCOPE	×	設定されている場合には、漢字ターミナルはビデオ・ターミナルである。
SET_CURSOR_COL_ROW		設定されている場合には、漢字ターミナルはカラム/行アドレッシングを使用する。
SIXEL_GRAPHICS	×	設定されている場合には、漢字ターミナルは ReGIS で定義されている SIXEL グラフィック・プロトコルを使用してグラフィックを表示できる。
SOFT_CHARACTERS	×	設定されている場合には、漢字ターミナルはユーザ定義文字集合 (Dynamic Replacable Character Set) をロードできる。
UNDERLINE	×	設定されている場合には、漢字ターミナルは重ね書きでない下線機能を備えている。

たとえば、次の TERMTABLE エントリは VT300 シリーズ・ターミナルの 2 つの属性を記述しています。

```
NAME = "VT300_series"
  BOOLEAN
    ansi_crt = 1,      dec_crt = 1
```

このエントリは、漢字ターミナルが ANSI CRT プログラミング標準規格と DEC VT300 シリーズ標準規格に準拠することを指定しています。

## 5.2.2 数値機能フィールド

数値機能は、引数として数値を受け付ける機能です。たとえば、ターミナル画面のカラム数などを引数として受け付けます。

数値機能フィールドの形式は次のとおりです。

```
NUMERIC
  {numeric-capability = value} [, ...]
```

各要素の意味は次のとおりです。

- numeric-capability 表 5-2 に示す機能フィールドのいずれか
- value 指定した数値機能の値

表 5-2 は数値機能を示しています。

表 5-2 数値機能

VMS 名	SMG による 使用	説明
COLUMNS		1 行のカラム数を指定する。
CR_FILL	×	復帰改行 (キャリッジ・リターン) の後に必要なフィル文字の数を指定する。
LF_FILL	×	改行 (ライン・フィード) の後に必要なフィル文字の数を指定する。
FRAME	×	入力または出力される各文字に対してターミナル・ドライバが期待するデータ・ビットの数を制御する (値は 5 ~ 8 の範囲でなければならない)。
NUMBER_FN_KEYS	×	ファンクション・キーの数を指定する。
PRIVATE_NUM_1 to 10	×	設定されている場合には、これらのフィールドはユーザ定義機能 1 ~ 10 を示す。
ROWS	×	画面の行数を指定する。
WIDE_SCREEN_COLUMNS		ワイド・モードで使用できるカラム数を指定する。

たとえば、次に示す TERMTABLE エントリは、VT300 シリーズ・ターミナルの 2 つの属性を示しています。

```
NAME = "VT300_series"  
  NUMERIC  
    rows = 24,      columns = 80
```

### 5.2.3 文字列機能フィールド

文字列機能フィールドは複数の機能を備えています。これらのフィールドを使用すれば、次の操作が可能です。

- ラインを描くための描画文字を指定できる。
- プログラムがキャリッジ制御文字 (たとえば、フォーム・フィード) を実行するのではなく、表示できるようにアイコンを提供できる。
- 任意のタイプの漢字ターミナルで特定の操作 (たとえば、ERASE\_TO\_END\_OF\_LINE) を実行するための文字シーケンスを指定できる。
- 特定の漢字ターミナルで、特殊キー (たとえばファンクション・キー) によって戻される文字列を指定できる。
- 最大 255 バイトの文字列を指定できる (文字列の最大長はその文字列の複雑さに応じて異なるが、どの場合も 255 バイト以下である)。

表 5-3 は文字列機能を示しています。

表 5-3 文字列機能

VMS 名	SMG による 使用	説明
BEGIN_ALTERNATE_CHAR	×	代替文字集合を開始する。
BEGIN_AUTOPRINT_MODE	×	自動プリント・モードを開始する。
BEGIN_AUTOREPEAT_MODE	×	自動リピート・モードを開始する。
BEGIN_AUTOWRAP_MODE	×	自動ラップ・モードを開始する。
BEGIN_BLINK		点滅文字を開始する。
BEGIN_BOLD		ボールド文字を開始する。
BEGIN_DELETE_MODE	×	削除モードを開始する。
BEGIN_INSERT_MODE	×	挿入モードを開始する。
BEGIN_LINE_DRAWING_CHAR		ライン描画文字集合の使用を開始する。
BEGIN_NORMAL_RENDITION		通常ビデオ属性の使用を開始する。
BEGIN_REVERSE		文字の反転表示を開始する。
BEGIN_UNDERSCORE		下線付き文字を開始する。
BOTTOM_T_CHAR		ライン描画文字の " " を表示する。
CLEAR_TAB	×	現在のカラムのタブを消去する。
CR_GRAPHIC		制御文字を実行するのではなく、表現するときにキャリッジ・リターンを示すための文字を定義する。
CROSS_CHAR		垂直線の交差を表現するための文字を定義する。
CURSOR_DOWN	×	カーソルを n 行だけ下に移動する (スクロールは実行されない)。
CURSOR_LEFT	×	カーソルを n カラムだけ左に移動する。
CURSOR_NEXT_LINE	×	引数 n を受け付け、カーソルを n 行後の最初の位置に移動する。
CURSOR_POSITION_REPORT	×	2 つ引数を使用して現在の位置を報告する。
CURSOR_PRECEDING_LINE	×	引数 n を受け付けカーソルを n 行前の最初の位置に移動する。
CURSOR_RIGHT	×	引数 n を受け付けカーソルを右に n カラムだけ移動する。
CURSOR_UP	×	引数 n を受け付け、カーソルを上を n 行移動する (スクロールは実行されない)。
DARK_SCREEN		画面の背景色を暗くする (通常表示)。
DELETE_CHAR	×	引数 n を受け付け、n 文字を削除する。
DELETE_LINE	×	引数 n を受け付け n 行を削除する。
DEVICE_ATTRIBUTES	×	"What are you?" シーケンスに対する漢字ターミナルの応答。

(次ページに続く)

表 5-3 (続き) 文字列機能

VMS 名	SMG による 使用	説明
DOUBLE_HIGH_BOTTOM		行を縦横倍角文字の下半分に変更する。
DOUBLE_HIGH_TOP		行を縦横倍角文字の上半分に変更する。
DOUBLE_WIDE		行を横倍角文字に変更する。
END_ALTERNATE_CHAR	×	代替文字集合を終了する。
END_AUTOPRINT_MODE	×	自動プリント・モードを終了する。
END_AUTOREPEAT_MODE	×	自動リピート・モードを終了する。
END_AUTOWRAP_MODE	×	自動ラップ・モードを終了する。
END_BLINK	×	点滅文字を終了する。
END_BOLD	×	ボールド文字モードを終了する。
END_DELETE_MODE	×	削除モードを終了する。
END_INSERT_MODE	×	挿入モードを終了する。
END_LINE_DRAWING_CHAR		ライン描画文字を終了する。
END_REVERSE	×	文字の反転表示を終了する。
END_UNDERSCORE	×	下線付けを終了する。
ERASE_DISPLAY_TO_CURSOR	×	ディスプレイを仮想カーソルの位置まで消去する。
ERASE_LINE_TO_CURSOR	×	行を仮想カーソルの位置まで消去する。
ERASE_TO_END_DISPLAY	×	ディスプレイの最後まで消去する。
ERASE_TO_END_LINE		行の最後まで消去する。
ERASE_WHOLE_DISPLAY		ディスプレイ全体を消去する。
ERASE_WHOLE_LINE	×	行全体を消去する。
ERROR_ICON		エラーを示す文字を定義する。
FF_GRAPHIC		制御文字を実行するのではなく、表示するときに、改ページ(フォーム・フィールド)を示すためにこの文字を使用する。
HOME		カーソルのホーム・ポジションを定義する。
HORIZONTAL_BAR		ライン描画文字の " " を表示する。
HT_GRAPHIC		制御文字を実行するのではなく、表示するときに、水平タブを示すためにこの文字を使用する。
INDEX	×	カラム位置を変更せずに、カーソルを 1 行だけに下に移動する (必要に応じて画面の内容は上にスクロールされる)。
INIT_STRING		ターミナル初期化文字列を定義する。
INSERT_CHAR	×	引数 n を受け付け、n 文字を挿入する。
INSERT_LINE	×	引数 n を受け付け、n 行を挿入する。
INSERT_PAD	×	引数 n を受け付け、挿入した文字の後に n 文字のパッド文字を挿入する。

(次ページに続く)

表 5-3 (続き) 文字列機能

VMS 名	SMG による 使用	説明
KEY_0		アプリケーション・モードでキーパッドの 0 によって戻される。
KEY_1		アプリケーション・モードでキーパッドの 1 によって戻される。
KEY_2		アプリケーション・モードでキーパッドの 2 によって戻される。
KEY_3		アプリケーション・モードでキーパッドの 3 によって戻される。
KEY_4		アプリケーション・モードでキーパッドの 4 によって戻される。
KEY_5		アプリケーション・モードでキーパッドの 5 によって戻される。
KEY_6		アプリケーション・モードでキーパッドの 6 によって戻される。
KEY_7		アプリケーション・モードでキーパッドの 7 によって戻される。
KEY_8		アプリケーション・モードでキーパッドの 8 によって戻される。
KEY_9		アプリケーション・モードでキーパッドの 9 によって戻される。
KEY_BACKSPACE	×	バックスペース・キーによって戻される。
KEY_COMMA		キーパッドのコンマ・キーによって戻される。
KEY_DOWN_ARROW		□キーによって戻される。
KEY_E1		E1(編集キー 1) によって戻される。
KEY_E2		E2(編集キー 2) によって戻される。
KEY_E3		E3(編集キー 3) によって戻される。
KEY_E4		E4(編集キー 4) によって戻される。
KEY_E5		E5(編集キー 5) によって戻される。
KEY_E6		E6(編集キー 6) によって戻される。
KEY_ENTER (k)		キーパッドのEnter キーによって戻される。
KEY_F1		F1キー (ファンクション・キー 1) によって戻される。
~		
KEY_F20		F20キー (ファンクション・キー 20) によって戻される。
KEY_LABEL_F1	×	F1キー (ファンクション・キー 1) 上の記号。
~		

(次ページに続く)

表 5-3 (続き) 文字列機能

VMS 名	SMG による 使用	説明
KEY_LABEL_F20	×	<input type="checkbox"/> F20 キー (ファンクション・キー 20) 上の記号。
KEY_LEFT_ARROW		<input type="checkbox"/> キーによって戻される。
KEY_MINUS		キーパッドのマイナス・キーによって戻される。
KEY_PERIOD		キーパッドのピリオド・キーによって戻される。
KEY_PF1		<input type="checkbox"/> PF1 キーによって戻される。
KEY_PF2		<input type="checkbox"/> PF2 キーによって戻される。
KEY_PF3		<input type="checkbox"/> PF3 キーによって戻される。
KEY_PF4		<input type="checkbox"/> PF4 キーによって戻される。
KEY_RIGHT_ARROW		<input type="checkbox"/> キーによって戻される。
KEY_UP_ARROW		<input type="checkbox"/> キーによって戻される。
LEFT_T_CHAR		ライン描画文字の " " を表示する。
LF_GRAPHIC		制御文字を実行するのではなく、表示するときに、改行 (ライン・フィード) を示すためにこの文字を使用する。
LIGHT_SCREEN		画面の背景色を明るくする (反転表示)。
LOWER_LEFT_CORNER		ライン描画文字の " " を表示する。
LOWER_RIGHT_CORNER		ライン描画文字の " " を表示する。
NAME		ターミナル名を定義する。エントリ内の最初のフィールドでなければならない。
NEWLINE_CHAR	×	復帰改行 (ニューライン) 文字を定義する。
NEXT_LINE	×	次の行を表示する。
NO_PRINTER	×	プリンタが接続されていない状態を定義する。
PAD_CHAR	×	パッド文字を定義する (NULL 以外の場合)。
PRINT_SCREEN	×	画面の内容をプリントする。
PRINTER_READY	×	プリンタの使用可能 (ready) 状態を定義する。
PRINTER_NOT_READY	×	プリンタの使用不可能 (not ready) 状態を定義する。
PRIVATE_STR_1	×	ユーザ定義機能 1
~		~
PRIVATE_STR_10	×	ユーザ定義機能 10
REQUEST_CURSOR_POSITION	×	現在のカーソルの位置を要求する。

(次ページに続く)

表 5-3 (続き) 文字列機能

VMS 名	SMG による 使用	説明
REQUEST_PRINTER_STATUS	×	接続されているプリンタの状態を要求する。
RESTORE_CURSOR	×	前に保存した位置にカーソルを戻す。
REVERSE_INDEX	×	前の行の同じカラムにカーソルを移動する (必要に応じて、画面の内容は下にスクロールされる)。
RIGHT_T_CHAR		ライン描画文字の " " を表示する。
SAVE_CURSOR	×	カーソルの位置を保存する。
SCROLL_FORWARD	×	引数 n を受け付け、前向きに n 行をスクロールする。
SCROLL_REVERSE		引数 n を受け付け、逆向きに n 行をスクロールする。
SEL_ERASE_TO_END_DISPLAY	×	カーソルの位置からディスプレイの最後まで選択的に消去する (属性は変更しない)。
SEL_ERASE_TO_END_LINE	×	カーソルの位置から行の最後まで選択的に消去する (属性は変更しない)。
SEL_ERASE_WHOLE_DISPLAY	×	ディスプレイ全体を選択的に消去する (属性は変更しない)。
SEL_ERASE_WHOLE_LINE	×	行全体を選択的に消去する (属性は変更しない)。
SET_APPLICATION_KEYPAD		アプリケーション・キーパッド・モードを開始する。
SET_CHAR_NOT_SEL_ERASE	×	後続のすべての文字を選択的に消去不可能として指定する。
SET_CHAR_SEL_ERASE	×	後続のすべての文字を選択的に消去可能として指定する。
SET_CURSOR_ABS		カーソルのアドレスを指定する (行引数とカラム引数を受け付ける)。
SET_CURSOR_OFF		カーソルを表示されない状態に設定する。
SET_CURSOR_ON		カーソルを表示される状態に設定する。
SET_JUMP_SCROLL		スクロールをジャンプ・スクロールに設定する。
SET_NUMERIC_KEYPAD		キーパッド・アプリケーション・モードを終了する (数値モードを再開する)。
SET_ORIGIN_ABSOLUTE	×	現在のスクロール領域の外部にカーソルを移動することを許可する。
SET_ORIGIN_RELATIVE	×	現在のスクロール領域の外部にカーソルを移動することを禁止する。
SET_PRINTER_OUTPUT	×	画面ではなく、プリンタ・ポートに出力を送信する。
SET_SCREEN_OUTPUT	×	ターミナル画面に出力を送信する。

(次ページに続く)

表 5-3 (続き) 文字列機能

VMS 名	SMG による 使用	説明
SET_SCROLL_REGION		スクロール領域を設定する (引数として上マージンと下マージンを受け付ける)。
SET_SMOOTH_SCROLL		スクロールを連続スクロールに設定する。
SET_TAB	×	現在のコラムにタブを設定する。
SINGLE_HIGH		この行の高さと幅を通常表示に変更する。
TAB_CHAR	×	タブ文字を定義する (Ctrl/I またはパッドを含むタブ以外の文字)。このフィールドは、ASCII 以外の漢字ターミナルの場合にだけ使用しなければならない。
TOP_T_CHAR		ライン描画文字の " " を表示する。
TRUNCATION_ICON		オーバーフロー文字が切り捨てられたことを示す文字を定義する。
UNDERLINE_CHAR	×	文字に下線を付ける。
UPPER_LEFT_CORNER		ライン描画文字の " " を表示する。
UPPER_RIGHT_CORNER		ライン描画文字の " " を表示する。
VERTICAL_BAR		ライン描画文字の " " を表示する。
VT_GRAPHIC		制御文字を実行するのではなく、表示するときに、垂直タブを示す文字を定義する。
WIDTH_NARROW		漢字ターミナルの幅を通常幅 (80 コラム) に設定する。
WIDTH_WIDE		漢字ターミナルの幅を拡大幅 (132 コラム) に設定する。

表 5-4 は STRING\_2 機能を示しています。SMG\$M\_USER1 から SMG\$M\_USER8 のユーザ定義レンディションのいずれかを作成する場合には、STRING\_2 機能を使用して TERMTABLE.TXT ファイルに適切な定義を登録しなければなりません。

表 5-4 String\_2 機能

VMS 名	SMG による 使用	説明
BEGIN_STATUS_LINE		ハードウェア状態行への出力を開始する。
BEGIN_USER1		最初のユーザ定義属性を開始する。
BEGIN_USER2		2 番目のユーザ定義属性を開始する。
BEGIN_USER3		3 番目のユーザ定義属性を開始する。
BEGIN_USER4		4 番目のユーザ定義属性を開始する。
BEGIN_USER5		5 番目のユーザ定義属性を開始する。

(次ページに続く)

表 5-4 (続き) String\_2 機能

VMS 名	SMG による 使用	説明
BEGIN_USER6		6 番目のユーザ定義属性を開始する。
BEGIN_USER7		7 番目のユーザ定義属性を開始する。
BEGIN_USER8		8 番目のユーザ定義属性を開始する。
END_STATUS_LINE		ハードウェア状態行への出力を終了する。
END_USER1	×	最初のユーザ定義属性を終了する。
END_USER2	×	2 番目のユーザ定義属性を終了する。
END_USER3	×	3 番目のユーザ定義属性を終了する。
END_USER4	×	4 番目のユーザ定義属性を終了する。
END_USER5	×	5 番目のユーザ定義属性を終了する。
END_USER6	×	6 番目のユーザ定義属性を終了する。
END_USER7	×	7 番目のユーザ定義属性を終了する。
END_USER8	×	7 番目のユーザ定義属性を終了する。
BLACK_SCREEN		画面の背景色を黒に設定する。
BLUE_SCREEN		画面の背景色を青に設定する。
CYAN_SCREEN		画面の背景色を青緑に設定する。
GREEN_SCREEN		画面の背景色を緑に設定する。
MAGENTA_SCREEN		画面の背景色を紫に設定する。
RED_SCREEN		画面の背景色を赤に設定する。
WHITE_SCREEN		画面の背景色を白に設定する。
YELLOW_SCREEN		画面の背景色を黄色に設定する。
USER1_SCREEN		ユーザ定義背景色。
USER2_SCREEN		ユーザ定義背景色。

文字列機能フィールドには、印字不可能な文字が含まれることが多いため、これらの文字を機能文字列に容易に挿入できるようにするために、次に示す置換文字を使用します。プリントされない文字を表現するには特殊文字を使用します。

特殊文字	プリントされない文字	意味
\$	ESCAPE	10 進数で 27 の ASCII 文字
^	CONTROL	コントロール・シフト
&	CSI	10 進数で 155 の ASCII 文字
@	SS3	10 進数で 143 の ASCII 文字

したがって、エスケープ文字を含む機能文字列を作成する場合には、その位置に単にドル記号を指定します。制御文字を含む機能文字列を作成する場合には、文字の前にサーカンフレックス(^)を指定します。次の例を参照してください。

## フォーリン・ターミナルのサポート

### 5.2 機能フィールド

```
NAME = "VT300_series"
.
.
.
STRING
    begin_alternate_char = "^N",
    end_alternate_char = "^O",
    erase_whole_display = "$[2J"
.
.
.
END
```

機能文字列内で文字を使用し、その文字の通常の ASCII 値を使用する場合には、その文字の前にアンダースコアを指定します (たとえば、“\_ \$”と指定すると、アンダースコアの後にエスケープ文字が挿入されるのではなく、1つのドル記号が挿入されます)。次に示す文字を通常の ASCII テキストとして取り扱うには、その文字の前にアンダースコアを指定しなければなりません。

アンパーサンド	&
アポストロフィ	'
アットマーク	@
引用符	"
サーカンフレックス	^
ドル記号	\$
感嘆符	!
左括弧	(
アンダースコア	_

日本語 SMG はライン描画文字集合 (たとえば, `bottom_t_char`, `top_t_char` など) を表示するために, 必要なグラフィック・モードを自動的に起動します。しかし, フォーリン・ターミナル・ルーチンを直接呼び出す場合には, ユーザが必要なグラフィック・モードを起動しなければなりません。

漢字ターミナルがコマンドを実行するための十分な時間を確保できるように, パッド文字 (たとえば, NULL 文字) をターミナル・コマンドに追加しなければならないことがあります。必要なパッド文字の数は, 使用する漢字ターミナルとポー・レートに応じて異なります。パッド文字機能フィールドは将来の拡張に備えて準備されているものであり, このリリースでは機能しません。パッド文字を挿入する操作はユーザが実行しなければなりません。

フォーリン・ターミナル・サポート・ルーチンを直接呼び出す場合には, 多くの文字列機能フィールドは引数を使用し, その引数の値を実行時に指定しなければなりません。さらに, 引数の中には, 引数に対して値を代入するときに算術演算を実行しなければならないものがあります。この後の節では, 引数の置換と算術演算について説明します。

### 5.2.4 引数の置き換え

多くの場合、ターミナル・コマンド文字列内の値を置き換えなければなりません。たとえば、スクロール領域を設定する場合や、カーソルを右に 10 カラムだけ移動する場合には、実行時に値を置き換える必要があります。これらの値は TERMTABLE ターミナル定義には登録できません。TERMTABLE は文字列を置き換えるために、!UL を受け付けます。これは \$FAO 形式のディレクティブです。!UL ディレクティブは、その場所に値を挿入することを示します。

TERMTABLE インターフェイス・ルーチンは機能フィールド文字列 (および戻されるコマンド文字列) 内の値を置き換える前に、符号なしロングワードを受け付け、それを ASCII の数値に変換します。次の例を参照してください。

```
NAME = "VT300_series"
.
.
.
STRING
set_cursor_abs = "$[!UL;!ULH"
.
.
.
END
```

SET\_CURSOR\_ABS に対して定義される文字列には、2 つの !UL ディレクティブに対して置き換えられる値を指定しておかなければなりません。これらの値は、カーソルを設定する行番号とカラム番号を示します。これらの実行時引数は、SMG\$GET\_TERM\_DATA ルーチンに対する省略可能なロングワード・ベクタ引数として指定します。ベクタ内の最初のエントリには、その後続く引数の数が指定されます。したがって、最初のエントリが 2 の場合には、2 番目のエントリは行番号を示し、3 番目のエントリはカラム番号です。SMG\$GET\_TERM\_DATA ルーチンは最初のデータ項目 (ベクタ内の 2 番目の項目) を ASCII の対応する数値に変換し、最初の !UL ディレクティブに対してその ASCII 値を挿入します。その後、2 番目のデータ項目を変換し、その結果を 2 番目の !UL ディレクティブに対して挿入します。以下も同様です。

### 5.2.5 算術演算

引数の置き換えの他に、ターミナル・コマンド・シーケンスは算術演算も必要とすることがあります。引数の置き換えと算術演算を実行する場合には、TERMTABLE エントリは、引数変換および置換の場合と異なる方式を必要とします。

引数の置き換えと算術演算のどちらも実行する場合には、左括弧、パーセント記号 (置換の場所を示す)、算術演算子、オペランド、右括弧を使用します。次の例を参照してください。

```

NAME = "VT52"
.
.
.
STRING
set_cursor_abs = "$Y(%1+31)(%2+31)"
.
.
.
END

```

この例では、VT52 でカーソルを直接移動する文字列を示しています。ただし、パイアスを行番号とカラム番号に加算しなければなりません。数式内で置換される値は、引数置換を単独で実行する場合と同じ方法で、SMG\$GET\_TERM\_DATA ルーチンによって渡されます。パーセント記号の後に常に整数が続きます。この整数は引数を置換する順序を示します。

次の表は算術演算で使用する文字を示しています。

文字	意味
(	数式の先頭
%n	n番目のユーザ引数を置換する
+	加算演算子
-	減算演算子
*	乗算演算子
/	除算演算子
)	数式の最後

画面の座標を表現するには、ロングワード整数で十分です。数式は左から右に評価されます。演算子に優先順位はありません。

項目間のスペースは意味がありません。数式を読みやすくするために、スペースを任意に使用してもかまいません。機能文字列の長さは128バイトに制限されています。

### 5.3 OpenVMS ターミナル機能ファイルの作成

ターミナル機能データベースのソース・コードはTERMTABLE.TXT という名前のASCIIファイルです。このファイルには、各タイプの漢字ターミナルに対してエントリが登録されています。各エントリには、漢字ターミナルの機能をはじめ、初期化シーケンスや画面のサイズなどの装置固有の情報が登録されています。ひとつのTERMTABLE エントリからファイル内の複数のレコードを参照できます。ターミナル定義の追加はTERMTABLE.TXT ファイルを編集することにより可能です。ただし、その後SYS\$SYSTEM:SMGBLDTRM.EXEを実行し、TERMTABLE.TXTを再処理しなければなりません。

TERMTABLE.TXT は任意のテキスト・エディタを使用して作成できます。  
TERMTABLE エントリは 1 つのターミナル名、任意の数の機能フィールドとその値で構成されます (機能フィールドについての詳しい説明は、第 5.2 節を参照してください)。TERMTABLE.TXT は、コンパイルのために適切な形式に設定しなければなりません。機能名はその機能をわかりやすく表現しており、簡単に理解できます。ターミナル名は固有の名前でなければなりません。たとえば、フォーリン・ターミナルに対して複数の定義が必要な場合には、複数の名前を使用しなければなりません。

TERMTABLE ルーチンが最初にターミナル・エントリを検索する場合には、TERMSTABLOC という論理名の領域から TERMTABLE.EXE を検索しようとします。指定されたターミナル・エントリをその領域から検索できない場合には、ルーチンは SYSS\$SYSTEM から TERMTABLE.EXE を検索します。ある漢字ターミナルに対して、システム定義と異なるターミナル定義を使用する場合には、TERMTABLE.TXT と TERMTABLE.EXE のプライベート・コピーを作成できます。そうすれば、SYSS\$SYSTEM:TERMTABLE.EXE に登録されている定義と異なる定義を使用してその漢字ターミナルを定義でき、他の漢字ターミナルに対しては、標準的なシステム定義を使用できます。

TERMTABLE エントリの形式は次のとおりです。

```
NAME = "terminal-name"  
      capability-field [...]  
END
```

TERMTABLE.TXT ファイルでは、REQUIRE ディレクティブを使用できます。REQUIRE ディレクティブを使用すれば、TERMTABLE.TXT ファイルに別のソース・ファイルを取り込むことができます。その形式は次のとおりです。

```
REQUIRE "filespec"
```

この形式で、“filespec”は適切な OpenVMS ファイル指定です。

---

## 5.4 例

```
!  
! Private versions of DIGITAL terminal definitions  
!  
NAME = 'myvt300'  
  
BOOLEAN  
ansi_crt = 1,          dec_crt = 1  
  
NUMERIC  
rows = 24,            columns = 80,  
wide_screen_columns = 132
```

```

STRING
begin_alterate_char = "^N",
end_alterate_char = "^O",
erase_whole_display = "$[2J",
init_string = "$_(B",
set_cursor_abs = "$[!UL;!ULH"
END
NAME = "MYVT52"
BOOLEAN
ansi_crt = 0,          dec_crt = 1
NUMERIC
rows = 24,            columns = 80,
wide_screen_columns = 80
STRING
begin_alterate_char = "$F"
end_alterate_char = "$G",
erase_whole_display = "$Y(32)(32)$J", !position to 1,1; then erase
set_cursor_abs = "$Y(%1+31)(%2+31)"
END

```

VT300 シリーズ (MYVT300) に対して登録されているカーソル設定シーケンスの場合、戻される文字列は、SMG\$GET\_TERM\_DATA ルーチン呼び出すときに指定した引数ベクタ内の値に応じて異なります。たとえば、カーソルを 3 行目、12 カラム目に移動する場合には、これらのロングワード値をベクタ内の 2 番目のエン트리と 3 番目のエントリに指定します (最初のエントリはその後に続く値の個数です)。SMG\$GET\_TERM\_DATA ルーチンはこれらのロングワード値を ASCII 値に変換し、変換した値に対応する !UL ディレクティブの位置に戻される文字列に挿入します。

VT52 (MYVT52) に対して登録されているカーソル設定シーケンスの場合には、戻される文字列は引数置換によってではなく、算術演算に応じて変化します (VT52 はバイアス操作を必要とするため)。SMG\$GET\_TERM\_DATA ルーチンに対する引数ベクタのエントリ 2 とエントリ 3 に指定した行番号とカラム番号に 31 (10 進数) を加算するために、算術演算子を使用しています。

MYVT300 内の INIT\_STRING フィールドは、括弧が、通常は数式を示す特殊文字として取り扱われることを示しています。括弧を通常のテキスト文字として解釈するには、その前にアンダースコアを指定しなければなりません。したがって、“\$\_(B” という文字列は ESC(B になります。これは ASCII 文字集合を G0 に設定するためのコマンドです。

MYVT52 に対する ERASE\_WHOLE\_DISPLAY シーケンスは、特定の機能を提供するために、シーケンスを組み合わせなければならない可能性があることを示しています。VT52 には画面全体を消去するコマンドがありません。しかし、カーソルをホーム・ポジションに移動し、現在の位置から画面の最後まで消去するコマンドを使用すれば、画面全体を消去できます。

---

## 5.5 TERMTABLE.EXE の作成

各プログラムが ASCII ファイルをアクセスするのは非効率的です。これは、ASCII テキストを、漢字ターミナルに対して使用可能な文字列として渡すには、その前にバイナリ情報として処理しておかなければならないからです。各イメージを開始するときはこの処理を実行する必要がないようにするために、TERMTABLE.TXT は必要なバイナリ形式に“プリコンパイル”されています。この場合、画面アプリケーションはプリコンパイルされたバイナリの機能ファイルからターミナル・シーケンスを検索できます。

TERMTABLE.TXT をコンパイルするには、SYSSSYSTEM:SMGBLDTRM .EXE プログラムを実行します。このユーティリティは入力ファイル TERMTABLE.TXT を受け付け、TERMSTABLOC という論理名によって示される装置とディレクトリに、出力ファイルとして TERMTABLE.EXE を作成します。

コンパイルされたターミナル機能は、グローバル・セクションにマップされるファイルにテーブルとして登録されます。したがって、ユーザ・プログラムはグローバル・セクションを参照すれば良く、機能データのコピーを独自に持つ必要がありません。

ユーザが自分のディレクトリから個人用の TERMTABLE.TXT をコンパイルする場合には、インターフェイス・ルーチンはそれを一時的なセクションとしてマッピングすることによりアクセスします。TERMTABLE インターフェイス・ルーチンはシステムのグローバル・セクションを検索する前に、一時的なセクションから定義を検索します。

システム管理者は、標準でない定義がユーザの個人領域に制限されるようにするために、ターミナル定義を調整しなければならないことがあります。

ほとんどのユーザはグローバル・セクションを作成するための特権を与えられていません。SYSSSYSTEM:SMGMAPTRM.EXE という短いプログラムはコンパイルされた TERMTABLE をグローバル・セクションとしてマップするプログラムであり、標準的な OpenVMS スタートアップ・プロシージャの一部です。変更した TERMTABLE.EXE をグローバル・セクションとしてマップするには、既存のグローバル・セクションを最初に削除しておかなければなりません。システムが動作中にグローバル・セクションを削除すると、ユーザ・プログラムが異常終了する可能性があります。したがって、変更した TERMTABLE.EXE を省略時の設定として定義するには、システムを再ブートしなければなりません。

コンパイル時間と作成されるグローバル・セクションのサイズを削減するために、SYSSSYSTEM:TERMTABLE.TXT 内のターミナル定義は最低限に抑えなければなりません。実際にコンピュータ・システムに接続される漢字ターミナルのタイプだけを定義するようにしてください。

---

## 5.6 画面管理が使用する機能フィールド

第 5.2 節の表は、日本語 SMG が特定の機能文字列を要求できるかどうかを示しています。ライン描画などの一部の機能は、ラインを描く日本語 SMG ルーチン呼び出した場合などにだけ要求されます。通常のテキストを画面に書き込む操作だけが実行しない場合には、次のフィールドだけを定義しておけば十分です。

基本的な機能

- NAME
- SET\_CURSOR\_ABS

SET\_CURSOR\_ABS を省略した場合には、日本語 SMG は漢字ターミナルをハード・コピー装置として取り扱います (ハード・コピー装置に対する日本語 SMG の使用は、本バージョンではサポートされません)。

次のオプション機能を指定した場合には、日本語 SMG 操作はより効率的になります。

- ERASE\_TO\_END\_DISPLAY
- ERASE\_TO\_END\_LINE
- SET\_SCROLLING\_REGION

ERASE\_TO\_END\_DISPLAY, ERASE\_TO\_END\_LINE, SET\_SCROLLING\_REGION を指定しなかった場合には、日本語 SMG はこれらの機能を実行するために空白を挿入します。しかし、空白を挿入する操作は時間のかかる操作です。同様に、ハードウェアのスクロール機能を使用すれば、出力の速度を向上できます。スクロール機能を使用できない場合には、日本語 SMG は画面全体を描き直さなければなりません。

日本語 SMG は ASCII 文字集合を使用します。漢字ターミナルにライン描画文字セットが備えられている場合には、ライン描画文字 (bottom\_t\_char, horizontal\_bar など) を定義しなければなりません。ライン描画文字が定義されていない場合には、SMG は境界を描くために通常の ASCII 文字を使用します。

また日本語 SMG は、ターミナル・ドライバを管理しているターミナル属性にも依存します。これらの属性は、DCL の SET TERMINAL コマンドを使用して変更できます。たとえば、SET TERMINAL/NOTAB と入力すると、日本語 SMG は漢字ターミナルにタブを送信しなくなります。

---

## 5.7 フォーリン・ターミナルに対する入力サポート

フォーリン・ターミナルとは、OpenVMS が認識する標準的なターミナル以外の装置タイプの漢字ターミナル、あるいは ANSI\_CRT 属性が設定されていない漢字ターミナルです。

## フォーリン・ターミナルのサポート

### 5.7 フォーリン・ターミナルに対する入力へのサポート

ANSI ターミナルを使用する場合には、ファンクション・キーやキーボード・キーなどの特殊なキーを入力すると、エスケープ・シーケンス (ANSI 標準規格によって定義されているシーケンス) が OpenVMS ターミナル・ドライバに送信されます。OpenVMS ターミナル・ドライバはこの ANSI 標準規格を認識し、この標準規格に従って、エスケープ・シーケンスを解釈します。したがって、OpenVMS ターミナル・ドライバは、エスケープ・シーケンスの長さ、そのシーケンス内のどの位置でどの文字が認められるかを認識しています。

OpenVMS ターミナル・ドライバは、エスケープ・シーケンス内のプリント文字をエコー表示しません。これは、これらの文字がエスケープ・シーケンスの一部として特殊な意味を持つものとして解釈されるからです。ただし、TRMSM\_TM\_NOECHO 修飾子を指定しない限り、通常のキーはエコー表示されます。

OpenVMS ターミナル・ドライバはエスケープ・シーケンス、シーケンスの長さ、ファンクション・キーが押される前に入力された文字数を日本語 SMG ルーチンに戻します。日本語 SMG ルーチンは、シーケンスとその長さを TERMTABLE.EXE 内の特定の漢字ターミナルに対するキー定義のリストと比較することにより、どのキーが押されたかを判断します。このコードは SMG\$K\_TRM\_xxx という形式でユーザに戻されます。ただし、xxx は特定のキーを指します。

フォーリン・ターミナルでファンクション・キーやキーボード・キーなどの特殊なキーを押した場合には、ANSI 標準ではないシーケンスが OpenVMS ターミナル・ドライバに送信されます。このシーケンスが制御文字から始まる場合には、OpenVMS ターミナル・ドライバはこの文字を終了文字として解釈します (省略時の設定では、特に他の目的で終了文字マスクを使用しない限り、すべての制御文字は終了文字になります)。その後、ターミナル・ドライバは文字の読み込みを停止し、その文字、長さ 1、およびファンクション・キーが押される前に入力された文字数を日本語 SMG ルーチンに戻します。

日本語 SMG ルーチンは、戻された文字を調べます。その文字が制御文字の場合には、日本語 SMG ルーチンは先読みバッファにシーケンスの次の文字が登録されていないかどうかを確認します。先読みバッファに文字が登録されている場合には、日本語 SMG ルーチンは先読みバッファから 1 文字を読み込み、それを前に読み込んだ制御シーケンスの後に追加し、この新しいシーケンスを TERMTABLE.EXE 内のこのターミナルに対するキー定義のリストと比較し、どのキーが押されたかを判断します。シーケンスが一致しない場合には、先読みバッファから次の文字が読み込まれます。一致するものが検出されるまで、または先読みバッファが空になるまで、この処理が継続的に実行されます。

ターミナル・ドライバはこのシーケンスに関して知識を持たないため、エコー表示を禁止する修飾子をユーザが指定しない限り、シーケンス内のプリント可能な文字はターミナル・ドライバによってエコー表示されます。日本語 SMG ルーチンにはどの文字がこのシーケンスを構成するのかわからないため、シーケンスを構成する実際の文字を日本語 SMG ルーチンに戻すことができるようにするために、行編集機能が禁止されます。

特殊キーが制御文字から始まらないシーケンスを送信する漢字ターミナルは、入力の対象として日本語 SMG ルーチンでサポートされません。ファンクション・キー・シーケンスを開始する制御文字を除外するように終了文字マスクを変更する操作はサポートされません。さらに、入力を実行するフォーリン・ターミナルの性能は、同じ入力を実行する ANSI ターミナルの性能と一致しません。これは、日本語 SMG ルーチンが OpenVMS ターミナル・ドライバのかわりにシーケンスを解析しなければならないからです。

---

## 5.8 SET TERMINAL コマンドと SHOW TERMINAL コマンドのサポート

DCL の SET TERMINAL コマンドは、TERMTABLE 定義に従うように漢字ターミナルを設定します。SET TERMINAL コマンドを使用すると、TERMTABLE データベースから次の 3 つのフィールドが検索され、漢字ターミナルに対して設定されます。

1. ターミナル・タイプ - システムが割り当てる、特定の装置タイプに関連する符号付き整数
2. 幅 - 物理画面のカラム数
3. ページ・サイズ - 画面の行数

さらに、対応する論理値機能がターミナル定義に設定されている場合には、次のフラグが設定されます。

- ADVANCED\_VIDEO
- ANSI\_CRT
- BLOCK\_MODE
- DEC\_CRT
- EDIT
- EIGHT\_BIT
- FORM
- FULLDUP
- LOWERCASE
- REGIS
- SCOPE
- SIXEL\_GRAPHICS
- SOFT\_CHARACTERS
- TAB

これらのフィールドのいずれかが定義から脱落している場合には、その属性に対しては、以前の設定が保存されます。SET TERMINAL コマンドは漢字ターミナルに対してその属性を設定しようとしません。予測できない設定が発生するのを防止するには、ここに示したすべての機能を定義に指定しなければなりません。

SET TERMINAL コマンドは、VT300 シリーズ、VT200 シリーズ、VT100、および VT52 などの認識される漢字ターミナルに対しては、常に同様に動作します。SET TERMINAL が認識しない装置名を検出した場合には、TERMTABLE からその名前を持つ定義を検索します。ユーザ固有の定義は、SET TERMINAL が現在認識する名前以外の名前ではなければなりません。現在認識される漢字ターミナルは次に示すとおりです。

LA12	LQP02	VT102
LA34	VT05	VT125
LA36	VT52	VT131
LA38	VT55	VT132
LA100	VT80	VT200-SERIES
LA120	VT100	VT300-SERIES
Unknown	VT101	FT1 から FT8

SET TERMINAL が独自の内部テーブルから装置名を検索する場合には、TERMTABLE データベースを検索しません。

SET TERMINAL コマンドは装置名の最初の 15 文字だけを認識するため、ターミナル名は 15 文字に制限することが必要です。

SET TERMINAL/DEVICE\_TYPE=name コマンドを使用した場合、指定された漢字ターミナルが OpenVMS オペレーティング・システムによって認識されないときは、TERMTABLE データベースからその漢字ターミナルが検索されます。

SET TERMINAL/DEVICE\_TYPE=name コマンドはその後、次の表に示すように、TERMTABLE データベース内の機能の存在をもとに、いろいろなターミナル属性を設定します。

機能フィールド	ターミナル属性
LOWERCASE	LOWERCASE
PHYSICAL_TABS	TABS
SCOPE	SCOPE
EIGHT_BIT	EIGHTBIT
PHYSICAL_FF	FORM
FULLDUP	FULLDUP
SIXEL_GRAPHICS	SIXEL
SOFT_CHARACTERS	SOFT
ANSI_CRT	ANSI_CRT
REGIS	REGIS
BLOCK_MODE	BLOCK
ADVANCED_VIDEO	AVO
EDIT_MODE	EDIT
DEC_CRT	DEC_CRT

TERMTABLE ターミナルに対しては、SET TERMINAL/DEVICE\_TYPE=形式のコマンドを使用しなければなりません。SET TERMINAL/name は古い形式のコマンドであり、一部の装置名に対してだけしか機能せず、OpenVMS の前のバージョンとの互換性を維持する目的でのみ使用できます。

## 日本語 SMG を使用したプログラム開発

この章では、新しいプログラムを開発するための日本語 SMG の望ましい使用方法について説明します。この章の説明では特に、日本語 SMG ルーチンが AST リエントラントでないことに注意してください。

アプリケーションから日本語 SMG ルーチンを呼び出す方法としては、次の 2 種類の方法があります。

- 直接呼び出し

日本語 SMG を直接呼び出すアプリケーションは、ペーストボードと仮想ディスプレイを使用している。

- 間接呼び出し

この種のアプリケーションは日本語 SMG を直接使用しないが、他のルーチン中で、日本語 SMG を使用する。

今後、これまでより多くの呼び出し可能ルーチンが日本語 SMG を使用して出力を作成できるようになります。そのようになると、アプリケーションがこのカテゴリに分類されるかどうかを判断するのは困難になります。

どちらの場合も、呼び出しルーチンは何らかの時点でサブシステムを呼び出すことにより、サブシステムが画面にデータを書き込むことができるようになります。

その後、端末ユーザは、サブシステム固有のディスプレイを消去する必要があるでしょう。しかしサブシステムが、データを表示するために仮想ディスプレイを作成し使用した場合には、ディスプレイ識別子は呼び出しプログラムからは利用できません。したがって、呼び出しプログラムはディスプレイを消去できません。さらに、呼び出しプログラムが日本語 SMG を使っていない場合、画面のペーストボード識別子も呼び出しプログラムから利用できません。

この問題を解決するには、日本語 SMG を直接的または間接的に使用するすべての呼び出し可能ルーチンが、pasteboard-id 引数に対して (省略可能な) 入力引数と、仮想 display-id 引数に対して (省略可能な) 出力引数を使用するようにしなければなりません。ペーストボード識別子とディスプレイ識別子を渡すことにより、呼び出しプログラムから消去できないサブシステム固有のデータが、画面に蓄積されるのを防止できます。これらのガイドラインをまとめると、次のようになります。

- pasteboard-id 引数が呼び出しプログラムから提供される場合には、
  1. 呼び出されるプログラムは、独自のペーストボードを作成してはならない。

2. 呼び出されるプログラムは、呼び出しプログラムから提供されるペーストボードにすべての出力を送信しなければならない。つまり、呼び出されるプログラムは、pasteboard-id 引数によって指定されるペーストボードにのみディスプレイをペーストできる。
  3. 呼び出されるプログラムは、SMG\$DELETE\_VIRTUAL\_DISPLAY ルーチンを呼び出すことにより、作成した仮想ディスプレイを削除できるが、ペーストボードを削除してはならない。  
仮想ディスプレイを後で再利用できると考えて、SMG\$UNPASTE\_VIRTUAL\_DISPLAY ルーチンを呼び出されるプログラムから単純に呼び出ししてはならない。呼び出されるプログラムと呼び出しプログラムはペーストボードを共用するため、呼び出しプログラムは SMG\$POP\_VIRTUAL\_DISPLAY ルーチンを使用して、呼び出されたプログラムが作成したすべてのディスプレイを削除できる。
  4. 呼び出されるプログラムは、そのプログラムが呼び出すルーチンに pasteboard-id 引数を渡さなければならない。このようにして、すべての出力は指定されたペーストボードに送られる。
- pasteboard-id 引数が呼び出しプログラムから提供されない場合には、
    1. 呼び出されるプログラムは、独自にペーストボードを作成しなければならない。呼び出されるプログラムは、他の方法によって特定の装置に出力を送る場合を除き、ペーストボードに対する物理装置を割り当てることができる。  
呼び出されるプログラムは、SMG\$CREATE\_PASTEBOARD ルーチン呼び出しの状態を確認することにより、固有のペーストボード識別子を作成したのか、既存のペーストボードのペーストボード識別子を受信したのかを判断しなければならない。ペーストボードがすでに存在する場合には、呼び出されるプログラムはそのペーストボードを削除してはならない。
    2. 呼び出されるルーチンがペーストボードを作成し、次にペーストボードを使用する可能性のあるサブルーチンを呼び出す場合には、pasteboard-id 引数をサブルーチンに渡さなければならない。
    3. 呼び出されるプログラムは、SMG\$UNPASTE\_VIRTUAL\_DISPLAY ルーチンを使用して物理画面から消去し、再起動される可能性がある場合には、将来の起動のために再利用できるようにディスプレイを保存できる。しかし、SMG\$UNPASTE\_VIRTUAL\_DISPLAY ルーチンを使用するのは、呼び出されるプログラムが独自のペーストボードを作成する場合に制限しなければならない。この場合、呼び出しプログラムは呼び出されるプログラムが作成した仮想ディスプレイを削除できないからである。
  - 仮想 display-id 引数を呼び出しプログラムが与える場合には、呼び出しプログラムは呼び出されるプログラムが作成した仮想ディスプレイを削除しなければならない。呼び出されるプログラムは最初にペーストした仮想ディスプレイの識別子を呼び出しプログラムに戻さなければならない。呼び出しプログラムは SMG\$POP\_VIRTUAL\_DISPLAY ルーチンを呼び出すことにより、この仮想ディスプレイとその後ペーストするすべての仮想ディスプレイを削除できる。

- 仮想 display-id 引数を呼び出しプログラムが与えない場合には、呼び出されるプログラムはペーストボードにペーストしたすべての仮想ディスプレイを削除しなければならない。

次のガイドラインに従えば、モジュール方式でアプリケーションを開発できます。

- 呼び出しプログラムは、どの情報をペーストボードにペーストするかを制御する。ペーストボード識別子は階層構造の上から下に流れ、各ルーチンは呼び出しプログラムが与えた `pasteboard-id` 引数を使用し、それをサブルーチンに渡す。
- 呼び出しプログラムが仮想 display-id 引数を与え、呼び出されるプログラムがその引数に識別子を格納する場合には、呼び出しプログラムは、呼び出されるプログラムが作成したディスプレイを削除しなければならないと解釈する。

呼び出されるプログラムは、最初にペーストした仮想ディスプレイの `display-id` 引数を戻すことにより、呼び出しプログラムが `SMG$POP_VIRTUAL_DISPLAY` ルーチンを呼び出すことで、このディスプレイと、その後ペーストしたすべてのディスプレイを削除できるようにする。

- 仮想ディスプレイは、それを必要とするルーチン内で作成される（およびペーストされる）。呼び出しプログラムが `display-id` 引数を与えない場合には、そのディスプレイを作成したルーチン内でディスプレイはアンペーストされるか、または削除される。

---

## 6.1 日本語 SMG を使用しないルーチンの呼び出し

日本語 SMG を使用せずに画面に書き込みを実行するサブルーチン（またはサブシステム）を呼び出す場合には、別の状況が発生します。日本語 SMG を使用しない場合には（つまり、画面上で日本語 SMG によって制御されない領域にテキストが配置される場合）、画面を更新しようとするときに問題が発生します。

この理由から、日本語 SMG には、画面管理を使用しないプログラムに画面（またはその一部）を一時的に渡すルーチンと、制御が `SMG$` ルーチン以外のルーチンから戻された後、前の状態に画面を復元するためのルーチンがあります。これらのルーチンは `SMG$SAVE_PHYSICAL_SCREEN` ルーチンと `SMG$RESTORE_PHYSICAL_SCREEN` ルーチンです。

`SMG$` 以外の入出力を、画面に対して実行するルーチンを呼び出す場合には、その前に `SMG$SAVE_PHYSICAL_SCREEN` ルーチンを呼び出し、画面のどの部分を `SMG$` 以外のルーチンに渡すかを指定します。 `SMG$SAVE_PHYSICAL_SCREEN` ルーチンは指定された領域を消去し、ターミナルの物理スクロール領域をこの領域に設定し、物理カーソルを領域の行 1、カラム 1 に設定します。 `SMG$` 以外のコードが順次入出力だけしか実行しない場合には（つまり、カーソル・アドレスを直接指定しない場合には）、出力は画面の指定された領域に制限されます。

## 日本語 SMG を使用したプログラム開発

### 6.1 日本語 SMG を使用しないルーチンの呼び出し

制御が SMG\$ 以外のルーチンから戻された後、SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチン呼び出してください。このルーチンは、SMG\$SAVE\_PHYSICAL\_SCREEN ルーチン呼び出す前の状態に画面イメージを復元します。

## 日本語 SMG ルーチンの呼び出しの例

この章では、VMS Calling Standard に準拠したおもな言語で SMG\$READ\_KEYSTROKE ルーチン呼び出す方法を示します。他の日本語 SMG ルーチン (たとえば、SMG\$CREATE\_VIRTUAL\_DISPLAY, SMG\$CREATE\_PASTEBOARD, SMG\$CREATE\_VIRTUAL\_KEYBOARD, SMG\$PASTE\_VIRTUAL\_DISPLAY, SMG\$PUT\_LINE) もこれらの例の中で使用しています。

例 7-1 の FORTRAN プログラムでは、SMG\$READ\_KEYSTROKE ルーチンの他、SMG\$CREATE\_VIRTUAL\_DISPLAY, SMG\$CREATE\_PASTEBOARD, SMG\$PASTE\_VIRTUAL\_DISPLAY, SMG\$CREATE\_VIRTUAL\_KEYBOARD, SMG\$PUT\_LINE ルーチンも使用しています。

### 例 7-1 FORTRAN での SMG\$ルーチンの使用

```

C+
C This routine creates a virtual display and writes ti to the PASTEBOARD.
C Data is placed in the virtual display using the routine SMG$PUT_CHARS.
C Include the SMG definitions. In particular, we want SMG$M_BORDER,
C-
      INCLUDE '($SMGDEF)'
      INTEGER SMG$CREATE_VIRTUAL_DISPLAY, SMG$CREATE_PASTEBOARD
      INTEGER SMG$PASTE_VIRTUAL_DISPLAY,
1      SMG$CREATE_VIRTUAL_KEYBOARD
      INTEGER SMG$READ_KEYSTROKE, SMG$PUT_LINE
      INTEGER DISPLAY1, PASTE1, KEYBOARD1, ROWS, COLUMNS,
1      TERM_CHAR
      CHARACTER*3 TEXT
      CHARACTER*27 TEXT_OUTPUT

C+
C Create the virtual display with a border.
C-
      ROWS = 7
      COLUMNS = 60

      ISTATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)

C+
C Create the pasteboard.
C-
      ISTATUS = SMG$CREATE_PASTEBOARD (PASTE1)

```

(次ページに続く)

## 日本語 SMG ルーチンの呼び出しの例

### 例 7-1 (続き) FORTRAN での SMG\$ルーチンの使用

```
C+
C Create a virtual keyboard
C-
      ISTATUS = SMG$CREATE_VIRTUAL_KEYBOARD (KEYBOARD1)
C+
C Paste the virtual display at row 3, column 9.
C-
      ISTATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 3, 9)
      ISTATUS = SMG$PUT_LINE (DISPLAY1,
1      'Enter the character K after the >> prompt.')
      ISTATUS = SMG$PUT_LINE (DISPLAY1,
1      'This character will not be echoed as you type it.')
      ISTATUS = SMG$PUT_LINE (DISPLAY1,
1      'The terminal character equivalent of K is displayed.')
      ISTATUS = SMG$PUT_LINE (DISPLAY1, ' ')
C+
C Read a keystroke from the virtual pasteboard.
C-
      ISTATUS = SMG$READ_KEYSTROKE ( KEYBOARD1, TERM_CHAR, '>>', ,
1      DISPLAY1)
      ISTATUS = SMG$PUT_LINE (DISPLAY1, ' ')
C+
C Convert the decimal value of TERM_CHAR to a decimal ASCII text string.
C-
      ISTATUS = OTS$CVT_L_TI( TERM_CHAR, TEXT)
      TEXT_OUTPUT = ' TERMINAL CHARACTER IS: ' // TEXT
C+
C Print the decimal ASCII text string.
C-
      ISTATUS = SMG$PUT_LINE (DISPLAY1, TEXT_OUTPUT)
      ISTATUS = SMG$PUT_CHARS (DISPLAY1, TEXT, 7, 25)

      END
```

# 第2部

---

## リファレンス・セクション

第2部では、日本語画面管理ライブラリ・ルーチンについて詳しく説明します。ルーチン名はアルファベット順に並んでいます。

---

日本語 SMG リファレンス

---

## SMG\$ADD\_KEY\_DEF

### Add Key Definition

Add Key Definition ルーチンは、キーパッド・キーの定義をキー定義テーブルに追加登録します。

---

#### 形式

SMG\$ADD\_KEY\_DEF *key-table-id* ,*key-name* [*,if-state*] [*,attributes*] [*,equivalence-string*] [*,state-string*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

key-table-id

OpenVMS 用法	identifier
データ型	longword(unsigned)
アクセス	read only
受け渡し方	by reference

キー定義を追加するキー・テーブルを指定します。key-table-id 引数は、キー・テーブル識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_KEY\_TABLE ルーチンによって与えられます。

key-name

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

値を定義するキーを指定します。key-name 引数は、このキー名を示すディスクリプタのアドレスです。SMG\$ADD\_KEY\_DEF ルーチンは、文字列を大文字に変更し、後続の空白を削除します。

使用できるキー名については、第 3 章の表 3-1 を参照してください。

## if-state

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

key-name 引数によって指定されるキーが押されたときに戻される値を条件指定します。if-state 引数は、状態文字列を示すディスクリプタのアドレスです。

if-state 引数を指定した場合には、key-name 引数の定義は、現在の状態が指定された if-state 文字列と一致する場合にだけ使用されます。if-state 引数の長さは 1 ~ 31 バイトでなければなりません。この引数を省略した場合には、if-state 引数の省略時の設定は "DEFAULT" になります。

## attributes

OpenVMS 用法 mask\_longword  
 データ型 longword(unsigned)  
 アクセス read only  
 受け渡し方 by reference

キー定義の追加属性を指定するロングワード・ビット・マスク。attributes 引数は、属性マスクを格納した符号なしロングワードのアドレスです。この引数を省略した場合には、マスクは 0 になります。

attributes 引数に対して指定できる値は次のとおりです。

SMG\$M_KEY_NOECHO	このビットが設定されている場合には、このキーが押されたときに、equivalence-string 引数はエコー表示されません。 このビットがクリアされている場合には、equivalence-string 引数はエコー表示されます。SMG\$M_KEY_TERMINATE が設定されていない場合には、SMG\$M_KEY_NOECHO は無視されます。
SMG\$M_KEY_TERMINATE	このビットが設定されている場合には、このキーが押されたときに (if-state 引数の条件指定にしたがって)、入力行は終了し、追加文字を受け付けられないことを指定します。 このビットがクリアされている場合には、追加文字を受け付けることができます。つまり、このビットが設定されている場合は、equivalence-string 引数は終了文字として取り扱われます。
SMG\$M_KEY_LOCK	このビットが設定されており、state-string 引数が指定されている場合には、state-string 引数によって指定される状態名は、定義に state-string 引数を含む後続のキーストロークによって明示的に変更されるまで、現在の状態のまま変更されません。 このビットがクリアされている場合には、state-string 引数によって指定される状態名は、次の定義済みキーストロークに対してのみ有効になります。
SMG\$M_KEY_PROTECTED	このビットが設定されている場合には、このキー定義を変更または削除できないことを指定します。クリアされている場合には、キー定義を変更または削除できます。

他のビットは未定義であり，0 でなければなりません。一度に複数の属性を設定するために，これらの値の対して論理和操作を実行できます。

#### equivalence-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

戻される行内のキーストロークに対して置換される文字列。equivalence-string 引数は，この同値文字列を示すディスクリプタのアドレスです。

SMG\$M\_KEY\_NOECHO が設定されていない限り，equivalence-string 引数はエコー表示されます。equivalence-string 引数を省略した場合には，このキーに対して同値文字列は定義されません。

#### state-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

このキーが押されたときに現在の状態として設定される，新しい状態名を格納します。state-string 引数は，新しい状態文字列を示すディスクリプタのアドレスです。

この引数を省略した場合には，新しい状態は定義されません。現在の状態が一時的な状態の場合には（つまり，最後に押された定義済みキーに対して SMG\$M\_KEY\_LOCK が指定されていない場合），現在の state-string 引数は DEFAULT になります。

---

## 説明

SMG\$ADD\_KEY\_DEF ルーチンは，キー定義をキー定義テーブルに登録します。テーブルは SMG\$CREATE\_KEY\_TABLE ルーチンを呼び出すことにより作成しておかなければなりません。SMG\$ADD\_KEY\_DEF ルーチンを実行した後，SMG\$READ\_COMPOSED\_LINE ルーチンに対する応答として，ユーザが指定されたキーを入力すると，指定された同値文字列が戻されます。

VT100，VT200 シリーズ，および VT300 シリーズのキーボードとキーパッドのすべてのキーを定義できます。ただし，次のキーは定義できません。

- VT200 シリーズと VT300 シリーズ・キーボードの Compose Character キー
- ESCAPE キー
- SHIFT キー
- VT200 シリーズおよび VT300 シリーズ・キーボードの  $\boxed{F1}$  ~  $\boxed{F5}$  キー

一部のキーとキー定義はユーザが定義できますが、なるべくこのような定義は実行しないでください。ユーザがこのような定義をキー名として使用した場合、日本語 SMG ルーチンはエラーを戻すわけではありませんが、次に示すように、DCL レベルで次の特殊な方法で漢字ターミナルを設定しない限り、これらのキーの組み合わせに対して割り当てた定義は実行されません。

- `Ctrl/C`, `Ctrl/O`, `Ctrl/X`, および `F6`

これらのキーに対して割り当てた定義を使用するには、まず、DCL の SET TERMINAL/PASTHRU コマンドを入力しなければならない。

- `Ctrl/T`, `Ctrl/Y`

これらのキーに割り当てた定義を使用するには、まず、DCL の SET TERMINAL/PASTHRU コマンドと SET NOCONTROL コマンドのどちらか一方または両方を入力しなければならない。

- `Ctrl/S`, `Ctrl/Q`

これらのキーに対して割り当てた定義を使用するには、まず、DCL の SET TERMINAL/ NOTTSYNC コマンドを入力しなければならない。

これらの特殊なターミナル設定はなるべく使用しないでください。ターミナル・ドライバに制御を渡すために、省略時の設定を変更した結果がどのようなものかを十分理解していない限り、このような特殊なターミナル設定を使用すると、予測できない結果が発生することがあります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDEFATT	キー定義属性に誤りがあります。
SMG\$_KEYDEFPRO	キー定義が変更または削除できないように保護されています。
SMG\$_INVKEYNAM	key-name 引数に誤りがあります。
SMG\$_INVKTB_ID	key-table-id 引数に誤りがあります。
SMG\$_PREDEFREP	正常終了。前のキー定義は新しいキー定義に変更されました。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	LIBSSCOPY_DXD から戻された状態値。

---

## SMG\$BEGIN\_DISPLAY\_UPDATE

### Begin Batching of Display Updates

Begin Batching of Display Updates ルーチンは、対応する SMG\$END\_DISPLAY\_UPDATE の呼び出しを検出するまで、仮想ディスプレイに対するすべての出力を保存、つまりバッチ処理します。

---

#### 形式

SMG\$BEGIN\_DISPLAY\_UPDATE *display-id*

---

#### 戻り値

VMS 用法: cond\_value  
データ型: longword (unsigned)  
アクセス: write only  
受け渡し方: by value

---

#### 引数

*display-id*  
OpenVMS 用法 identifier  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

出力をバッチ処理する仮想ディスプレイを指定します。display-id 引数はディスプレイ識別子を格納した符号なしロングワードのアドレスです。

display-id 引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンから戻されます。

---

#### 説明

SMG\$BEGIN\_DISPLAY\_UPDATE は、ディスプレイに対して複数の変更処理を実行します。変更結果は、すべての変更処理が終了した後で一括して表示されます。したがって、ユーザから見ると、ディスプレイは初期状態から最終状態に直接変化し、途中の状態はまったく表示されません。

バッチ処理が終了するのは、特定のディスプレイに対して SMG\$BEGIN\_DISPLAY\_UPDATE と同じ回数だけ、SMG\$END\_DISPLAY\_UPDATE が呼び出されたときです。日本語 SMG は、指定されたディスプレイに対するバッチ処理の状況を追跡し、管理します。したがって、SMG\$BEGIN\_DISPLAY\_UPDATE と SMG\$END\_DISPLAY\_UPDATE を同じモジュール内で呼び出す必要はありません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_BATWAS_ON	正常終了。バッチ処理はすでに開始されています。
SMG\$_WRONUMARG	引数の数が誤っています。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。

---

## SMG\$BEGIN\_PASTEBOARD\_UPDATE

### Begin Batching of Pasteboard Updates

Begin Batching of Pasteboard Updates ルーチンは、対応する SMG\$END\_PASTEBOARD\_UPDATE ルーチンの呼び出しが検出されるまで、ペーストボードに対するすべての出力を保存、つまりバッチ処理します。

---

#### 形式

SMG\$BEGIN\_PASTEBOARD\_UPDATE *pasteboard-id*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

*pasteboard-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

出力がバッチ処理されるペーストボードを指定します。*pasteboard-id* 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

---

#### 説明

SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチンを使用すれば、ペーストボードに対して複数の変更処理を実行し、すべての変更処理が終了した後で変更結果を表示することができます。したがって、ユーザから見ると、ペーストボードは初期状態から最終状態に直接変化し、途中の状態は表示されません。

バッチ処理が終了するのは、特定のペーストボードに対して SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチンと同じ回数だけ、SMG\$END\_PASTEBOARD\_UPDATE ルーチン呼び出したときです。日本語 SMG は、指定されたペーストボードに対するバッチ処理の状況を追跡し、管理します。したがって、SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチンと SMG\$END\_PASTEBOARD\_UPDATE ルーチンを同じモジュール内で呼び出す必要はありません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_BATWAS_ON	正常終了。バッチ処理はすでに開始されています。
SMG\$_WRONUMARG	引数の数が誤っています。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。

---

## SMG\$CANCEL\_INPUT

### Cancel Input Request

Cancel Input Request ルーチンは、SMG\$READ\_COMPOSED\_LINE、SMG\$READ\_KEYSTROKE、SMG\$READ\_STRING、SMG\$READ\_VERIFY の各ルーチンが要求した読み込み操作をただちに取り消します。

---

#### 形式

SMG\$CANCEL\_INPUT *keyboard-id*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

keyboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

入力操作を取り消す対象となる仮想キーボードを指定します。keyboard-id 引数は、キーボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンによって与えられます。

---

#### 説明

SMG\$CANCEL\_INPUT ルーチンを実行すると、漢字ターミナルからの SMG\$READ\_COMPOSED\_LINE、SMG\$READ\_KEYSTROKE、SMG\$READ\_STRING、SMG\$READ\_VERIFY の各ルーチンによる入力操作はただちに終了しま

す。SMG\$CANCEL\_INPUT ルーチンを使用した場合には、これらのルーチンに対して、SS\$\_CANCEL または SS\$\_ABORT が状態コードとして戻されます。

---

#### 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVKBD_ID	keyboard-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$CHANGE\_PBD\_CHARACTERISTICS

## Change Pasteboard Characteristics

Change Pasteboard Characteristics ルーチンは、ペーストボードに割り当てた属性を変更します。

---

### 形式

SMG\$CHANGE\_PBD\_CHARACTERISTICS

*pasteboard-id* [*,desired-width*] [*,width*] [*,desired-height*] [*,height*]  
[*,desired-background-color*] [*,background-color*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

属性を変更するペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

desired-width

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

ペーストボードの新しい幅。desired-width 引数は、希望する幅を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、幅は変更されません。

## width

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	write only
受け渡し方	by reference

ペーストボードの物理幅が書き込まれます。width 引数は、ペーストボードの実際の幅が書き込まれる符号付きロングワードのアドレスです。

漢字ターミナルの幅を正確に desired-width 引数に設定できない場合には、width 引数は desired-width 引数より大きくなる可能性があります。漢字ターミナルの物理幅が desired-width 引数より小さい場合には、width 引数が desired-width 引数より小さくなる可能性があります。

## desired-height

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

ペーストボードの新しい高さ。desired-height 引数は、希望する高さを格納した符号付きロングワードのアドレスです。この引数を省略した場合には、高さは変更されません。

## height

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	write only
受け渡し方	by reference

ペーストボードの物理的な高さが書き込まれます。height 引数は、ペーストボードの実際の高さが書き込まれる符号付きロングワードのアドレスです。

漢字ターミナルが高さを正確に desired-height 引数に設定できない場合には、height 引数は desired-height 引数より大きくなる可能性があります。漢字ターミナルの物理的な高さが desired-height 引数より小さい場合には、height 引数は desired-height 引数より小さくなる可能性があります。

## desired-background-color

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

希望する背景色のシンボル名。desired-background-color 引数は、希望する背景色を格納した符号なしロングワードのアドレスです。

次に示すシンボルは\$SMGDEF に定義されています。desired-background-color 引数に対して指定できる値は次のとおりです。

## SMG\$CHANGE\_PBD\_CHARACTERISTICS

SMG\$C_COLOR_WHITE	白の背景
SMG\$C_COLOR_BLACK	黒の背景
SMG\$C_COLOR_BLUE	青の背景
SMG\$C_COLOR_CYAN	青緑の背景
SMG\$C_COLOR_GREEN	緑の背景
SMG\$C_COLOR_MAGENTA	紫の背景
SMG\$C_COLOR_RED	赤の背景
SMG\$C_COLOR_YELLOW	黄色の背景
SMG\$C_COLOR_LIGHT	明るい背景
SMG\$C_COLOR_DARK	暗い背景
SMG\$C_COLOR_USER1	ユーザ定義背景色 1
SMG\$C_COLOR_USER2	ユーザ定義背景色 2

desired-background-color 引数を省略した場合や、指定された背景色をターミナル・ハードウェアがサポートしない場合には、背景色は変更されません。

background-color

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

選択された背景色が書き込まれます。background-color 引数は、背景色が書き込まれる符号なしロングワードのアドレスです。

このルーチンは、desired-background-color 引数の説明に示されている値、または SMG\$C\_COLOR\_UNKNOWN を戻します。desired-background-color 引数を省略した場合には、background-color 引数の値は変更されません。

---

## 説明

SMG\$CHANGE\_PBD\_CHARACTERISTICS ルーチンを使用すれば、ペーストボードに割り当てた幅、高さ、背景色を変更できます。

必要な場合には、このルーチンは、DCL の SHOW TERMINAL コマンドが入力されたときに表示されるターミナル属性を更新することにより、ペーストボード属性が変更されたことを OpenVMS オペレーティング・システムに通知します。

バッチ処理されるペーストボードに対して、SMG\$CHANGE\_PBD\_CHARACTERISTICS ルーチンを使用することはできません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SS\$_xxx	\$QIOW から戻されたエラー。
SMG\$_INVCOLARG	指定された背景色が認識されません。
SMG\$_INVPAGARG	希望する高さとして 0 を指定することはできません。
SMG\$_INVWIDARG	希望する幅として 0 を指定することはできません。
SMG\$_PBDIN_USE	バッチ処理が有効に設定された状態でペーストボード属性を変更することはできません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$CHANGE\_RENDITION

### Change Default Rendition

Change Default Rendition ルーチンは、仮想ディスプレイ全体または一部のビデオ属性を変更します。

---

#### 形式

SMG\$CHANGE\_RENDITION *display-id ,start-row ,start-column ,number-of-rows ,number-of-columns*  
*[,rendition-set] [,rendition-complement]*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

省略時の属性を変更する対象となる仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

新しい属性を受け付ける開始行番号。start-row 引数は、開始行番号を格納した符号付きロングワードのアドレスです。

## start-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

新しい属性を受け付ける開始カラム番号。start-column 引数は、開始カラム番号を格納した符号付きロングワードのアドレスです。start-column 引数が全角文字の右半分である場合は、その文字の左半分も新しい属性になります。

## number-of-rows

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

新しい属性を受け付ける行番号。number-of-rows 引数は、影響を受ける行番号を格納した符号付きロングワードのアドレスです。

## number-of-columns

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

新しい属性を受け付けるカラム数。number-of-columns 引数は、影響を受けるカラム数を格納した符号付きロングワードのアドレスです。

## rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスク内の各属性を設定すると、ディスプレイ内の対応する属性が設定されます。rendition-set 引数を使用して以下の属性を指定できます。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想ディスプレイの現在の省略時のレンディションとは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## rendition-complement

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスク内の各属性を設定すると、ディスプレイ内の対応する属性が補足されます。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数で補足できます。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの2つの引数を組み合わせて使用すれば、1回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性をオフに設定する。

---

 説明

このルーチンは、指定された仮想ディスプレイ内にすでに存在する長方形のテキスト・ブロックの省略時のビデオ属性を変更します。たとえば、このルーチンを使用して特定の行を反転表示することができます。

DEC 漢字または Super DEC 漢字文字集合が使用され、start-column 引数が全角文字の右半分である場合には、その文字の左半分も新しい属性になります。また、長方形が全角文字の左半分で終了する場合には、その文字の右半分も新しい属性になります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVARG	行数またはカラム数が誤っているか、rendition-set 引数が認識されないか、またはrendition-complement 引数が認識されません。
SMG\$_INVCOL	start-column 引数に誤りがあります。指定されたカラムは仮想ディスプレイの外部です。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_INVROW	start-row 引数に誤りがあります。指定された行は仮想ディスプレイの外部です。
SMG\$_NO_CHADIS	仮想ディスプレイは変更されませんでした。
SMG\$_WRONUMARG	引数の数が誤っています。

SMG\$CHANGE\_RENDITION

<html\_segment>(seg\_smg)

---

# SMG\$CHANGE\_VIEWPORT

## Change the Viewport Associated with a Virtual Display

Change the Viewport Associated with a Virtual Display ルーチンは、仮想ディスプレイ内の既存のビューポートのサイズを変更します。ビューポートに現在表示されているテキストは、新しいサイズに収まるように再マップされます。

---

### 形式

```
SMG$CHANGE_VIEWPORT display-id [,viewport-row-start] [,viewport-column-start]
                    [,viewport-number-rows] [,viewport-number-columns]
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

変更するビューポートを含む仮想ディスプレイの識別子。display-id 引数は、この識別子を格納した符号なしロングワードのアドレスです。

viewport-row-start

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

変更後のビューポートの 1 行目となる仮想ディスプレイの行番号 (省略可能)。viewport-row-start 引数は行番号を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、現在の viewport-row-start 引数の値が使用されます。

## viewport-column-start

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

変更後のビューポートで 1 カラム目となる仮想ディスプレイ内のカラム番号 (省略可能)。viewport-column-start 引数は、カラム番号を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、現在の viewport-column-start 引数の値が使用されます。

viewport-column-start 引数で指定されたカラムが、全角文字の右半分であった場合には、そのカラムは表示されません。

## viewport-number-rows

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

変更後のビューポートの行数 (省略可能)。viewport-number-rows 引数は、行数を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、現在の viewport-number-rows 引数の値が使用されます。

## viewport-number-columns

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

変更後のビューポートのカラム数 (省略可能)。viewport-number-columns 引数は、カラム数を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、現在の viewport-number-columns 引数の値が使用されます。

---

 説明

SMG\$CHANGE\_VIEWPORT ルーチンを使用すれば、仮想ディスプレイ内の既存のビューポートのサイズを変更できます。このビューポート内に現在表示されているテキストは、新しいサイズに収まるように再マッピングされ、viewport-row-start 引数と viewport-column-start 引数によって指定される位置から始まります。この位置はまた、仮想カーソルの新しい位置も指定します。

viewport-column-start 引数で指定されたカラムが、全角文字の右半分であった場合には、そのカラムは表示されません。また、ビューポートが全角文字の左半分で終了した場合にも、そのカラムは表示されません。

---

 戻される状態値

SS\$ _NORMAL	正常終了。
SMG\$ _INVARG	行数またはカラム数が 0 より小さい値です。
SMG\$ _INVCOL	指定されたカラム番号に誤りがあります。
SMG\$ _INVDIS_ID	display-id 引数に誤りがあります。
SMG\$ _INVROW	指定された行番号に誤りがあります。
SMG\$ _NO_WINASSOC	仮想ディスプレイにビューポートが対応づけられていません。
SMG\$ _WRONUMARG	引数の数が誤っています。

---

 例

```

1.
C+
C This FORTRAN example program demonstrates the use of
C SMG$CHANGE_VIEWPORT.
C-

  IMPLICIT INTEGER (A-Z)
  INCLUDE '($SMGDEF)'

C Create the virtual display. Give it a border.

  ROWS = 9
  COLUMNS = 50

  STATUS = SMG$CREATE_VIRTUAL_DISPLAY
           1 (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER )
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%val(STATUS))

C Create the pasteboard.

  STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%val(STATUS))

C Put data in the virtual display.

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
  1 'This is row 1 in a virtual display with 9 rows.',1,1)
  IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
  1 'This is row 2 in a virtual display with 9 rows.',2,1)
  IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
  1 'This is row 3 in a virtual display with 9 rows.',3,1)
  IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
  1 'This is row 4 in a virtual display with 9 rows.',4,1)
  IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))

```

## SMG\$CHANGE\_VIEWPORT

```
STATUS = SMG$PUT_CHARS ( DISPLAY1,  
1 'This is row 5 in a virtual display with 9 rows.',5,1)  
IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))  
  
STATUS = SMG$PUT_CHARS ( DISPLAY1,  
1 'This is row 6 in a virtual display with 9 rows.',6,1)  
IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))  
  
STATUS = SMG$PUT_CHARS ( DISPLAY1,  
1 'This is row 7 in a virtual display with 9 rows.',7,1)  
IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))  
  
STATUS = SMG$PUT_CHARS ( DISPLAY1,  
1 'This is row 8 in a virtual display with 9 rows.',8,1)  
IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))  
  
STATUS = SMG$PUT_CHARS ( DISPLAY1,  
1 'This is row 9 in a virtual display with 9 rows.',9,1)  
IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))  
  
C Paste the virtual display.  
  
STATUS = SMG$COPY_VIRTUAL_DISPLAY(DISPLAY1,DISPLAY2)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
STATUS = SMG$LABEL_BORDER (DISPLAY1, 'Full Display',,,SMG$M_BOLD)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
STATUS = SMG$LABEL_BORDER (DISPLAY2, 'Viewport',,,SMG$M_BOLD)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 2, 10)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY2, PASTE1, 13, 10)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
CALL LIB$WAIT (4.0)  
  
STATUS = SMG$CREATE_VIEWPORT ( DISPLAY2, 2, 1, 5, 21)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
CALL LIB$WAIT (4.0)  
  
STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY2, PASTE1, 13, 10)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
CALL LIB$WAIT (4.0)  
  
STATUS = SMG$CHANGE_VIEWPORT ( DISPLAY2, 4, 8, 3, 15)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
call lib$wait (4.0)  
  
END
```

この FORTRAN の例で生成される出力は以下の図に示すとおりです。図 8-1 では、プログラムは初期仮想ディスプレイを、“Viewport” というラベルの付いた 2 番目の仮想ディスプレイにコピーしています。

図 8-1 仮想ディスプレイがペーストされた後に生成される出力

```
----- Full Display -----
This is row 1 in a virtual display with 9 rows.
This is row 2 in a virtual display with 9 rows.
This is row 3 in a virtual display with 9 rows.
This is row 4 in a virtual display with 9 rows.
This is row 5 in a virtual display with 9 rows.
This is row 6 in a virtual display with 9 rows.
This is row 7 in a virtual display with 9 rows.
This is row 8 in a virtual display with 9 rows.
This is row 9 in a virtual display with 9 rows.

----- Viewport -----
This is row 1 in a virtual display with 9 rows.
This is row 2 in a virtual display with 9 rows.
This is row 3 in a virtual display with 9 rows.
This is row 4 in a virtual display with 9 rows.
This is row 5 in a virtual display with 9 rows.
This is row 6 in a virtual display with 9 rows.
This is row 7 in a virtual display with 9 rows.
This is row 8 in a virtual display with 9 rows.
This is row 9 in a virtual display with 9 rows.
```

ZK-6423/1-GE

2つの同じ仮想ディスプレイをペーストした後、プログラムは2番目の(コピーした)仮想ディスプレイにビューポートを作成します。2番目のディスプレイが“再ペースト”された後、ビューポート内の一部だけが表示される状態になります。この結果は図 8-2 に示されています。

図 8-2 ビューポートを作成した後に生成される出力

```
----- Full Display -----
This is row 1 in a virtual display with 9 rows.
This is row 2 in a virtual display with 9 rows.
This is row 3 in a virtual display with 9 rows.
This is row 4 in a virtual display with 9 rows.
This is row 5 in a virtual display with 9 rows.
This is row 6 in a virtual display with 9 rows.
This is row 7 in a virtual display with 9 rows.
This is row 8 in a virtual display with 9 rows.
This is row 9 in a virtual display with 9 rows.

----- Viewport -----
This is row 1 in a vi
This is row 2 in a vi
This is row 3 in a vi
This is row 4 in a vi
This is row 5 in a vi
This is row 6 in a vi
```

ZK-6423/2-GE

SMG\$CHANGE\_VIEWPORT ルーチンを呼び出すことにより、仮想ディスプレイの中でビューポートを通じて表示される部分は変化します。これは図 8-3 に示すとおりです。

図 8-3 SMG\$CHANGE\_VIEWPORT ルーチン呼び出し後に生成される出力

```
----- Full Display -----  
This is row 1 in a virtual display with 9 rows.  
This is row 2 in a virtual display with 9 rows.  
This is row 3 in a virtual display with 9 rows.  
This is row 4 in a virtual display with 9 rows.  
This is row 5 in a virtual display with 9 rows.  
This is row 6 in a virtual display with 9 rows.  
This is row 7 in a virtual display with 9 rows.  
This is row 8 in a virtual display with 9 rows.  
This is row 9 in a virtual display with 9 rows.  
  
----- Viewport -----  
row 4 in a vir  
row 5 in a vir  
row 6 in a vir
```

ZK-6423/3-GE

---

# SMG\$CHANGE\_VIRTUAL\_DISPLAY

## Change Virtual Display

Change Virtual Display ルーチンは、仮想ディスプレイのサイズ、境界、ビデオ属性を変更します。

---

### 形式

SMG\$CHANGE\_VIRTUAL\_DISPLAY

*display-id* [*number-of-rows*] [*number-of-columns*] [*display-attributes*]  
[*video-attributes*] [*character-set*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

属性を変更する仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

number-of-rows

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

仮想ディスプレイの新しい行数。number-of-rows 引数は、仮想ディスプレイの行数を格納した符号付きロングワードのアドレスです。

## number-of-columns

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

仮想ディスプレイの新しいカラム数。number-of-columns 引数は、仮想ディスプレイのカラム数を格納した符号付きロングワードのアドレスです。

## display-attributes

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

仮想ディスプレイの属性。display-attributes 引数は、仮想ディスプレイの属性を格納したロングワード・ビット・マスクのアドレスです。この引数に対して指定できる値は次のとおりです。

SMG\$M_BORDER	境界付き仮想ディスプレイを指定します。省略した場合は、仮想ディスプレイに境界は表示されません。
SMG\$M_BLOCK_BORDER	ブロック形の境界付き仮想ディスプレイを指定します。省略した場合は、仮想ディスプレイに境界は表示されません。
SMG\$M_DISPLAY_CONTROLS	漢字ターミナルがグラフィック文字をサポートする場合は、キャリッジ・リターンやライン・フィードなどの制御文字をグラフィック文字として表示することを指定します。
SMG\$M_TRUNC_ICON	仮想ディスプレイの幅をこえる行を切り捨てた際に、その場所にアイコン (通常は菱形) を表示することを指定します。

## video-attributes

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

仮想ディスプレイの出力に適用される属性のシステム設定値。

video-attributes 引数は、ビデオ属性マスクを格納した符号なしロングワードのアドレスです。特定の出カルーチン呼び出さない限り、仮想ディスプレイのすべての出力に適用されます。

たとえば、属性を指定して SMG\$PUT\_CHARS を呼び出した場合は、レンディションのシステム設定値は無効になります。

この引数に対して指定できるビットは次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。

SMG\$M_REVERSE	文字を反転表示します。つまり、仮想ディスプレイの現在の省略時の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

1 回の呼び出しで複数の属性の組み合わせを指定できます。他のビットはすべて HP が使用するために予約されており、0 でなければなりません。

#### character-set

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

仮想ディスプレイに対する文字集合のシステム設定値。character-set 引数は、文字集合指定子を格納した符号なしロングワードのアドレスです。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンは、既存の仮想ディスプレイのサイズまたはシステム設定値の属性を変更します。仮想ディスプレイのサイズを変更する場合は、日本語 SMG は新しいサイズ (1 行目/1 カラム目から始まる) に適合するように、仮想ディスプレイに関連するテキストを再マップします。

仮想ディスプレイの新しいサイズが前のサイズより小さい場合は、一部のテキストが切り捨てられる可能性があります。全角文字の右半分が切り捨てられた場合は、その文字の左半分は未定義文字になります。仮想ディスプレイの新しいサイズが前のサイズより大きい場合は、テキストの右側に空白が挿入されることがあります。

仮想ディスプレイのサイズを変更した場合は、仮想ディスプレイの仮想カーソルは、1 行目、1 カラム目に移動されます。ラベル付き境界を仮想ディスプレイで使用する

際に、サイズの変更された仮想ディスプレイ上にラベルを表示できない場合は、ラベルは削除されます。

オリジナルの SMG ルーチンは、ASCII 文字集合と DEC 特殊文字集合のみをサポートします。日本語 SMG ではこの他に SMG\$C\_SUPPLEMENTAL、SMG\$C\_KANJI、SMG\$C\_JIS\_KANA、SMG\$C\_SDK が指定できます。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INSVIRMEM	要求されたバッファを再び割り当てるために必要な仮想メモリが不足しています。
SMG\$_INVARG	ビデオ属性または仮想ディスプレイの属性に誤りがあります。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$CHECK\_FOR\_OCCLUSION

### Check for Occlusion

Check for Occlusion ルーチンは、仮想ディスプレイが別の仮想ディスプレイによって覆われているかどうかを確認します。

---

#### 形式

SMG\$CHECK\_FOR\_OCCLUSION *display-id ,pasteboard-id ,occlusion-state*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

確認したい仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

確認したいペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

## occlusion-state

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

ディスプレイが重なっているかどうかを示す値が書き込まれます。

occlusion-state 引数は、重なっているかどうかを示す状態が書き込まれる符号付きロングワードのアドレスです。ディスプレイが重なっている場合には、occlusion-state 引数は 1 に設定され、指定されたペーストボードでディスプレイが重なっていない場合には 0 に設定されます。プロシージャが SSS\_NORMAL を戻さなかった場合には、occlusion-state 引数の内容は未定義です。

---

 説明

SMG\$CHECK\_FOR\_OCCLUSION ルーチンは、指定された仮想ディスプレイを指定されたペーストボードにペーストしたときに、その仮想ディスプレイが別の仮想ディスプレイと重なり合っているかどうかを確認します。

---

 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_NOTPASTED	仮想ディスプレイが指定されたペーストボードにペーストされていません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

 例

```

1.
C+
C This FORTRAN example program demonstrates the use of
C SMG$CHECK_FOR_OCCLUSION.
C
C This routine creates a virtual display and writes it to the
C pasteboard. Data is placed in the virtual display using SMG$PUT_CHARS.
C-
```

## SMG\$CHECK\_FOR\_OCCLUSION

```
INTEGER SMG$CREATE_VIRTUAL_DISPLAY, SMG$CREATE_PASTEBOARD
INTEGER SMG$PASTE_VIRTUAL_DISPLAY, SMG$PUT_CHARS
INTEGER SMG$CHECK_FOR_OCCLUSION
INTEGER DISPLAY1, DISPLAY2, PASTE1, PASTE2, ROWS, COLUMNS, BORDER
INTEGER OCCLUSION, STATUS
CHARACTER*29 TEXT
```

C+

C Include the SMG definitions. In particular, we want SMG\$M\_BORDER.

C-

```
INCLUDE '($SMGDEF)'
```

C+

C Create two virtual displays using SMG\$CREATE\_VIRTUAL\_DISPLAY.

C Give them borders.

C-

```
ROWS = 6
COLUMNS = 50

STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1 (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

ROWS = 5
COLUMNS = 30

STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1 (ROWS, COLUMNS, DISPLAY2, SMG$M_BORDER)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
```

C+

C Create the pasteboard using SMG\$CREATE\_PASTEBOARD.

C-

```
STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
```

C+

C Use SMG\$PUT\_CHARS to put data into the virtual displays.

C-

```
STATUS = SMG$PUT_CHARS ( DISPLAY1,
1 ' This virtual display has 6 rows and 50 columns.', 2, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1 ' This is a bordered virtual display.', 3, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1 ' SMG$PUT_CHARS puts data in this virtual display.', 4,
1 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1 ' This text should be partially occluded.', 5, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
```

```

        STATUS = SMG$PUT_CHARS ( DISPLAY1,
1         ' So should part of this row.', 6, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        STATUS = SMG$PUT_CHARS ( DISPLAY2, ' This is virtual', 3, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        STATUS = SMG$PUT_CHARS ( DISPLAY2,
1         ' display #2.', 4, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        STATUS = SMG$PUT_CHARS ( DISPLAY2,
1         ' This is just some more text.', 5, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual display.
C-

        STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY2, PASTE1, 8, 15)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Check the two virtual displays for occlusion by calling
C SMG$CHECK_FOR_OCCLUSION.
C-

        TEXT = 'This display is not occluded.'

        STATUS = SMG$CHECK_FOR_OCCLUSION (DISPLAY1, PASTE1, OCCLUSION)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        IF (OCCLUSION .EQ. 0) THEN
            STATUS = SMG$PUT_CHARS (DISPLAY1, TEXT, 1, 1)
            IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
        ELSE
            STATUS = SMG$PUT_CHARS (DISPLAY1, 'Occluded.', 1, 1)
            IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
        END IF

        STATUS = SMG$CHECK_FOR_OCCLUSION (DISPLAY2, PASTE1, OCCLUSION)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

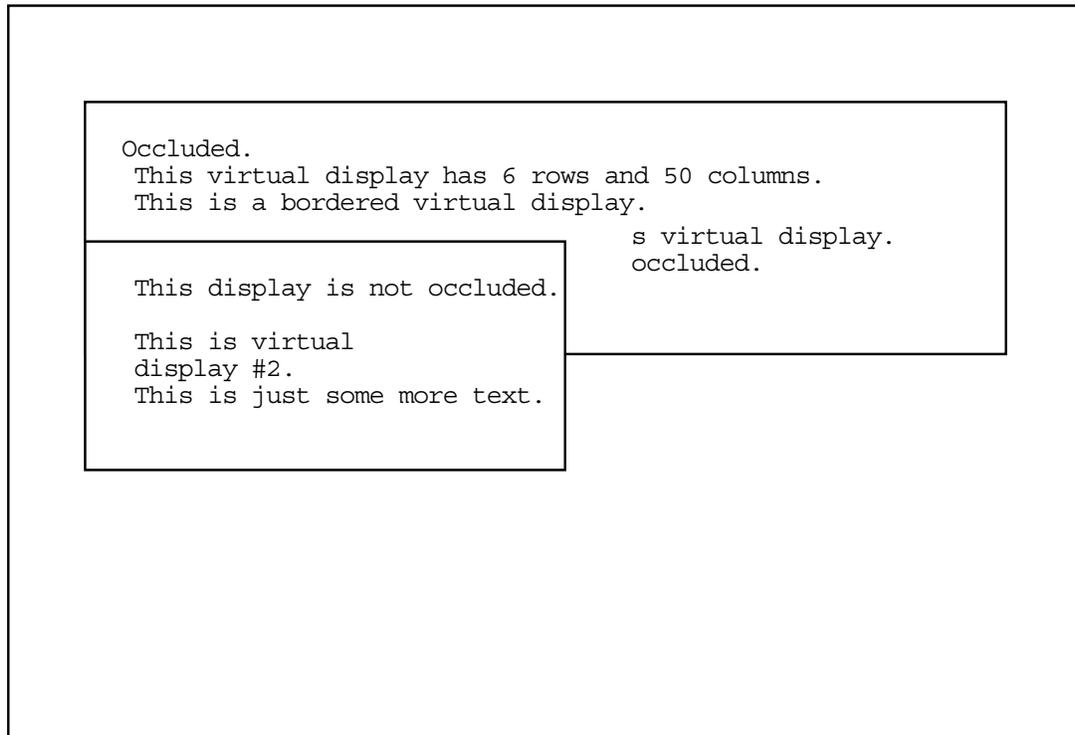
        IF (OCCLUSION .EQ. 0) THEN
            STATUS = SMG$PUT_CHARS (DISPLAY2, TEXT, 1, 1)
            IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
        ELSE
            STATUS = SMG$PUT_CHARS (DISPLAY2, 'Occluded.', 1, 1)
            IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
        END IF

        END

```

この FORTRAN プログラムから生成される出力は図 8-4 に示すとおりです。

図 8-4 SMG\$CHECK\_FOR\_OCCLUSION ルーチン呼び出す FORTRAN プログラムが生成する出力



ZK-4128-GE

---

## SMG\$CONTROL\_MODE

### Control Mode

Control Mode ルーチンはペーストボードのモードを制御します。これには、バッファ操作、画面更新、ペーストボード削除時のターミナル画面の消去、ターミナル画面管理のためのタブの使用が含まれます。

---

#### 形式

SMG\$CONTROL\_MODE *pasteboard-id* [*new-mode*] [*old-mode*] [*buffer-size*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

変更されるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

new-mode

OpenVMS 用法	mask_longword
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

制御モードの新設定。new-mode 引数は省略可能で、設定モードを格納した符号なしロングワード・ビット・マスクのアドレスです。ビットを 1 に設定すると各モードが設定され、0 に設定すると解除されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BUF_ENABLED	バッファリングを行います。
SMG\$M_CLEAR_SCREEN	ペーストボードが消去されずにプログラムが終了した際にターミナル画面を消去します。
SMG\$M_MINUPD	更新を最小限にします (省略時の設定値)。
SMG\$M_NOTABS	日本語 SMG がターミナル画面管理を行う際にタブを使わないようにします。
SMG\$M_PROTECT	ペーストボード操作を AST 割り込みから保護します (省略時の設定値)。

他のビットはすべて HP が使用するために予約されており、0 でなければなりません。

#### old-mode

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス write only  
 受け渡し方 by reference

制御モードの旧設定。old-mode 引数は省略可能で、旧設定モードが書き込まれる符号なしロングワード・ビット・マスクのアドレスです。このプロシージャが呼ばれる前に使われていた設定モードです。ビットが 1 の場合は各モードが設定されており、0 の場合は設定されていません。

#### buffer-size

OpenVMS 用法 word\_unsigned  
 データ型 word (unsigned)  
 アクセス read only  
 受け渡し方 by reference

バッファ・サイズ (バイト長)。buffer-size 引数は省略可能で、バッファ・サイズを格納した符号なしワードのアドレスです。buffer-size 引数はバッファリング可能 (SMG\$M\_BUF\_ENABLED の指定がある) な場合に適用されます。システム設定値である最小バッファ・サイズは 256 バイトです。最大バッファ・サイズは 65535 バイトです。buffer-size 引数の値は UAF 値に依存し、その最大値は SYSGEN パラメータ (MAXBUF) によって決まります。

---

## 説明

SMG\$CONTROL\_MODE ルーチンを使用することにより、指定したペーストボード・モードに対する日本語 SMG 操作の設定モードを得たり、その設定値を変更することができます。また、new-mode 引数と old-mode 引数を組み合わせて、次のような場合に使用することができます。

- 現在の設定モードに関する情報を得る場合  
 《形式》 SMG\$CONTROL\_MODE (pasteboard\_id „old\_mode)

- 現在の設定モードとは関係なくモードを設定する場合  
《形式》 SMG\$CONTROL\_MODE (pasteboard\_id ,new\_mode)
- 現在の設定を保存して新モードを設定した後で、旧設定を復元する場合  
《形式 1 》 SMG\$CONTROL\_MODE (pasteboard\_id ,new\_mode ,old\_mode)

この形式では、現在のモード設定を保存し new-mode 引数に従って新モードを設定します。

この後で旧設定を復元する場合は次の形式を指定してください。以前に保存した設定に従って新モードを設定します。

《形式 2 》 SMG\$CONTROL\_MODE (pasteboard\_id ,old\_mode)

---

### 注意

---

old-mode 引数と new-mode 引数の両方を省略した場合には情報は与えられません。

---

SMG\$CONTROL\_MODE ルーチンを使用することにより、次のモードに関して、情報を得たり変更を加えることができます。

- バッファリング

このモードでは効率的なシステム QIO を使用するために、日本語 SMG はすべての出力をバッファに格納します。バッファが一杯になった時点で日本語 SMG はバッファを漢字ターミナルに書き込みます。また、SMG\$FLUSH\_BUFFER ルーチン呼び出すことにより、漢字ターミナルに書き込まれていないペーストボード・バッファの内容をターミナル画面に出力することができます。

- 最小限の画面更新

日本語 SMG のシステム設定では、漢字ターミナルに出力する文字数を最小限におさえます。これは、日本語 SMG がペーストボードの現在の内容と新しい内容を比べ、変更のあった文字だけを送り出しているからです。この機能を使用しない場合は、変更を加える文字だけではなく、その文字を含む行が書き換えられます。その際には、変更される最初の文字以降が書き換えられます。

- 画面消去

日本語 SMG のシステム設定値では、ペーストボードが削除されずにプログラムが終了した場合、ターミナル画面は消去されません。システム設定による動作を変更するために、このモードを使用します。

- タブ解除

このモードが設定されている場合は、漢字ターミナルのタブ設定は使用されません。このモードが設定されていない場合には、最小限の画面更新プロシージャに漢字ターミナルのタブ機能が使用されます。ただし、タブ・ストップの設定は DIGITAL の標準位置 (8 カラムごと) が仮定されています。漢字ターミナルのタブ設定に依存せずにアプリケーションを使用したい場合は“no tabs”に指定し

てください。省略時にはビットはクリアされています。ターミナル設定で SET TERM/NOTABS と指定すると、システム設定値は使用されません。

---

戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVARG	引数が誤っています。下記に相当しないビットが new-mode 引数に設定されているか、またはバッファ・サイズが 256 バイト以下です。  SMGSM_BUF_ENABLED SMGSM_MINUPD SMGSM_CLEAR_SCREEN SMGSM_NOTABS
SMG\$_INVPAS_ID	pasteboard-id 引数が誤っています。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$COPY\_VIRTUAL\_DISPLAY

## Copy a Virtual Display

Copy a Virtual Display ルーチンは、既存の仮想ディスプレイのコピーを作成し、そのコピーに新しい仮想ディスプレイ識別子を割り当てます。

---

### 形式

SMG\$COPY\_VIRTUAL\_DISPLAY *current-display-id* ,*new-display-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

current-display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

コピーされる仮想ディスプレイの識別子を表示します。current-display-id 引数はディスプレイ識別子を格納した符号なしロングワードのアドレスです。

new-display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

新しく生成される仮想ディスプレイのディスプレイ識別子を書き込まれます。new-display-id 引数は、新しいディスプレイ識別子を格納する符号なしロングワードのアドレスです。

---

## 説明

SMG\$COPY\_VIRTUAL\_DISPLAY ルーチンは既存の仮想ディスプレイのコピーを作成し、そのコピーに新しい仮想ディスプレイ番号を割り当てます。この新しく生成される仮想ディスプレイはどこにもペーストされません。新しく生成される仮想ディスプレイをペーストするには、SMG\$PASTE\_VIRTUAL\_DISPLAY ルーチンと new-display-id 識別子を使用します。SMG\$COPY\_VIRTUAL\_DISPLAY ルーチンを呼び出すときに、コピーされる既存のディスプレイをペーストする必要はありません。

---

## 戻される状態値

SS\$NORMAL	正常終了。
LIB\$INSVIRMEM	必要なバッファに割り当てるための仮想メモリが不足しています。

## 例

## 1.

```

C+
C This FORTRAN example program demonstrates the use of
C SMG$COPY_VIRTUAL_DISPLAY.
C
C This routine creates a virtual display and writes it to the
C pasteboard. Data is placed in the virtual display using SMG$PUT_CHARS.
C-

        IMPLICIT INTEGER (A-Z)
        CHARACTER*29 TEXT

C+
C Include the SMG definitions. In particular, we want SMG$M_BORDER.
C-

        INCLUDE '($SMGDEF)'

C+
C Create two virtual displays using SMG$CREATE_VIRTUAL_DISPLAY.
C Give them borders.
C-

        ROWS = 6
        COLUMNS = 50

        STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1         (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        ROWS = 5
        COLUMNS = 30

        STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1         (ROWS, COLUMNS, DISPLAY2, SMG$M_BORDER)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Create the pasteboard using SMG$CREATE_PASTEBOARD.
C-

        STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$PUT_CHARS to put data into the virtual displays.
C-

        STATUS = SMG$PUT_CHARS ( DISPLAY1,
1         ' This virtual display has 6 rows and 50 columns.', 2, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        STATUS = SMG$PUT_CHARS ( DISPLAY1,
1         ' This is a bordered virtual display.', 3, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

```

## SMG\$COPY\_VIRTUAL\_DISPLAY

```
STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' SMG$PUT_CHARS puts data in this virtual display.', 4,
1      1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This text should be partially occluded.', 5, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' So should part of this row.', 6, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY2, ' This is virtual', 3, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY2,
1      ' display #2.', 4, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY2,
1      ' This is just some more text.', 5, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual display.
C-

STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY2, PASTE1, 8, 15)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Copy the first virtual display, the one that is partially occluded.
C-

STATUS = SMG$COPY_VIRTUAL_DISPLAY ( DISPLAY1, NEW_DISPLAY)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

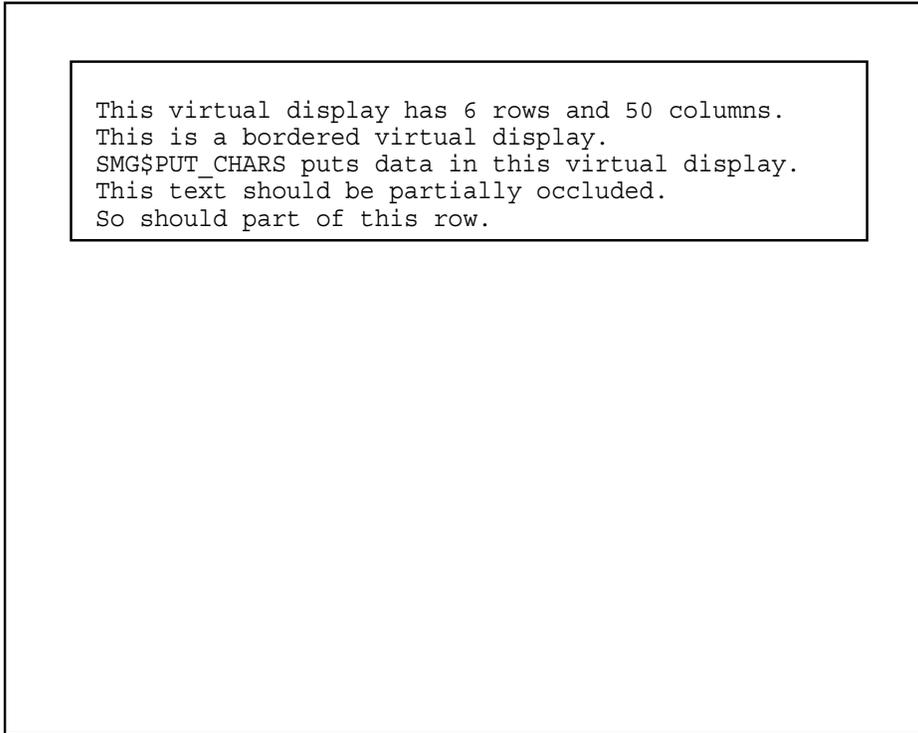
C+
C Now paste this new virtual display so that it occludes the other displays.
C-

STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( NEW_DISPLAY, PASTE1, 4, 20)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

END
```

この FORTRAN の例で生成される最初の仮想ディスプレイは 図 8-5 に示すとおりです。

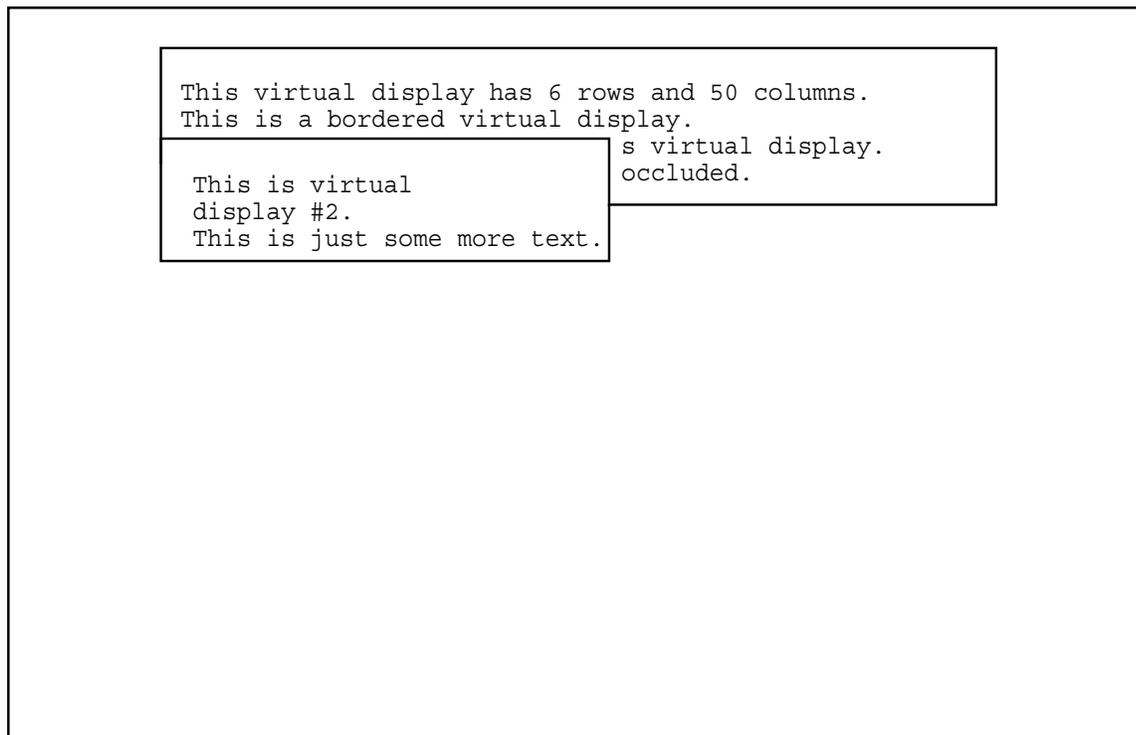
図 8-5 この例で生成される最初の仮想ディスプレイ



ZK-4808-GE

この FORTRAN の例で生成される 2 番目の仮想ディスプレイは 図 8-6 に示すと  
おりです。

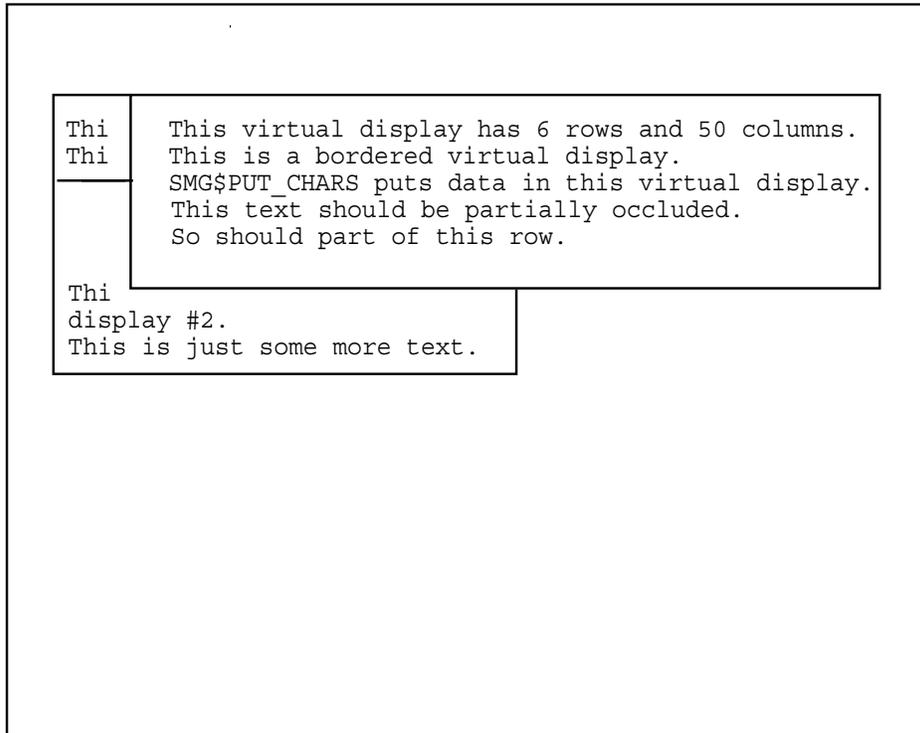
図 8-6 この例で生成される 2 番目の仮想ディスプレイ



ZK-4809-GE

SMG\$COPY\_VIRTUAL\_DISPLAY ルーチン呼び出した後に生成される出力は  
図 8-7 に示すとおりです。

図 8-7 SMG\$COPY\_VIRTUAL\_DISPLAY ルーチン呼び出した後で生成される出力



ZK-4810-GE

---

## SMG\$CREATE\_KEY\_TABLE

### Create Key Table

Create Key Table ルーチンは、キー定義を登録するためのテーブルを作成します。

---

#### 形式

SMG\$CREATE\_KEY\_TABLE *key-table-id*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

*key-table-id*

OpenVMS 用法	identifier
データ型	longword(unsigned)
アクセス	write only
受け渡し方	by reference

新しく生成されるキー・テーブルの識別子が書き込まれます。key-table-id 引数は、キー・テーブル識別子が書き込まれる符号なしロングワードのアドレスです。

---

#### 説明

SMG\$CREATE\_KEY\_TABLE ルーチンは、キー定義テーブルを作成します。この後、キー定義は SMG\$ADD\_KEY\_DEF, SMG\$LOAD\_KEY\_DEFS, SMG\$DEFINE\_KEY ルーチンを使用してこのテーブルに追加登録でき、また、SMG\$READ\_COMPOSED\_LINE ルーチンによって使用されます。

---

戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INSVIRMEM	仮想メモリが不足しています。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$CREATE\_MENU

### Fill the Virtual Display with a Menu

Fill the Virtual Display with a Menu ルーチンは、指定された仮想ディスプレイに、指定された行から選択メニューを表示します。

---

#### 形式

```
SMG$CREATE_MENU display-id ,choices [,menu-type] [,flags] [,row] [,rendition-set]
                [,rendition-complement]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

メニューを作成する仮想ディスプレイを指定します。display-id 引数はディスプレイ識別子を格納した符号なしロングワードのアドレスです。

choices

OpenVMS 用法	static array of char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

メニューに表示される各々の項目に対応する静的配列。choices 引数は文字列の静的配列を示すディスクリプタのアドレスです。空白のメニュー項目は認識されないの  
で、注意してください。



レイ内の対応する属性が設定されます。rendition-set 引数を使用して以下の属性を指定できます。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想ディスプレイの現在の省略時の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

#### rendition-complement

OpenVMS 用法	mask_longword
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

属性補足指定子。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、ディスプレイ内の対応する属性は補足されます。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数によって補足できます。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの2つの引数を組み合わせて使用すれば、1回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性をシステム設定値にする。
1	1	属性を無効に設定する。

## 説明

SMG\$CREATE\_MENU ルーチンは、仮想ディスプレイのスクロール領域上の指定された行から、メニュー選択のリストを表示します。メニュー項目は、仮想ディスプレ

イの 2 カラム目から表示が開始されます。複数のメニュー項目が同じ行にある場合には、4 カラム分の間隔をあけて分かれています。

選択は、以下のどれかの形式で、指定された属性で表示されます。

Vertical	各々のメニュー項目が各々の行に表示されます。
Horizontal	複数のメニュー項目がすべてひとつの行に表示されます。
Block	メニュー項目がマトリックス形式で表示されます。

仮想ディスプレイの境界の中に入らないメニュー項目は、ディスプレイに表示されませんが、SMG\$SELECT\_FROM\_MENU ルーチンによって後からスクロールされるために保存されます。省略時設定では、メニュー項目はシングル・スペースで表示されますが、ダブル・スペースに変更することも可能です。各々の項目は、一列に 4 カラムずつスペースをあけて表示されます。また、最も大きい文字列のサイズに等しいカラムの幅に統一して表示することも可能です。

SMG\$CREATE\_MENU ルーチンを呼び出したあとで、ユーザは、作成されたメニューを含む行を乱すような文字を、ディスプレイに出力してはいけません。メニューを乱すような文字が出力された場合には、予期しない結果になる場合があります。このメニューから項目を選択する場合には SMG\$SELECT\_FROM\_MENU ルーチンを使用してください。

SMG\$CREATE\_MENU ルーチンは、1 次元の NCA、VSA 文字配列をサポートしています。NCA、VSA 文字配列とは、HP ada、HP BASIC、HP COBOL、HP Fortran、HP Pascal と HP PL/I で作成される文字配列を含みます (しかし、SMG\$CREATE\_MENU ルーチンを HP BASIC から呼び出す場合は、配列宣言に MAP は必要です)。

メニュー項目の文字集合属性は、メニュー項目が表示される仮想ディスプレイの文字集合属性によって、決められます。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
LIB\$_xxxx	LIB\$CREATE_VM_ZONE、LIB\$GET_VM、LIB\$FREE_VM から戻される状態値。
SMG\$_xxxx	SMG\$PUT_CHARS、SMG\$BEGIN_DISPLAY_UPDATE、SMG\$END_DISPLAY_UPDATE ルーチンから戻される状態値。 注：SMG\$BEGIN_DISPLAY_UPDATE ルーチンと SMG\$END_DISPLAY_UPDATE ルーチンはサポートしていません。

---

例

SMG\$CREATE\_MENU ルーチンを使用した例については、SMG\$SELECT\_FROM\_MENU ルーチンを参照してください。

---

# SMG\$CREATE\_PASTEBOARD

## Create a Pasteboard

Create Pasteboard ルーチンはペーストボードを作成し、ペーストボード識別子を返します。

---

### 形式

```
SMG$CREATE_PASTEBOARD pasteboard-id [output-device] [number-of-pasteboard-rows]
                        [number-of-pasteboard-columns] [flags] [type-of-terminal]
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

新規生成されるペーストボードの識別子。pasteboard-id 引数は、ペーストボード識別子を格納する符号なしロングワードのアドレスです。

output-device

VMS 用法:	device_name
データ型:	character string
アクセス:	read only
受け渡し方:	by descriptor

ペーストボードに結びついた出力を書き出すファイル指定または論理名。output-device 引数は、出力装置名を示すディスクリプタのアドレスです。この引数を省略した場合の値は SYSSOUTPUT です。

---

#### 注意

---

日本語 SMG では、ファイルへの出力はサポートされていません。

---

## number-of-pasteboard-rows

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

output-device 引数で指定された装置の行数。number-of-pasteboard-rows 引数は、output-device 引数で指定された装置の行数を格納する符号付きロングワードのアドレスです。この値は、ペーストボードの行数となる場合もあります。

## number-of-pasteboard-columns

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

output-device 引数で指定された装置のカラム数。number-of-pasteboard-columns 引数は output-device 引数で指定された装置のカラム数を格納する符号付きロングワードのアドレスです。

## flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

ペーストボードで使用される属性。flags 引数は省略可能で、フラグを格納した符号なしロングワードのアドレスです。システムで設定されている動作では、ペーストボードが生成される時点でターミナル画面を消去します。以下の値が指定できます。

SMG\$M\_KEEP\_CONTENTS 初期化時に、画面を消去しません。日本語 SMG はすべてのターミナル画面を管理する際に最良の機能を提供するため、SMG\$M\_KEEP\_CONTENTS の使用は望ましくありません。

## type-of-terminal

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス write only  
 受け渡し方 by reference

ペーストボード関連の出力が書かれる SMG 内部の装置タイプ。type-of-terminal 引数はターミナル・タイプを格納する符号なしロングワードのアドレスです。

返される値は次のとおりです。

SMG\$K\_UNKNOWN  
 SMG\$K\_VTFOREIGN  
 SMG\$K\_HARDCOPY  
 SMG\$K\_VTTERMTABLE

SMG\$K\_VTTERMTABLE 以外の値が返される場合、SMG\$SNAPSHOT ルーチンを使用してペーストボードの内容を出力しなければなりません (現在の日本語 SMG では、SMG\$SNAPSHOT ルーチンはサポートされていません)。

---

## 説明

SMG\$CREATE\_PASTEBOARD ルーチンは新しいペーストボードを作成し、output-device 引数で指定された装置と結び付け、割り当てられた pasteboard-id 引数を返します。すでにペーストボードが割り当てられている装置にペーストボードが要求された場合は、このルーチンは既存のペーストボードの識別子を返し、SMG\$\_PASALREXI 状態コードを返します。

プログラムが SMG\$CREATE\_PASTEBOARD ルーチンと SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンを呼ぶ場合、SMG\$CREATE\_PASTEBOARD ルーチンが先に呼ばれることを確認してください。SMG\$CREATE\_PASTEBOARD ルーチンより先に SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンを呼ぶと、プログラムは正しく機能しません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INSVIRMEM	要求されたバッファを割り当てるのに必要な仮想メモリが不足しています。
SMG\$_PASALREXI	終了。この装置にはすでにペーストボードが存在しています。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	LIB\$GET_EF , LIB\$GET_VM , \$QIO , \$GETDVI \$ASSIGN から戻される状態値

---

## SMG\$CREATE\_SUBPROCESS

### Create and Initialize a Subprocess

Create and Initialize a Subprocess ルーチンは、DCL サブプロセスを生成し、そのサブプロセスを仮想ディスプレイに対応づけます。

---

#### 形式

SMG\$CREATE\_SUBPROCESS *display-id* [,AST-routine] [,AST-argument]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

新しく生成したサブプロセスに対応づけられる、仮想ディスプレイの識別子。display-id 引数は、この識別子を格納した符号なしロングワードのアドレスです。

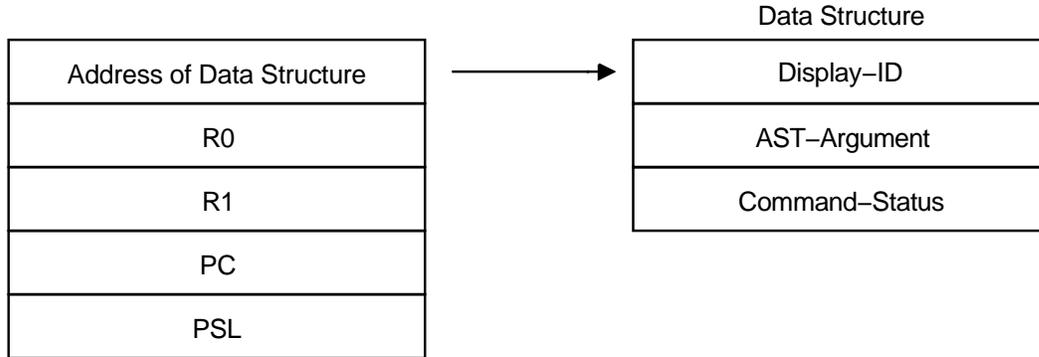
この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

AST-routine

OpenVMS 用法	ast_procedure
データ型	procedure entry mask
アクセス	call without stack unwinding
受け渡し方	by value

現在実行中のコマンドが終了したときに呼び出される AST ルーチン (省略可能)。AST-routine 引数はルーチンのプロシージャ・エントリ・マスクです。

AST ルーチンは 5 つの引数を使用して呼び出されます。最初の引数は、display-id 引数、AST-argument 引数、command-status 引数の値を格納したデータ構造を示すポインタです。AST ルーチンに対する他の 4 つの引数は R0、R1、PC、PSL です。



ZK-6508-GE

AST-routine 引数を指定した場合には、SMG\$EXECUTE\_COMMAND ルーチンはそのルーチンに渡されたコマンドをバッファに登録し、各コマンドを順に実行し、各コマンドが終了すると、指定された AST ルーチンを呼び出します。AST-routine 引数を指定しなかった場合には、SMG\$EXECUTE\_COMMAND ルーチンは、指定されたコマンドが終了するのを待ち、その後、制御をユーザに戻します。

#### AST-argument

OpenVMS 用法 user\_arg  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by value

AST ルーチンに対して指定する引数 (省略可能)。AST-argument 引数は、AST ルーチンに渡される値を格納した符号なしロングワードです。

---

## 説明

SMG\$CREATE\_SUBPROCESS ルーチンを使用すれば、DCL サブプロセスを生成し、このサブプロセスを仮想ディスプレイに対応づけることができます (サブプロセスは SET NOVERIFY と SET NOON の DCL コマンドを使用して初期化されます)。その後、メイン・プロセスから SMG\$EXECUTE\_COMMAND ルーチンを使用してサブプロセスが実行するコマンドを指定できます。

プロセス間通信はメール・ボックスを使用して実行されます。したがって、入力コマンドと出力テキストを制御できます。コマンドをバッファに登録する場合には、オプションとして AST ルーチンを使用することにより、コマンドが終了したことをメイン・プロセスに通知します。このルーチンを使用するために、ブロードキャスト・トラッピングと要求されていない入力を禁止する必要はありません。

サブプロセスを生成する前に、日本語 SMG は、必要なメールボックスとサブプロセスを作成するための十分な資源があるかどうかを確認します。BYTLM の残存値は少なくとも 5,000 でなければならず、PRCLM の残存値は少なくとも 1 でなければなりません。

日本語 SMG は、ユーザが SMG\$DELETE\_SUBPROCESS ルーチンを呼び出さずに処理を終了した場合、サブプロセスを削除する終了ハンドラを宣言します。しかし、場合によっては、これらの機能が提供する終了ハンドラが実行されないことがあります。この場合には、DCL の SHOW PROCESS/SUB コマンドと DCL の STOP コマンドを使用してサブプロセスを削除しなければなりません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INSQUOCRE	サブプロセスを生成するのに必要なクォータが不足しています。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_SUBALREXI	この display-id 引数に対応するサブプロセスがすでに存在します (代替正常終了状態)。
SS\$_xxxx	\$GETDVI, \$GETJPI, \$DCLEXH, または \$CREMBX から戻された状態値。
LIB\$_xxxx	LIB\$SPAWN, LIB\$GET_EF, LIB\$GET_VM から戻された状態値。

---

# SMG\$CREATE\_VIEWPORT

## Create a Virtual Viewport

Create a Virtual Viewport ルーチンは、ビューポートを作成し、そのビューポートを仮想ディスプレイに対応づけます。ビューポートの位置とサイズは呼び出し側が指定します。

---

### 形式

```
SMG$CREATE_VIEWPORT display-id ,viewport-row-start ,viewport-column-start ,viewport-number-rows
,viewport-number-columns
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

新しく作成されるビューポートに対応づけられる、仮想ディスプレイのディスプレイ識別子。display-id 引数は、この識別子を格納した符号なしロングワードのアドレスです。

viewport-row-start

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

ビューポートの 1 行目となる仮想ディスプレイの行番号。viewport-row-start 引数は、行番号を格納した符号付きロングワードのアドレスです。

## viewport-column-start

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

ビューポートの 1 カラム目となる仮想ディスプレイのカラム番号。viewport-column-start 引数は、カラム番号を格納した符号付きロングワードのアドレスです。

viewport-column-start 引数で指定されたカラムが、全角文字の右半分であった場合には、そのカラムは表示されません。

## viewport-number-rows

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

ビューポート内の行数。viewport-number-rows 引数は、新しく作成されるビューポート内の行数を格納した符号付きロングワードのアドレスです。

## viewport-number-columns

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

ビューポート内のカラム数。viewport-number-columns 引数は、新しく作成されるビューポート内のカラム数を格納した符号付きロングワードのアドレスです。

---

 説明

SMG\$CREATE\_VIEWPORT ルーチンはビューポートを作成し、そのビューポートを特定の仮想ディスプレイに対応づけます。仮想ディスプレイはビューポートを作成する前に作成しておかなければならず、各仮想ディスプレイに対してビューポートを 1 つだけ作成できます。ビューポートを表示するには、SMG\$PASTE\_VIRTUAL\_DISPLAY ルーチンを呼び出すことにより、仮想ディスプレイをペーストしなければなりません。仮想ディスプレイのビューポートの中に入る部分だけが表示されます。ビューポートは SMG\$DELETE\_VIEWPORT ルーチンを使用して削除できます。

viewport-column-start 引数で指定されたカラムが全角文字の右半分であった場合には、そのカラムは表示されません。また、ビューポートが全角文字の左半分で終了した場合にも、そのカラムは表示されません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVARG	行数またはカラム数が 0 より小さい値です。
SMG\$_INVCOL	誤ったカラム番号が指定されました。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_INVROW	誤った行番号が指定されました。
SMG\$_WINEXISTS	ビューポートはすでに仮想ディスプレイ内に存在します (代替正常終了状態)。
SMG\$_WRONUMARG	引数の数が誤っています。

---

**例**

```
1. C+
C This FORTRAN example creates two virtual displays, one
C being a copy of the other. The initial virtual display is
C filled and pasted to the pasteboard. The second virtual
C display is assigned a viewport and then pasted to the
C pasteboard. Therefore, only the section of the second
C virtual display that falls inside the viewport is visible.
C-
  IMPLICIT INTEGER (A-Z)
  INCLUDE '($SMGDEF)'

C Create the Virtual Display. Give it a border.
  ROWS = 9
  COLUMNS = 32

  STATUS = SMG$CREATE_VIRTUAL_DISPLAY
    1 (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER )
  IF (.NOT. STATUS) call lib$signal(%val(STATUS))

C Create the Pasteboard
  STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
  IF (.NOT. STATUS) call lib$signal(%val(STATUS))

C Put data in the Virtual Display
  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 1, you see.', 1, 1)
  IF (.not. STATUS) call lib$signal(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 2, you see.', 2, 1)
  IF (.not. STATUS) call lib$signal(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 3, you see.', 3, 1)
  IF (.not. STATUS) call lib$signal(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 4, you see.', 4, 1)
  IF (.not. STATUS) call lib$signal(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 5, you see.', 5, 1)
  IF (.not. STATUS) call lib$signal(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 6, you see.', 6, 1)
  IF (.not. STATUS) call lib$signal(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 7, you see.', 7, 1)
  IF (.not. STATUS) call lib$signal(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 8, you see.', 8, 1)
  IF (.not. STATUS) call lib$signal(%val(STATUS))
```

```

STATUS = SMG$PUT_CHARS ( DISPLAY1,
  1 'This is row number 9, you see.', 9, 1)
IF (.not. STATUS) call lib$signal(%val(STATUS))

C Paste the Virtual Display
STATUS = SMG$PASTE_VIRTUAL_DISPLAY (DISPLAY1, PASTE1, 2, 2)
IF (.NOT. STATUS) call lib$signal(%VAL(STATUS))

      STATUS = SMG$LABEL_BORDER (DISPLAY1, 'Full Display',,,SMG$M_BOLD)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$COPY_VIRTUAL_DISPLAY (DISPLAY1, DISPLAY2)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$LABEL_BORDER (DISPLAY2, 'Viewport',,,SMG$M_BOLD)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$CREATE_VIEWPORT ( DISPLAY2, 3, 9, 3, 12)
IF (.NOT. STATUS) call lib$signal(%VAL(STATUS))

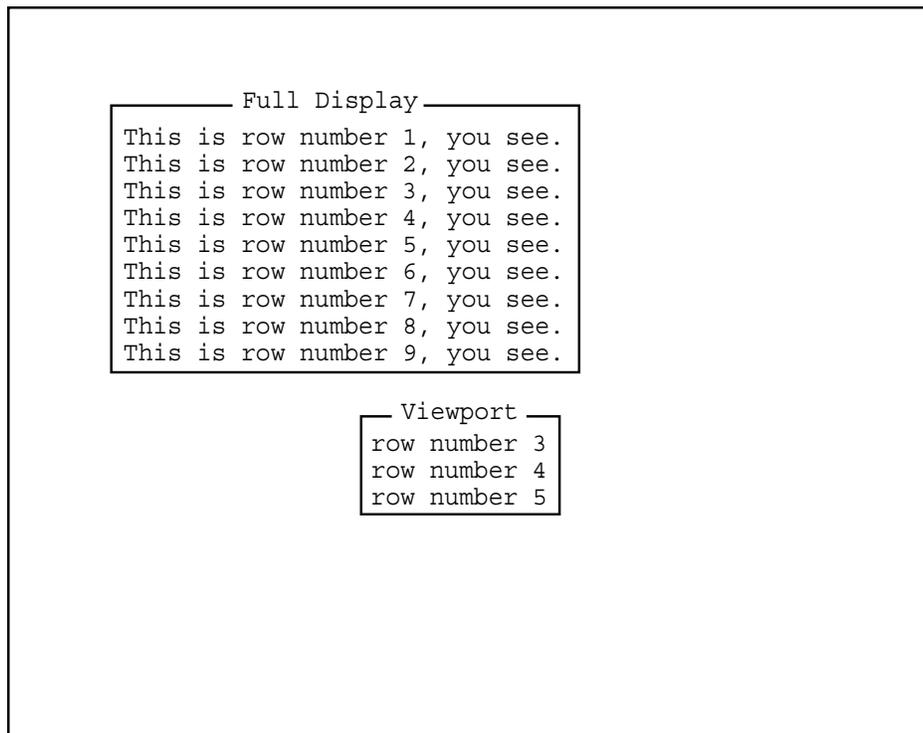
STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY2, PASTE1, 15, 20)
IF (.NOT. STATUS) call lib$signal(%VAL(STATUS))

END

```

この FORTRAN の例では、最初の仮想ディスプレイが 2 番目の仮想ディスプレイにコピーされます。2 番目の仮想ディスプレイにはビューポートが対応づけられています。2 番目の仮想ディスプレイをペーストすると、仮想ディスプレイの中でビューポートの内部だけが表示されます。これは図 8-8 に示すとおりです。

図 8-8 ビューポートを作成することにより生成される出力



ZK-6424-GE

---

# SMG\$CREATE\_VIRTUAL\_DISPLAY

## Create a Virtual Display

Create a Virtual Display ルーチンは、仮想ディスプレイを作成してその識別子を返します。

---

### 形式

SMG\$CREATE\_VIRTUAL\_DISPLAY

*number-of-rows ,number-of-columns ,display-id [,display-attributes]  
[,video-attributes] [,character-set]*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

number-of-rows

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

新規に作成される仮想ディスプレイの行数。number-of-rows 引数は、行数を格納した符号付きロングワードのアドレスです。

number-of-columns

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

新規に作成する仮想ディスプレイのカラム数。number-of-columns 引数は、カラム数を格納した符号付きロングワードのアドレスです。

## display-id

OpenVMS 用法 identifier  
 データ型 longword (unsigned)  
 アクセス write only  
 受け渡し方 by reference

新規に作成した仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を書き込まれる符号なしロングワードのアドレスです。

## display-attributes

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

新規に作成する仮想ディスプレイの属性。display-attributes 引数は、仮想ディスプレイの属性を格納したロングワード・ビット・マスクのアドレスです。この引数に対して指定できる値は次のとおりです。

SMG\$M_BORDER	境界付き仮想ディスプレイを指定します。省略した場合には、仮想ディスプレイ面に境界は表示されません。
SMG\$M_BLOCK_BORDER	ブロック形境界付き仮想ディスプレイを指定します。省略した場合は、仮想ディスプレイに境界は表示されません。
SMG\$M_DISPLAY_CONTROLS	漢字ターミナルがグラフィック文字をサポートする場合は、キャリッジ・リターンやライン・フィードなどの制御文字をグラフィック文字として表示することを指定します。
SMG\$M_TRUNC_ICON	仮想ディスプレイの幅をこえる行を切り捨てた際に、その場所にアイコン (通常は菱形) を表示することを指定します。

## video-attributes

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

仮想ディスプレイの全出力に適用される属性のシステム設定値。video-attributes 引数は、ビデオ属性マスクを格納した符号なしロングワードのアドレスです。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

挿入されるテキストの文字集合。character-set 引数は、文字集合指定子を格納した符号なしロングワードのアドレスです。この引数を省略した場合、SMG\$DEFAULT\_CHARACTER\_SET 論理名が用いられます。この論理名が無効な文字集合を指していたり、定義されていなかった場合には、システム設定値は未定義になります。この引数に対して指定できる文字集合は、以下のとおりです。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

SMG\$DEFAULT\_CHARACTER\_SET 論理名については、表 2-2 を参照してください。

---

 説明

SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンは、新しい仮想ディスプレイを作成し、仮想ディスプレイの識別子を返します。仮想ディスプレイを作成したばかりの状態では、その仮想ディスプレイには空白が含まれ、仮想カーソルは 1 行目、1 カラム目に設定されます。仮想スクロール領域は仮想ディスプレイ全体に設定されます。仮想ディスプレイを表示可能な状態に設定するには、SMG\$PASTE\_VIRTUAL\_DISPLAY ルーチンを使用します。

オリジナルの SMG ルーチンは、ASCII 文字集合と DEC 特殊文字集合のみをサポートします。日本語 SMG ではこの他に SMG\$C\_SUPPLEMENTAL、SMG\$C\_KANJI、SMG\$C\_JIS\_KANA、SMG\$C\_SDK が指定できます。

## 戻される状態値

SS\$ _NORMAL	正常終了。
LIB\$ _INSVIRMEM	仮想メモリが不足しています。
SMG\$ _DSPIN_USE	同じディスプレイへの日本語 SMG ルーチンコールを中断させた AST ルーチンから、日本語 SMG ルーチンが呼ばれました。
SMG\$ _INVARG	引数が誤っています。video-attributes 引数または display-attributes 引数に認識されない値が指定されています。
SMG\$ _WRONUMARG	引数の数が誤っています。

## 例

```

1. C+
   C This FORTRAN example program demonstrates the use of
   C SMG$ERASE_PASTEBOARD.
   C-

      IMPLICIT INTEGER*4 (A-Z)
      CHARACTER*80   OUT_STR,TRIM_STR
      CHARACTER*18   PROMPT           /'Please enter data '/

      SMG$M_BOLD = 1
      SMG$M_REVERSE = 2
      SMG$M_BLINK = 4
      SMG$M_UNDERLINE = 8

C+
C Establish the terminal keyboard as the virtual keyboard
C by calling SMG$CREATE_VIRTUAL_KEYBOARD.
C-

      STATUS = SMG$CREATE_VIRTUAL_KEYBOARD(KEYBOARD_ID,,,)
      IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Establish the terminal screen as a pasteboard using
C SMG$CREATE_PASTEBOARD.
C-

      STATUS = SMG$CREATE_PASTEBOARD (NEW_PID,,,)
      IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Establish a virtual display region by
C calling SMG$CREATE_VIRTUAL_DISPLAY.
C-

```

```

        STATUS = SMG$CREATE_VIRTUAL_DISPLAY (5,80,DISPLAY_ID,,)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Paste the virtual display to the screen, starting at
C row 10, column 15. To paste the virtual display, use
C SMG$PASTE_VIRTUAL_DISPLAY.
C-
        STATUS = SMG$PASTE_VIRTUAL_DISPLAY(DISPLAY_ID,NEW_PID,10,15)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Prompt the user for input, and accept that input using
C SMG$READ_STRING.
C-
        STATUS = SMG$READ_STRING(KEYBOARD_ID,OUT_STR,PROMPT,,,,,,,,)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Clear the screen using SMG$ERASE_PASTEBOARD.
C-
        STATUS = SMG$ERASE_PASTEBOARD (NEW_PID)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Trim any trailing blanks from the user input
C by calling STR$TRIM.
C-
        STATUS = STR$TRIM(TRIM_STR,OUT_STR,STR_LEN)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Display the data input by the user using SMG$PUT_CHARS
C and SMG$PUT_LINE.
C-
        STATUS = SMG$PUT_CHARS(DISPLAY_ID,'You entered: ',,,,,,,,,)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
        STATUS = SMG$PUT_LINE(DISPLAY_ID,TRIM_STR(1:STR_LEN),,
1                               SMG$M_REVERSE,0,,)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
        END

```

このFORTRANプログラムの例から生成される出力は、図 8-9 に示すとおりです。

## SMG\$CREATE\_VIRTUAL\_DISPLAY

図 8-9 SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチン呼び出す FORTRAN のプログラムが生成する出力



ZK-4100-GE

---

# SMG\$CREATE\_VIRTUAL\_KEYBOARD

## Create a Virtual Keyboard

Create a Virtual Keyboard ルーチンは、仮想キーボードを作成してキーボード識別子を返します。

---

### 形式

SMG\$CREATE\_VIRTUAL\_KEYBOARD  
*keyboard-id* [,*input-device*] [,*default-filespec*] [,*resultant-filespec*] [,*recall-size*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

keyboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

新規に作成したキーボードの識別子。keyboard-id 引数は、キーボード識別子を格納する符号なしロングワードのアドレスです。

input-device

VMS 用法:	char_string
データ型:	character string
アクセス:	read only
受け渡し方:	by descriptor

仮想キーボードで使用される、ファイル指定、ファイル論理名、または漢字ターミナルの論理名。input-device 引数は、ファイル指定を示すディスクリプタのアドレスです。省略時の設定値は SYSSINPUT です。

---

**注意**

---

日本語 SMG では、ファイルからの入力はサポートされていません。

---

**default-filespec**

VMS 用法: char\_string  
データ型: character string  
アクセス: read only  
受け渡し方: by descriptor

ファイル指定のシステム設定値。default-filespec 引数は、ファイル指定のシステム設定値を示すディスクリプタのアドレスです。default-filespec 引数を省略した場合は、空文字列を使用します。

input-device 引数でファイル名とファイル・タイプを指定し、default-filespec 引数で装置とディレクトリを指定することもできます。

---

**注意**

---

日本語 SMG では、ファイルからの入力はサポートされていません。

---

**resultant-filespec**

VMS 用法: char\_string  
データ型: character string  
アクセス: write only  
受け渡し方: by descriptor

実際に使用されたファイル指定。resultant-filespec 引数は、実際に使用されたファイル指定が、完全に展開された形式で書き込まれる文字列を示すディスクリプタのアドレスです。

---

**注意**

---

日本語 SMG では、ファイルからの入力はサポートされていません。

---

**recall-size**

OpenVMS 用法 byte\_unsigned  
データ型 byte (unsigned)  
アクセス read only  
受け渡し方 by reference

再呼び出しのために保存される入力行数。recall-size 引数は省略可能で、行数を格納した符号なしバイトのアドレスです。0 に設定すると入力行を再び呼び出すことができなくなります。recall-size 引数を省略した場合は、20 行が保存されます。

---

 説明

SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンは、ファイル指定 (ターミナル名または RMS ファイル) と仮想キーボードの関係を設定します。入力ストリームを識別するため、キーボード識別子は他の SMG\$ プロシージャに渡されます。SMG\$CREATE\_PASTEBOARD ルーチンも同時に呼んでいる場合には、それが SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンを呼び出す前に呼ばれていなければなりません。

プロシージャは漢字ターミナルにチャンネルを割り当て、ターミナル・キーボードにアプリケーション・モード (サポート時のみ) を設定します。仮想キーボードが削除される際に、これらの属性にはもとの値が復元されます。仮想キーボードはイメージの終了時に自動的に削除されます。SMG\$DELETE\_VIRTUAL\_KEYBOARD ルーチンで削除することもできます。

---

 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INSEF	イベント・フラグ数が不足しています。
LIB\$_INSTRDES	仮想ディスクリプタが誤っています。
LIB\$_INSVIRMEM	仮想メモリが不足しています。
SMG\$_FILTOOLON	ファイル指定が長すぎます (最大 255 バイト)。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	\$OPEN または \$CONNECT により RMS から戻される状態値。 \$GETDVIW, \$ASSIGN, または \$DCLEXH から戻される状態値。

---

 例

```

1. C+
   C This FORTRAN example program demonstrates the use of
   C SMG$CREATE_VIRTUAL_KEYBOARD, SMG$CREATE_KEY_TABLE,
   C SMG$ADD_KEY_DEF, and SMG$READ_COMPOSED_LINE.
   C-

       INTEGER SMG$CREATE_VIRTUAL_KEYBOARD, SMG$CREATE_KEY_TABLE
       INTEGER SMG$ADD_KEY_DEF, SMG$READ_COMPOSED_LINE
       INTEGER SMG$DELETE_KEY_DEF, KEYBOARD, KEYS, STATUS

   C+
   C Include the SMG definitions. In particular, we want SMG$M_KEY_NOECHO
   C and SMG$M_KEY_TERMINATE.
   C-
  
```

## SMG\$CREATE\_VIRTUAL\_KEYBOARD

```
        INCLUDE '($SMGDEF)'  
  
C+  
C Create a virtual keyboard (using SMG$CREATE_VIRTUAL_KEYBOARD)  
C and create a key table (using SMG$CREATE_KEY_TABLE).  
C-  
  
        STATUS = SMG$CREATE_VIRTUAL_KEYBOARD (KEYBOARD)  
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
        STATUS = SMG$CREATE_KEY_TABLE (KEYTABLE)  
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
C+  
C Prompt the user with the following instructions.  
C-  
  
        WRITE (6,*) 'When you see the prompt (->), strike the following'  
        WRITE (6,*) 'keys (on the KEYPAD): '  
        WRITE (6,*) '          PF1 '  
        WRITE (6,*) '          5 '  
        WRITE (6,*) '          PF3 '  
        WRITE (6,*) ' '  
        WRITE (6,*) 'When you have done this, the following sentence'  
        WRITE (6,*) '(and nothing more) should appear following the'  
        WRITE (6,*) 'prompt: '  
        WRITE (6,*) '(PF3 should act as a carriage return.)'  
        WRITE (6,*) ' '  
        WRITE (6,*) 'NOW IS THE TIME FOR ALL TEXT TO APPEAR.'  
  
C+  
C Add key definitions by calling SMG$ADD_KEY_DEF.  
C-  
  
        STATUS = SMG$ADD_KEY_DEF (KEYTABLE, 'PF1', , ,  
1 'NOW IS THE TIME FOR '  
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
        STATUS = SMG$ADD_KEY_DEF (KEYTABLE, 'KP5', , ,  
1 'TEXT TO APPEAR. '  
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
        STATUS = SMG$ADD_KEY_DEF (KEYTABLE, 'PF3', , ,  
1 SMG$M_KEY_NOECHO + SMG$M_KEY_TERMINATE ,  
1 'THIS SHOULD NOT BE ECHOED. IF YOU CAN  
1 SEE THIS, AN ERROR EXISTS. '  
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
C+  
C Call SMG$READ_COMPOSED_LINE to read a line of input.  
C-  
  
        WRITE (6,*) ' '  
        STATUS = SMG$READ_COMPOSED_LINE (KEYBOARD, KEYTABLE, R_TEXT,  
1 '->')  
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
        END
```

出力例

\$ RUN example

When you see the prompt (->), strike the following  
keys (on the KEYPAD):

PF1

5

PF3

When you have done this, the following sentence  
(and nothing more) should appear following the  
prompt:

(PF3 should act as a carriage return.)

NOW IS THE TIME FOR ALL TEXT TO APPEAR.

->NOW IS THE TIME FOR ALL TEXT TO APPEAR.

\$

---

## SMG\$CURSOR\_COLUMN

### Return Cursor Column Position

Return Cursor Column Position ルーチンは、指定された仮想ディスプレイ内の仮想カーソルの、現在のカラム位置を戻します。

---

#### 形式

SMG\$CURSOR\_COLUMN *display-id*

---

#### 戻り値

OpenVMS 用法 longword\_unsigned  
データ型 longword (unsigned)  
アクセス write only  
受け渡し方 by value

SMG\$CURSOR\_COLUMN ルーチンは現在の仮想カーソルのカラム位置を戻します。

---

#### 引数

*display-id*

OpenVMS 用法 identifier  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

カラム位置が戻される対象となるディスプレイ。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

---

#### 説明

SMG\$CURSOR\_COLUMN ルーチンは、指定された仮想ディスプレイ内の、現在の仮想カーソルのカラム位置の値を格納するロングワードを戻します。display-id 引数を省略した場合には、このルーチンは SMG\$\_WRONUMARG を通知しま

す。display-id 引数が誤っている場合には、このルーチンは SMG\$\_INVDIS\_ID を通知します。

---

#### 戻される状態値

SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$CURSOR\_ROW

### Return Cursor Row Position

Return Cursor Row Position ルーチンは、指定された仮想ディスプレイ内で仮想カーソルの現在の行番号を戻します。

---

#### 形式

SMG\$CURSOR\_ROW *display-id*

---

#### 戻り値

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

SMG\$CURSOR\_ROW ルーチンは現在の行番号を戻します。

---

#### 引数

*display-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

行番号が戻される対象となるディスプレイ。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

---

#### 説明

SMG\$CURSOR\_ROW ルーチンは、指定された仮想ディスプレイの現在の仮想カーソルの行番号の値を格納するロングワードを戻します。display-id 引数を省略した場合には、このルーチンは SMG\$\_WRONUMARG を通知します。display-id 引数が誤っている場合には、このルーチンは SMG\$\_INVDIS\_ID を通知します。

---

戻される状態値

SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$DEFINE\_KEY

### Perform a DEFINE/KEY Command

Perform a DEFINE/KEY Command ルーチンは、ユーザが指定した DEFINE/KEY コマンドを実行します。

---

#### 形式

SMG\$DEFINE\_KEY *key-table-id* ,*command-string*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword(unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

*key-table-id*

OpenVMS 用法	identifier
データ型	longword(unsigned)
アクセス	read only
受け渡し方	by reference

DEFINE/KEY コマンドが実行されるキー定義テーブルの識別子。key-table-id 引数は、キー・テーブル識別子を格納した符号なしロングワードのアドレスです。

key-table-id 引数は SMG\$CREATE\_KEY\_TABLE ルーチンによって与えられます。

*command-string*

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

実行される DEFINE/KEY コマンドを格納した文字列。command-string 引数は、実行されるコマンドを示すディスクリプタのアドレスです。

DEFINE/KEY コマンドに対して指定できる修飾子は次のとおりです。

- /TERMINATE

- /NOECHO
- /LOCK
- /IF\_STATE
- /SET\_STATE

DEFINE/KEY コマンドの修飾子に対して、次の 2 つの制約事項が適用されます。

- /LOCK 修飾子を使用する場合には、/SET\_STATE 修飾子も使用しなければならない。
- /SET\_STATE 修飾子と/TERMINATE 修飾子をどちらとも使用する場合には、/LOCK 修飾子は使用できない。

---

## 説明

SMG\$DEFINE\_KEY ルーチンは、DEFINE/KEY コマンドを解析し、実行します。DEFINE/KEY コマンドを受け付けるだけで、そのコマンドを独自に解析しないプログラムは、このルーチンを使用できます。

SMG\$DEFINE\_KEY ルーチンは、CLISDCL\_PARSE ルーチンを呼び出すことにより、コマンド行を解析し、その後、必要に応じて SMG\$ADD\_KEY\_DEF ルーチンを呼び出します。もとのコマンドは、CLISDCL\_PARSE を呼び出すことにより復元できます。このプロシージャを使用する場合には、DCL コマンド言語インタプリタのもとでイメージを実行していなければなりません。

---

## 戻される状態値

SSS_NORMAL	正常終了。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	LIBSSCOPY_DXDX, CLIS\$ルーチン, SMG\$ADD_KEY_DEF から戻された状態値。

---

## SMG\$DEL\_TERM\_TABLE

### Delete Terminal Table

Delete Terminal Table ルーチンは、個人用の TERMTABLE.EXE に対するアクセスを終了し、仮想アドレス空間の割り当てを解除します。

---

#### 形式

SMG\$DEL\_TERM\_TABLE

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

なし

---

#### 説明

SMG\$DEL\_TERM\_TABLE ルーチンは、個人用の TERMTABLE.EXE に対するアクセスを終了します。このルーチンの呼び出しは必ずしも必要ありません。このルーチンは、個人用の TERMTABLE.EXE によって使用されていた仮想アドレス空間を、呼び出しプログラムが再利用しなければならないときに便利です。このルーチンを使用しなければならないのは、漢字ターミナルに対して直接に日本語 SMG ルーチン以外の入出力を実行する場合だけです。

---

#### 戻される状態値

SSS_NORMAL	正常終了。
------------	-------

---

# SMG\$DELETE\_CHARS

## Delete Characters

Delete Characters ルーチンは仮想ディスプレイのカラムを削除します。

---

### 形式

SMG\$DELETE\_CHARS *display-id ,number-of-columns ,start-row ,start-column*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

カラムが削除される仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

number-of-columns

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

削除カラム数。number-of-columns 引数は、削除するカラム数を格納した符号付きロングワードのアドレスです。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only

受け渡し方      by reference

削除開始行。start-row 引数は、削除操作を開始する行の位置を格納した符号付きロングワードのアドレスです。

start-column

OpenVMS 用法    longword\_signed  
データ型        longword (signed)  
アクセス        read only  
受け渡し方      by reference

削除開始カラム。start-column 引数は、削除操作を開始するカラムの位置を格納した符号付きロングワードのアドレスです。start-column 引数が全角文字の右半分である場合は、その文字の左半分は未定義文字になります。

---

## 説明

SMG\$DELETE\_CHARS ルーチンは、指定された行/カラム位置から、指定されたカラム数だけ削除します。この行の残りのカラムは左に移動し、削除された部分を埋めます。単一行中のカラムだけを削除します。

削除できるカラム数より多くのカラム数を指定した場合、SMG\$DELETE\_CHARS ルーチンは指定されたカラムから行の最後までのカラムを削除します。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、全角文字の右半分から削除操作が始まる場合は、その文字の左半分は未定義文字になります。また、全角文字の左半分で削除操作が終了する場合は、その文字の右半分は未定義文字になります。

このルーチンを終了した後に、仮想カーソルは削除された最初のカラム位置に移動します。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVARG	引数が誤っています。指定されたカラム数が仮想ディスプレイの幅をこえます。
SMG\$_INVCOL	カラム位置が誤っています。指定されたカラムは仮想ディスプレイの内部に含まれません。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行の位置が誤っています。指定された行は仮想ディスプレイの内部に含まれていません。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。

SMG\$\_WRONUMARG

引数の数が誤っています。

## 例

```

1. C+
   C This FORTRAN example program demonstrates the use of
   C SMG$DELETE_CHARS.
   C-

       INTEGER SMG$CREATE_VIRTUAL_DISPLAY, SMG$CREATE_PASTEBOARD
       INTEGER SMG$PASTE_VIRTUAL_DISPLAY, SMG$PUT_CHARS
       INTEGER SMG$DELETE_CHARS, DISPLAY1, PASTE1
       INTEGER ROWS, COLUMNS, BORDER, STATUS

   C+
   C Create the virtual display by calling SMG$CREATE_VIRTUAL_DISPLAY.
   C To give it a border, set BORDER = 1. No border would be BORDER = 0.
   C-

       ROWS = 7
       COLUMNS = 50
       BORDER = 1

       STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, BORDER)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
   C-

       STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Use SMG$PUT_CHARS to put data in the virtual display.
   C-

       STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This virtual display has 7 rows and 50 columns.', 2, 1)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

       STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This is a bordered virtual display.', 4, 1)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

       STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' SMG$PUT_CHARS puts data in this virtual display.', 6, 1)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Paste the virtual display to the pasteboard using
   C SMG$PASTE_VIRTUAL_DISPLAY.
   C-

       STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

```

## SMG\$DELETE\_CHARS

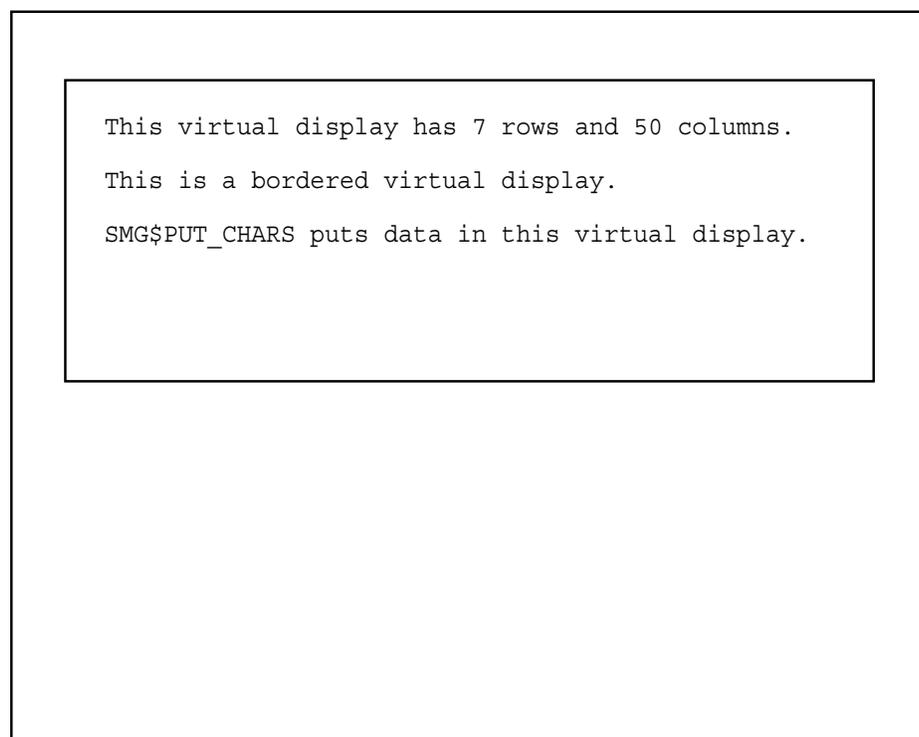
```
C+
C Call SMG$DELETE_CHARS to delete 4 characters from row 4
C starting from character (column) 14, removing the characters
C "rder" from the word "bordered".
C-

      STATUS = SMG$DELETE_CHARS ( DISPLAY1, 4, 4, 14)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      END
```

この FORTRAN プログラムによって、SMG\$DELETE\_CHARS ルーチン呼び出す前に生成される出力は図 8-10 に示すとおりです。

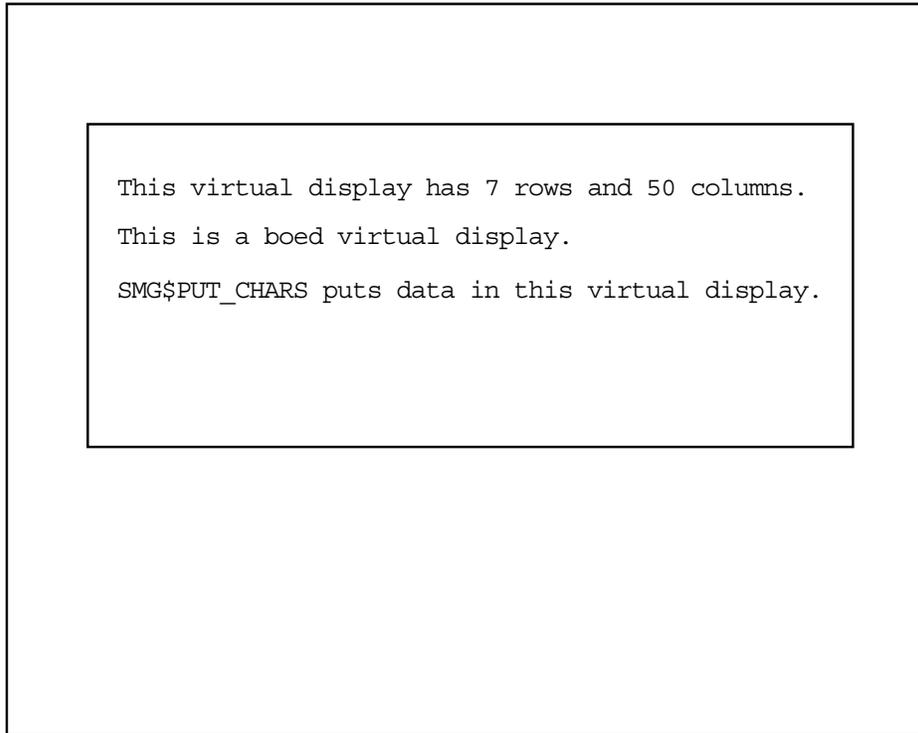
図 8-10 SMG\$DELETE\_CHARS ルーチン呼び出す前に生成される出力



ZK-4101-GE

SMG\$DELETE\_CHARS ルーチン呼び出した後で生成される出力は、図 8-11 に示すとおりです。

図 8-11 SMG\$DELETE\_CHARS ルーチン呼び出し後で生成される出力



ZK-4107-GE

---

## SMG\$DELETE\_KEY\_DEF

### Delete Key Definition

Delete Key Definition ルーチンは、指定されたキー定義テーブルからキー定義を削除します。

---

#### 形式

```
SMG$DELETE_KEY_DEF key-table-id ,key-name [,if-state]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword(unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

*key-table-id*

OpenVMS 用法	identifier
データ型	longword(unsigned)
アクセス	read only
受け渡し方	by reference

キー定義を削除するキー・テーブルを指定します。key-table-id 引数は、キー・テーブル識別子を格納した符号なしロングワードのアドレスです。

*key-name*

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

定義を削除するキーの名前を格納した文字列。key-name 引数は、キー名を示すディスクリプタのアドレスです。実際に使用される前に、key-name 引数から後続の空白は削除され、大文字に変換されます。

使用できるキー名については、本書の第3章の表 3-1 を参照してください。

if-state

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

key-name 引数をさらに条件指定する状態名を格納した文字列。if-state 引数は、状態名を示すディスクリプタのアドレスです。この引数を省略した場合には、無状態が使用されます。したがって、キーに複数の定義を割り当てて、if-state 引数の値に応じて適切な定義を使用する場合には、このルーチンがそれらの定義の中から 1 つの定義だけを削除します。

---

## 説明

SMG\$DELETE\_KEY\_DEF ルーチンは、指定されたキー定義テーブルからキー定義を削除します。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVKEYNAM	key-name 引数に誤りがあります。
SMG\$_INVKTB_ID	key-table-id 引数に誤りがあります。
SMG\$_KEYDEFPRO	キー定義が保護されています。
SMG\$_KEYNOTDEF	キーが定義されていません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$DELETE\_LINE

### Delete Line

Delete Line ルーチンは仮想ディスプレイの行を削除します。

---

#### 形式

SMG\$DELETE\_LINE *display-id ,start-row [,number-of-rows]*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

行が削除される仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

削除開始行。start-row 引数は、削除する最初の行の番号を格納した符号付きロングワードのアドレスです。

number-of-rows

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only

受け渡し方      by reference

削除行数。number-of-rows 引数は、削除する行数を格納した符号付きロングワードのアドレスです。

---

## 説明

SMG\$DELETE\_LINE ルーチンは、1 行以上を仮想ディスプレイから削除し、残りの行を上スクロールさせ削除された部分を埋めます。仮想ディスプレイの下部には空行が挿入されます。仮想カーソルは、start-row 引数で指定された行の最初のコラムに移動します。

---

## 戻される状態値

SS\$NORMAL	正常終了。
SMG\$INVARG	引数が誤っています。
SMG\$INVDIS_ID	display-id 引数が誤っています。
SMG\$INVROW	行の位置が誤っています。
SMG\$WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$WRONUMARG	引数の数が誤っています。

---

## 例

```

1. C+
   C This FORTRAN example program demonstrates the use of SMG$DELETE_LINE.
   C-
       IMPLICIT INTEGER (A-Z)
   C+
   C Create the virtual display by calling SMG$CREATE_VIRTUAL_DISPLAY.
   C To give it a border, set BORDER = 1. No border would be BORDER = 0.
   C-
       ROWS = 7
       COLUMNS = 50
       BORDER = 1
       STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, BORDER)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
   C+
   C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
   C-

```

## SMG\$DELETE\_LINE

```
STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$PUT_CHARS to put data in the virtual display.
C-

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This virtual display has 7 rows and 50 columns.', 2, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This is a bordered virtual display.', 4, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' SMG$PUT_CHARS puts data in this virtual display.', 6, 1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Paste the virtual display to the pasteboard using
C SMG$PASTE_VIRTUAL_DISPLAY.
C-

STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

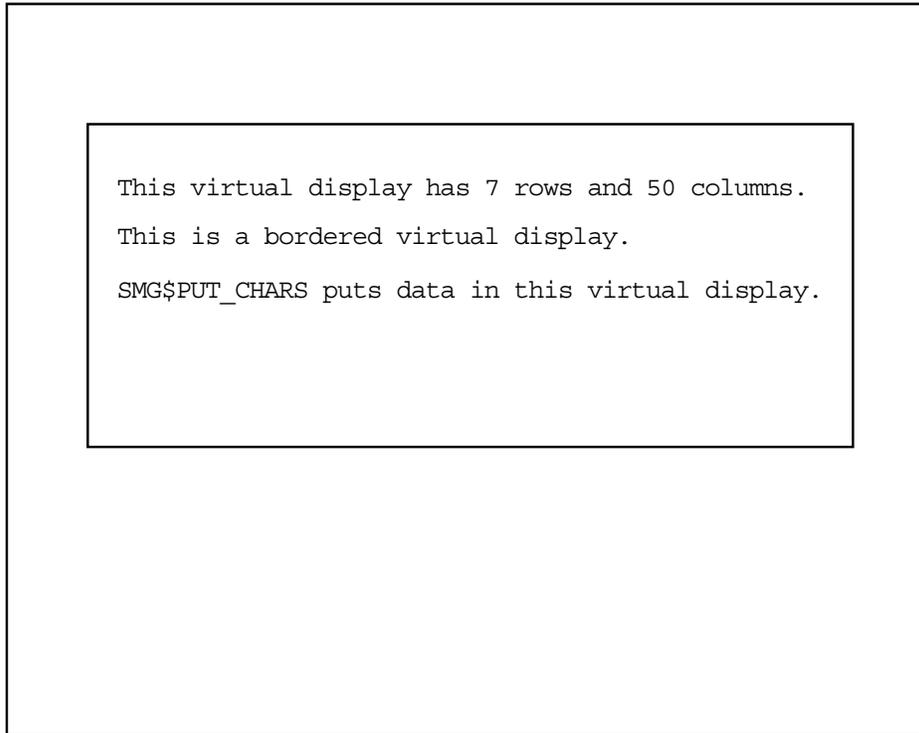
C+
C Call SMG$DELETE_LINE to delete rows 3, 4, and 5.
C-

STATUS = SMG$DELETE_LINE ( DISPLAY1, 3, 3)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

END
```

このFORTRANプログラムによって、SMG\$DELETE\_LINEルーチン呼び出す前に生成される出力は、図 8-12 に示すとおりです。

図 8-12 SMG\$DELETE\_LINE ルーチンを呼び出す前に FORTRAN プログラムが生成する出力

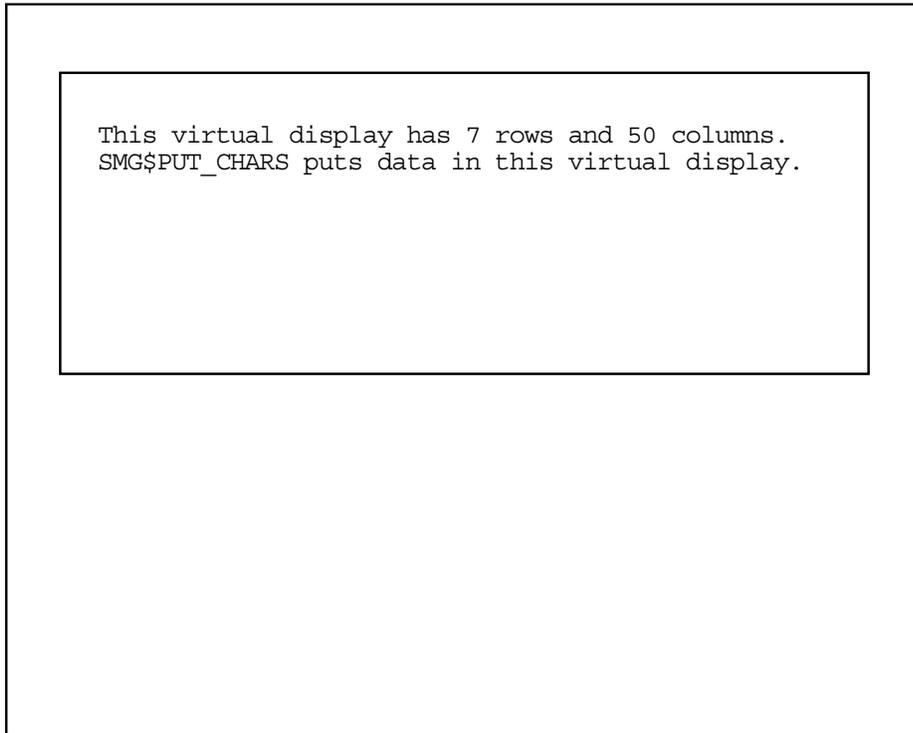


ZK-4103-GE

SMG\$DELETE\_LINE ルーチンを呼び出した後に生成される出力は、図 8-13 に示すとおりです。

## SMG\$DELETE\_LINE

図 8-13 SMG\$DELETE\_LINE ルーチンを呼び出した後に生成される出力



ZK-4109-GE

---

# SMG\$DELETE\_MENU

## End Access to a Menu in the Virtual Display

End Access to a Menu in the Virtual Display ルーチンは、指定された仮想ディスプレイのメニュー選択へのアクセスを終了します。

---

### 形式

SMG\$DELETE\_MENU *display-id* [*flags*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

メニュー選択を画面表示する仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

flags

OpenVMS 用法	mask_longword
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

省略可能なビットマスクであり、メニューを削除するときに可能なアクションを指定します。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。この引数に対して指定できる値は、SMG\$M\_ERASE\_MENU のみです。この値が指定された場合は、メニュー項目が含まれている行すべてが削除されます。

---

**説明**

SMG\$DELETE\_MENU ルーチンは、指定された仮想ディスプレイでのメニュー選択を中断します。flags 引数は省略可能で、メニューが削除された場合に、ディスプレイからメニュー項目を削除することを指定できます。

---

**戻される状態値**

SS\$_NORMAL	正常終了。
LIB\$_xxxx	LIB\$FREE_VM から戻された状態値。
SMG\$_xxxx	SMG\$ERASE_DISPLAY ルーチンから戻された状態値。

---

# SMG\$DELETE\_PASTEBOARD

## Delete Pasteboard

Delete Pasteboard ルーチンはペーストボードを削除します。

---

### 形式

SMG\$DELETE\_PASTEBOARD *pasteboard-id* [*flags*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

削除されるペーストボード識別子。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

flags

VMS 用法:	mask_longword
データ型:	longword (unsigned)
アクセス:	read only
受け渡し方:	by reference

ペーストボード削除後にターミナル画面消去を行うかどうかを指定するフラグ。flags 引数は省略可能で、フラグを格納した符号なしロングワードのアドレスです。指定できるのは、以下の値です。

0 画面を消去しない。

## SMG\$DELETE\_PASTEBOARD

SMG\$M\_ERASE\_PBD

画面を消去する (省略時の設定値)。

---

### 説明

SMG\$DELETE\_PASTEBOARD ルーチンは仮想ディスプレイに全出力を書き出し、ペーストボードの使用をすべて終了させて、ペーストボードに割り当てられていた資源を解放します。

---

### 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVPAS_ID	pasteboard-id 引数が誤っています。
SMG\$_NOTPASTED	指定された仮想ディスプレイは指定されたペーストボードにペーストされていません。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	\$DASSGN, LIB\$FREE_VM, LIB\$FREE_EF または SMG\$FLUSH_BUFFER から戻される状態値。

---

# SMG\$DELETE\_SUBPROCESS

## Terminate a Subprocess

Terminate a Subprocess ルーチンは、SMG\$CREATE\_SUBPROCESS ルーチンを使用して生成したサブプロセスを削除します。

---

### 形式

SMG\$DELETE\_SUBPROCESS *display-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

*display-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

削除するサブプロセスに対応づけられている仮想ディスプレイの識別子。display-id 引数は、この仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

---

### 説明

SMG\$DELETE\_SUBPROCESS ルーチンは、SMG\$CREATE\_SUBPROCESS ルーチン呼び出すことにより生成されたサブプロセスを削除します。日本語 SMG は独自の終了ハンドラを備えているので、ユーザ独自の終了ハンドラの内部から SMG\$DELETE\_SUBPROCESS ルーチンを起動しないでください。詳しい説明は、第 4 章の 第 4.4 節を参照してください。

戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_NOSUBEXI	削除するプロセスが存在しません。
SS\$_xxxx	\$DELPRC から戻された状態値。
LIB\$_xxxx	LIB\$FREE_VM から戻された状態値。

---

# SMG\$DELETE\_VIEWPORT

## Delete a Viewport

Delete a Viewport ルーチンは、ペーストされているペーストボードから指定されたビューポートを削除します。

---

### 形式

SMG\$DELETE\_VIEWPORT *display-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

削除するビューポートに対応づけられている仮想ディスプレイの識別子。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

---

### 説明

SMG\$DELETE\_VIEWPORT ルーチンはビューポートを削除します。ビューポートは、それがペーストされているペーストボードから自動的に“アンペースト”されます。しかし、削除されるビューポートに対応づけられている仮想ディスプレイは削除されません。この仮想ディスプレイを表示するには、SMG\$PASTE\_VIRTUAL\_DISPLAY ルーチンを使用して仮想ディスプレイをペーストボードにペーストしなければなりません。この仮想ディスプレイを削除するには、SMG\$DELETE\_VIRTUAL\_DISPLAY ルーチンを使用します。

---

戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_NO_WINASSOC	仮想ディスプレイに対応づけられているビューポートはありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$DELETE\_VIRTUAL\_DISPLAY

## Delete Virtual Display

Delete Virtual Display ルーチンは仮想ディスプレイを削除します。

---

### 形式

SMG\$DELETE\_VIRTUAL\_DISPLAY *display-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

削除する仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

---

### 説明

SMG\$DELETE\_VIRTUAL\_DISPLAY ルーチンは仮想ディスプレイを削除し、その仮想ディスプレイがペーストされているペーストボードから、仮想ディスプレイを削除します。また、仮想ディスプレイに割り当てられているバッファ空間の割り当てを解除します。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_NOTPASTED	指定された仮想ディスプレイが指定されたペーストボードにペーストされていません。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	LIB\$FREE_VM から戻される状態値。 DELPRC から戻される状態値。

---

# SMG\$DELETE\_VIRTUAL\_KEYBOARD

## Delete Virtual Keyboard

Delete Virtual Keyboard ルーチンは仮想キーボードを削除します。

---

### 形式

SMG\$DELETE\_VIRTUAL\_KEYBOARD *keyboard-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

*keyboard-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

削除されるキーボード識別子。keyboard-id 引数は、キーボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンによって与えられます。

---

### 説明

SMG\$DELETE\_VIRTUAL\_KEYBOARD ルーチンは、仮想キーボードを削除します。仮想キーボード作成時に設定されたターミナル属性を以前の値に設定し、キーボード・モード (数値またはアプリケーション・キーパッド・モード) は最初の状態に設定されます。さらに、チャンネルの割り当ても解除されます。

日本語 SMG ルーチンは独自の終了ハンドラを備えているため、ユーザ独自の終了ハンドラからこのルーチンを呼ばないでください。

---

戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVKBD_ID	keyboard-id 引数が誤っています。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$DISABLE\_BROADCAST\_TRAPPING

## Disable Broadcast Trapping

Disable Broadcast Trapping ルーチンは、指定された漢字ターミナルに対して、ブロードキャスト・メッセージのトラッピングを禁止します。

---

### 形式

SMG\$DISABLE\_BROADCAST\_TRAPPING *pasteboard-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

*pasteboard-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

操作の対象となる漢字ターミナルのペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

---

### 説明

SMG\$DISABLE\_BROADCAST\_TRAPPING ルーチンは、指定された漢字ターミナルに対して、ブロードキャスト・メッセージのトラッピングを禁止します。SMG\$DISABLE\_BROADCAST\_TRAPPING ルーチンは、SMG\$SET\_BROADCAST\_TRAPPING ルーチンを使用して設定されたメールボックスの割り当てを解除し、ターミナル属性を再設定します。したがって、ユーザは LIB\$SPAWN ルーチンを呼び出すことができるようになります。

SMG\$SET\_BROADCAST\_TRAPPING ルーチンを使用して設定したブロードキャスト・トラッピングを禁止する場合には、このルーチンを使用しなければなりません。

ブロードキャスト・トラッピングを禁止する場合には、漢字ターミナルのキューに登録されたブロードキャスト・メッセージは失われます。SMG\$SET\_BROADCAST\_TRAPPING ルーチンを使用してブロードキャスト・トラッピングを許可した後、イメージが終了する前に、SMG\$DISABLE\_BROADCAST\_TRAPPING ルーチンを使用してブロードキャスト・トラッピングを禁止しなかった場合には、漢字ターミナルにブロードキャストされたメッセージは、イメージが終了するときに失われます。

ブロードキャスト・トラッピングと、要求されていない入力のトラッピングをどちらも許可した場合には、メールボックスの割り当てを解除するために、SMG\$DISABLE\_BROADCAST\_TRAPPING ルーチンと SMG\$DISABLE\_UNSOLICITED\_INPUT ルーチンの両方を起動しなければなりません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	\$QIOW から戻される状態値。

---

# SMG\$DISABLE\_UNSOLICITED\_INPUT

## Disable Unsolicited Input

Disable Unsolicited Input ルーチンは、要求されていない入力のトラッピングを禁止します。

---

### 形式

SMG\$DISABLE\_UNSOLICITED\_INPUT *pasteboard-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

要求されていない入力禁止されるキーボード (指定されたペーストボードに対応づけられたキーボード) を指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

---

### 説明

SMG\$DISABLE\_UNSOLICITED\_INPUT ルーチンは、指定されたペーストボードに対して、要求されていない入力の AST を禁止します。SMG\$DISABLE\_UNSOLICITED\_INPUT ルーチンは、SMG\$ENABLE\_UNSOLICITED\_INPUT ルーチンによって設定されたメールボックスの割り当てを解除し、ターミナル属性を再設定します。したがって、ユーザは LIB\$SPAWN ルー

## SMG\$DISABLE\_UNSOLICITED\_INPUT

チンを呼び出すことができるようになります。SMG\$ENABLE\_UNSOLICITED\_INPUT ルーチンを使用して許可した、要求されていない入力のトラッピングを禁止する場合には、このルーチンを使用しなければなりません。

要求されていない入力のトラッピングとブロードキャスト・メッセージのトラッピングの両方を許可した場合には、メールボックスの割り当てを解除するために、SMG\$DISABLE\_UNSOLICITED\_INPUT ルーチンと SMG\$DISABLE\_BROADCAST\_TRAPPING ルーチンの両方を起動しなければなりません。

---

### 戻される状態値

SSS_NORMAL	正常終了。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	\$QIOW から戻される状態値。

---

# SMG\$DRAW\_CHAR

## Draw a Character in a Virtual Display

Draw a Character in a Virtual Display ルーチンは、仮想ディスプレイ内の指定された位置に文字を描きます。

---

### 形式

SMG\$DRAW\_CHAR *display-id ,flags [,row] [,column] [,rendition-set] [,rendition-complement]*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

仮想ディスプレイの識別子。display-id 引数は、この識別子を格納した符号なしロングワードのアドレスです。

flags

OpenVMS 用法	mask_longword
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

描く文字を示すビット・マスク (省略可能)。Flags 引数は、フラグを格納した符号なしロングワードのアドレスです。Flags 引数には以下の値を指定できます。

- SMG\$M\_UP
- SMG\$M\_DOWN
- SMG\$M\_LEFT

- SMG\$M\_RIGHT

論理和演算を実行することにより、T文字、角を形成する文字、交差を形成する文字などを描くことができます。値が0の場合には、菱形文字( )が描かれます。

row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

指定された文字を描く位置を指定する行番号 (省略可能)。row 引数は、行番号を格納した符号付きロングワードのアドレスです。row 引数を省略した場合には、文字は現在の仮想カーソルの行位置に描かれます。

column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

指定された文字を描く位置を指定するカラム番号 (省略可能)。column 引数は、カラム番号を格納した符号付きロングワードのアドレスです。column 引数を省略した場合には、文字は現在の仮想カーソルのカラム位置に描かれます。column 引数が全角文字の右半分の位置になる場合、その文字の左半分は未定義文字になります。また、column 引数が全角文字の左半分の位置になる場合、その文字の右半分は未定義文字になります。

rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、仮想ディスプレイ内の対応する属性が設定されます。rendition-set 引数を使用して以下の属性を指定できます。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想ディスプレイの現在の省略時の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## rendition-complement

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、ディスプレイ内の対応する属性は補足されます。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数によって補足できます。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの2つの引数を組み合わせて使用すれば、1回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性をオフに設定する。

---

 説明

SMG\$DRAW\_CHAR ルーチンは、指定された仮想ディスプレイの指定された位置に指定された文字を描きます。このルーチンは仮想カーソルの位置を変更しません。描かれる文字は漢字ターミナルのタイプに応じて異なります。たとえば、SMG\$は、可能な場合にはターミナルのライン描画文字集合を使用します。その文字集合を使用できない場合には、SMG\$はラインを描くために +, -, および | を使用します。column 引数が全角文字の右半分の場合、その文字の左半分は未定義文字になります。また、column 引数が全角文字の左半分の場合、その文字の右半分は未定義文字になります。

---

 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVCOL	カラム番号に誤りがあります。
SMG\$_INVROW	行番号に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## 例

```

1. C+
   C This FORTRAN example demonstrates the use of
   C SMG$DRAW_CHAR to use the terminal line drawing
   C characters.
   C-
   IMPLICIT INTEGER (A-Z)
   INCLUDE '($SMGDEF)'

   s = SMG$CREATE_PASTEBOARD(p_id)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$CREATE_VIRTUAL_DISPLAY(17,7,d_id,SMG$_BORDER)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$PASTE_VIRTUAL_DISPLAY(d_id,p_id,4,30)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$SET_CURSOR_REL(d_id,1,3)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))

   s = SMG$DRAW_CHAR(d_id,SMG$_UP,1,4,SMG$_BOLD)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_DOWN,2,4,0,SMG$_REVERSE)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_LEFT,3,4,SMG$_BLINK)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_RIGHT,4,4,0,0)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_UP + SMG$_DOWN,5)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_UP + SMG$_LEFT,6)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_UP + SMG$_RIGHT,7)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_DOWN + SMG$_LEFT,8)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_DOWN + SMG$_RIGHT,9)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_LEFT + SMG$_RIGHT,10)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_UP + SMG$_DOWN + SMG$_LEFT,11)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
   s = SMG$DRAW_CHAR(d_id,SMG$_UP + SMG$_DOWN + SMG$_RIGHT,12)
   IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))

```

```
s = SMG$DRAW_CHAR(d_id,SMG$M_DOWN + SMG$M_LEFT + SMG$M_RIGHT,13)
IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
s = SMG$DRAW_CHAR(d_id,SMG$M_UP + SMG$M_LEFT + SMG$M_RIGHT,14)
IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
s = SMG$DRAW_CHAR(d_id,SMG$M_UP + SMG$M_DOWN + SMG$M_RIGHT +
1          SMG$M_LEFT, 15)
IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))
s = SMG$DRAW_CHAR(d_id,0,16)
IF (.NOT. s) CALL LIB$SIGNAL(%VAL(s))

END
```

この例では、1つのカラム位置にライン描画文字を生成しています。

---

## SMG\$DRAW\_LINE

### Draw a Line

Draw a Line ルーチンは、水平線または垂直線を描きます。

---

#### 形式

SMG\$DRAW\_LINE *display-id ,start-row ,start-column ,end-row ,end-column [,rendition-set]  
[,rendition-complement]*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ラインを描く仮想ディスプレイを指定します。display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

ライン描画を開始する行を指定します。start-row 引数は、ライン描画を開始する行番号を格納した符号付きロングワードのアドレスです。

start-column

OpenVMS 用法	longword_signed
データ型	longword (signed)

アクセス read only  
受け渡し方 by reference

ライン描画を開始するカラムを指定します。start-column 引数は、ライン描画を開始するカラム番号を格納した符号付きロングワードのアドレスです。

水平線の描画時に、start-column 引数が全角文字の右半分から始まった場合、左半分は、未定義文字になります。垂直線描画時に、start-column 引数が全角文字の右半分あるいは左半分のバイト上にあった場合には、残りの部分は未定義文字になります。

end-row

OpenVMS 用法 longword\_signed  
データ型 longword (signed)  
アクセス read only  
受け渡し方 by reference

ライン描画を終了する行を指定します。end-row 引数は、ライン描画を終了する行番号を格納した符号付きロングワードのアドレスです。

end-column

OpenVMS 用法 longword\_signed  
データ型 longword (signed)  
アクセス read only  
受け渡し方 by reference

ライン描画を終了するカラムを指定します。end-column 引数は、ライン描画を終了するカラム番号を格納した符号付きロングワードのアドレスです。

水平線の描画時に、end-column 引数が全角文字の左半分で終わった場合、右半分は、未定義文字になります。垂直線描画時に、end-column 引数が全角文字の右半分あるいは左半分にあった場合には、残りの部分は未定義文字になります。

rendition-set

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、ディスプレイ内の対応する属性が設定されます。rendition-set 引数を使用して以下の属性を指定できます。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想ディスプレイの現在の省略時の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。

SMG\$M\_INVISIBLE 表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。

SMG\$M\_USER1 から ユーザ定義属性を表示します。  
SMG\$M\_USER8

#### rendition-complement

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、仮想ディスプレイ内の対応する属性が補足されます。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数を使用して補足できます。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの2つの引数を組み合わせて使用すれば、1回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性をオフに設定する。

## 説明

SMG\$DRAW\_LINE ルーチンは、指定された開始行/カラムから指定された終了行/カラムまでラインを描きます。このルーチンは仮想カーソルの位置を変更しません。このルーチンでは、水平線または垂直線だけを描くことができます。ラインを描くために使用される文字はターミナルのタイプに応じて異なります。可能な場合には、日本語 SMG ルーチンは漢字ターミナルのライン描画文字集合を使用します。その文字集合を使用できない場合には、日本語 SMG ルーチンはラインを描くために +, -, および | を使用します。

DEC 漢字文字集合が使用されていた場合、描画される線が全角文字の左半分あるいは右半分にあった場合には残りの部分は未定義文字になります。

---

戻される状態値

SS\$ _NORMAL	正常終了。
SMG\$ _DIALINNOT	斜めのラインを描くことはできません。
SMG\$ _INVCOL	カラム番号が誤っています。指定されたカラムは仮想ディスプレイの外部です。
SMG\$ _INVDIS_ID	display-id 引数に誤りがあります。
SMG\$ _INVROW	行番号が誤っています。指定された行は仮想ディスプレイの外部です。
SMG\$ _WRONUMARG	引数の数が誤っています。

---

## 例

```

1. C+
   C This FORTRAN example program demonstrates the use of SMG$DRAW_LINE.
   C-
           INTEGER SMG$CREATE_VIRTUAL_DISPLAY, SMG$CREATE_PASTEBOARD
           INTEGER SMG$PASTE_VIRTUAL_DISPLAY, SMG$DRAW_LINE
           INTEGER DISPLAY1, PASTE1, ROWS, COLUMNS, BORDER, STATUS
   C+
   C First, create the virtual display using SMG$CREATE_VIRTUAL_DISPLAY.
   C To give it a border, set BORDER = 1. No border would be BORDER = 0.
   C-
           ROWS = 7
           COLUMNS = 50
           BORDER = 1
           STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1           (ROWS, COLUMNS, DISPLAY1, BORDER)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
   C+
   C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
   C-
           STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
   C+
   C Draw a vertical line using SMG$DRAW_LINE.
   C Start at row 2, column 20. End at row 6.
   C-
           STATUS = SMG$DRAW_LINE (DISPLAY1, 2, 20, 6, 20)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

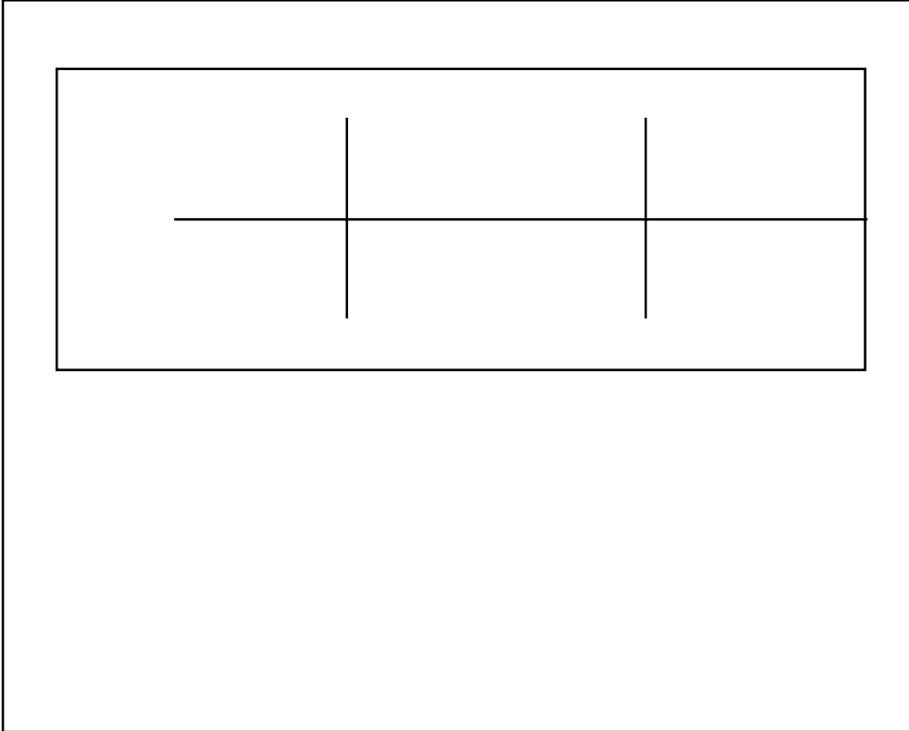
```

## SMG\$DRAW\_LINE

```
C+
C Now, use SMG$DRAW_LINE to draw a vertical line.
C Start at row 6, column 40. End at row 2.
C This is similar to the line drawn above, but we are drawing the
C line in the reverse direction.
C-
      STATUS = SMG$DRAW_LINE (DISPLAY1, 6, 40, 2, 40)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
C+
C Draw a horizontal line now, again calling SMG$DRAW_LINE.
C Start at row 4, column 8. End at column 50.
C-
      STATUS = SMG$DRAW_LINE (DISPLAY1, 4, 8, 4, 50)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
C+
C Paste the virtual display using SMG$PASTE_VIRTUAL_DISPLAY.
C-
      STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      END
```

この FORTRAN プログラムの例で生成される出力は図 8-14 に示すとおりです。

図 8-14 SMG\$DRAW\_LINE ルーチン呼び出す FORTRAN プログラムによって生成される出力



ZK-4110-GE

---

# SMG\$DRAW\_RECTANGLE

## Draw a Rectangle

Draw a Rectangle ルーチンは長方形を描きます。

---

### 形式

SMG\$DRAW\_RECTANGLE *display-id ,start-row ,start-column ,end-row ,end-column [,rendition-set] [,rendition-complement]*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

長方形を描く仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

長方形の左上の角の行番号を指定します。start-row 引数は、長方形の左上の角の行番号を格納した符号付きロングワードのアドレスです。

start-column

OpenVMS 用法	longword_signed
データ型	longword (signed)

アクセス read only  
受け渡し方 by reference

長方形の左上の角の列番号を指定します。start-column 引数は、長方形の左上の角の列番号を格納した符号付きロングワードのアドレスです。start-column 引数が全角文字上にあった場合、その残りの部分は、未定義文字になります。

end-row

OpenVMS 用法 longword\_signed  
データ型 longword (signed)  
アクセス read only  
受け渡し方 by reference

長方形の右下の角の行番号を指定します。end-row 引数は、長方形の右下の角の行番号を格納した符号付きロングワードのアドレスです。

end-column

OpenVMS 用法 longword\_signed  
データ型 longword (signed)  
アクセス read only  
受け渡し方 by reference

長方形の右下の角の列番号を指定します。end-column 引数は、長方形の右下の角の列番号を格納した符号付きロングワードのアドレスです。end-column 引数が全角文字上にあった場合、その残りの部分は、未定義文字になります。

rendition-set

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、仮想ディスプレイ内の対応する属性が設定されます。rendition-set 引数を使用して以下の属性を指定できます。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想ディスプレイの現在の省略時の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

rendition-complement

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only

受け渡し方      by reference

属性補足指定子。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、仮想ディスプレイ内の対応する属性が補足されます。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数を使用して補足できます。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの 2 つの引数を組み合わせて使用すれば、1 回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性をオフに設定する。

## 説明

SMG\$DRAW\_RECTANGLE ルーチンは、指定された左上の角と右下の角の位置をもとに、仮想ディスプレイ内に長方形を描きます。このルーチンは仮想カーソルの位置を変更しません。長方形を構成するラインを描くために使用される文字は漢字ターミナルのタイプに応じて異なります。可能な場合には、日本語 SMG ルーチンは漢字ターミナルのライン描画文字集合を使用します。その文字集合を使用できない場合には、日本語 SMG ルーチンはラインを描くために、+、-、および | を使用します。

DEC 漢字または Super DEC 漢字文字集合を使用した場合に、長方形が全角文字上に描かれた場合には、全角文字の残された部分は未定義文字になります。

## 戻される状態値

SS\$ _NORMAL	正常終了。
SMG\$ _INVCOL	カラム番号が誤っています。指定されたカラムは仮想ディスプレイの外部です。

SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_INVROW	行番号が誤っています。指定された行は仮想ディスプレイの外部です。
SMG\$_WRONUMARG	引数の数が誤っています。

---

 例

```

1.
C+
C This FORTRAN example program demonstrates the use of
C SMG$DRAW_RECTANGLE.
C
C This routine creates a virtual display and uses SMG$DRAW_RECTANGLE
C to draw a rectangle inside the bordered virtual display.
C-

C+
C Include the SMG definitions. In particular, we want SMG$_M_BORDER.
C-

        INCLUDE '($SMGDEF)'
        INTEGER SMG$CREATE_VIRTUAL_DISPLAY, SMG$CREATE_PASTEBOARD
        INTEGER SMG$PASTE_VIRTUAL_DISPLAY, SMG$DRAW_RECTANGLE
        INTEGER DISPLAY1, PASTE1, ROWS, COLUMNS, STATUS

C+
C Create a virtual display with a border by calling
C SMG$CREATE_VIRTUAL_DISPLAY.
C-

        ROWS = 7
        COLUMNS = 50

        STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$_M_BORDER)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$CREATE_PASTEBOARD to create the pasteboard.
C-

        STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Using SMG$DRAW_RECTANGLE, draw a rectangle inside the bordered region.
C-

        STATUS = SMG$DRAW_RECTANGLE (DISPLAY1, 2, 10, 6, 20)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Paste the virtual display by calling SMG$PASTE_VIRTUAL_DISPLAY.
C-

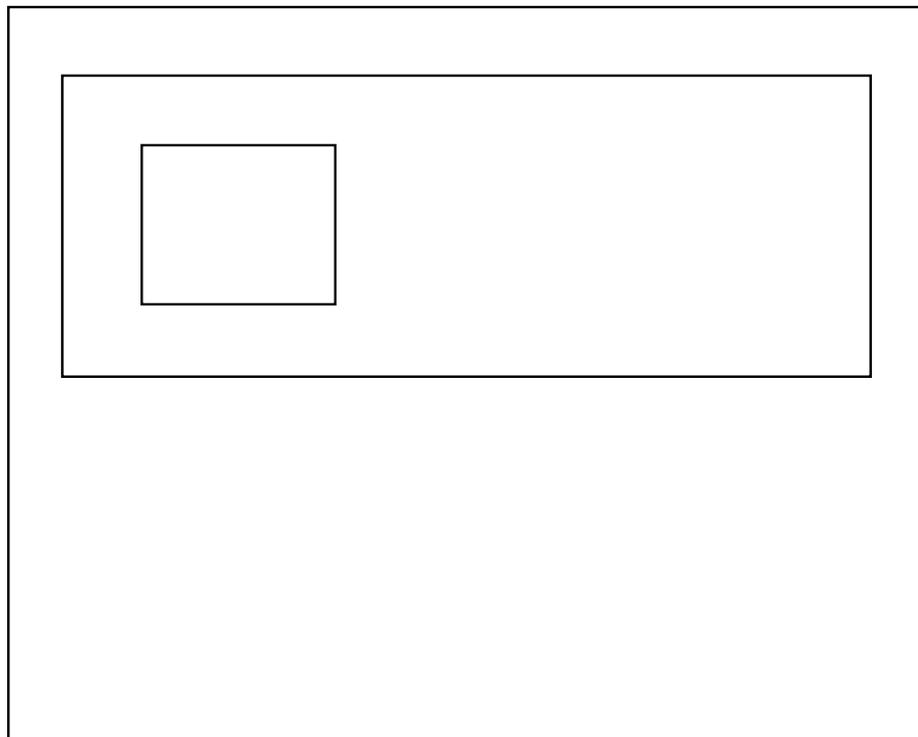
```

## SMG\$DRAW\_RECTANGLE

```
STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
END
```

この FORTRAN プログラムの例で生成される出力は図 8-15 に示すとおりです。

図 8-15 SMG\$DRAW\_RECTANGLE ルーチン呼び出す FORTRAN プログラムによって生成される出力



ZK-4111-GE

---

## SMG\$ENABLE\_UN SOLICITED\_INPUT

### Enable Unsolicited Input

Enable Unsolicited Input ルーチンは、要求されていない入力を検出し、AST ルーチン呼び出します。

---

#### 形式

SMG\$ENABLE\_UN SOLICITED\_INPUT *pasteboard-id* ,AST-routine [,AST-argument]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

要求されていない入力を許可する対象となるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

AST-routine

OpenVMS 用法	ast_procedure
データ型	procedure entry mask
アクセス	read only
受け渡し方	by value

漢字ターミナルから要求されていない入力を受信したときに呼び出される AST ルーチンのアドレス。AST-routine 引数には、ルーチンのプロシージャ・エントリ・マスクが格納されます。SMG\$ENABLE\_UN SOLICITED\_INPUT ルーチンは、要求されていない入力の存在を検出し、6 つの引数を使用して AST ルーチン呼び出しま

す。6つの引数とは、pasteboard-id 引数、AST-argument 引数、R0、R1、PC、および PSL です。AST ルーチン引数は図 8-16 に示すとおりです。

図 8-16 AST ルーチン引数

Pasteboard ID
AST Argument
R0
R1
PC
PSL

ZK-4802-GE

#### AST-argument

OpenVMS 用法 user\_arg  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by value

AST ルーチンに渡される値。AST-argument 引数には、AST ルーチンに渡される値が格納されています。

---

## 説明

SMG\$ENABLE\_UNSOLICITED\_INPUT ルーチンは要求されていない入力是否存在を検出し、AST ルーチンを呼び出します。

このルーチンは入力文字を読み込むわけではなく、単に AST ルーチンを呼び出すことにより、SMG\$READ\_COMPOSED\_LINE、SMG\$READ\_KEYSTROKE、SMG\$READ\_STRING、SMG\$READ\_VERIFY ルーチンを使用して読み込み操作を実行しなければならないことを、アプリケーションに“通知”するだけです。要求されていない入力を読み込むのは、ユーザの役割です。

SMG\$ENABLE\_UNSOLICITED\_INPUT ルーチンは、ユーザ・イメージにとって注意を必要とするターミナル関連イベントが発生したときに、メッセージを受信するメールボックスを設定します。このメールボックスは、ターミナル・データではなく、状態メッセージをドライバからユーザ・プログラムに渡します。この状態メッセージは、先読みバッファに要求されていないデータが存在するときにメールボックスに送信されます。この場合、ユーザ・プロセスは、要求されていないデータ・メッセージ

を受信した後、漢字ターミナルとの対話を開始します。この対話が終了した後、日本語 SMG は最後の入出力交換時に、要求されないデータ・メッセージ機能を再度許可します。読み込み操作と次の読み込み操作の間には、1 つのメッセージだけが送信されます。SYSGEN パラメータである DEFMBXBUFQUO と DEFMBXMXMSG が、メールボックス作成時に使用されます。

漢字ターミナルとメールボックスの相互作用についての詳しい説明は、『OpenVMS I/O User's Reference Manual』を参照してください。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	\$QIOW から戻される状態値。

---

## SMG\$END\_DISPLAY\_UPDATE

### End Display Update

End Display Update ルーチンは、仮想ディスプレイに対するバッチ更新処理を終了します。

---

#### 形式

SMG\$END\_DISPLAY\_UPDATE *display-id*

---

#### 戻り値

VMS 用法: cond\_value  
データ型: longword (unsigned)  
アクセス: write only  
受け渡し方: by value

---

#### 引数

display-id  
OpenVMS 用法 identifier  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

影響を受ける仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

display-id 引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンから戻されます。

---

#### 説明

SMG\$END\_DISPLAY\_UPDATE は SMG\$BEGIN\_DISPLAY\_UPDATE と組み合わせて使用し、特定の仮想ディスプレイに対する出力のバッチ処理を制御します。SMG\$BEGIN\_DISPLAY\_UPDATE を呼び出すたびに、“バッチ・カウント”が増分され、それに対し、SMG\$END\_DISPLAY\_UPDATE を呼び出すと、そのたびにこのカウントは減分されます。バッチ・カウントが 0 になると、仮想ディスプレイはバッチ処理中に実行されたすべての操作によって更新され、仮想ディスプレイがペーストされている場合には、ペーストボードに書き込まれます。

バッチ・カウントが0のときに SMG\$END\_DISPLAY\_UPDATE を呼び出す操作は正しい操作です。したがって、正常終了状態が戻されます。

---

### 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_BATSTIPRO	正常終了。バッチ処理はまだ実行中です。
SMG\$_BATWASOFF	正常終了。バッチ処理はすでに終了しています。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$END\_PASTEBOARD\_UPDATE

### End Pasteboard Update

End Pasteboard Update ルーチンは、ペーストボードに対するバッチ更新処理を終了します。

---

#### 形式

SMG\$END\_PASTEBOARD\_UPDATE *pasteboard-id*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

*pasteboard-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

バッチ・カウントを減分させるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。バッチ・カウントが 0 になると、バッファに格納されていた指定されたペーストボードに対するすべての出力がペーストボードに書き込まれます。

---

#### 説明

SMG\$END\_PASTEBOARD\_UPDATE ルーチンは SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチンと組み合わせて使用し、特定のペーストボードに対する出力バッチ処理を制御します。SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチンを呼び出すと、そのたびに“バッチ・カウント”が増分されます。それに対し、SMG\$END\_

PASTEBOARD\_UPDATE ルーチン呼び出すと、そのたびにこのカウントは減分されます。バッチ・カウントが0になると、ペーストボードは画面に出力されます。

バッチ・カウントが0のときに SMG\$END\_PASTEBOARD\_UPDATE ルーチン呼び出すのは有効な操作です。このときは、正常終了の状態値が戻されます。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_BATSTIPRO	正常終了。バッチ処理はまだ実行中です。
SMG\$_BATWASOFF	正常終了。バッチ処理はすでに終了しています。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$ERASE\_CHARS

### Erase Characters

Erase Characters ルーチンは、仮想ディスプレイ内のカラムを消去してそれらを空白と置き換えます。

---

#### 形式

SMG\$ERASE\_CHARS *display-id ,number-of-columns ,start-row ,start-column*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

カラムが消去される仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

number-of-columns

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

消去するカラム数。number-of-columns 引数は、消去するカラム数を格納した符号付きロングワードのアドレスです。

start-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

カラム消去開始行。start-row 引数は、消去操作を開始する行の位置を格納した符号付きロングワードのアドレスです。

start-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

カラム消去操作の開始カラム。start-column 引数は、消去操作を開始するカラム位置を格納した符号付きロングワードのアドレスです。

指定カラムが全角文字の右半分である場合は、その文字の右半分は空白になり、左半分は未定義文字になります。

---

## 説明

SMG\$ERASE\_CHARS ルーチンは、仮想ディスプレイのカラムを空白に置換することにより、それらのカラムを消去します。消去操作では指定された行だけが変更されます。number-of-columns 引数の値がその行のカラム数より大きい場合は、指定された開始位置からその行の最後までのカラムが消去されます。このルーチンを終了した後、仮想カーソルは消去された最初のカラム位置に移動します。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、全角文字の右半分から消去操作が始まる場合は、その文字の右半分は空白になり、左半分は未定義文字になります。また、全角文字の左半分で消去操作が終了する場合は、その文字の右半分は未定義文字になります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVCOL	カラム位置が誤っています。指定されたカラムは仮想ディスプレイの内部に含まれていません。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行の位置が誤っています。
SMG\$_WRONUMARG	引数の数が誤っています。

## 例

```

1. C+
   C This FORTRAN example demonstrates the use of SMG$ERASE_CHARS.
   C
   C Include the SMG definitions. In particular, we want SMG$M_BORDER.
   C-
           IMPLICIT INTEGER (A-Z)
           INCLUDE '($SMGDEF)'

   C+
   C Create a virtual display with a border by calling
   C SMG$CREATE_VIRTUAL_DISPLAY.
   C-
           ROWS = 7
           COLUMNS = 50
           STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1          (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
   C-
           STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Using SMG$PUT_CHARS, put data in the virtual display.
   C-
           STATUS = SMG$PUT_CHARS ( DISPLAY1,
1          ' This virtual display has 7 rows and 50 columns.', 2, 1)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
           STATUS = SMG$PUT_CHARS ( DISPLAY1,
1          ' This is a bordered virtual display.', 4, 1)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
           STATUS = SMG$PUT_CHARS ( DISPLAY1,
1          ' SMG$PUT_CHARS puts data in this virtual display.', 6, 1)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Call SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual display.
   C-
           STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

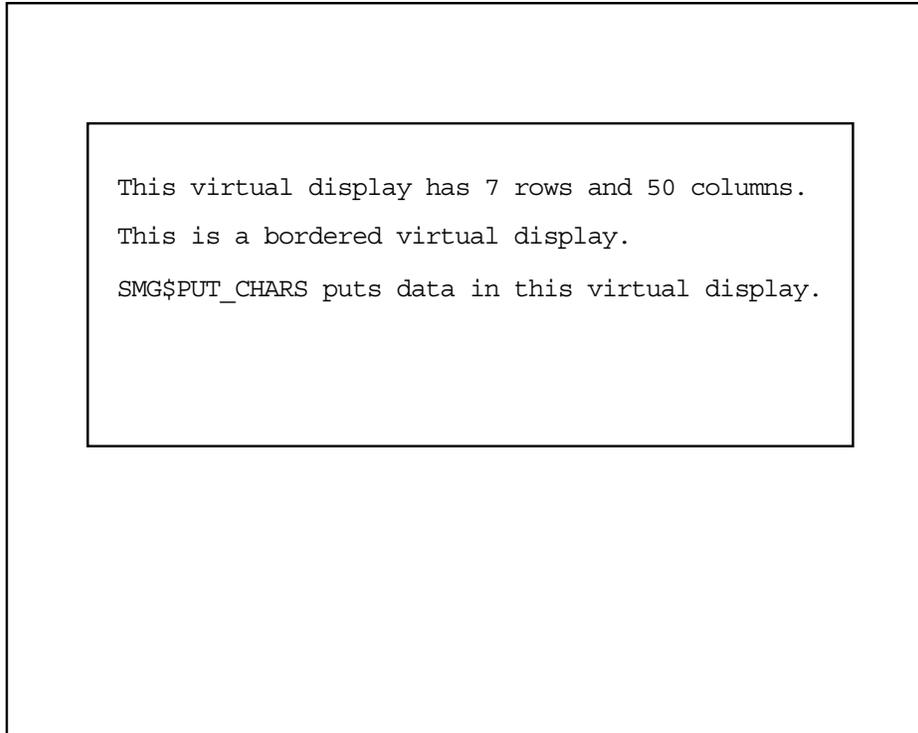
   C+
   C Erase 4 characters on row 4 starting from character (column) 14 by
   C calling SMG$ERASE_CHARS. This will remove the characters "rder"
   C from the word "bordered".
   C-

```

```
STATUS = SMG$ERASE_CHARS ( DISPLAY1, 4, 4, 14)  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
END
```

このFORTRANプログラムによって生成される最初の出力は図 8-17 に示すとおりです。

図 8-17 SMG\$ERASE\_CHARS ルーチン呼び出す前に生成される出力

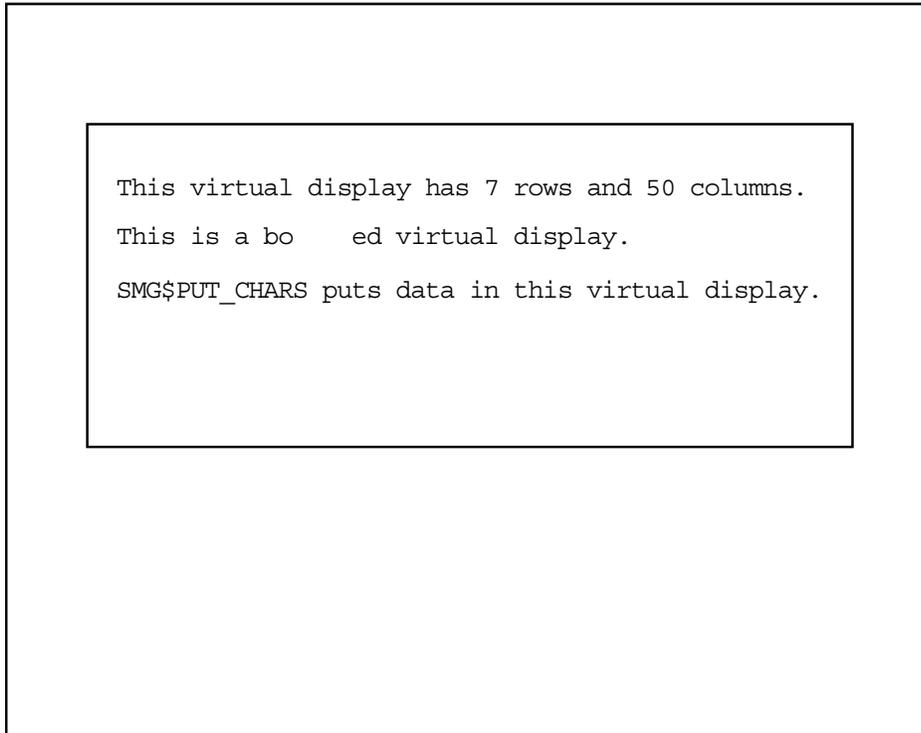


ZK-4105-GE

SMG\$ERASE\_CHARS ルーチン呼び出した後に生成される出力は図 8-18 に示すとおりです。

## SMG\$ERASE\_CHARS

図 8-18 SMG\$ERASE\_CHARS ルーチン呼び出し後に生成される出力



ZK-4113-GE

---

# SMG\$ERASE\_COLUMN

## Erase Column from Display

Erase Column From Display ルーチンは、仮想ディスプレイの指定された部分を、指定された位置からカラムの最後まで消去します。

---

### 形式

SMG\$ERASE\_COLUMN *display-id* [*start-row*] [*column-number*] [*end-row*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

影響を受ける仮想ディスプレイの識別子。display-id 引数は、この仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

消去操作を開始する行番号(省略可能)。start-row 引数は、行番号を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、column-number 引数は無視され、消去操作はその仮想ディスプレイの仮想カーソルの現在の位置から開始されます。

column-number

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

消去操作が開始されるカラム番号 (省略可能)。column-number 引数は、カラム番号を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、start-row 引数は無視され、消去操作はその仮想ディスプレイの仮想カーソルの現在の位置から開始されます。column-number 引数が、全角文字の右半分である場合には、その文字の左半分は未定義文字になります。また、column-number 引数が、全角文字の左半分である場合には、その文字の右半分は未定義文字になります。

end-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

消去操作を終了する行番号 (省略可能)。end-row 引数は、行番号を格納した符号付きロングワードのアドレスです。

---

## 説明

SMG\$ERASE\_COLUMN ルーチンを使用すれば、仮想ディスプレイのカラムを、指定した位置からそのカラムの最後まで消去できます。位置を指定しなかった場合には、消去操作は指定された仮想ディスプレイの仮想カーソルの現在の位置から開始されます。消去操作が正常終了した後、このルーチンは消去した最初の文字位置に仮想カーソルを移動します。

DEC 漢字または Super DEC 漢字文字集合が使用され、削除操作が全角文字の右半分で行われる場合、文字の左半分は未定義文字になります。また、削除操作が全角文字の左半分で行われる場合、文字の右半分は未定義文字になります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVCOL	カラム番号が誤っています。指定されたカラムは仮想ディスプレイの外部です。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_INVROW	行番号が誤っています。指定された行は仮想ディスプレイの外部です。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$ERASE\_DISPLAY

## Erase Virtual Display

Erase Virtual Display ルーチンは、仮想ディスプレイ全体または一部を消去しテキスト文字を空白で置き換えます。

---

### 形式

SMG\$ERASE\_DISPLAY *display-id* [*,start-row*] [*,start-column*] [*,end-row*] [*,end-column*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

消去される仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

消去操作を始める行。start-row 引数は、消去操作が開始される行の位置を格納した符号付きロングワードのアドレスです。

start-row 引数を省略した場合は、start-column 引数も無視されて仮想ディスプレイ全体が消去されます。start-row 引数と start-column 引数を指定しなかった場合は、end-row 引数と end-column 引数は無視されて、仮想ディスプレイ全体が消去されます。

start-column

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

消去操作を始めるカラム。start-column 引数は、消去操作が開始されるカラム位置を格納した符号付きロングワードのアドレスです。

start-column 引数を省略した場合は、start-row 引数も無視されて仮想ディスプレイ全体が消去されます。start-row 引数と start-column 引数を省略した場合は、end-row 引数と end-column 引数は無視されて仮想ディスプレイ全体が消去されます。

全角文字の右半分から消去操作が始まる場合は、その文字の左半分は未定義文字になります。

end-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

消去操作を終了する行。end-row 引数は、最後に消去する行の位置を格納した符号付きロングワードのアドレスです。最後に消去する行を指定します。

end-row 引数を省略した場合は、end-column 引数も無視されて仮想ディスプレイ内の残りのすべての行が消去されます。

end-column

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

消去操作を終了するカラム。end-column 引数は、最後に消去するカラム位置を格納した符号付きロングワードのアドレスです。最後に消去するカラムを指定します。

end-column 引数を省略した場合は、end-row 引数は無視されて仮想ディスプレイ内の残りのすべてのカラムが消去されます。

全角文字の左半分で消去操作が終了する場合は、その文字の右半分は未定義文字になります。

---

 説明

SMG\$ERASE\_DISPLAY ルーチンを実行すると、仮想ディスプレイ全体またはその一部が消去され、テキスト文字は空白に置換されます。開始位置を省略した場合は 1 行目、1 カラム目に設定されます。終了位置を省略した場合は、仮想ディスプレイの最後の行または最後のカラムになります。したがって、display-id 引数だけを指定すれば、仮想ディスプレイ全体を消去できます。この操作を実行した後、カーソルは、消去した部分の最初の位置に設定されます。仮想ディスプレイ全体を消去した場合は、仮想カーソルは 1 行目、1 カラム目に設定されます。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、全角文字の右半分から消去操作が始まる場合には、その文字の左半分は未定義文字になります。また、全角文字の左半分で消去操作が終了する場合は、その文字の右半分は未定義文字になります。

---

 戻される状態値

SSS_NORMAL	正常終了。
SMG\$_INVCOL	カラム位置が誤っています。指定されたカラムは仮想ディスプレイの内部に含まれていません。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行の位置が誤っています。指定された行は仮想ディスプレイの内部に含まれていません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

 例

```

1.
  C+
  C This FORTRAN example program demonstrates the use of SMG$ERASE_DISPLAY.
  C-

      IMPLICIT INTEGER (A-Z)

  C+
  C Call SMG$CREATE_VIRTUAL_DISPLAY to create the virtual
  C display. To give it a border, set BORDER = 1.
  C No border would be BORDER = 0.
  C-

      ROWS = 7
      COLUMNS = 50
      BORDER = 1
  
```

## SMG\$ERASE\_DISPLAY

```

        STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1          (ROWS, COLUMNS, DISPLAY1, BORDER)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Using SMG$CREATE_PASTEBOARD, create the pasteboard.
C-

        STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$PUT_CHARS to put data in the virtual display.
C-

        STATUS = SMG$PUT_CHARS ( DISPLAY1,
1          ' This virtual display has 7 rows and 50 columns.', 2, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        STATUS = SMG$PUT_CHARS ( DISPLAY1,
1          ' This is a bordered virtual display.', 4, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        STATUS = SMG$PUT_CHARS ( DISPLAY1,
1          ' SMG$PUT_CHARS puts data in this virtual display.', 6, 1)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Paste the virtual display by calling SMG$PASTE_VIRTUAL_DISPLAY.
C-

        STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

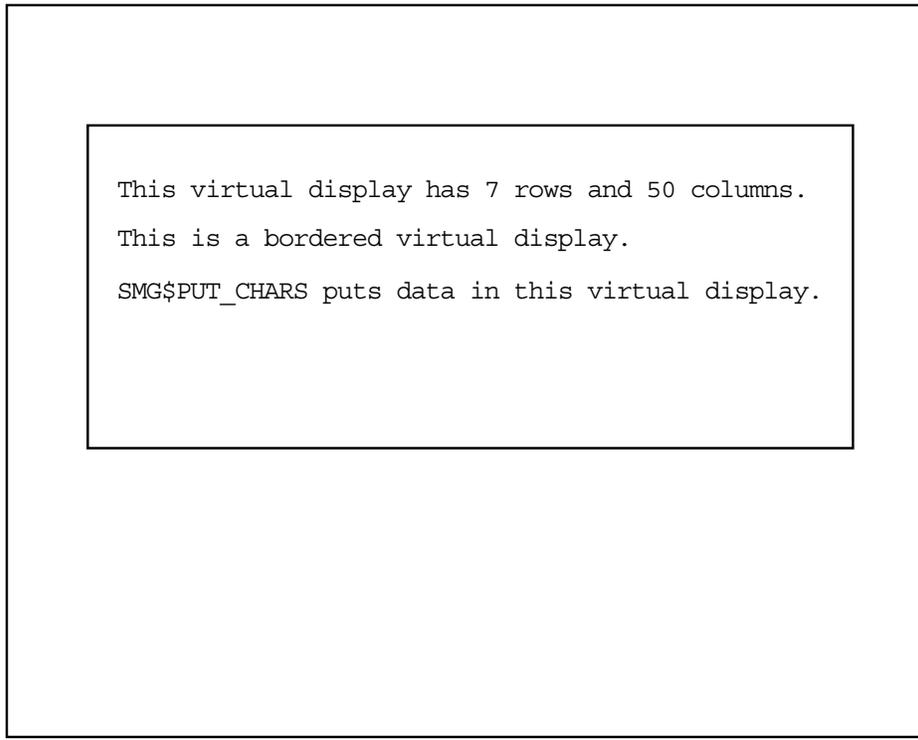
C+
C Call SMG$ERASE_DISPLAY to erase the display from row 2,
C column 6, through row 4, column 28.
C-

        STATUS = SMG$ERASE_DISPLAY ( DISPLAY1, 2, 6, 4, 28)
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        END
```

この FORTRAN プログラムによって生成される最初の出力は、図 8-19 に示すとおりです。

図 8-19 SMG\$ERASE\_DISPLAY ルーチンを呼ぶ FORTRAN プログラムによって生成される最初の出力

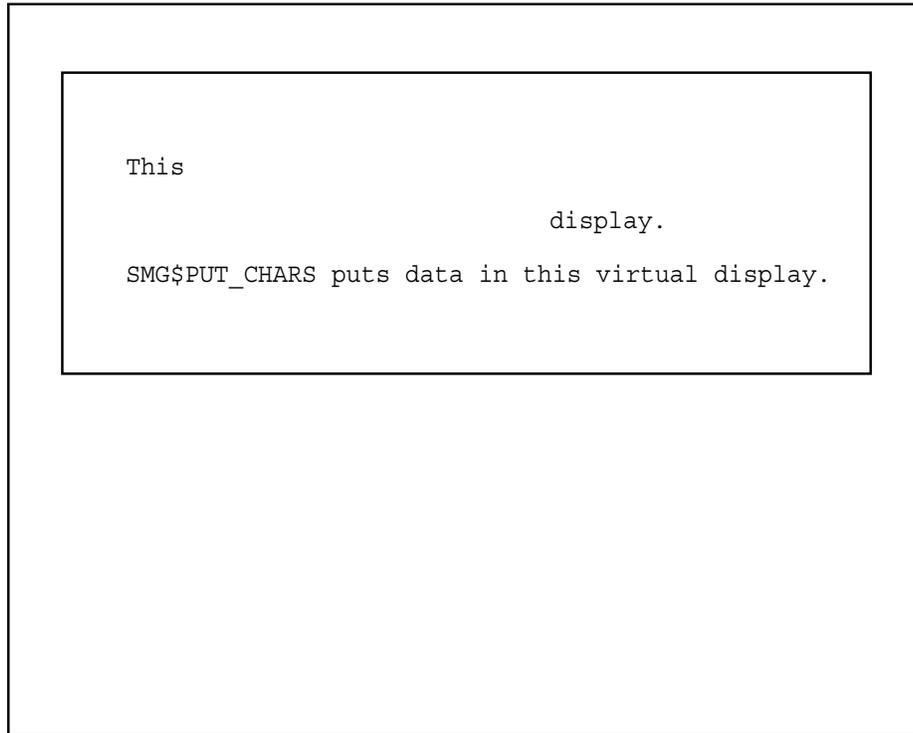


ZK-4105-GE

SMG\$ERASE\_DISPLAY ルーチンを呼び出した後に生成される出力は、図 8-20 に示すとおりです。

## SMG\$ERASE\_DISPLAY

図 8-20 SMG\$ERASE\_DISPLAY ルーチン呼び出し後に生成される出力



ZK-4115-GE

---

# SMG\$ERASE\_LINE

## Erase Line

Erase Line ルーチンは、仮想ディスプレイの 1 行全体またはその一部を消去します。

---

### 形式

SMG\$ERASE\_LINE *display-id* [*,start-row*] [*,start-column*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

行が消去される仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

消去開始行。start-row 引数は、消去操作を開始する行の位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、start-column 引数も無視されて現在のカーソル位置が使用されます。

start-column

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

消去開始カラム。start-column 引数は、消去操作が開始されるカラム位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、start-row 引数も無視されて現在のカーソル位置が使用されます。

---

## 説明

SMG\$ERASE\_LINE ルーチンは、指定された開始位置からその行の最後までを消去します。開始位置を省略した場合は、SMG\$ERASE\_LINE ルーチンは現在の仮想カーソルの位置から行の最後までまでのテキストを消去します。仮想カーソルは、消去されたテキストの最初の空白に設定されます。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、全角文字の右半分から消去操作が始まる場合は、その文字の左半分は未定義文字になります。

---

 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVCOL	カラム位置が誤っています。指定されたカラムは仮想ディスプレイの外部です。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行の位置が誤っています。指定された行は仮想ディスプレイの外部です。
SMG\$_WRONUMARG	引数の数が誤っています。

---

 例

```

1.
C+
C This FORTRAN example program demonstrates the use of
C SMG$ERASE_LINE.
C-

      IMPLICIT INTEGER (A-Z)
      INCLUDE '($SMGDEF)'

C+
C Use SMG$CREATE_VIRTUAL_DISPLAY to create a virtual display
C with a border.
C-

      ROWS = 7
      COLUMNS = 50

      STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
C-

      STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Put data in the virtual display by calling SMG$PUT_CHARS.
C-

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This virtual display has 7 rows and 50 columns.', 2, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This is a bordered virtual display.', 4, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' SMG$PUT_CHARS puts data in this virtual display.', 6, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

```

## SMG\$ERASE\_LINE

```
C+
C Use SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual display.
C-

      STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$ERASE_LINE to erase line 2, and then again to
C erase the last 4 words on line 4.
C-

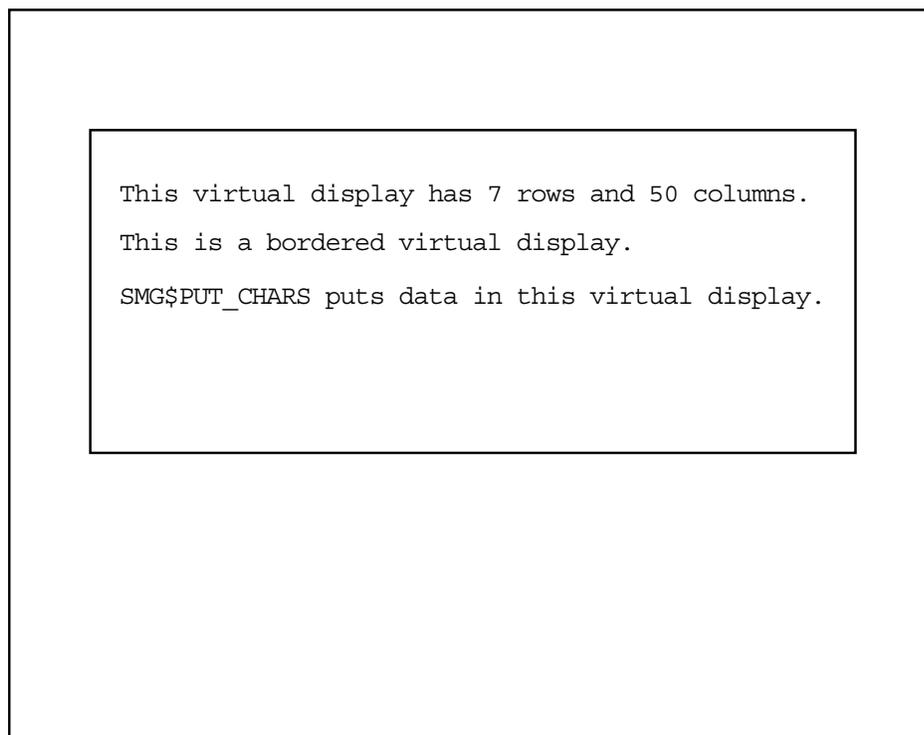
      STATUS = SMG$ERASE_LINE ( DISPLAY1, 2, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$ERASE_LINE ( DISPLAY1, 4, 9)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      END
```

このFORTRANプログラムによって生成される最初の出力は、図 8-21 に示すとおりです。

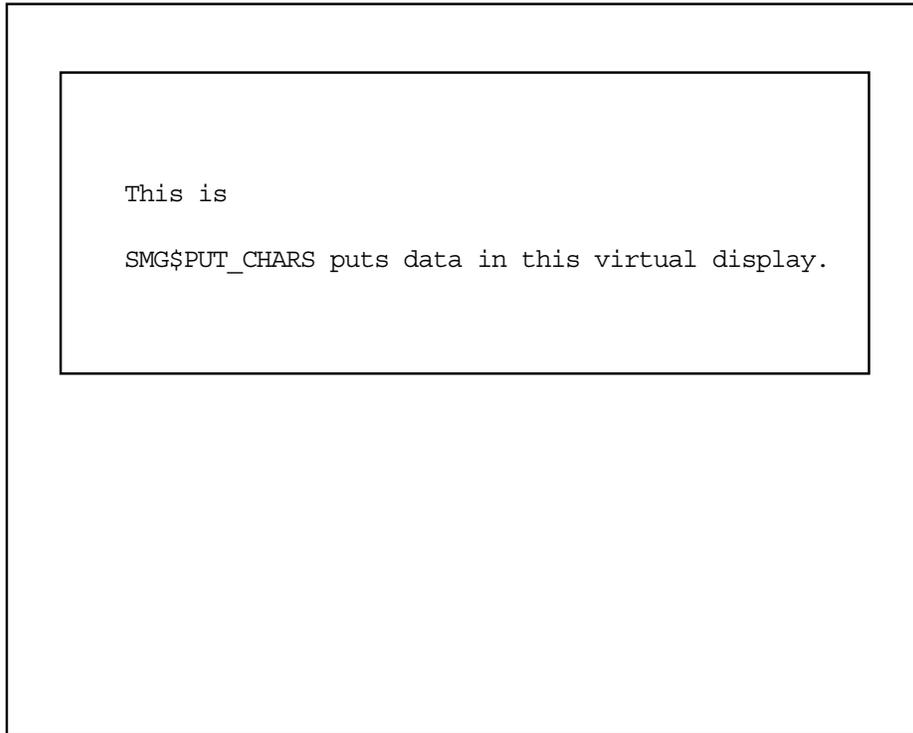
図 8-21 SMG\$ERASE\_LINE ルーチン呼び出す FORTRAN プログラムによって生成される最初の出力



ZK-4108-GE

SMG\$ERASE\_LINE ルーチン呼び出した後で生成される出力は図 8-22 に示すとおりです。

図 8-22 SMG\$ERASE\_LINE ルーチン呼び出し後に生成される出力



ZK-4117-GE

---

# SMG\$ERASE\_PASTEBOARD

## Erase Pasteboard

Erase Pasteboard ルーチンは、ペーストボードの内容を消去します。

---

### 形式

SMG\$ERASE\_PASTEBOARD *pasteboard-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

*pasteboard-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

消去されるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

---

### 説明

SMG\$ERASE\_PASTEBOARD ルーチンは、指定されたペーストボードの内容を消去します。物理カーソルは左上の角 (1 行目, 1 カラム目) に設定されます。ペーストボードにペーストされる仮想ディスプレイがある場合には、次の日本語 SMG による、ペーストボードへの出力時に書き出されます。

---

 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_BATWAS_ON	ペーストボードがバッチ処理されています。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。
SS\$_xxxx	\$QIOW から戻されたエラー。

---

## 例

```

1.
C+
C This FORTRAN example program demonstrates the use of
C SMG$ERASE_PASTEBOARD.
C-

      IMPLICIT INTEGER*4 (A-Z)
      CHARACTER*80   OUT_STR,TRIM_STR
      CHARACTER*18   PROMPT           /'Please enter data '/

      SMG$_BOLD = 1
      SMG$_REVERSE = 2
      SMG$_BLINK = 4
      SMG$_UNDERLINE = 8

C+
C Establish the terminal keyboard as the virtual keyboard
C by calling SMG$CREATE_VIRTUAL_KEYBOARD.
C-

      STATUS = SMG$CREATE_VIRTUAL_KEYBOARD(KEYBOARD_ID,,,)
      IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Establish the terminal screen as a pasteboard using
C SMG$CREATE_PASTEBOARD.
C-

      STATUS = SMG$CREATE_PASTEBOARD (NEW_PID,,,)
      IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Establish a virtual display region by
C calling SMG$CREATE_VIRTUAL_DISPLAY.
C-

      STATUS = SMG$CREATE_VIRTUAL_DISPLAY (5,80,DISPLAY_ID,,,)
      IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Paste the virtual display to the screen, starting at
C row 10, column 15. To paste the virtual display, use
C SMG$PASTE_VIRTUAL_DISPLAY.
C-

```

## SMG\$ERASE\_PASTEBOARD

```
STATUS = SMG$PASTE_VIRTUAL_DISPLAY(DISPLAY_ID,NEW_PID,10,15)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Prompt the user for input, and accept that input using
C SMG$READ_STRING.
C-
STATUS = SMG$READ_STRING(KEYBOARD_ID,OUT_STR,PROMPT,,,,,,)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Clear the screen using SMG$ERASE_PASTEBOARD.
C-
STATUS = SMG$ERASE_PASTEBOARD (NEW_PID)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Trim any trailing blanks from the user input
C by calling STR$TRIM.
C-
STATUS = STR$TRIM(TRIM_STR,OUT_STR,STR_LEN)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
C+
C Display the data input by the user using SMG$PUT_CHARS
C and SMG$PUT_LINE.
C-
STATUS = SMG$PUT_CHARS(DISPLAY_ID,'You entered: ',,,,,,)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
STATUS = SMG$PUT_LINE(DISPLAY_ID,TRIM_STR(1:STR_LEN),,
1 SMG$M REVERSE,0,,)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
END
```

---

## SMG\$EXECUTE\_COMMAND

### Execute Command in a Subprocess

Execute Command in a Subprocess ルーチンは、SMG\$CREATE\_SUBPROCESS ルーチンを使用して生成したサブプロセスの内部で、指定されたコマンドを実行します。

---

#### 形式

SMG\$EXECUTE\_COMMAND *display-id* ,*command-desc* [*flags*] [*ret-status*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

サブプロセスに対応づけられた仮想ディスプレイのディスプレイ識別子。display-id 引数は、この識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

command-desc

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

コマンド文字列。command-desc 引数はコマンド文字列を示すディスクリプタのアドレスです。

## SMG\$EXECUTE\_COMMAND

### flags

OpenVMS 用法	mask_longword
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

オプションとして実行される動作を指定するビット・マスク (省略可能)。flags 引数はフラグを格納した符号なしロングワードのアドレスです。flags 引数に対して指定できる値は次のとおりです。

### SMG\$M\_DATA\_FOLLOWS

この後に入力データが続きます。次の SMG\$EXECUTE\_COMMAND ルーチンの呼び出しには、現在実行中のコマンドに対する入力データが含まれています。これが最後の入力データ項目である場合には、この値を指定しないでください。この値を指定した場合には、ret-status 引数は戻されません。

### SMG\$M\_SEND\_EOF

エンド・オブ・ファイル・マーカを送信します。エンド・オブ・ファイル・マーカはサブプロセスに送信されます。

### ret-status

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

実行したコマンドの状態 (省略可能)。コマンドがバッファに登録されないときに戻されます。ret-status 引数は、この状態を格納する符号なしロングワードのアドレスです。

---

## 説明

SMG\$EXECUTE\_COMMAND ルーチンは、SMG\$CREATE\_SUBPROCESS ルーチンを使用して生成したサブプロセス内で、指定されたコマンドを実行します。コマンドがバッファに登録される場合には、このルーチンはコマンドをバッファに登録した後、制御を戻し、コマンドが終了した後、ユーザが指定した AST ルーチンを起動します。コマンドがバッファに登録されない場合には、SMG\$EXECUTE\_COMMAND ルーチンは、コマンドが実行を終了するのを待ち、その後、コマンドの状態を戻します。

コマンド文字列を指定する場合には、DCL コマンドの 1 文字目としてドル記号 (\$) を指定しなければなりません。ドル記号から始まらないコマンド文字列は、前のコマンドに対する入力データとして解釈されます。

SMG\$EXECUTE\_COMMAND ルーチンは、コマンドを実行するときに、そのコマンドとコマンドからの出力を指定された仮想ディスプレイに出力します。指定した仮想ディスプレイに対して入出力を実行しないでください。SPAWN コマンド、GOTO

コマンド、LOGOUT コマンドをコマンド文字列として使用することはできず、これらのコマンドを使用した場合には、予測できない結果が発生します。

入出力はターミナル・ドライバを通じてではなく、メールボックスを使用して実行されるため、コマンド・プロンプトや、`Ctrl/C`、`Ctrl/Y`、`Ctrl/Z`などの単一文字コマンドは使用できません。`Ctrl/Z`をサブプロセスに送信する場合には、`flags` 引数に対して `SMG$M_SEND_EOF` を指定しなければなりません。詳しい説明は、『OpenVMS I/O User's Reference Manual』のメールボックス・ドライバに関する節を参照してください。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INPTOOLON	入力の長さが 255 バイトを超えています。
SMG\$_INVIDS_ID	display-id 引数に誤りがあります。
SMG\$_NOSUBEXI	サブプロセスが存在しません。
SS\$_xxxx	\$QIO、\$DCLAST、または\$SYNCH から戻された状態値。
LIB\$_xxxx	LIB\$ANALYZE_SDESC ルーチンから戻された状態値。
SMG\$_xxxx	SMG\$PUT_LINE ルーチンから戻された状態値。

---

## SMG\$FIND\_CURSOR\_DISPLAY

### Find Display That Contains the Cursor

Find Display that Contains the Cursor ルーチンは、物理カーソルが現在格納されている、最後にペーストされた仮想ディスプレイの識別子を戻します。

---

#### 形式

SMG\$FIND\_CURSOR\_DISPLAY *pasteboard-id* ,*display-id* [,*pasteboard-row*] [,*pasteboard-column*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

物理カーソルが検索される対象となるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

物理カーソルが存在する仮想ディスプレイの識別子を書き込まれます。

display-id 引数は、ディスプレイ識別子を書き込まれる符号なしロングワードのアドレスです。

## pasteboard-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

物理カーソルの検索を開始する行番号。pasteboard-row 引数は省略可能な引数であり、ペーストボードの行番号を格納した符号付きロングワードのアドレスです。物理カーソルの行番号の代わりに pasteboard-row 引数を使用できます。

## pasteboard-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

物理カーソルが検索を開始するカラム番号。pasteboard-column 引数は省略可能な引数であり、ペーストボード・カラム番号を格納した符号付きロングワードのアドレスです。物理カーソルのカラム番号の代わりに pasteboard-column 引数を使用できます。

---

 説明

SMG\$FIND\_CURSOR\_DISPLAY ルーチンは、指定されたペーストボードの物理カーソルがどの仮想ディスプレイに格納されているかを判断し、仮想ディスプレイの識別子を戻します。SMG\$FIND\_CURSOR\_DISPLAY ルーチンは、物理カーソルが格納されている最後にペーストされた仮想ディスプレイの display-id 引数を戻します。仮想ディスプレイに物理カーソルが格納されていない場合には、このルーチンは 0 を戻します。これは、ディスプレイ識別子としては無効な値です。

---

 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$FLUSH\_BUFFER

## Flush Buffer

Flush Buffer ルーチンは、バッファに保存されているすべての出力を漢字ターミナルに送信します。

---

### 形式

SMG\$FLUSH\_BUFFER *pasteboard-id*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

*pasteboard-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

出力するペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

---

### 説明

SMG\$FLUSH\_BUFFER ルーチンを使用すると、バッファに保存されていてまだ送信されていない出力はすべて、ただちにペーストボードに送信されます。日本語 SMG は、通常バッファが満杯になったときにテキストを出力します。したがってこのルーチンを使用しなければならないのは、まだ満杯になっていないバッファを出力しなければならない場合だけです。呼び出しプログラムは通常、何らかの CPU に負

担のかかる演算を実行する直前や、ペーストボードを最新の状態に更新しなければならないときに、このルーチンを呼び出します。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_WRONUMARG	引数の数が誤っています。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SS\$_xxxx	\$QIOW から戻されたエラー。

---

## SMG\$GET\_BROADCAST\_MESSAGE

### Get Broadcast Message

Get Broadcast Message ルーチンは、メッセージがペーストボードにブロードキャストされたかどうかを判断し、メッセージを受信している場合には、そのメッセージを戻します。

---

#### 形式

SMG\$GET\_BROADCAST\_MESSAGE  
*pasteboard-id* [*message*] [*message-length*] [*message-type*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ブロードキャスト・メッセージが存在するかどうかを確認するペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納する符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

message

VMS 用法:	char_string
データ型:	character string
アクセス:	write only
受け渡し方:	by descriptor

ブロードキャスト・メッセージが存在した場合、そのメッセージが書き込まれる文字列。message 引数は、メッセージ・テキストが書き込まれる文字列を示すディスクリ

ブタのアドレスです。この引数を省略した場合には、ブロードキャスト・メッセージは破棄されます。

#### message-length

OpenVMS 用法 word\_unsigned  
 データ型 word (unsigned)  
 アクセス write only  
 受け渡し方 by reference

ブロードキャスト・メッセージの実際の長さが書き込まれます。

message-length 引数は、メッセージの長さが書き込まれる符号なしワードのアドレスです。

#### message-type

OpenVMS 用法 word\_unsigned  
 データ型 word (unsigned)  
 アクセス write only  
 受け渡し方 by reference

ブロードキャスト・メッセージのタイプが書き込まれます。message-type 引数は、メッセージのタイプが書き込まれる符号なしワードのアドレスです。message-type 引数の値は\$MSGDEF ライブラリ定義によって定義されています。message-type 引数の値が MSG\$\_TRMBRDCST でない場合には、戻される状態値は SMG\$\_NOBRDMSG です。

---

## 説明

SMG\$GET\_BROADCAST\_MESSAGE ルーチンは、ブロードキャスト・トラッピングが許可されているときに、ブロードキャスト・メッセージが指定されたペーストボードに送信されたかどうかを判断し、送信された場合には、message 引数にメッセージを戻します。すべてのブロードキャスト・メッセージが戻されるまで、このルーチンを繰り返し呼び出すことができます。すべてのブロードキャスト・メッセージが戻されたならば、SMG\$GET\_BROADCAST\_MESSAGE ルーチンは正常終了状態として SMG\$\_NO\_MORMSG を戻します。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_NO_MORMSG	正常終了。受信したメッセージはすべて戻されました。
SMG\$_NONBRDMSG	ブロードキャスト・メッセージは戻されませんでした。
SMG\$_WRONUMARG	引数の数が誤っています。

SMG\$GET\_BROADCAST\_MESSAGE

その他

LIB\$SCOPY\_DXDX から戻される状態値。

---

# SMG\$GET\_DISPLAY\_ATTR

## Get Display Attributes

Get Display Attributes ルーチンは仮想ディスプレイに関連する属性を戻します。

---

### 形式

SMG\$GET\_DISPLAY\_ATTR *display-id* [*height*] [*width*] [*display-attributes*] [*video-attributes*]  
 [*character-set*] [*flags*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

属性情報を要求する仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

height

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	write only
受け渡し方	by reference

仮想ディスプレイの行数。height 引数は省略可能で、仮想ディスプレイの行数が書き込まれる符号付きロングワードのアドレスです。

width

OpenVMS 用法	longword_signed
データ型	longword (signed)

アクセス write only  
受け渡し方 by reference

仮想ディスプレイのカラム数。width 引数は省略可能で、仮想ディスプレイのカラム数を書き込まれる符号付きロングワードのアドレスです。

## display-attributes

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス write only  
受け渡し方 by reference

現在の仮想ディスプレイの属性。display-attributes 引数は省略可能で、現在の仮想ディスプレイの属性が書き込まれる符号なしロングワードのアドレスです。

この引数に対して書き込まれる値は次のとおりです。

SMG\$M_BORDER	境界付き仮想ディスプレイを指示します。
SMG\$M_BLOCK_BORDER	ブロック形境界付き仮想ディスプレイを指示します。
SMG\$M_DISPLAY_CONTROLS	漢字ターミナルがグラフィック文字をサポートする場合には、キャリッジ・リターンやライン・フィードなどの制御文字をグラフィック文字として表示することを指示します。
SMG\$M_TRUNC_ICON	仮想ディスプレイの幅をこえる行を切り捨てた際に、その場所にアイコン (通常は菱形) を表示することを指示します。

## video-attributes

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス write only  
受け渡し方 by reference

現在のビデオ属性。video-attributes 引数は省略可能で、現在のビデオ属性が書き込まれる符号なしロングワードのアドレスです。

この引数に対して書き込まれる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## character-set

OpenVMS 用法 longword\_unsigned  
データ型 longword (unsigned)  
アクセス write only

受け渡し方      by reference

使用されている文字集合。character-set 引数は省略可能で、現在の文字集合属性が書き込まれる符号なしロングワードのアドレスです。文字集合が未定義の場合、SMG\$C\_UNKNOWN が戻ります。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_UNKNOWN	未定義文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

flags

OpenVMS 用法    mask\_longword  
 データ型        longword (unsigned)  
 アクセス        write only  
 受け渡し方      by reference

仮想ディスプレイ属性を格納するビット・マスク。flags 引数は省略可能で、フラグを格納する符号なしロングワードのアドレスです。この引数に格納される値は次のとおりです。

SMG\$M_SUBPROCESS	仮想ディスプレイにサブプロセスがアタッチされています。
SMG\$M_MENU	仮想ディスプレイにメニューが含まれています。
SMG\$M_VIEWPORT	仮想ディスプレイにビューポートが含まれています。

## 説明

SMG\$GET\_DISPLAY\_ATTR ルーチンは仮想ディスプレイの属性を返します。オリジナルの SMG ルーチンは、ASCII 文字集合と DEC 特殊文字集合のみをサポートしているため、SMG\$GET\_DISPLAY\_ATTR ルーチンは character-set 引数に、SMG\$C\_ASCII または SMG\$C\_SPEC\_GRAPHICS を戻します。日本語 SMG では、その他の文字集合をサポートしているため、SMG\$GET\_DISPLAY\_ATTR ルーチンは character-set 引数に日本語 SMG でサポートしている文字集合を戻します。

---

戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$GET\_KEY\_DEF

### Get Key Definition

Get Key Definition ルーチンは、指定されたキーに対するキー定義を戻します。

---

#### 形式

SMG\$GET\_KEY\_DEF *key-table-id* ,*key-name* [*,if-state*] [*,attributes*] [*,equivalence-string*] [*,state-string*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword(unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

key-table-id

OpenVMS 用法	identifier
データ型	longword(unsigned)
アクセス	read only
受け渡し方	by reference

定義を検索するキー・テーブルを指定します。key-table-id 引数は、キーテーブル識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_KEY\_TABLE ルーチンによって与えられます。

key-name

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

定義に対応するキーの名前を指定します。key-name 引数は、キー名を示すディスクリプタのアドレスです。

指定できるキーについては、第 3 章の表 3-1 を参照してください。

## if-state

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

キーが押された後に有効になる現在の状態名を指定します。if-state 引数は、状態名を示すディスクリプタのアドレスです。

詳しい説明は、SMG\$ADD\_KEY\_DEF ルーチンの説明を参照してください。

## attributes

OpenVMS 用法 mask\_longword  
 データ型 longword(unsigned)  
 アクセス write only  
 受け渡し方 by reference

このキー定義に対する属性ビット・マスクが書き込まれます。attributes 引数は、キーの属性を記述するビット・マスクが書き込まれるロングワードのアドレスです。

書き込まれる値は次のとおりです。

SMG\$M_KEY_NOECHO	このビットが設定されている場合には、このキーが押されたときに、equiv-string 引数がエコー表示されないことを指定します。 このビットがクリアされている場合には、equiv-string 引数はエコー表示されます。SMG\$M_KEY_TERMINATE が設定されていない場合には、SMG\$M_KEY_NOECHO は無視されます。
SMG\$M_KEY_TERMINATE	このビットが設定されている場合には、このキーが押されたときに (if-state 引数の条件指定にしたがって)、入力行は終了し、追加文字を受け付けられないことを指定します。 このビットがクリアされている場合には、追加文字を受け付けることができます。
SMG\$M_KEY_LOCK	このビットが設定されており、state-string 引数が指定されている場合には、state-string 引数によって指定される状態名は、定義に state-string 引数を含む後続のキーストロークによって明示的に変更されるまで、現在の状態のまま変更されません。 このビットがクリアされている場合には、state-string 引数によって指定される状態名は、次の定義済みキーストロークに対してのみ有効になります。
SMG\$M_KEY_PROTECTED	このビットが設定されている場合には、このキー定義を変更または削除できないことを指定します。 このビットがクリアされている場合には、キー定義を変更または削除できます。

## equivalence-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス write only  
 受け渡し方 by descriptor

このキー定義に対する同値文字列が書き込まれます。equivalence-string 引数は、同値文字列が書き込まれる文字列を示すディスクリプタのアドレスです。

state-string

OpenVMS 用法	char_string
データ型	character string
アクセス	write only
受け渡し方	by descriptor

このキー定義によって設定される新しい状態名が書き込まれます。state-string 引数は、新しい状態文字列が書き込まれることを示すディスクリプタのアドレスです。

---

## 説明

SMG\$GET\_KEY\_DEF ルーチンは、指定された key-name 引数と if-state 引数に対応するキー定義を戻します。このキー定義は SMG\$READ\_COMPOSED\_LINE ルーチンの呼び出しで使用できます。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVKEYNAM	key-name 引数に誤りがあります。
SMG\$_INVKTB_ID	key-table-id 引数に誤りがあります。
SMG\$_KEYNOTDEF	キーが定義されていません。
SMG\$_WRONUMARG	引数の数が誤っています。
その他	LIB\$SCOPY_DXDX から戻された状態値。

---

## SMG\$GET\_KEYBOARD\_ATTRIBUTES

### Get Keyboard Attributes

Get Keyboard Attributes ルーチンは、仮想キーボードに関する情報を検索し、その情報をユーザ指定領域 (キーボード情報テーブル (KIT)) に格納します。

---

#### 形式

SMG\$GET\_KEYBOARD\_ATTRIBUTES *keyboard-id* , *keyboard-info-table* ,*keyboard-info-table-size*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

keyboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

キーボード識別子。keyboard-id 引数は、読み込みが行われる仮想キーボードの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンによって与えられません。

keyboard-info-table

OpenVMS 用法	unspecified
データ型	unspecified
アクセス	write only
受け渡し方	by reference, array reference

キーボード情報が書き込まれます。keyboard-info-table引数は、キーボード属性が書き込まれるデータブロックのアドレスです。

キーボード情報テーブル (KIT) は、そのサイズとフィールド参照が \$SMGDEF に記述されているバイト・ブロックです。ユーザは正しいサイズのブロックを割り当て、そのアドレスをこのルーチンに渡さなければなりません。

keyboard-info-table 引数の値は次のシンボリック名を使用してアクセスできます。

SMG\$L_DEV_CHAR	装置特性 (ロングワード)
SMG\$L_DEV_DEPEND	装置依存特性 1
SMG\$L_DEV_DEPEND2	装置依存特性 2
SMG\$B_DEV_CLASS	装置クラス (バイト) - たとえば、DC\$_TERM
SMG\$B_RECALL_NUM	再呼び出しバッファのサイズ (バイト)*
SMG\$B_DEVTYPE	物理装置タイプ (バイト) - たとえば、TTS_VT100
SMG\$B_TYPEAHD_CHAR	先読みバッファ内の最初の文字 (バイト)*
SMG\$W_NUM_COLUMNS	漢字ターミナルの幅 (ワード)
SMG\$W_TYPEAHD_CNT	先読みバッファ内の文字数 (ワード)*

アスタリスク (\*) を指定した項目は、装置が漢字ターミナルである場合 (DEVCLASS=DC\$\_TERM) を除き、0 になります。

keyboard-info-table-size

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

キーボード情報テーブルのサイズ。keyboard-info-table-size 引数は、KIT のサイズ (バイト数) を格納した符号なしロングワードのアドレスです。

正確なサイズを指定しなければなりません。このサイズは、シンボル定数 SMG\$\_KEYBOARD\_INFO\_BLOCK を使用して指定できます。

---

## 説明

SMG\$GET\_KEYBOARD\_ATTRIBUTES ルーチンは、仮想キーボードに関する情報を検索し、この情報をキーボード情報テーブル (KIT) に格納します。KIT は 1 バイト・ブロックで構成されるユーザ指定領域です。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVARG	KIT のサイズに誤りがあります。
SMG\$_INVKBD_ID	keyboard-id 引数に誤りがあります。

---

## SMG\$GET\_NUMERIC\_DATA

### Get Numeric Terminal Data

Get Numeric Terminal Data ルーチンは、TERMTABLE.EXE をアクセスし、指定された論理値機能または数値機能に対応する値を戻します。

---

#### 形式

SMG\$GET\_NUMERIC\_DATA *termtable-address* ,*request-code* ,*buffer-address*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

*termtable-address*

OpenVMS 用法	address
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

情報を得たい漢字ターミナルの、TERMTABLE エントリのアドレスを指定します。*termtable-address* 引数は、ターミナル機能テーブル (TERMTABLE) のアドレスを格納した符号なしロングワードのアドレスです。

SMG\$GET\_NUMERIC\_DATA ルーチンを呼び出す前に、SMG\$INIT\_TERM\_TABLE ルーチンまたは SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチンを呼び出すことにより、このターミナル機能テーブルのアドレスを得ておかなければなりません。

*request-code*

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

情報を得たい機能を指定する要求コード。request-code 引数は、この要求コードを格納した符号なしロングワード定数です。要求コードは SMG\$K\_code という形式であり、code はたとえば ANSI\_CRT などのターミナル機能テーブル (TERMTABLE) 内のキーワードに対応します。SMG\$K\_code 定数は、HP の提供する SSMGTRMPTR ライブラリに定義されています。

使用できる機能フィールドについては、本書の第 5 章の表 5-1、表 5-2、表 5-3、表 5-4 を参照してください。

buffer-address

OpenVMS 用法	address
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

SMG\$GET\_NUMERIC\_DATA ルーチンが、数値機能データを書き込むロングワードの 1 バイト目のアドレス。buffer-address 引数は、このバッファのアドレスを格納する符号なしロングワードです。

---

## 説明

SMG\$GET\_NUMERIC\_DATA ルーチンは、指定されたターミナル機能テーブルから要求された数値情報を検索します。SMG\$GET\_NUMERIC\_DATA ルーチンを呼び出す前に、SMG\$INIT\_TERM\_TABLE ルーチンまたは SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチンを呼び出すことにより、そのターミナル機能テーブルのアドレスを得ておかなければなりません。このルーチンを使用しなければならないのは、独自の TERMTABLE アクセスを実行する場合と、漢字ターミナルに対して直接日本語 SMG によるもの以外の入出力を実行する場合だけです。

---

## 戻される状態値

SS\$NORMAL	正常終了。
SMG\$INVREQCOD	要求コードに誤りがあります。
SMG\$INVERTAB	ターミナル機能テーブル・アドレスに誤りがあります。

---

# SMG\$GET\_PASTEBOARD\_ATTRIBUTES

## Get Pasteboard Attributes

Get Pasteboard Attributes ルーチンは、ペーストボード属性を検索し、それをペーストボード情報テーブルに格納します。

---

### 形式

```
SMG$GET_PASTEBOARD_ATTRIBUTES
    pasteboard-id ,pasteboard-info-table ,pasteboard-info-table-size
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

情報が要求されるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

pasteboard-info-table

OpenVMS 用法	unspecified
データ型	unspecified
アクセス	write only
受け渡し方	by reference, array reference

ペーストボード属性が書き込まれます。pasteboard-info-table 引数は、ペーストボード属性が書き込まれるデータ構造のアドレスです。

SMG\$L_DEVCHAR	装置属性 (ロングワード)
SMG\$L_DEVDEPEND	装置依存特性 1(ロングワード)
SMG\$L_DEVDEPEND2	装置依存特性 2(ロングワード)
SMG\$B_DEVCLASS	装置クラス (バイト) - たとえば, DC\$_TERM
SMG\$B_SMG_DEVTYPE	内部 SMG 装置タイプ (バイト)。SMG\$B_SMG_DEVTYPE の値は次のいずれかです。  SMG\$K_UNKNOWN SMG\$K_VTForeign SMG\$K_HARDCOPY SMG\$K_VTTERMTABLE
SMG\$B_PHY_DEVTYPE	物理装置タイプ (バイト) - たとえば, TTS_VT100。 SMG\$B_PHY_DEVTYPE の値は STARLET の \$TTDEF に定義されています。
SMG\$B_ROWS	ペーストボードの行数 (バイト)
SMG\$W_WIDTH	ペーストボードの幅 (ワード)
SMG\$B_COLOR	背景色の設定 (バイト)。SMG\$B_COLOR の値は次のいずれかです。  SMG\$C_COLOR_UNKNOWN            認識されない背景色 SMG\$C_COLOR_WHITE            白の背景色 SMG\$C_COLOR_BLACK            黒の背景色 SMG\$C_COLOR_BLUE            青の背景色 SMG\$C_COLOR_CYAN            青緑の背景色 SMG\$C_COLOR_GREEN            緑の背景色 SMG\$C_COLOR_MAGENTA        紫の背景色 SMG\$C_COLOR_RED            赤の背景色 SMG\$C_COLOR_YELLOW        黄色の背景色 SMG\$C_COLOR_LIGHT        明るい背景色 SMG\$C_COLOR_DARK        暗い背景色 SMG\$C_COLOR_USER1        ユーザ定義背景色 1 SMG\$C_COLOR_USER2        ユーザ定義背景色 2
SMG\$B_PARITY	パリティ属性 (バイト) - ペーストボードが漢字ターミナルでない場合には, このフィールドは 0 です。
SMG\$W_SPEED	漢字ターミナルの通信速度 (ワード) - ペーストボードが漢字ターミナルでない場合には, このフィールドは 0 です。
SMG\$W_FILL	フィル属性 (ワード) - ペーストボードがターミナルでない場合には, このフィールドは 0 です。
SMG\$W_PHYS_CURSOR_ROW	物理カーソルの現在のペーストボード行番号 (ワード)
SMG\$W_PHYS_CURSOR_COL	物理カーソルの現在のペーストボード・カラム番号 (ワード)
SMG\$L_CURSOR_DID	物理カーソルが格納されている一番上のディスプレイのディスプレイ識別子 (ロングワード)

pasteboard-info-table 引数内の値は, 次のシンボリック名を使用してアクセスできません。

## SMG\$GET\_PASTEBOARD\_ATTRIBUTES

pasteboard-info-table-size

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ペーストボード情報テーブルのバイト数を指定します。pasteboard-info-table-size 引数は、ペーストボード情報テーブルのサイズ (バイト数) を格納した符号なしロングワードのアドレスです。

指定するサイズは正確な値でなければなりません。このサイズは、SMG\$C\_PASTEBOARD\_INFO\_BLOCK というシンボル定数を使用して指定できます。

---

### 説明

SMG\$GET\_PASTEBOARD\_ATTRIBUTES ルーチンは、ペーストボード属性を検索し、それをペーストボード情報テーブルに格納します。

---

### 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVARG	pasteboard-info-table-size 引数に指定されたサイズに誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$GET\_PASTING\_INFO

### Return Pasting Information

指定された仮想ディスプレイが現在ペーストされている場合には、Return Pasting Information ルーチンは、ペーストボードの行とカラム番号を返します。

---

#### 形式

SMG\$GET\_PASTING\_INFO *display-id ,pasteboard-id ,flags [,pasteboard-row] [,pasteboard-column]*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ペースト情報を調べる仮想ディスプレイの識別子。display-id 引数は、この仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

仮想ディスプレイがペーストされているペーストボードの識別子。pasteboard-id 引数は、このペーストボードの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

## flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス write only  
 受け渡し方 by reference

指定されたペーストボードに対して指定された仮想ディスプレイの状態を示すビット・マスク。flags 引数は、フラグが書き込まれる符号なしロングワードのアドレスです。flags 引数に対して指定できる値は次のとおりです。

0 仮想ディスプレイは指定されたペーストボードにペーストされていません。  
 SMG\$M\_DISPLAY\_PASTED display-id 引数によって指定される仮想ディスプレイは、pasteboard-id 引数によって指定されるペーストボードにペーストされています。

## pasteboard-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

指定された仮想ディスプレイの 1 行目があるペーストボードの行。pasteboard-row 引数は省略可能な引数であり、仮想ディスプレイの最初の行のあるペーストボードの行番号が書き込まれる符号付きロングワードのアドレスです。

## pasteboard-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

指定された仮想ディスプレイの 1 カラム目があるペーストボードのカラム。pasteboard-column 引数は省略可能な引数であり、仮想ディスプレイの最初のカラムのあるペーストボード・カラムのカラム番号が書き込まれる符号付きロングワードのアドレスです。

---

 説明

SMG\$GET\_PASTING\_INFO ルーチンは、まず display-id 引数によって指定される仮想ディスプレイが、pasteboard-id 引数によって指定されたペーストボードにペーストされているかどうかを確認します。この仮想ディスプレイがこのペーストボードにペーストされている場合には、SMG\$GET\_PASTING\_INFO ルーチンは、ペーストされている仮想ディスプレイの 1 行目、および 1 カラム目に対応するペーストボードの行番号とカラム番号を戻します。

---

戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_ILLBATFNC	ディスプレイはバッチ処理されています。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$GET\_TERM\_DATA

### Get Terminal Data

Get Terminal Data ルーチンは、TERMTABLE.EXE にアクセスして、漢字ターミナルに指定した操作を実行させる文字シーケンスを返します。

---

#### 形式

```
SMG$GET_TERM_DATA termtable-address ,request-code ,maximum-buffer-length ,return-length
                  ,buffer-address [input-argument-vector]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

*termtable-address*

OpenVMS 用法	address
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

情報を得たい漢字ターミナルの TERMTABLE エントリのアドレス。termtable-address 引数は、ターミナル機能テーブル (TERMTABLE) のアドレスを格納したロングワードのアドレスです。

この引数は SMG\$INIT\_TERM\_TABLE または SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチンによって与えられます。

*request-code*

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

情報を得たい機能。code 引数は SMG\$K\_code の形式をした要求コードを格納した符号なしロングワードのアドレスです。SMG\$K\_code 定数は、弊社の提供する \$SMGTRMPTR ライブラリに定義されています。

指定できる要求コードに関しては本書の第 5 章の表 5-1, 表 5-2, 表 5-3, 表 5-4 を参照してください。

#### maximum-buffer-length

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

機能データが書き込まれるバッファの最大長。maximum-buffer-length 引数は、バッファに書き込める最大バイト数を格納した符号付きロングワードのアドレスです。

#### return-length

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	write only
受け渡し方	by reference

バッファに実際に書き込まれたバイト数。return-length 引数は、バッファに転送されたバイト数が書き込まれる符号付きロングワードのアドレスです。

#### buffer-address

OpenVMS 用法	address
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

機能データを受信するバッファの 1 バイト目のアドレス。buffer-address 引数は、バッファのアドレスが書き込まれる符号なしロングワードのアドレスです。

#### input-argument-vector

OpenVMS 用法	vector_longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference, array reference

特定の機能操作の文字シーケンスを作成するために使われる引数を渡す、ロングワード・リストのアドレス。input-argument-vector 引数は、文字シーケンスを作成するための引数を格納した符号なしロングワードの配列のアドレスです。ある機能を実行する文字シーケンスを作成するためには引数が必要になる場合があります。input-argument-vector 引数は、これらの引数を渡すためのロングワード・リストのアドレスを指定します。最初のロングワードは次に続く引数の数を格納していなければなりません。

---

## 説明

SMG\$GET\_TERM\_DATA ルーチンは、漢字ターミナルへの入出力を直接行う場合、すなわち日本語 SMG ルーチンを使用せずに漢字ターミナルへの入出力を行う場合に限り使用します。指定されたターミナル・タイプの TERMTABLE.EXE エントリに

## SMG\$GET\_TERM\_DATA

アクセスし、特定の操作を実行する文字シーケンスを返します。この文字シーケンスの漢字ターミナルへの出力は行いません。

---

### 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVREQCOD	要求コードが誤っています。
SMG\$_INVERTTAB	TERMTABLE アドレスが誤っています。

---

# SMG\$GET\_VIEWPORT\_CHAR

## Get Characteristics of Display Viewport

Get Characteristics of Display Viewport ルーチンは、指定されたビューポートの属性を戻します。

---

### 形式

SMG\$GET\_VIEWPORT\_CHAR

*display-id* [*viewport-row-start*] [*viewport-column-start*]  
 [*viewport-number-rows*] [*viewport-number-columns*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ビューポートに対応づけられている仮想ディスプレイの識別子。display-id 引数は、この識別子を格納した符号なしロングワードのアドレスです。

viewport-row-start

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	write only
受け渡し方	by reference

ビューポートの開始行番号を受け取る引数 (省略可能)。viewport-row-start 引数は、この行番号が書き込まれる符号付きロングワードのアドレスです。

## viewport-column-start

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

指定されたビューポートの開始カラム番号を受け取る引数 (省略可能)。viewport-column-start 引数は、このカラム番号が書き込まれる符号付きロングワードのアドレスです。

## viewport-number-rows

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

指定されたビューポート内の行数を受け取る省略可能な引数。viewport-number-rows 引数は、この行数が書き込まれる符号付きロングワードのアドレスです。

## viewport-number-columns

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

指定されたビューポート内のカラム数を受け取る省略可能な引数。viewport-number-columns 引数は、このカラム数が書き込まれる符号付きロングワードのアドレスです。

---

 説明

SMG\$GET\_VIEWPORT\_CHAR ルーチンは、指定されたビューポートの要求された属性を戻します。

---

 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_NO_WINASSOC	仮想ディスプレイに対応づけられているビューポートはありません。
SMG\$_WRONUMARG	引数の数が誤っています。

## 例

```

1. C+
C This FORTRAN example demonstrates the use of SMG$GET_VIEWPORT_CHAR.
C The viewport created will start at row 3, column 4. It will consist of
C 7 rows and 29 columns. Note the parameters used in the SMG$CREATE_VIEWPORT
C routine. I request 26 rows and 55 columns, but my viewport is truncated
C to fit.
C-
  IMPLICIT INTEGER (A-Z)
  INCLUDE '($SMGDEF)'

C Create the virtual display. Give it a border.

  ROWS = 4
  COLUMNS = 34

  STATUS = SMG$CREATE_VIRTUAL_DISPLAY
    1 ( ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER )
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%val(STATUS))

C Create the pasteboard.

  STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%val(STATUS))

C Put data in the virtual display.

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 1 of 4, you see', 1, 1 )
  IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 2 of 4, you see', 2, 1 )
  IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 3 of 4, you see', 3, 1 )
  IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))

  STATUS = SMG$PUT_CHARS ( DISPLAY1,
    1 'This is row number 4 of 4, you see', 4, 1 )
  IF (.not. STATUS) CALL LIB$SIGNAL(%val(STATUS))

C Paste the virtual display.

  STATUS = SMG$COPY_VIRTUAL_DISPLAY(DISPLAY1,DISPLAY2)
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

  STATUS = SMG$LABEL_BORDER (DISPLAY1, 'Full Display',,,SMG$M_BOLD)
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

  STATUS = SMG$LABEL_BORDER (DISPLAY2,'Viewport',,,SMG$M_BOLD)
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

  STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 2, 2 )
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

  STATUS = SMG$CREATE_VIEWPORT ( DISPLAY2, 1, 5, 26, 55 )
  IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

```

## SMG\$GET\_VIEWPORT\_CHAR

```
STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY2, PASTE1, 8, 2 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

CALL SMG$SET_PHYSICAL_CURSOR(PASTE1, 16, 1)
TYPE *, ' '
TYPE *, LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$GET_VIEWPORT_CHAR ( DISPLAY2, A, B, C, D )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

TYPE *, ' '
WRITE(5,7) A,B
7 FORMAT(1X,'Row start = ',I2,8X,'Column start = ',I2)
TYPE *, ' '
WRITE(5,8) C,D
8 FORMAT(1X,'Number of rows = ',I2,4X,'Number of columns = ',I3)

END
```

このプログラムの出力は図 8-23 に示すとおりです。

図 8-23 SMG\$GET\_VIEWPORT\_CHAR ルーチンによって生成される出力

```
Full Display
This is row number 1 of 4, you see
This is row number 2 of 4, you see
This is row number 3 of 4, you see
This is row number 4 of 4, you see

Viewport
is row number 1 of 4, you see
is row number 2 of 4, you see
is row number 3 of 4, you see
is row number 4 of 4, you see

%SMG-S-WINTRUNCFIT, Viewport truncated to fit
2146623360

Row start = 1           Column start = 5
Number of rows = 4     Number of columns = 30
```

ZK-6425-GE

---

# SMG\$HOME\_CURSOR

## Home Cursor

Home Cursor ルーチンは、仮想カーソルを仮想ディスプレイの指定した角に移動します。

---

### 形式

SMG\$HOME\_CURSOR *display-id* [*position-code*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

仮想カーソルが移動される仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納したロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

position-code

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

仮想カーソルの移動先を指定します。position-code 引数は、位置コードを格納したロングワードのアドレスです。

position-code 引数に対して指定できる値は次のとおりです。

コード	意味
SMG\$C_UPPER_LEFT	1 行目, 1 カラム目 (左上の角)。 position-code 引数を指定しなかった場合には、これが省略時の設定である。
SMG\$C_LOWER_LEFT	n 行目, 1 カラム目 (ただし, n はディスプレイ内の行数である)。つまり, 左下の角である。入力を受け付けるときに, 仮想ディスプレイを上向きにスクロールする場合には, このコードを使用すると便利である。
SMG\$C_UPPER_RIGHT	1 行目, m カラム目 (ただし, m はディスプレイ内のカラム数である) つまり, 右上の角である。
SMG\$C_LOWER_RIGHT	n 行目, m カラム目 (ただし, n はディスプレイ内の行数であり, m はディスプレイ内のカラム数である)。つまり, 右下の角である。

## 説明

SMG\$HOME\_CURSOR ルーチンは, position-code 引数に指定されたコードに従って, 仮想カーソルを指定された仮想ディスプレイの角に移動します。仮想ディスプレイのサイズを知っておく必要はなく, また, 仮想カーソルの位置も必要ありません。position-code 引数を省略した場合には, SMG\$HOME\_CURSOR ルーチンは仮想カーソルを仮想ディスプレイの左上の角に移動します。

## 戻される状態値

SS\$ _NORMAL	正常終了。
SMG\$ _INVARG	引数に誤りがあります。
SMG\$ _INVDIS_ID	display-id 引数に誤りがあります。
SMG\$ _WRONUMARG	引数の数が誤っています。

---

## SMG\$INIT\_TERM\_TABLE

### Initialize Terminal Table

Initialize Terminal Table ルーチンは、指定された漢字ターミナルに対して、TERMTABLE データベースを初期化することにより、その後の SMG\$GET\_TERM\_DATA の呼び出しで、その漢字ターミナルに対する情報とコマンド文字列を検索できるようにします。

---

#### 形式

SMG\$INIT\_TERM\_TABLE *terminal-name, termtable-address*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

*terminal-name*

OpenVMS 用法	device_name
データ型	character string
アクセス	read only
受け渡し方	by reference

漢字ターミナルの名前を指定します。terminal-name 引数は、ターミナル名を示すディスクリプタのアドレスです。名前は TERMTABLE.EXE 内のエン트리でなければなりません。

*termtable-address*

OpenVMS 用法	address
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

TERMTABLE.EXE 内の特定のタイプの漢字ターミナルのエントリのアドレス。termtable-address 引数は、ターミナル機能テーブルのアドレスが書き込まれる符号なしロングワードのアドレスです。

指定されたタイプの漢字ターミナルに対して SMG\$GET\_TERM\_DATA ルーチン呼び出す場合には、このアドレスを使用します。termtable-address 引数は SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチンによっても与えられます。

---

## 説明

SMG\$INIT\_TERM\_TABLE ルーチンは、その後で SMG\$GET\_TERM\_DATA ルーチン呼び出すときに、その漢字ターミナルに対して情報とコマンド文字列を検索できるように、指定された漢字ターミナルに対して TERMTABLE データベースを初期化します。このルーチンを使用しなければならないのは、漢字ターミナルに対して、直接日本語 SMG によるもの以外の入出力を実行する場合だけです。

SMG\$INIT\_TERM\_TABLE ルーチンは、まず TERM\$TABLOC という論理名の領域から TERMTABLE.EXE を検索します。TERMTABLE.EXE をその領域から検索できない場合には、このルーチンはグローバル・セクション SMG\$TERMTABLE を検索します。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_GBLSECMAP	正常終了。定義はグローバル・セクション TERMTABLE から検出されました。
SMG\$_PRISECMAP	正常終了。定義は個人用の TERMTABLE から検出されました。
SMG\$_UNDTERNAM	ターミナル名が定義されていません。
SMG\$_UNDTERNOP	未定義ターミナル。漢字ターミナルに対する定義を検索できず、個人用の TERMTABLE も見つかりませんでした。
SMG\$_UNDTERNOS	未定義ターミナル。漢字ターミナルに対する定義を検索できず、システム TERMTABLE も見つかりませんでした。

---

# SMG\$INIT\_TERM\_TABLE\_BY\_TYPE

## Initialize TERMTABLE by VMS Terminal Type

Initialize TERMTABLE by VMS Terminal Type ルーチンは、指定された漢字ターミナルに対して TERMTABLE データベースを初期化することにより、その後で SMG\$GET\_TERM\_DATA ルーチンを呼び出したときに、その漢字ターミナルに対する情報とコマンド文字列を検索できるようにします。

---

### 形式

```
SMG$INIT_TERM_TABLE_BY_TYPE
                                terminal-type
                                ,termtable-address
                                [,terminal-name]
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

terminal-type

OpenVMS 用法	byte_signed
データ型	byte (signed)
アクセス	read only
受け渡し方	by reference

漢字ターミナルの装置タイプであり、VMS シンボリック・ターミナル・タイプまたは \$GETDVI システム・サービスが戻す値によって指定されます。terminal-type 引数は、ターミナル・タイプを格納した符号付きバイトのアドレスです。

termtable-address

OpenVMS 用法	address
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

TERMTABLE.EXE 内の特定のターミナル・タイプのエントリのアドレス。termtable-address 引数は、ターミナル・エントリのアドレスが書き込まれる符号なしロングワードのアドレスです。

指定されたタイプの漢字ターミナルに対して、SMG\$GET\_TERM\_DATA プロシージャを呼び出す場合には、このアドレスを使用します。

terminal-name

OpenVMS 用法	device_name
データ型	character string
アクセス	write only
受け渡し方	by descriptor

装置タイプに対応するターミナル名が書き込まれる文字列。terminal-name 引数は、ターミナル名が書き込まれる文字列を示すディスクリプタのアドレスです。

---

## 説明

SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチンは、指定されたターミナル・タイプに対して、TERMTABLE データベースを初期化することにより、その後で SMG\$GET\_TERM\_DATA ルーチンを呼び出したときに、そのタイプの漢字ターミナルに対して情報とコマンド文字列を検索できるようにします。このルーチンを使用しなければならないのは、漢字ターミナルに対して直接日本語 SMG によるもの以外の入出力を実行する場合だけです。

SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチンはまず、TERM\$TABLOC という論理名の領域から TERMTABLE.EXE を検索します。その領域から TERMTABLE.EXE を検索できなかった場合には、このルーチンはグローバル・セクション SMG\$TERMTABLE を検索します。

---

## 戻される状態値

SSS_NORMAL	正常終了。
SMG\$_GBLSECMAP	正常終了。定義はグローバル・セクション TERMTABLE から検索されました。
SMG\$_PRISECMAP	正常終了。定義は個人用の TERMTABLE から検索されました。
SMG\$_UNDTERNAM	ターミナル名が定義されていません。
SMG\$_UNDTERNOP	未定義ターミナル。漢字ターミナルに対する定義を検索できず、個人用の TERMTABLE も見つかりませんでした。

SMG\$\_UNDTERNOS

未定義ターミナル。漢字ターミナルに対する定義を検索できず、システム TERMTABLE も見つかりませんでした。

---

## SMG\$INSERT\_CHARS

### Insert Characters

Insert Characters ルーチンは仮想ディスプレイに文字を挿入します。

---

#### 形式

SMG\$INSERT\_CHARS *display-id ,character-string ,start-row ,start-column [,rendition-set]  
[,rendition-complement] [,character-set]*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

文字が挿入される仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

character-string

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

挿入文字列。character-string 引数は、挿入される文字列を示すディスクリプタのアドレスです。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)

アクセス read only  
受け渡し方 by reference

挿入操作の開始行。start-row 引数は、行番号を格納した符号付きロングワードのアドレスです。

start-column

OpenVMS 用法 longword\_signed  
データ型 longword (signed)  
アクセス read only  
受け渡し方 by reference

挿入操作の開始カラム。start-column 引数は、カラム位置を格納した符号付きロングワードのアドレスです。

rendition-set

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

rendition-complement

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる属性は、すべてrendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その

後，rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより，ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御することができます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性を有効にする。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性を無効にする。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

挿入されるテキストの文字集合。character-set 引数は，文字集合コードを格納した符号なしロングワードのアドレスです。

この引数を省略した場合，SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$INSERT\_CHARS ルーチンは，start-row 引数と start-column 引数によって指定される位置に指定される文字列を挿入します。挿入された文字列の右側の文字は右にシフトされますが，その行に格納できない文字は切り捨てられます。仮想カーソルは，最後に挿入された文字位置に設定されます。

オリジナルの SMG ルーチンは，ASCII 文字集合と DEC 特殊文字集合のみをサポートします。日本語 SMG ではこの他に SMG\$C\_SUPPLEMENTAL，SMG\$C\_KANJI，SMG\$C\_JIS\_KANA，SMG\$C\_SDK が指定できます。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、全角文字の右半分から挿入操作が始まる場合は、その分割された文字は2つの未定義文字になり、挿入された文字列の初めと終わりに置かれます。また、行の最後の全角文字の右半分が切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

## 戻される状態値

SS\$ _NORMAL	正常終了。
LIB\$ _INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$ _INVARG	認識されない属性コードが指定されています。
SMG\$ _INVCOL	カラム位置が誤っています。指定されたカラムは仮想ディスプレイの外部です。
SMG\$ _INVDIS_ID	display-id 引数が誤っています。
SMG\$ _INVROW	行の位置が誤っています。指定された行は仮想ディスプレイの外部です。
SMG\$ _WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$ _WRONUMARG	引数の数が誤っています。

---

## 例

```

1. C+
   C This FORTRAN example program demonstrates the use of SMG$INSERT_CHARS.
   C-

       IMPLICIT INTEGER (A-Z)
       INCLUDE '($SMGDEF)'

C+
C Use SMG$CREATE_VIRTUAL_DISPLAY to create a virtual display
C with a border.
C-

       ROWS = 7
       COLUMNS = 50

       STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
C-

       STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Put data in the virtual display by calling SMG$PUT_CHARS.
C-

```

## SMG\$INSERT\_CHARS

```
STATUS = SMG$PUT_CHARS ( DISPLAY1,
1   ' This virtual display has 7 rows and 50 columns.', 2, 1 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1   ' This is a bordered virtual display.', 4, 1 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_CHARS ( DISPLAY1,
1   ' SMG$PUT_CHARS puts data in this virtual display.', 6, 1 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual display.
C-

STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$INSERT_CHARS to add a row 1 of text, starting at column 6.
C Underline these characters.
C-

STATUS = SMG$INSERT_CHARS ( DISPLAY1,
1   'This is a new row.', 1, 6, SMG$M_UNDERLINE )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

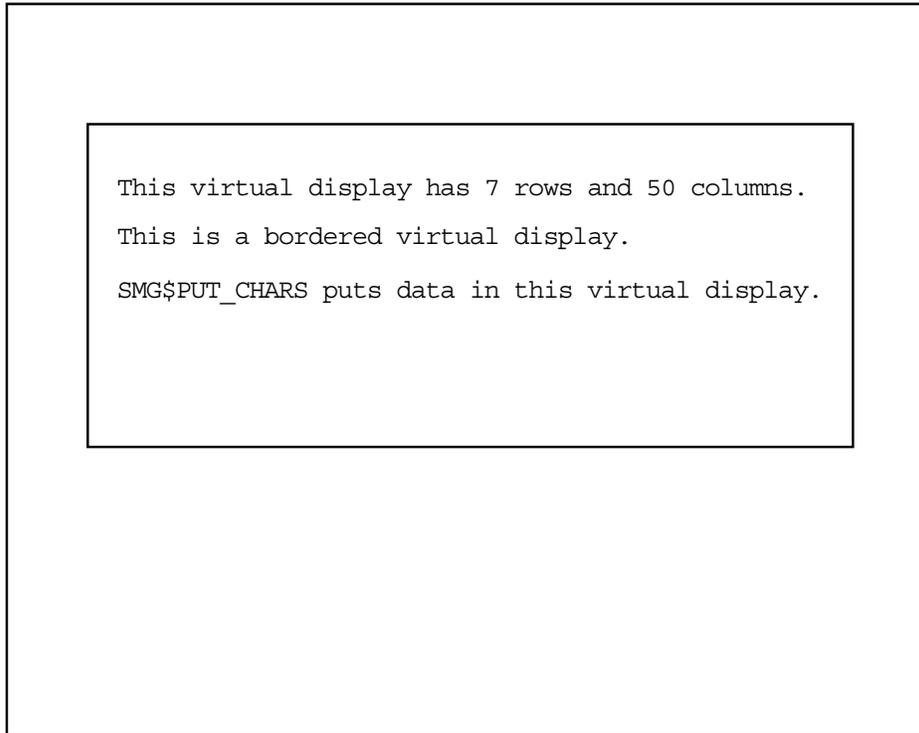
C+
C Calling SMG$INSERT_CHARS again, add text to row 6.
C Note that there will be some characters that will no
C longer fit on the line. They will be discarded. The
C new text will be bolded.
C-

STATUS = SMG$INSERT_CHARS ( DISPLAY1,
1   'to this bordered display.', 6, 28, SMG$M_BOLD )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

END
```

この FORTRAN プログラムが SMG\$INSERT\_CHARS ルーチン呼び出す前に生成する出力は、図 8-24 に示すとおりです。

図 8-24 SMG\$INSERT\_CHARS ルーチン呼び出す前に生成される出力

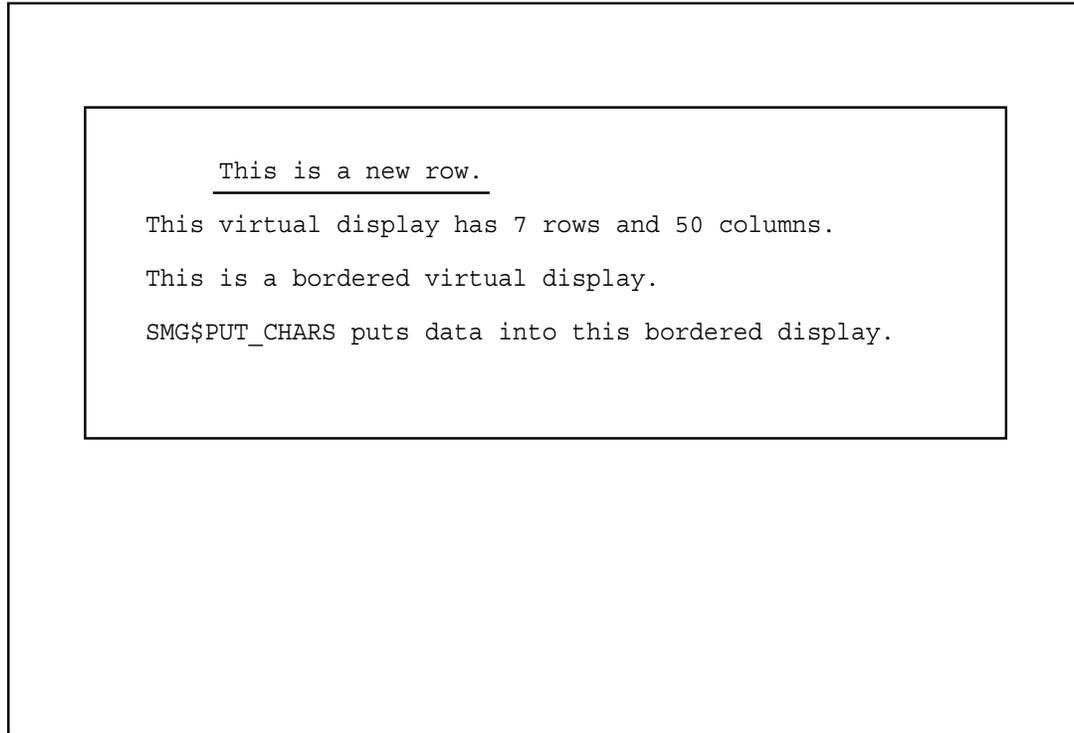


ZK-4132-GE

この FORTRAN プログラムが SMG\$INSERT\_CHARS ルーチンを実行後に生成する出力は、図 8-25 に示すとおりです。

## SMG\$INSERT\_CHARS

図 8-25 SMG\$INSERT\_CHARS ルーチンを実行した後に生成される出力



ZK-4144-GE

---

## SMG\$INSERT\_LINE

### Insert Line

Insert Line ルーチンは、仮想ディスプレイに 1 行を挿入して仮想ディスプレイをスクロールします。

---

#### 形式

SMG\$INSERT\_LINE *display-id* ,*start-row* [*character-string*] [*direction*] [*rendition-set*]  
 [*rendition-complement*] [*flags*] [*character-set*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

行が挿入される仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

文字列挿入によるスクロール開始行。start-row 引数は、行の位置を格納した符号付きロングワードのアドレスです。

## character-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

SMG\$INSERT\_LINE ルーチンによって挿入される文字列。character-string 引数はこの文字列を示すディスクリプタのアドレスです。

## direction

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

スクロール方向。direction 引数は、方向コードを格納したロングワード・ビット・マスクのアドレスです。この引数に対して指定できる値は SMG\$M\_UP と SMG\$M\_DOWN です。この引数を省略した場合は、SMG\$M\_UP が設定されます。

## rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## rendition-complement

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定できます。rendition-set 引数で指定できる属性はすべて、rendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの2つの引数を組み合わせて使用することにより、ユーザは1回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御できます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に戻す。
1	0	属性を有効にする。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性を無効にする。

#### flags

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

動作指定のビット・マスク。flags 引数は省略可能で、フラグを格納した符号なしロングワードのアドレスです。テキストを 1 行に格納できない場合に、どのような動作を実行するかを指定します。この引数に対して指定できる値は次のとおりです。

0 自動改行しないことを指定します  
 (省略時の設定値)。  
 SMG\$M\_WRAP\_CHAR 行の最後の文字で自動改行します。  
 SMG\$M\_WRAP\_WORD 行の最後の空白で自動改行します。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

出力されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$INSERT\_LINE ルーチンは、仮想ディスプレイの最初の行と最後の行を除く行に 1 行挿入できます。既存の行は指定された方向にスクロールされ、挿入のための空

間が作成されます。character-string 引数を指定した場合は、その文字列が生成される空間に書き込まれますが、省略した場合は空白になります。文字列が仮想ディスプレイの幅より短い場合は空白が埋め込まれます。

flags 引数が SMG\$M\_WRAP\_WORD または SMG\$M\_WRAP\_CHAR で、指定した文字列が仮想ディスプレイの幅より長い場合は、SMG\$INSERT\_LINE ルーチンは別の行をスクロールし、あふれた文字を生成される空間に書き込みます。flags 引数が 0 の場合は、あふれた文字は切り捨てられます。仮想カーソルは、最後に書き込まれた文字の次の文字位置に設定されます。

挿入操作で DEC 漢字または Super DEC 漢字文字集合を指定し、あふれた文字が全角文字の右半分の部分から切り捨てられる場合は、その文字の左半分は未定義文字になります。

仮想ディスプレイの最初の行または最後の行に行を挿入してスクロールする場合には、SMG\$PUT\_LINE を参照してください。

オリジナルの SMG ルーチンは、ASCII 文字集合と DEC 特殊文字集合のみをサポートします。日本語 SMG ではこの他に SMG\$C\_SUPPLEMENTAL, SMG\$C\_KANJI, SMG\$C\_JIS\_KANJI, SMG\$C\_SDK が指定できます。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVARG	引数が誤っています。指定された方向が上または下ではありません。
SMG\$_INVCOL	カラム位置が誤っています。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行の位置が誤っています。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## 例

```

1. C+
   C This FORTRAN example program demonstrates the use of SMG$INSERT_LINE.
   C
   C Include the SMG definitions. In particular, we want SMG$M_BORDER,
   C SMG$M_UNDERLINE, and SMG$M_UP.
   C-

       IMPLICIT INTEGER (A-Z)
       INCLUDE '($SMGDEF)'

```

## SMG\$INSERT\_LINE

```
C+
C Use SMG$CREATE_VIRTUAL_DISPLAY to create a virtual display
C with a border.
C-

      ROWS = 7
      COLUMNS = 50

      STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
C-

      STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$PUT_CHARS to put data in the virtual display.
C-

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This virtual display has 7 rows and 50 columns.', 2, 1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This is a bordered virtual display.', 4, 1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' SMG$PUT_CHARS puts data in this virtual display.', 6, 1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Paste the virtual display by calling SMG$PASTE_VIRTUAL_DISPLAY.
C-

      STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

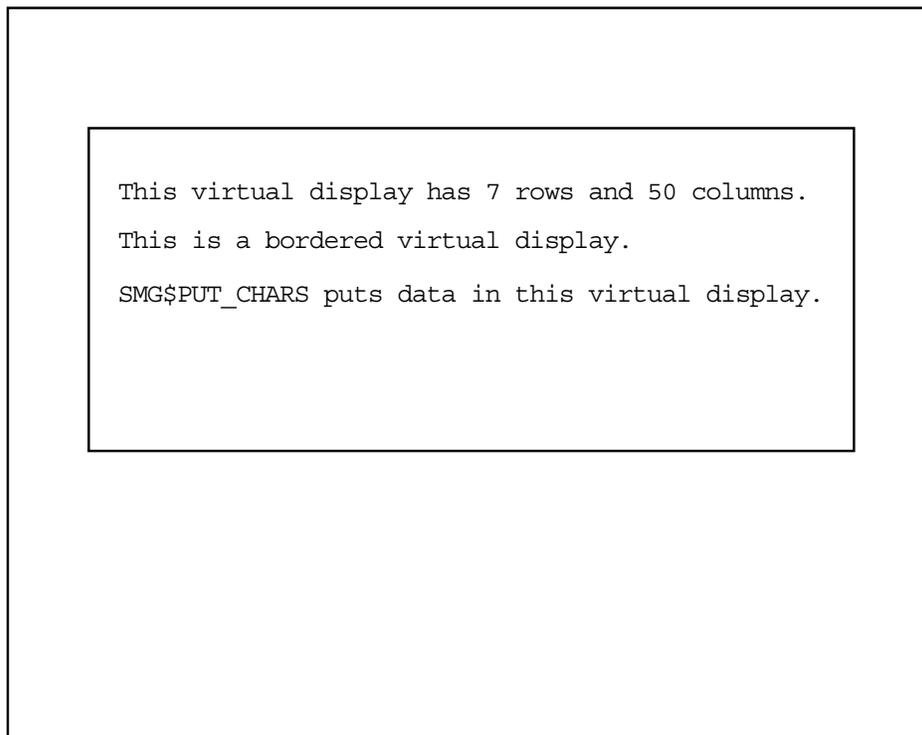
C+
C Call SMG$INSERT_LINE to add a line of text after line 6 and scroll
C the display. Also, underline the new characters.
C-

      STATUS = SMG$INSERT_LINE ( DISPLAY1, 7,
1      'This is a new line.', SMG$M_UP, SMG$M_UNDERLINE )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      END
```

FOTRAN プログラムによって生成される最初の出力は図 8-26 に示すとおりです。

図 8-26 SMG\$INSERT\_LINE ルーチン呼び出す前に生成される出力

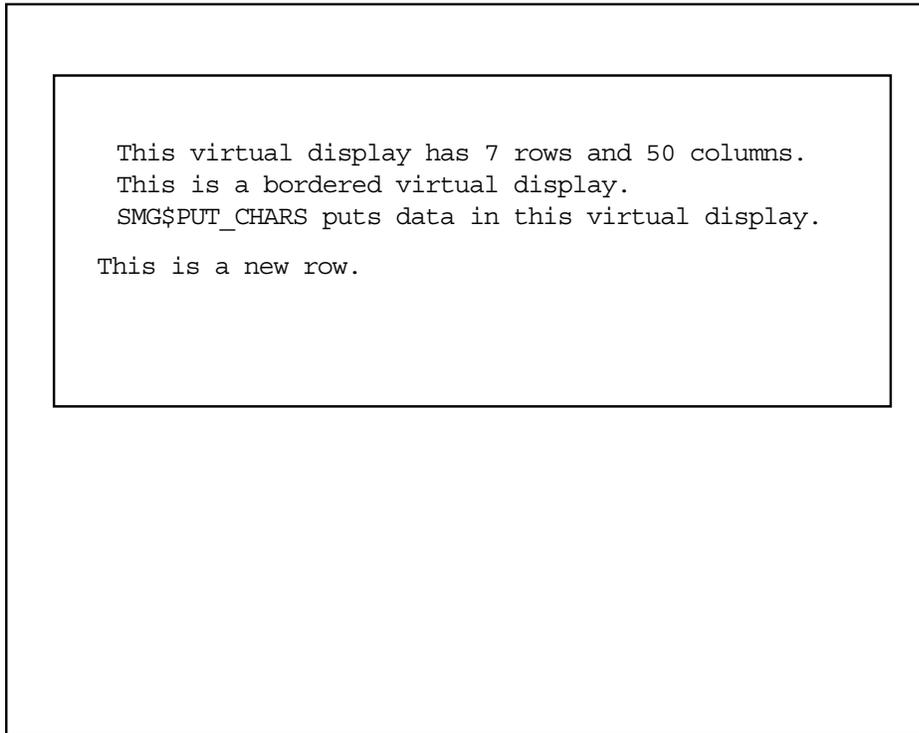


ZK-4132-GE

SMG\$INSERT\_LINE ルーチン呼び出した後に生成される出力は図 8-27 に示すとおりです。

## SMG\$INSERT\_LINE

図 8-27 SMG\$INSERT\_LINE ルーチン呼び出し後に生成される出力



ZK-4131-GE

---

# SMG\$INVALIDATE\_DISPLAY

## Mark a Display as Invalid

Mark a Display as Invalid ルーチンは、ディスプレイを無効状態としてマークし、ディスプレイ全体を再表示します。

---

### 形式

SMG\$INVALIDATE\_DISPLAY *display-id*

---

### 戻り値

VMS 用法: cond\_value  
データ型: longword (unsigned)  
アクセス: write only  
受け渡し方: by value

---

### 引数

*display-id*  
OpenVMS 用法 identifier  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

影響を受ける仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

display-id は SMG\$CREATE\_VIRTUAL\_DISPLAY から戻されます。

---

### 説明

SMG\$INVALIDATE\_DISPLAY はディスプレイを無効状態としてマークし、ディスプレイ全体を再表示します。このルーチンは通常、日本語 SMG を利用せずに、出力がディスプレイに書き込まれたことを検出したときに使用します。

このルーチンは、仮想ディスプレイの“フットプリント”のペーストボードの内容を無効にすることにより、仮想ディスプレイを再表示します。ディスプレイが隠されている場合には、仮想ディスプレイの隠されている部分だけが再表示されます (ディスプレイ全体が再表示されるわけではありません)。

---

戻される状態値

SS\$\_NORMAL

正常終了。

SMG\$\_INVDIS\_ID

display-id 引数に誤りがあります。

---

# SMG\$KEYCODE\_TO\_NAME

## Translate a Key Code into a Key Name

Translate a Key Code into a Key Name ルーチンは、キーボードのキーのキー・コードを対応するキー名に変換します。

---

### 形式

SMG\$KEYCODE\_TO\_NAME *key-code ,key-name*

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

key-code

OpenVMS 用法	word_unsigned
データ型	word (unsigned)
アクセス	read only
受け渡し方	by reference

キー名に変換するキー・コードを指定します。key-code 引数は、キー・コードを格納した符号なしワードのアドレスです。

key-code 引数は、SMG\$READ\_COMPOSED\_LINE、SMG\$READ\_KEYSTROKE、SMG\$READ\_STRING、SMG\$READ\_VERIFY ルーチンから戻される word-terminator-code 引数と同じです。

key-name

OpenVMS 用法	char_string
データ型	character string
アクセス	write only
受け渡し方	by descriptor

key-code 引数が変換されるキーの名前を格納する文字列。key-name 引数は、キー名を格納する文字列を示すディスクリプタのアドレスです。key-name 引数はキーの名前です。たとえば、COMMA や PERIOD、KP4 などです。

---

## 説明

SMG\$KEYCODE\_TO\_NAME ルーチンは、キーボードのキーのキー・コードを対応するキー名に変換します。このキー・コードは、SMG\$READ\_KEYSTROKE ルーチンが word-terminator-code 引数に戻すコードと同じです。key-code 引数の形式は SMG\$K\_TRM\_keyname です (たとえば、SMG\$K\_TRM\_DELETE)。

終了文字の値についての詳しい説明は、第 3 章の表 3-1 を参照してください。

---

## 戻される状態値

SS\$NORMAL	正常終了。
SMG\$INVKEYNAM	key-name 引数に誤りがあります。

---

# SMG\$LABEL\_BORDER

## Label a Virtual Display Border

Label a Virtual Display Border ルーチンは、仮想ディスプレイの境界上のラベルを指定します。

---

### 形式

SMG\$LABEL\_BORDER *display-id* [*text*] [*position-code*] [*units*] [*rendition-set*] [*rendition-complement*]  
[*character-set*]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

境界上にラベルが置かれる仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

text

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

境界上の新ラベル。text 引数はラベル・テキストを示すディスクリプタのアドレスです。この引数を省略した場合は、仮想ディスプレイにラベルは表示されません。

## position-code

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ラベル付け位置。position-code 引数は、位置コードを格納した符号なしロングワードのアドレスです。上下左右のどの境界にラベルを付けるかを指定します。この引数に対して指定できる値は次のとおりです。

SMG\$K_TOP	上側境界にラベルを表示します。
SMG\$K_BOTTOM	下側境界にラベルを表示します。
SMG\$K_RIGHT	右側境界にラベルを表示します。
SMG\$K_LEFT	左側境界にラベルを表示します。

この引数を省略した場合は、ラベルは上側境界上に表示されます。

---

**注意**


---

SMG\$K\_RIGHT と SMG\$K\_LEFT は、文字集合が DEC 漢字または Super DEC 漢字文字集合のときは使用できません。

---

## units

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

境界上のラベル付け開始位置。units 引数は文字位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、ラベルは指定した境界上で中寄せされます。

## rendition-set

OpenVMS 用法	mask_longword
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。

SMG\$M\_UNDERLINE      下線を付けた文字を表示します。  
 SMG\$M\_USER1 から      ユーザ定義属性を表示します。  
 SMG\$M\_USER8

#### rendition-complement

OpenVMS 用法    mask\_longword  
 データ型        longword (unsigned)  
 アクセス        read only  
 受け渡し方      by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる属性はすべて、rendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御できます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性を有効にする。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性を無効にする。

#### character-set

OpenVMS 用法    longword\_unsigned  
 データ型        longword (unsigned)  
 アクセス        read only  
 受け渡し方      by reference

境界上にラベルとして置かれるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$LABEL\_BORDER ルーチンは、仮想ディスプレイに表示するラベルのテキストを指定するために使用します。指定した仮想ディスプレイに境界表示属性 (SMG\$M\_BORDER) が設定されていない場合は、この属性が強制的に設定されます。ラベル文字列を指定すると、現在のラベル・テキストと置き換えられます。空 (null) のラベル文字列を指定した場合は、境界にラベルは表示されません。指定した位置からラベル・テキストを境界内に格納できない場合は、このルーチンは SMG\$INVARG を戻します。

position-code 引数と units 引数は、境界内のラベル・テキストの開始位置を指定します。position-code 引数を省略した場合は、上側境界にラベルを表示します。units 引数を省略した場合は、ラベルの置かれる位置に応じて、水平方向または垂直方向にテキストを中寄せするように開始位置を選択します。position-code 引数と units 引数の両方を省略した場合には、テキストは上側境界内で中寄せされます。

垂直方向のラベルを使用できるのは、文字集合が SMG\$C\_ASCII と SMG\$C\_SPEC\_GRAPHICS の場合のみです。

オリジナルの SMG ルーチンは、ASCII 文字集合と DEC 特殊文字集合のみをサポートします。日本語 SMG ではこの他に SMG\$C\_SUPPLEMENTAL, SMG\$C\_KANJI, SMG\$C\_JIS\_KANA, SMG\$C\_SDK が指定できます。

## 戻される状態値

SS\$NORMAL	正常終了。
SMG\$INVARG	引数が誤っています。position-code 引数, units 引数, text 引数を組み合わせて使用した結果, その位置は境界領域の外域になります。
SMG\$INVDIS_ID	display-id 引数が誤っています。

SMG\$\_WRONUMARG

引数の数が誤っています。

## 例

```

1. C+
   C This FORTRAN example program demonstrates the use of SMG$LABEL_BORDER.
   C-

   C+
   C Include the SMG definitions. In particular, we want SMG$_BORDER,
   C SMG$_K_TOP, SMG$_K_BOTTOM, and SMG$_K_RIGHT.
   C-

           IMPLICIT INTEGER (A-Z)
           INCLUDE '($SMGDEF)'

   C+
   C Call SMG$CREATE_VIRTUAL_DISPLAY to create virtual display number 1.
   C Give it a border.
   C-

           ROWS = 4
           COLUMNS = 30

           STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1           (ROWS, COLUMNS, DISPLAY1, SMG$_BORDER)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Call SMG$CREATE_VIRTUAL_DISPLAY to create virtual display number 2.
   C Give it a border.
   C-

           ROWS = 3
           COLUMNS = 30

           STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1           (ROWS, COLUMNS, DISPLAY2, SMG$_BORDER)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Create virtual display number 3. Do NOT give it a border.
   C-

           ROWS = 4
           COLUMNS = 35

           STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1           (ROWS, COLUMNS, DISPLAY3)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Use SMG$CREATE_PASTEBOARD to create the pasteboard.
   C-

           STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

```

## SMG\$LABEL\_BORDER

```
C+
C Call SMG$PUT_CHARS to put data into the virtual displays.
C-
    STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' A bordered virtual display.', 2, 1 )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

    STATUS = SMG$PUT_CHARS ( DISPLAY2,
1      ' A bordered virtual display.', 1, 1 )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

    STATUS = SMG$PUT_CHARS ( DISPLAY3,
1      ' Started as an unbordered display.', 2, 1 )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$LABEL_BORDER to label the virtual display borders.
C-
    STATUS = SMG$LABEL_BORDER ( DISPLAY1, 'Side', SMG$K_RIGHT )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

    STATUS = SMG$LABEL_BORDER ( DISPLAY2, 'LABEL Bottom',
1      SMG$K_BOTTOM, 1 )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

    STATUS = SMG$LABEL_BORDER ( DISPLAY3, 'Forced bordering ',
1      SMG$K_TOP )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual displays.
C-
    STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 2, 10 )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

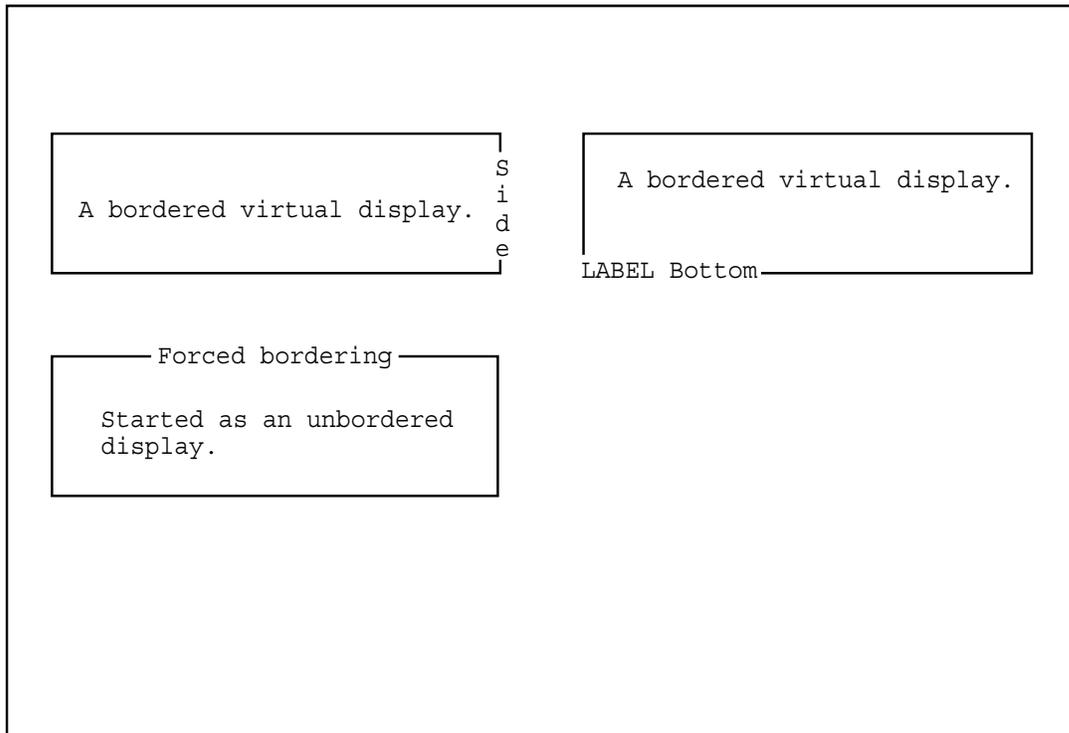
    STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY2, PASTE1, 2, 45 )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

    STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY3, PASTE1, 10, 5 )
    IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

    END
```

このプログラムから生成される出力は図 8-28 に示すとおりです。

図 8-28 SMG\$LABEL\_BORDER ルーチン呼び出すプログラムが生成する出力



ZK-4127-GE

---

## SMG\$LIST\_KEY\_DEFS

### List Key Definitions

List Key Definitions ルーチンは、1 度に 1 つずつ、指定されたキー・テーブル内の指定されたキーに対応する定義 (同値文字列) を戻します。

---

#### 形式

```
SMG$LIST_KEY_DEFS key-table-id ,context [,key-name] [,if-state] [,attributes] [,equivalence-string]
                  [,state-string]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword(unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

key-table-id

OpenVMS 用法	identifier
データ型	longword(unsigned)
アクセス	read only
受け渡し方	by reference

キー定義が検索されるキー・テーブルを指定します。key-table-id 引数は、キー・テーブル識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_KEY\_TABLE ルーチンによって与えられます。

context

OpenVMS 用法	context
データ型	longword(unsigned)
アクセス	modify
受け渡し方	by reference

キー・テーブルから一連のキー定義を検索するための方法を指定します。context 引数は、コンテキスト変数を格納した符号なしロングワードのアドレスです。このルーチンを最初に呼び出す場合には、context 引数を 0 に設定しておかなければなりません。

context 引数は、SMG\$LIST\_KEY\_DEFS ルーチンによって1つずつ加算されます。したがって次の呼び出しでは、次のキー定義が戻されます。

key-name

OpenVMS 用法	char_string
データ型	character string
アクセス	modify
受け渡し方	by descriptor

値のリストを表示するキーを指定します。key-name 引数は、キー名を示すディスクリプタのアドレスです。

if-state

OpenVMS 用法	char_string
データ型	character string
アクセス	write only
受け渡し方	by descriptor

キー・テーブル内の次の定義を条件指定する状態名が書き込まれます。if-state 引数は、状態名が書き込まれる文字列を示すディスクリプタのアドレスです。

attributes

OpenVMS 用法	mask_longword
データ型	longword(unsigned)
アクセス	write only
受け渡し方	by reference

このキー定義の属性。attributes 引数は、キー属性が書き込まれる符号なしロングワードのアドレスです。

属性は次のとおりです。

SMG\$M\_KEY\_NOECHO

このビットが設定されている場合には、このキーが押されたときに、equiv-string 引数がエコー表示されないことを指定します。

このビットがクリアされている場合には、equiv-string 引数はエコー表示されます。SMG\$M\_KEY\_TERMINATE が設定されていない場合には、SMG\$M\_KEY\_NOECHO は無視されます。

SMG\$M\_KEY\_TERMINATE

このビットが設定されている場合には、このキーが押されたときに (if-state 引数の条件指定にしたがって)、入力行は終了し、追加文字が受け付けられないことを指定します。

このビットがクリアされている場合には、追加文字を受け付けることができます。

SMG\$M\_KEY\_LOCK

このビットが設定されており、state-string 引数が指定されている場合には、state-string 引数によって指定された状態名は、定義に state-string 引数を含む後続のキーストロークによって明示的に変更されるまで、現在の状態のまま変更されません。

このビットがクリアされている場合には、state-string 引数によって指定された状態名は、次の定義済みキーストロークに対してのみ有効になります。

## SMG\$LIST\_KEY\_DEFS

SMG\$M\_KEY\_PROTECTED      このビットが設定されている場合には、このキー定義を変更または削除できないことを指定します。  
このビットがクリアされている場合には、キー定義を変更または削除できます。

### equivalence-string

OpenVMS 用法    char\_string  
データ型        character string  
アクセス        write only  
受け渡し方      by descriptor

次のキー定義に対する同値文字列が書き込まれる文字列。equivalence-string 引数は、同値文字列が書き込まれる文字列を示すディスクリプタのアドレスです。

### state-string

OpenVMS 用法    char\_string  
データ型        character string  
アクセス        write only  
受け渡し方      by descriptor

次のキー定義によって設定される新しい状態名が書き込まれる文字列。state-string 引数は、状態名が書き込まれる文字列を示すディスクリプタのアドレスです。このキー定義が状態を設定する場合には、属性フラグ SMG\$M\_KEY\_SETSTATE も設定されます。

---

## 説明

SMG\$LIST\_KEY\_DEFS ルーチンを繰り返し呼び出した場合には、キー・テーブル内のすべての定義を検索できます。これらのキー定義は、SMG\$READ\_COMPOSED\_LINE ルーチンで使用できます。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVKEYNAM	key-name 引数に誤りがあります。
SMG\$_INVKTB_ID	key-table-id 引数に誤りがあります。
SMG\$_NOMOREKEYS	このテーブル内にこれ以上キーは存在しません。
その他	LIB\$SCOPY_DXD から戻された状態値。

---

## SMG\$LIST\_PASTING\_ORDER

### Return Virtual Display Pasting Information

Return Virtual Display Pasting Information ルーチンは、指定されたペーストボードにペーストされている仮想ディスプレイの識別子を戻します。また、オプションとして仮想ディスプレイのペーストボード上の原点位置 (1 行目, 1 カラム目) も戻すことができます。

---

#### 形式

SMG\$LIST\_PASTING\_ORDER *pasteboard-id ,context ,display-id [,pasteboard-row] [,pasteboard-column]*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

仮想ディスプレイがペーストされているペーストボードの識別子。pasteboard-id 引数は、このペーストボードの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

context

OpenVMS 用法	context
データ型	longword (unsigned)
アクセス	modify
受け渡し方	by reference

検索するコンテキスト。context 引数は、このコンテキストを格納する符号なしロングワードのアドレスです。このルーチンを初めて呼び出す場合には、context 引数を 0

に設定しておかなければなりません。SMG\$LIST\_PASTING\_ORDER ルーチンは、context 引数の値を更新します。ペーストされている次のディスプレイ識別子を検索するためには、更新された値を渡して再びルーチンを呼び出します。

display-id

OpenVMS 用法 identifier  
 データ型 longword (unsigned)  
 アクセス write only  
 受け渡し方 by reference

ペーストされている仮想ディスプレイの識別子。display-id 引数は、この仮想ディスプレイの識別子が書き込まれる符号なしロングワードのアドレスです。

pasteboard-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

指定された仮想ディスプレイの 1 行目があるペーストボードの行の行番号。

pasteboard-row 引数は省略可能な引数であり、仮想ディスプレイの最初の行があるペーストボードの行番号が書き込まれる、符号付きロングワードのアドレスです。

pasteboard-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス write only  
 受け渡し方 by reference

指定された仮想ディスプレイの 1 カラム目のあるペーストボードのカラム番号。

pasteboard-column 引数は省略可能な引数であり、仮想ディスプレイの最初のカラムがあるペーストボード・カラム番号が書き込まれる、符号付きロングワードのアドレスです。

---

## 説明

SMG\$LIST\_PASTING\_ORDER ルーチンは、指定されたペーストボードにペーストされている仮想ディスプレイの識別子を戻します。またオプションとして、仮想ディスプレイのペーストボード上の原点位置も戻すことができます。

SMG\$LIST\_PASTING\_ORDER ルーチンは、ペーストされている仮想ディスプレイの中で、最初の (一番下の) 仮想ディスプレイの識別子を戻します。このルーチンを、SMG\$\_NOTPASTED が返されるまで繰り返し呼び出すことによって、連続する仮想ディスプレイの識別子を返します。

このルーチンは、たとえば、多くの仮想ディスプレイを作成した後、アプリケーションにとって重要な仮想ディスプレイのディスプレイ識別子がわからなくなった場合などに使用すると便利です。

---

戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_NOTPASTED	ディスプレイはこれ以上ペーストされていません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$LOAD\_KEY\_DEFS

### Load Key Definitions

Load Key Definitions ルーチンは、キー定義ファイル (DEFINE/KEY コマンド) を指定されたキー・テーブルにロードします。

---

#### 形式

SMG\$LOAD\_KEY\_DEFS *key-table-id* ,*filespec* [,*default-filespec*] [,*flags*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword(unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

key-table-id

OpenVMS 用法	identifier
データ型	longword(unsigned)
アクセス	read only
受け渡し方	by reference

キー定義をロードするキー・テーブルを指定します。key-table-id 引数は、キー・テーブル識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_KEY\_TABLE ルーチンによって与えられます。

filespec

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

DEFINE/KEY コマンドを登録したファイルのファイル指定を格納した文字列。filespec 引数は、ファイル指定を示すディスクリプタのアドレスです。

## default-filespec

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

DEFINE/KEY コマンドを格納したファイルの省略時のファイル指定を格納した文字列。default-filespec 引数は、省略時のファイル指定を示すディスクリプタのアドレスです。この引数を省略した場合には、空文字列が使用されます。

## flags

OpenVMS 用法 mask\_longword  
 データ型 longword(unsigned)  
 アクセス read only  
 受け渡し方 by reference

省略可能なビット・マスクであり、filespec 引数が論理名として取り扱われるかどうかを指定します。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。flags 引数が設定されている場合には、filespec 引数を変換しなければなりません。この操作が不可能な場合には、空文字列が使用されます。

---

 説明

SMG\$LOAD\_KEY\_DEFS ルーチンは、DEFINE/KEY コマンドを登録したファイルをオープンし、読み込み、ファイル内の各コマンド行に対して SMG\$DEFINE\_KEY ルーチンを呼び出します。SMG\$LOAD\_KEY\_DEFS ルーチンを使用する場合には、呼び出しプログラムを DCL コマンド言語インタプリタのもとで実行していなければなりません。このルーチンは、コマンド行を処理するときに検出したエラーを通知します。詳しい説明は、SMG\$DEFINE\_KEY ルーチンを参照してください。

---

 戻される状態値

SS\$NORMAL	正常終了。
SMG\$_FILTOOLON	ファイル指定が長すぎます (最大 255 バイト)。
その他	SMG\$DEFINE_KEY, SOPEN から戻された状態値。

---

# SMG\$LOAD\_VIRTUAL\_DISPLAY

## Load a Virtual Display from a File

Load a Virtual Display from a File ルーチンは、新しい仮想ディスプレイを作成し、SMG\$SAVE\_VIRTUAL\_DISPLAY ルーチンを使用して保存した仮想ディスプレイを新しい仮想ディスプレイにロードします。

---

### 形式

SMG\$LOAD\_VIRTUAL\_DISPLAY *display-id* [,filespec]

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

#### display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by reference

保存されている仮想ディスプレイがロードされる新しい仮想ディスプレイの識別子。display-id 引数は、新しい仮想ディスプレイ識別子が書き込まれる符号なしロングワードのアドレスです。

#### filespec

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

指定された仮想ディスプレイが保存されているファイルのファイル指定を格納した文字列。filespec 引数はファイル指定を格納した文字列です。

filespec 引数を省略した場合には、SMG\$LOAD\_VIRTUAL\_DISPLAY ルーチンは省略時のファイル指定として SMGDISPLY.DAT を検索します。

---

## 説明

SMG\$LOAD\_VIRTUAL\_DISPLAY ルーチンは新しい仮想ディスプレイを作成し、SMG\$SAVE\_VIRTUAL\_DISPLAY ルーチンを使用して保存した仮想ディスプレイを、その新しい仮想ディスプレイにロードします。新しい仮想ディスプレイには、保存されている仮想ディスプレイからテキスト、ビデオ属性、文字集合、およびディスプレイ属性がロードされますが、メニュー、ビューポート、およびサブプロセスの内容はロードされません。新しい仮想ディスプレイはペーストボードにペーストされません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_xxxx	SMG\$CREATE_VIRTUAL_DISPLAY ルーチンから戻された条件値。
RMS\$_xxxx	\$OPEN, \$CONNECT, \$PUT, および \$CLOSE から戻されたエラー。

---

## SMG\$MOVE\_TEXT

### Move Text from One Virtual Display to Another

Move Text from One Virtual Display to Another ルーチンは、1つの仮想ディスプレイから別の仮想ディスプレイにテキスト領域を移動します。長方形の対角線上の2つの角を指定することにより、SMG\$MOVE\_TEXT ルーチンはテキストの幅と高さを判断します。

---

#### 形式

```
SMG$MOVE_TEXT display-id ,top-left-row ,top-left-column ,bottom-right-row ,bottom-right-column
               ,display-id2 [,top-left-row2] [,top-left-column2] [,flags]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

テキストが移動される仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

top-left-row

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

移動するテキスト領域の左上行。top-left-row 引数は、行の位置を格納した符号なしロングワードのアドレスです。

## top-left-column

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

移動するテキスト領域の左上カラム。top-left-column 引数は、カラム位置を格納した符号なしロングワードのアドレスです。

## bottom-right-row

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

移動するテキスト領域の右下行。bottom-right-row 引数は、行の位置を格納した符号なしロングワードのアドレスです。

## bottom-right-column

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

移動するテキスト領域の右下カラム。bottom-right-column 引数は、カラム位置を格納した符号なしロングワードのアドレスです。

## display-id2

OpenVMS 用法 identifier  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

テキスト移動先の仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

## top-left-row2

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

テキストの移動先領域の右上行。top-left-row2 引数は省略可能で、行の位置を格納した符号なしロングワードのアドレスです。この引数を省略した場合は、テキストは現在仮想カーソルが設定されている行に移動されます。

## top-left-column2

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

テキストの移動先領域の左上カラム。top-left-column2 引数は省略可能で、カラム位置を格納した符号なしロングワードのアドレスです。この引数を省略した場合は、テキストは仮想カーソルが現在設定されているカラムに移動されます。

flags

OpenVMS 用法	mask_longword
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

動作指定ビット・マスク。flags 引数は省略可能で、フラグを格納した符号なしロングワードのアドレスです。指定されたテキストを移動する際にどのような動作を実行するかを指定します。この引数に対して指定できる値は次のとおりです。

SMG\$M_TEXT_ONLY	テキストだけを移動し、属性を移動しないことを指定します。
SMG\$M_TEXT_SAVE	移動の後、テキストを消去しないことを指定します。

---

## 説明

SMG\$MOVE\_TEXT ルーチンは、1つの仮想ディスプレイから別の仮想ディスプレイにテキスト領域を移動します。移動の対象となる領域は、top-left-row 引数、top-left-column 引数、bottom-right-row 引数、bottom-right-column 引数で指定します。仮想カーソルの位置は変更されません。

図 8-29 テキスト移動のための引数

Top-left-row  
Top-left-column



Bottom-right-row  
Bottom-right-column

ZK-6451-GE

テキスト領域は、宛先となる仮想ディスプレイの現在の仮想カーソルの行およびカラムに移動されます。別の位置にテキストを移動する場合は、省略可能な top-left-row2 引数と top-left-column2 引数を使用します。

システム設定値では、最初(移動元)の仮想ディスプレイの属性も移動され、テキスト領域を移動した後、テキストは最初の仮想ディスプレイから消去されます。これらのシステム設定値を変更する場合は flags 引数を使用します。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、移動操作の結果として全角文字が半分ずつに分割される場合は、全角文字の分割された各部分は未定義文字になります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVCOL	カラム位置が誤っています。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行の位置が誤っています。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$MOVE\_VIRTUAL\_DISPLAY

## Move Virtual Display

Move Virtual Display ルーチンは、ペーストボード上で仮想ディスプレイを移動しますが、ペースト順序は変更しません。

---

### 形式

```
SMG$MOVE_VIRTUAL_DISPLAY
                                display-id
                                ,pasteboard-id
                                ,pasteboard-row
                                ,pasteboard-column
                                [,top-display-id]
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

移動する仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

移動操作が実行されるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

pasteboard-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

指定された仮想ディスプレイの、新しい位置を示すペーストボードの行番号を指定します。pasteboard-row 引数は、行番号を格納した符号付きロングワードのアドレスです。

pasteboard-column

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

指定された仮想ディスプレイの新しい位置を示すペーストボードのカラム番号を指定します。pasteboard-column 引数は、カラム番号を格納した符号付きロングワードのアドレスです。

top-display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

移動する display-id 引数がペーストされるときに、そのディスプレイの上になる仮想ディスプレイの識別子。top-display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。top-display-id 引数を使用できるのは、display-id 引数によって指定される仮想ディスプレイが現在ペーストされておらず、top-display-id 引数によって指定される仮想ディスプレイがペーストされている場合だけです。

---

## 説明

SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチンは、仮想ディスプレイを現在の位置から指定された位置に移動し、仮想ディスプレイがペーストされている場合には、ペースト順序をそのまま保存します。移動されるディスプレイが現在ペーストされていない場合には、SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチンは、ユーザに対して 2 つのオプションを示します。省略時の設定では、SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチンはペースト順序の 1 番上の、指定された位置に仮想ディスプレイをペーストします。

しかし、top-display-id 引数が指定された場合には、SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチンは移動する仮想ディスプレイを、top-display-id 引数によって指定される仮想ディスプレイの下にペーストします。この場合には、top-display-id 引数によって指定される仮想ディスプレイはすでにペーストされていなければなりません。

ディスプレイを異なるペーストボード間で移動することはできません。それにもかかわらず、pasteboard-id 引数を指定しなければならないのは、特定の仮想ディスプレイを複数のペーストボードにペーストできるからです。

仮想ディスプレイがバッチ処理される場合には、このルーチンを使用してはいけません。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_ILLBATFNC	ディスプレイがバッチ処理されています。 誤操作です。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## 例

SMG\$REPASTE\_VIRTUAL\_DISPLAY ルーチンに示されている FORTRAN プログラムの例を参照してください。

---

## SMG\$NAME\_TO\_KEYCODE

### Translate a Key Name into a Key Code

Translate a Key Name into a Key Code ルーチンは、キーボードのキーのキー名を対応するキー・コードに変換します。

---

#### 形式

SMG\$NAME\_TO\_KEYCODE *key-name* ,*key-code*

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

key-name

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

キー・コードに変換されるキーの名前を格納した文字列。key-name 引数は、キー名を格納した文字列を示すディスクリプタのアドレスです。key-name 引数は単なるキーの名前であり、たとえば、COMMA、PERIOD、KP4 などです。

key-code

OpenVMS 用法	word_unsigned
データ型	word (unsigned)
アクセス	write only
受け渡し方	by reference

key-name 引数を変換した後のキー・コードが書き込まれます。key-code 引数は、キー・コードを格納する符号なしワードのアドレスです。

key-code 引数は、SMG\$READ\_COMPOSED\_LINE、SMG\$READ\_KEYSTROKE、SMG\$READ\_STRING、SMG\$READ\_VERIFY ルーチンによって与えられる word-terminator-code 引数と同じです。

---

## 説明

SMG\$NAME\_TO\_KEYCODE ルーチンは、キーボードのキーのキー名を対応するキー・コードに変換します。このキー・コードは、SMG\$READ\_COMPOSED\_LINE, SMG\$READ\_KEYSTROKE, SMG\$READ\_STRING, SMG\$READ\_VERIFY ルーチンから word-terminator-code 引数に戻されるコードと同じです。このルーチンから戻されるキー・コードの形式は SMG\$K\_TRM\_keyname です。たとえば、キー名として ENTER を指定した場合には、このルーチンから戻されるキー・コードは SMG\$K\_TRM\_ENTER です。

終了文字コードについての詳しい説明は、第 3 章の表 3-1 を参照してください。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVKEYNAM	key-name 引数に誤りがあります。

---

# SMG\$PASTE\_VIRTUAL\_DISPLAY

## Paste Virtual Display

Paste Virtual Display ルーチンは、仮想ディスプレイをペーストボードにペーストします。

---

### 形式

```
SMG$PASTE_VIRTUAL_DISPLAY
                                display-id
                                ,pasteboard-id
                                [,pasteboard-row]
                                [,pasteboard-column]
                                [,top-display-id]
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ペーストされる仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

pasteboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

仮想ディスプレイがペーストされるペーストボードの識別子。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

#### pasteboard-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

指定された仮想ディスプレイの 1 行目が置かれるペーストボードの行。pasteboard-row 引数は、行の位置を格納した符号付きロングワードのアドレスです。

pasteboard-row 引数が省略された場合、仮想ディスプレイがペーストされた最後の行が用いられます。pasteboard-row 引数が省略され、かつ仮想ディスプレイが 1 度もペーストされなかったことがない場合には 1 行目が用いられます。

#### pasteboard-column

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

指定された仮想ディスプレイの 1 カラム目が置かれるペーストボードのカラム。pasteboard-column 引数は、カラム位置を格納した符号付きロングワードのアドレスです。

pasteboard-column 引数が省略された場合、仮想ディスプレイがペーストされた最後のカラムが用いられます。pasteboard-column 引数が省略され、かつ仮想ディスプレイが 1 度もペーストされなかったことがない場合には、1 カラム目が用いられます。

#### top-display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

display-id 引数で指定された仮想ディスプレイをその下にペーストする仮想ディスプレイの識別子。top-display-id 引数は省略可能で、この識別子を格納した符号なしロングワードのアドレスです。この引数で指定された仮想ディスプレイはすでにペーストされていないといけません。

---

## 説明

top-display-id 引数が指定されていない場合、SMG\$PASTE\_VIRTUAL\_DISPLAY はペーストボードに仮想ディスプレイを置き、仮想ディスプレイを表示可能にします。top-display-id 引数が指定されている場合は、SMG\$PASTE\_VIRTUAL\_DISPLAY は top-display-id 引数で指定された仮想ディスプレイの下に display-id 引数

## SMG\$PASTE\_VIRTUAL\_DISPLAY

で指定された仮想ディスプレイをペーストします。この場合、top-display-id 引数で指定された仮想ディスプレイはすでにペーストされていなくてはなりません。

仮想ディスプレイがバッチ処理されている場合はこのルーチンを使用してはいけません。

---

### 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_ILLBATFNC	仮想ディスプレイはバッチ処理されています。誤操作です。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVPAS_ID	pasteboard-id 引数が誤っています。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$POP\_VIRTUAL\_DISPLAY

## Delete a Series of Virtual Displays

Delete a Series of Virtual Displays ルーチンは、指定された仮想ディスプレイを削除し、さらに指定されたペーストボード上の指定された仮想ディスプレイの上にペーストされているすべてのディスプレイを削除します。

---

### 形式

```
SMG$POP_VIRTUAL_DISPLAY display-id ,pasteboard-id
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

*display-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

削除する一番下の (最初の) 仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。このディスプレイよりペースト順序が上のすべてのディスプレイ (つまり、指定されたディスプレイより後にペーストされたすべてのディスプレイ) も削除されます。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

*pasteboard-id*

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ディスプレイの削除操作が実行されるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

---

## 説明

SMG\$POP\_VIRTUAL\_DISPLAY ルーチンは、指定されたペーストボードから 1 つ以上のディスプレイを削除します (単にペーストを取り消すだけではありません)。この場合、指定されたディスプレイから削除を開始し、その後、そのディスプレイよりペースト順序が上にあるすべてのディスプレイも削除します (つまり、指定されたディスプレイの後でペーストされたすべてのディスプレイを削除します)。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_INVPAS_ID	pasteboard-id 引数に誤りがあります。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$PUT\_CHARS

## Write Characters to a Virtual Display

Write Characters to a Virtual Display ルーチンは、仮想ディスプレイ内に指定されたテキストを書き込みます。

---

### 形式

```
SMG$PUT_CHARS display-id ,text [,start-row] [,start-column] [,flags] [,rendition-set]
              [,rendition-complement] [,character-set]
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

文字が書き込まれる仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

text

VMS 用法:	char_string
データ型:	character string
アクセス:	read only
受け渡し方:	by descriptor

仮想ディスプレイへ書き込まれる文字。text 引数は、文字列を示すディスクリプタのアドレスです。

## start-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み開始行。start-row 引数は、行の位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、出力は現在の行から開始されます。

## start-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み開始カラム。start-column 引数は、カラム位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、出力は現在のカラムから開始されます。

全角文字の右半分からカラムが始まる場合は、その文字の左半分は未定義文字になります。

## flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み前の動作を指定する省略可能なビット・マスク。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。この引数に対して指定できる値は次のとおりです。

0	行を消去しないことを指定します (省略時の設定値)。
SMG\$M_ERASE_LINE	行全体を消去します。
SMG\$M_ERASE_TO_EOL	行の残りの部分を消去します。

## rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。

SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## rendition-complement

OpenVMS 用法	mask_longword
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる引数はすべて、rendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御できます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性を有効にする。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性を無効にする。

## character-set

OpenVMS 用法	longword_unsigned
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

挿入されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$PUT\_CHARS ルーチンはテキストを指定された仮想ディスプレイに書き込み、既存のテキストを置換します。このルーチンは 1 行だけを書き込みます。既存のテキストを保存して、新しいテキストを書き込む場合は、SMG\$INSERT\_CHARS ルーチンを使用します。

flags 引数を省略した場合は、SMG\$PUT\_CHARS ルーチンは、新しいテキストが書き込まれる文字位置だけを変更します。しかし、flags 引数に対して SMG\$M\_ERASE\_LINE を指定することにより、新しいテキストを書き込む前に行を消去することができます。また、flags 引数に SMG\$M\_ERASE\_TO\_EOL を指定することにより、テキストを書き込んだ後で残りの行を消去することも可能です。カーソルは、最後に書き込まれたテキストのすぐ後の文字位置に設定されます。仮想ディスプレイからあふれた分の文字は切り捨てられますが、仮想ディスプレイに対して SMG\$M\_TRUNC\_ICON 属性が設定されている場合には、SMG\$PUT\_CHARS ルーチンが行の一部を切り捨てたことを示すアイコンを出力します。

オリジナルの SMG ルーチンは、ASCII 文字集合と DEC 特殊文字集合のみをサポートします。日本語 SMG ではこの他に SMG\$C\_SUPPLEMENTAL、SMG\$C\_KANJI、SMG\$C\_JIS\_KANA、SMG\$C\_SDK が指定できます。

DEC 漢字または Super DEC 漢字文字集合を使用する際に、全角文字の右半分から書き込み操作が始まる場合は、その文字の左半分は未定義文字になります。また、全角文字の左半分で書き込み操作が終了する場合は、その文字の右半分は未定義文字になります。さらに、行の一部が切り捨てられる際に、全角文字の右半分から切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$_INVCOL	カラム位置が誤っています。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行の位置が誤っています。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

 例

```

1. C+
   C This FORTRAN example program demonstrates the use of
   C SMG$PUT_CHARS.
   C-
           INTEGER SMG$CREATE_VIRTUAL_DISPLAY, SMG$CREATE_PASTEBOARD
           INTEGER SMG$PASTE_VIRTUAL_DISPLAY, SMG$PUT_CHARS
           INTEGER DISPLAY1, PASTE1, ROWS, COLUMNS, BORDER, STATUS
   C+
   C Create the virtual display. To give it a border, set BORDER = 1.
   C No border would be BORDER = 0.
   C-
           ROWS = 7
           COLUMNS = 50
           BORDER = 1
           STATUS = SMG$CREATE_VIRTUAL_DISPLAY
   1           (ROWS, COLUMNS, DISPLAY1, BORDER)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
   C+
   C Create the pasteboard.
   C-
           STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
   C+
   C Put data in the virtual display.
   C-
           STATUS = SMG$PUT_CHARS ( DISPLAY1,
   1           ' This virtual display has 7 rows and 50 columns.', 2, 1 )
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
           STATUS = SMG$PUT_CHARS ( DISPLAY1,
   1           ' This is a bordered virtual display.', 4, 1 )
           IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

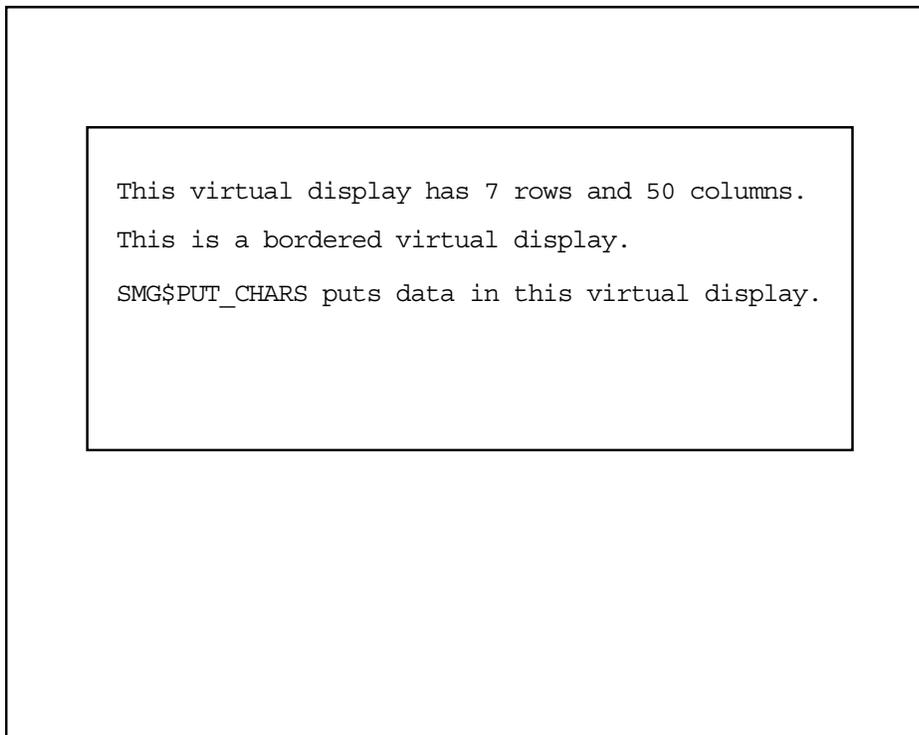
```

## SMG\$PUT\_CHARS

```
STATUS = SMG$PUT_CHARS ( DISPLAY1,  
1      ' SMG$PUT_CHARS puts data in this virtual display.', 6, 1 )  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
C+  
C Paste the virtual display.  
C-  
  
STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15 )  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
END
```

この FORTRAN プログラムによって生成される出力は、図 8-30 に示すとおりです。

図 8-30 SMG\$PUT\_CHARS ルーチンと呼ぶ FORTRAN プログラムによって生成される出力



ZK-4116-GE

<html\_segment>(seg\_smg)

---

# SMG\$PUT\_CHARS\_HIGHWIDE

## Write Double-Height Double-Width Characters

The Write Double-Height Double-Width Characters ルーチンは仮想ディスプレイに 4 倍角の文字を書き込みます。

---

### 形式

```
SMG$PUT_CHARS_HIGHWIDE display-id ,text [start-row] [start-column] [rendition-set]
                        [rendition-complement] [character-set]
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

4 倍角の文字が書き込まれる仮想ディスプレイを指定します。display-id 引数は、この仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

text

VMS 用法:	char_string
データ型:	character string
アクセス:	read only
受け渡し方:	by descriptor

仮想ディスプレイへ書き込まれる文字。text 引数は、文字列を示すディスクリプタのアドレスです。

## start-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み開始行。start-row 引数は、開始行番号を格納した符号付きロングワードのアドレスです。start-row 引数が省略されている場合、あるいはゼロに等しい場合は出力は現在の行から始まります。

## start-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み開始カラム。start-column 引数は、開始カラム番号を格納した符号付きロングワードのアドレスです。start-column 引数が省略されている場合、あるいはゼロに等しい場合は、出力は現在のカラムから始まります。全角文字の右半分からカラムが始まる場合は、その文字の左半分は未定義文字になります。

## rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスク内の各属性を設定すると、ディスプレイ内の対応する属性が設定されます。rendition-set 引数を使用して以下の属性を指定できます。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想ディスプレイの現在の省略時の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## rendition-complement

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能な、ロングワード・ビット・マスクのアドレスであり、このロングワード・ビット・マスク内の各属性を設定する

と、ディスプレイ内の対応する属性が補足されます。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数で補足できます。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの 2 つの引数を組み合わせて使用すれば、1 回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性をオフに設定する。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

挿入されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$PUT\_CHARS\_HIGHWIDE ルーチンは指定された仮想ディスプレイに 4 倍角の文字を書き込みます。対応するペーストボードの行は、半角/全角文字と 4 倍角文字

を混在して格納することはできません。その行が半角/全角文字を格納していた場合には、その部分は空白になります。テキストが1行よりも短い場合は、空白が埋め込まれます。仮想カーソルは、最後に書き込まれたテキストのすぐ後の文字位置に設定されます。

ターミナルが4倍角文字をサポートしていない場合は、同じ文字が2行にわたって表示されます。

仮想ディスプレイからあふれた分の文字は切り捨てられますが、仮想ディスプレイに対してSMG\$M\_TRUNC\_ICON属性が設定されている場合には、SMG\$PUT\_CHARS\_HIGHWIDEルーチンが行の一部を切り捨てたことを示すアイコンを出力します。

DEC 漢字または Super DEC 漢字文字集合を使用する際に、全角文字の右半分から書き込み操作が始まる場合は、その文字の左半分は未定義文字になります。また、全角文字の左半分で書き込み操作が終了する場合は、その文字の右半分は未定義文字になります。さらに、行の一部が切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$_INVARG	引数が誤っています。
SMG\$_INVCOL	カラム番号が誤っています。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行番号が誤っています。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

# SMG\$PUT\_CHARS\_MULTI

## Put Text with Multiple Renditions to Display

The Put Text with Multiple Renditions to Display ルーチンは、仮想ディスプレイに複数のレンディションで、テキストを書き込みます。

---

### 形式

```
SMG$PUT_CHARS_MULTI display-id ,text [,start-row] [,start-column] [,flags] [,rendition-string]
                    [,rendition-complement] [,character-set]
```

---

### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

影響を受ける仮想ディスプレイの識別子。display-id 引数は、この仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

text

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

出力されるテキスト。text 引数は、テキストを示すディスクリプタのアドレスです。

start-row

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only
受け渡し方	by reference

文字書き込み開始行。start-row 引数は、行の位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、出力は現在の行から開始されます。

start-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み開始カラム。start-column 引数は、カラム位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、出力は現在のカラムから開始されます。全角文字の右半分からカラムが始まる場合は、その文字の左半分は未定義文字になります。

flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み前の動作を指定する省略可能なビット・マスク。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。この引数に対して指定できる値は次のとおりです。

0	行を消去しないことを指定します (省略時の設定値)。
SMG\$M_ERASE_LINE	行全体を消去します。
SMG\$M_ERASE_TO_EOL	行の残りの部分を消去します。

rendition-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

ビデオ属性を制御するビット・マスク文字列 (省略可能)。rendition-string 引数は、ビット・マスク文字列を示すディスクリプタのアドレスです。仮想ディスプレイ上のテキスト文字列のカラムに対応する属性を指定することができます。この rendition-string 引数で指定できる属性は以下のとおりです。

全角文字の左半分と右半分に違う属性が指定された場合は、全角文字の左半分に指定された属性がその全角文字に適用されます。

SMG\$M_BLINK	点滅文字を指示します。
SMG\$M_BOLD	通常より明るい輝度の文字を指定します。
SMG\$M_REVERSE	反転表示文字を指示します。つまり、仮想ディスプレイの現在の属性の反対の状態で文字を表示しています。
SMG\$M_UNDERLINE	下線付き文字を指示します。

SMG\$M\_INVISIBLE 表示されない文字を指定します。つまり、文字は仮想ディスプレイ内に存在しますが、ペーストボードには表示されていません。

#### rendition-complement

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

ビデオ属性を制御するビット・マスク文字列 (省略可能)。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、仮想ディスプレイ内の対応するカラム属性は補足されます。全角文字の左半分と右半分に違う補足属性が指定された場合は、全角文字の左半分に指定された補足属性がその全角文字に適用されます。

rendition-string 引数と rendition-complement 引数の両方に、同じビットを指定した場合は、最初に rendition-string 引数が評価され、その後 rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御することができます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性をオフに設定する。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

挿入されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$PUT\_CHARS\_MULTI ルーチンは、仮想ディスプレイに多様なレンディションでテキストを書き込むことを可能にします。SMG\$PUT\_CHARS\_MULTI ルーチンは、指定された位置にある既存のテキストを上書きします。既存のテキストを保存し、新しいテキストを書き込む場合には、SMG\$INSERT\_CHARS ルーチンを使用してください。

省略時設定では、SMG\$PUT\_CHARS\_MULTI ルーチンは、新しいテキストが書かれている文字の位置のみを修正します。しかし、flags 引数に SMG\$M\_ERASE\_LINE を指定することによって、新しいテキストが書かれる前に行を消去することができます。また、flags 引数に SMG\$M\_ERASE\_TO\_EOL を指定することによって新しいテキストが書かれた後に、残りの行を消去することができます。カーソルは、最後のテキストが書かれたすぐ後の位置に残ります。

このルーチンは、1 行のみに書き込むということに注意してください。つまり、仮想ディスプレイからあふれた分の文字は切り捨てられます。しかし、仮想ディスプレイに対して SMG\$M\_TRUNC\_ICON 属性が設定されている場合には、SMG\$PUT\_CHARS\_MULTI ルーチンが行の一部を切り捨てたことを示すアイコンを出力します。

DEC 漢字または Super DEC 漢字文字集合を使用する際に、全角文字の右半分から書き込み操作が始まる場合は、その文字の左半分は未定義文字になります。また、全角文字の左半分で書き込み操作が終了する場合は、その文字の右半分は未定義文字になります。さらに、行の一部が切り捨てられる際に、全角文字の右半分から切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$_INVARG	引数が誤っています。
SMG\$_INVCOL	カラム番号の指定が誤っています。
SMG\$_INVROW	行番号の指定が誤っています。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$PUT\_CHARS\_WIDE

### Write Double-Width Characters

The Write Double-Width Characters ルーチンは、仮想ディスプレイに倍角の文字を書き込みます。

---

#### 形式

SMG\$PUT\_CHARS\_WIDE *display-id* ,*text* [*start-row*] [*start-column*] [*rendition-set*]  
[*rendition-complement*] [*character-set*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

倍角の文字が書き込まれる仮想ディスプレイを指定します。display-id 引数は仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられません。

text

VMS 用法:	char_string
データ型:	character string
アクセス:	read only
受け渡し方:	by descriptor

仮想ディスプレイに書き込まれる文字。text 引数は、テキストを示すディスクリプタのアドレスです。

## start-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み開始行。start-row 引数は、行の位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、出力は現在の行から開始されます。

## start-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

文字書き込み開始カラム。start-column 引数は、カラム位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、出力は現在のカラムから開始されます。全角文字の右半分からカラムが始まる場合は、その文字の左半分は未定義文字になります。

## rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## rendition-complement

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる引数はすべて、rendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御できます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性を有効にする。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性を無効にする。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

挿入されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$PUT\_CHARS\_WIDE ルーチンは、倍角のテキストを指定された仮想ディスプレイに書き込みます。既存のテキストがある場合には、そこに上書きします。1 行に半角/全角文字と倍角文字を混在して格納することはできません。その行に半角/全角文字が書かれていた場合には、空白に書き直されます。仮想カーソルは、テキストが書かれる最初の文字の位置におかれます。

仮想ディスプレイに対して SMG\$M\_TRUNC\_ICON 属性が設定されている場合には、SMG\$PUT\_CHAR\_WIDE ルーチンが行の一部を切り捨てたことを示すアイコンを出力します。

DEC 漢字または Super DEC 漢字文字集合を使用する際に、全角文字の右半分から書き込み操作が始まる場合は、その文字の左半分は未定義文字になります。また、全角文字の左半分で書き込み操作が終了する場合は、その文字の右半分は未定義文字になります。さらに、行の一部が切り捨てられる際に、全角文字の右半分から切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$_INVARG	引数が誤っています。
SMG\$_INVCOL	カラム位置が誤っています。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_INVROW	行の位置が誤っています。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$PUT\_HELP\_TEXT

### Output Help Text to the Display

The Output Help Text to the Display ルーチンは、仮想ディスプレイ上に指定されたトピックのヘルプ・テキストを出力します。

---

#### 形式

```
SMG$PUT_HELP_TEXT display-id [,keyboard-id] [,help-topic] [,help-library] [,rendition-set]
                  [,rendition-complement] [,character-set]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

ヘルプ・テキストが書き込まれる仮想ディスプレイを指定します。display-id 引数は仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

仮想ディスプレイはペーストされていないと、重ねたりバッチ処理されたりすることはできないので注意してください。

keyboard-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

入力するために使用される仮想キーボードの識別子 (省略可能)。keyboard-id 引数は、仮想キーボード識別子を格納した符号なしロングワードのアドレスです。keyboard-id 引数が指定されない場合は、入力できません。

## help-topic

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

省略可能なヘルプ・トピック。help-topic 引数は、ヘルプ・トピック文字列を示すディスクリプタのアドレスです。

## help-library

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

省略可能なヘルプ・ライブラリの名前。help-library 引数は、ヘルプ・ライブラリの名前を示すディスクリプタのアドレスです。省略時の設定値は SYSSHELP:HELPLIB.HLB です。

## rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このロングワード・ビット・マスクの各属性を設定すると、ディスプレイ内の対応する属性が設定されます。rendition-set 引数を使用して以下の属性を指定できます。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想ディスプレイの現在の省略時の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## rendition-complement

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数によって補足できます。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの2つの引数を組み合わせて使用すれば、1回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性をオフに設定する。

character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

出力されるヘルプ・テキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、仮想ディスプレイの文字集合が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$PUT\_HELP\_TEXT ルーチンは、指定された仮想ディスプレイに指定されたトピックのヘルプ・テキストを戻し、出力します。テキストは仮想スクロール領域に出力されます。keyboard-id 引数を指定した場合には、仮想スクロール領域に最後の行が出力されたときに、入力を要求するプロンプトが表示されます。このプロンプトに対して、`[Return]`キーを押せばそのまま継続して表示され、新しいトピックを入力すれば、それに対するヘルプ・テキストを受け取ることができます。

display-id 引数によって指定される仮想ディスプレイは、バッチ処理されたりビューポートを格納したりできません。keyboard-id 引数が指定された場合には、仮想ディスプレイは重ねることはできません。また、仮想ディスプレイは少なくとも 3 行なくてはなりません。

Super DEC 漢字、DEC 漢字文字、JIS カタカナ文字、ASCII 文字以外で構成されたヘルプ・トピック文字列はサポートしていません。

## 戻される状態値

SS\$NORMAL                      正常終了。

SMG\$_ILLBATFNC	仮想ディスプレイ、ペーストボードがバッチ処理されています。
SMG\$_INVARG	引数が誤っています。
SMG\$_INVDIS_ID	display-id 引数に誤りがあります。
SMG\$_NOTPASTED	display-id 引数によって指定された仮想ディスプレイがペーストされていません。
LIB\$_xxxx	LIB\$FIND_IMAGE_SYMBOL から戻された状態値。
LBR\$_xxxx	LBR\$OUTPUT_HELP から戻された状態値。
SMG\$_xxxx	SMG\$SET_CURSOR_ABS, SMG\$CHECK_FOR_OCCLUSION から戻された状態値。

---

## SMG\$PUT\_LINE

### Write Line to Virtual Display

Write Line to Virtual Display ルーチンは、1 行のテキストを仮想ディスプレイに書き込みます。

---

#### 形式

```
SMG$PUT_LINE display-id ,text [line-advance] [rendition-set] [rendition-complement] [flags]
             [character-set] [direction]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

行が書き込まれる仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

text

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

書き込まれる文字。text 引数は、文字列を示すディスクリプタのアドレスです。

line-advance

OpenVMS 用法	longword_signed
データ型	longword (signed)

アクセス read only  
受け渡し方 by reference

行書き込み後のスクロール行数。line-advance 引数は、進める行数を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、1 が設定されます。0 を指定すると、SMG\$PUT\_LINE ルーチンは既存のテキストに上書きします。

#### rendition-set

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定された属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

#### rendition-complement

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる引数はすべて、rendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御できます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性を有効にする。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性を無効にする。

#### flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

動作指定ビット・マスク。flags 引数は省略可能で、フラグを格納した符号なしロングワードのアドレスです。テキストが 1 行で格納できない場合に実行する動作を指定します。この引数に対して指定できる値は次のとおりです。

0 自動改行しないことを指定します (省略時の設定値)。  
 SMG\$M\_WRAP\_CHAR 行の最後の文字で自動改行します。  
 SMG\$M\_WRAP\_WORD 行の最後の空白で自動改行します。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

出力されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

#### direction

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

スクロール方向。direction 引数は、方向コードを格納したロングワード・ビット・マスクのアドレスです。この引数に対して指定できる値は SMG\$M\_UP と SMG\$M\_DOWN です。この引数を省略した場合は、SMG\$M\_UP が設定されます。

---

## 説明

SMG\$PUT\_LINE ルーチンは、現在の仮想カーソルの位置からテキストを仮想ディスプレイに書き込みます。テキストが 1 番下または 1 番上の行に到達した後 (スクロールの方向に応じて異なる) このルーチン呼び出すと、仮想ディスプレイはスクロールします。指定されたテキストが 1 行より短い場合は、必要に応じて空白が埋め込まれます。

flags 引数で自動改行を指定した場合は、line-advance 引数で指定した行数だけスクロールされ、あふれた文字を次の行に挿入するための空間が作成されます。次の行はスクロールの方向 (direction 引数) によって決定されます。flags 引数で自動改行を指定していない場合は、あふれた文字は切り捨てられます。

SMG\$PUT\_LINE ルーチン呼び出した後、仮想カーソルの位置は出力の対象となる次の行の 1 カラム目に設定されます。出力の対象となる次の行は、line-advance 引数と direction 引数によって決定されます。SMG\$PUT\_LINE ルーチンを続けて呼び出しても、前の出力の上に新しい出力が重ならないように、line-advance 引数のシステム設定値は 1 になっています。仮想ディスプレイ上にテキスト行を書き込むための日本語 SMG ルーチンとしては、他に SMG\$INSERT\_LINE ルーチンがあります。

オリジナルの SMG ルーチンは、ASCII 文字集合と DEC 特殊文字集合のみをサポートします。日本語 SMG ではこの他に SMG\$C\_SUPPLEMENTAL, SMG\$C\_KANJI, SMG\$C\_JIS\_KANA, SMG\$C\_SDK が指定できます。

DEC 漢字または Super DEC 漢字文字集合を使用している際に、現在の仮想カーソルの位置が全角文字の右半分である場合は、その文字の左半分は未定義文字になります。また、あふれる文字が切り捨てられる際に、全角文字の右半分から切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

## 戻される状態値

SS\$NORMAL	正常終了。
LIB\$INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$INVDIS_ID	display-id 引数が誤っています。
SMG\$WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$WRONUMARG	引数の数が誤っています。

---

**例**

```
1. C+
   C This FORTRAN example program demonstrates the use of SMG$PUT_LINE.
   C-

   C+
   C Include the SMG definitions. In particular, we want SMG$M_BORDER and
   C SMG$M_UNDERLINE.
   C-

       IMPLICIT INTEGER (A-Z)
       INCLUDE '($SMGDEF)'
       CHARACTER*30 TEXT(3)

   C+
   C Create a virtual display with a border.
   C-

       ROWS = 7
       COLUMNS = 50

       STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Create the pasteboard.
   C-

       STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Put data in the virtual display.
   C-

       TEXT(1) = 'This virtual display has 7'
       TEXT(2) = 'rows and 50 columns.'
       TEXT(3) = 'Text entered by SMG$PUT_LINE.'

   C+
   C After the first line of text is printed, call SMG$PUT_LINE to
   C advance two lines.
   C-

       STATUS = SMG$PUT_LINE ( DISPLAY1, TEXT(1), 2 )
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Now, use SMG$PUT_LINE to underline the next line of text.
   C Notice that 30 characters are being underlined. Advance 1
   C line of text after displaying the line.
   C-

       STATUS = SMG$PUT_LINE ( DISPLAY1, TEXT(2), 1, SMG$M_UNDERLINE )
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
```

```

C+
C Display the third line of text.
C-
      STATUS = SMG$PUT_LINE ( DISPLAY1, TEXT(3) )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

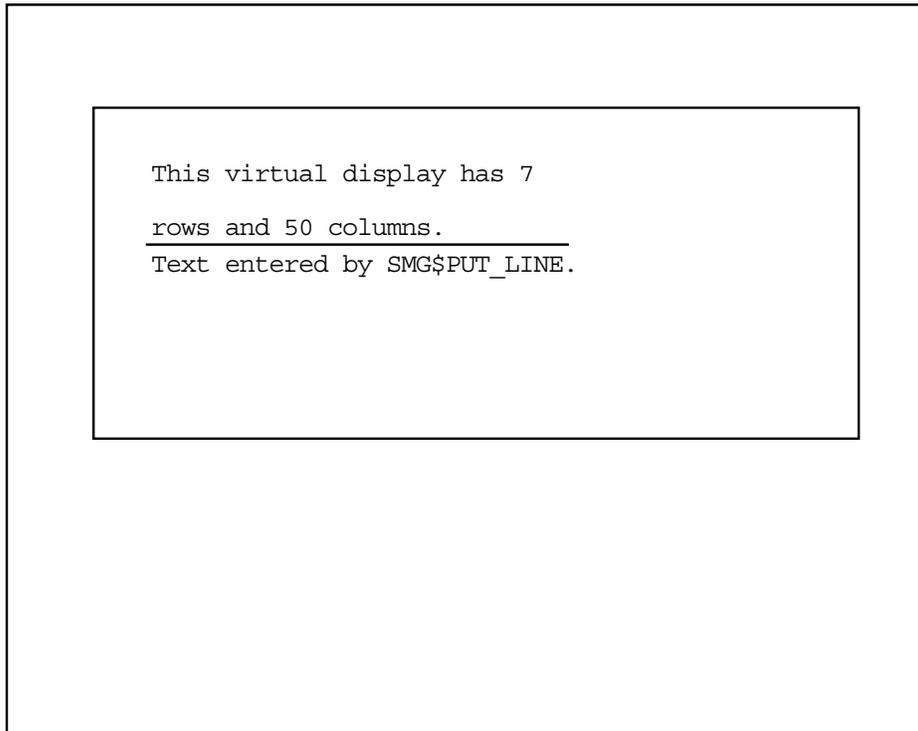
C+
C Paste the virtual display.
C-
      STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      END

```

この FORTRAN プログラムによって生成される出力は図 8-31 に示すとおりです。

図 8-31 SMG\$PUT\_LINE ルーチン呼び出す FORTRAN プログラムによって生成される出力



ZK-4135-GE

次のプログラムは、SMG\$PUT\_LINE ルーチンの新しい direction 引数の使用を示しています。この引数を用いるため、SMG\$PUT\_WITH\_SCROLL ルーチンは必要なくなりました。

## SMG\$PUT\_LINE

```
2. C+
C This FORTRAN example program demonstrates the use of DIRECTION
C parameter in the SMG$M_UNDERLINE routine.
C
C The DIRECTION parameter in SMG$PUT_LINE makes SMG$PUT_WITH_SCROLL
C an obsolete routine. This example is the same as the SMG$PUT_WITH_SCROLL
C routine, except that the calls to SMG$PUT_WITH_SCROLL have been
C replaced by calls to SMG$PUT_LINE.
C-
        INCLUDE '$SMGDEF'
        IMPLICIT INTEGER*4 (A-Z)

C+
C Call SMG$CREATE_PASTEBOARD to establish the terminal screen
C as a pasteboard.
C-
        STATUS = SMG$CREATE_PASTEBOARD (NEW_PID)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Using SMG$CREATE_VIRTUAL_DISPLAY, establish a virtual display region.
C-
        STATUS = SMG$CREATE_VIRTUAL_DISPLAY (5,80,DISPLAY_ID)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Paste the virtual display to the screen, starting at
C row 10, column 15, by calling SMG$CREATE_VIRTUAL_DISPLAY.
C-
        STATUS = SMG$CREATE_VIRTUAL_DISPLAY (DISPLAY_ID,NEW_PID,10,15)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Define a scrolling region through a call to
C SMG$SET_DISPLAY_SCROLL_REGION.
C-
        STATUS = SMG$SET_DISPLAY_SCROLL_REGION (DISPLAY_ID,1,5)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

C+
C Call SMG$PUT_LINE and SMG$ERASE_LINE to write three
C scrolling lines to the screen. The first line will be underlined.
C the second blinking, and the third in reverse video.
C-
        DO I = 1,10
            IF ((I/2) + (I/2) .EQ. I) THEN
                DIR = SMG$M_UP
            ERASE
                DIR = SMG$M_DOWN
            ENDIF
        STATUS = SMG$PUT_LINE (DISPLAY_ID,
            1 'This line is underlined',,SMG$M_UNDERLINE,,,DIR)
        IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
```

```
STATUS = SMG$ERASE_LINE(DISPLAY_ID)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

STATUS = SMG$PUT_LINE (DISPLAY_ID,'This line is blinking' , ,
1      SMG$M_BLINK,,,,DIR)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

STATUS = SMG$ERASE_LINE(DISPLAY_ID)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

STATUS = SMG$PUT_LINE (DISPLAY_ID,'This line is reverse
1      video' , ,SMG$M_REVERSE,,,,DIR)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))

STATUS = SMG$ERASE_LINE(DISPLAY_ID)
IF (.NOT. STATUS) CALL LIB$STOP(%VAL(STATUS))
ENDDO
END
```

---

## SMG\$PUT\_LINE\_HIGHWIDE

### Write Double-Height and Double-Width Line

The Write Double-Height and Double-Width Line ルーチンは 1 行のテキストを 4 倍角の文字で書き込みます。

---

#### 形式

SMG\$PUT\_LINE\_HIGHWIDE *display-id* ,*text* [*line-advance*] [*rendition-set*] [*rendition-complement*]  
 [*flags*] [*character-set*]

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

仮想ディスプレイ識別子。display-id 引数は仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

text

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

仮想ディスプレイへ書き込まれる文字。text 引数は、文字列を示すディスクリプタのアドレスです。

line-advance

OpenVMS 用法	longword_signed
データ型	longword (signed)
アクセス	read only

受け渡し方 by reference

行書き込み後のスクロール行数。line-advance 引数は、進める行数を格納した符号付きロングワードのアドレスです。この引数は省略可能です。

#### rendition-set

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定された属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

#### rendition-complement

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる引数はすべて、rendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御できます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。

設定	補足	動作
1	0	属性を有効にする。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性を無効にする。

#### flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

動作指定ビット・マスク。flags 引数は省略可能で、フラグを格納した符号なしロングワードのアドレスです。テキストが 1 行で格納できない場合に実行する動作を指定します。この引数に対して指定できる値は次のとおりです。

0 自動改行しないことを指定します  
 (省略時の設定値)。  
 SMG\$M\_WRAP\_CHAR 行の最後の文字で自動改行します。  
 SMG\$M\_WRAP\_WORD 行の最後の空白で自動改行します。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

挿入されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$PUT\_LINE\_HIGHWIDE ルーチンは、仮想ディスプレイに、4 倍角文字のテキスト行を書き込むために使用されます。このルーチンは、現在の仮想カーソルの位

置から行末まで書き込みます。テキストが1行よりも短い場合は、空白が埋め込まれます。

仮想ディスプレイの右側境界を越えたテキストは、flags 引数に制御されます。flags 引数で自動改行を指定した場合は、line-advance 引数で指定した行数だけスクロールされ、あふれた文字を次の行に挿入するための空間が作成されます。flags 引数で自動改行を指定していない場合は、あふれた文字は切り捨てられます。

SMG\$PUT\_LINE\_HIGHWIDE ルーチン呼び出した後で、仮想カーソルの位置は出力の対象となる、次の行の1カラム目に設定されます。出力の対象となる次の行は、line-advance 引数によって決定されます。line-advance 引数のシステム設定値は2になっていますので、引き続き SMG\$PUT\_LINE\_HIGHWIDE ルーチン呼び出しても、上書きされることはありません。

DEC 漢字または Super DEC 漢字文字集合を使用する際に、全角文字の右半分から書き込み操作が始まる場合は、その文字の左半分は未定義文字になります。また、行の一部が切り捨てられる際に、全角文字の右半分から切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$_INVARG	引数が誤っています。
SMG\$_WILUSERMS	ペーストボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$PUT\_LINE\_MULTI

### Write Line with Multiple Renditions to Display

The Write Line with Multiple Renditions to Display ルーチンは、複数の属性を含む行を仮想ディスプレイに書き込みます。カーソル移動がともなう場合もあります。

---

#### 形式

```
SMG$PUT_LINE_MULTI display-id ,text ,rendition-string [,rendition-complement] [,line-advance] [,flags]
                    [,direction] [,character-set]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

行が書き込まれる仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

text

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

仮想ディスプレイに書き込まれる文字。text 引数は、文字列を示すディスクリプタのアドレスです。

rendition-string

OpenVMS 用法	char_string
データ型	character string
アクセス	read only

受け渡し方 by descriptor

ビデオ属性を制御するビット・マスク文字列 (省略可能)。rendition-string 引数は、ビット・マスク文字列を示すディスクリプタのアドレスです。仮想ディスプレイ上のテキスト文字列のカラムに対応する属性を指定することができます。この rendition-string 引数で指定できる属性は以下のとおりです。

全角文字の左半分と右半分に違う属性が指定された場合は、全角文字の左半分に指定された属性がその全角文字に適用されます。

SMG\$M_BLINK	点滅文字を指示します。
SMG\$M_BOLD	通常より明るい輝度の文字を指定します。
SMG\$M_REVERSE	反転表示文字を指示します。つまり、仮想ディスプレイの現在の属性の反対の状態では文字を表示しています。
SMG\$M_UNDERLINE	下線付き文字を指示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、文字は仮想ディスプレイ内に存在しますが、ペーストボードには表示されていません。

rendition-complement

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

ビデオ属性を制御するビット・マスク文字列 (省略可能)。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。ビット・マスク内に属性が設定されている場合には、仮想ディスプレイ内の対応するカラム属性が補足されます。全角文字の左半分と右半分に違う補足属性が指定された場合は、全角文字の 1 バイト目に指定された補足属性がその全角文字に適用されます。

同じビットが rendition-set 引数と rendition-complement 引数の両方に指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの 2 つの引数を組み合わせて使用すれば、1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御できます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性をオンに設定する。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性をオフに設定する。

line-advance

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

行書き込み後のスクロール行数。line-advance 引数は、進める行数を格納した符号付きロングワードのアドレスです。

#### flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

テキストが 1 行に格納できない場合に、実行する動作を指定するビット・マスク (省略可能)。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。この引数に対して指定できる値は次のとおりです。

0 自動改行しないことを指定します (省略時の設定値)。  
 SMG\$M\_WRAP\_CHAR 行の最後の文字で自動改行します。  
 SMG\$M\_WRAP\_WORD 行の最後の空白で自動改行します。

#### direction

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

スクロールの方向を指定するビット・マスク (省略可能)。direction 引数は、方向コードを格納したロングワードのアドレスです。

指定できる値は SMG\$M\_UP, SMG\$M\_DOWN です。省略時の設定は SMG\$M\_UP です。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

出力されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

---

 説明

SMG\$PUT\_LINE\_MULTI ルーチンによって、複数の属性を含む行を、仮想ディスプレイに書き込むことができます。カーソル移動がともなう場合もあります。SMG\$PUT\_LINE\_MULTI ルーチンは、現在の仮想カーソルの位置からその行の行末まで書き込みます。テキストが 1 行より短い場合は、空白が埋め込まれます。

テキストが右側境界を超えている場合には、flags 引数の値によって制御されます。

flags 引数で自動改行を指定した場合は、line-advance 引数で指定した行数だけスクロールされ、あふれた文字を次の行に挿入するための空間が作成されます。次の行はスクロールの方向 ( direction 引数) によって決定されます。flags 引数で自動改行を指定していない場合は、あふれた文字は切り捨てられます。

SMG\$PUT\_LINE\_MULTI ルーチンを呼び出した後、仮想カーソルの位置は出力の対象となる次の行の 1 カラム目に設定されます。出力の対象となる次の行は、line-advance 引数と direction 引数によって決定されます。SMG\$PUT\_LINE\_MULTI ルーチンを続けて呼び出しても、前の出力の上に新しい出力が重ならないように、line-advance 引数のシステム設定値は 1 になっています。

DEC 漢字または Super DEC 漢字文字集合を使用する際に、全角文字の右半分から書き込み操作が始まる場合は、その文字の左半分は未定義文字になります。また、行の一部が切り捨てられる際に、全角文字の右半分から切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$_INVARG	line-advance 引数に負の値が指定されました。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_WILUSERMS	ペースト・ボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## SMG\$PUT\_LINE\_WIDE

### Write Double-Width Line

Write Double-Width Line ルーチンは、倍角のテキスト行を仮想ディスプレイに書き込みます。

---

#### 形式

```
SMG$PUT_LINE_WIDE display-id ,text [,line-advance] [,rendition-set] [,rendition-complement] [,flags]
                  [,character-set]
```

---

#### 戻り値

OpenVMS 用法	cond_value
データ型	longword (unsigned)
アクセス	write only
受け渡し方	by value

---

#### 引数

display-id

OpenVMS 用法	identifier
データ型	longword (unsigned)
アクセス	read only
受け渡し方	by reference

行が書き込まれる仮想ディスプレイの識別子。display-id 引数はディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

text

OpenVMS 用法	char_string
データ型	character string
アクセス	read only
受け渡し方	by descriptor

仮想ディスプレイへ書き込まれる文字。text 引数は、文字列を示すディスクリプタのアドレスです。

## line-advance

VMS 用法: longword\_signed  
 データ型: longword (signed)  
 アクセス: read only  
 受け渡し方: by reference

行書き込み後のスクロール行数。line-advance 引数は、進める行数を格納した符号付きロングワードのアドレスです。この引数は省略可能です。

## rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定された属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

SMG\$M_BLINK	点滅する文字を表示します。
SMG\$M_BOLD	通常の輝度より高い輝度で文字を表示します。
SMG\$M_INVISIBLE	表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。
SMG\$M_REVERSE	文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。
SMG\$M_UNDERLINE	下線を付けた文字を表示します。
SMG\$M_USER1 から SMG\$M_USER8	ユーザ定義属性を表示します。

## rendition-complement

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる引数はすべて、rendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御できます。各属性で次の変換が可能です。

設定	補足	動作
0	0	属性を省略時の状態に設定する。
1	0	属性を有効にする。
0	1	属性を省略時の逆の状態に設定する。
1	1	属性を無効にする。

#### flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

テキストが、1 行に格納できない場合に実行する動作を指定するビット・マスク (省略可能)。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。この引数に対して指定できる値は次のとおりです。

0 自動改行しないことを指定します (省略時の設定値)。  
 SMG\$M\_WRAP\_CHAR 行の最後の文字で自動改行します。  
 SMG\$M\_WRAP\_WORD 行の最後の空白で自動改行します。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

出力されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。

値	文字集合 (コードセット)
SMG\$C_ASCII	ASCII 文字集合
SMG\$C_SPEC_GRAPHICS	DEC 特殊文字集合
SMG\$C_SUPPLEMENTAL	補助文字集合 <sup>1</sup>
SMG\$C_KANJI	DEC 漢字文字集合
SMG\$C_JIS_KANA	JIS カタカナ文字集合
SMG\$C_SDK	Super DEC 漢字コードセット

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$PUT\_LINE\_WIDE ルーチンは、倍角のテキストを仮想ディスプレイに書き込みます。SMG\$PUT\_LINE\_WIDE ルーチンは、現在の仮想カーソルの位置からテキ

ストを書き始めます。テキストが 1 行より短い場合は、空白が埋め込まれます。

flags 引数が自動改行を指定した場合には、line-advance 引数で指定した行数だけスクロールされ、あふれた文字を次の行に挿入するための空間が作成されます。次の行はスクロールの方向 ( direction 引数) によって決定されます。flags 引数で自動改行を指定していない場合は、あふれた文字は切り捨てられます。

SMG\$PUT\_LINE\_WIDE ルーチン呼び出した後、仮想カーソルの位置は出力の対象となる次の行の 1 カラム目に設定されます。出力の対象となる次の行は、line-advance 引数によって決定されます。SMG\$PUT\_LINE ルーチンを続けて呼び出ししても、前の出力の上に新しい出力が重ならないように、line-advance 引数のシステム設定値は 1 になっています。

仮想ディスプレイ上にテキスト行を書き込むための日本語 SMG ルーチンとしては、他に SMG\$PUT\_LINE ルーチンと SMG\$PUT\_LINE\_HIGHWIDE ルーチンがあります。

DEC 漢字または Super DEC 漢字文字集合を使用する際に、全角文字の右半分から書き込み操作が始まる場合は、その文字の左半分は未定義文字になります。また、行の一部が切り捨てられる際に、全角文字の右半分から切り捨てられる場合は、その文字の左半分は未定義文字になります。

---

## 戻される状態値

SS\$_NORMAL	正常終了。
LIB\$_INVSTRDES	文字列ディスクリプタが誤っています。
SMG\$_INVARG	引数が誤っています。
SMG\$_INVDIS_ID	display-id 引数が誤っています。
SMG\$_WILUSERMS	ペースト・ボードがビデオ・ターミナルではありません。
SMG\$_WRONUMARG	引数の数が誤っています。

---

## 例

- ```

C+
C This FORTRAN example program demonstrates the use of
C SMG$PUT_LINE_WIDE.
C
C Include the SMG definitions. In particular, we want SMG$M_BORDER and
C SMG$M_UNDERLINE.
C-
```

## SMG\$PUT\_LINE\_WIDE

```
INCLUDE '($SMGDEF)'
INTEGER SMG$CREATE_VIRTUAL_DISPLAY, SMG$CREATE_PASTEBOARD
INTEGER SMG$PASTE_VIRTUAL_DISPLAY, SMG$PUT_LINE_WIDE
INTEGER DISPLAY1, PASTE1, ROWS, COLUMNS, STATUS
CHARACTER*34 TEXT(3)

C+
C Create a virtual display with a border by calling
C SMG$CREATE_VIRTUAL_DISPLAY.
C-

      ROWS = 7
      COLUMNS = 70

      STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
C-

      STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$PUT_LINE to put data in the virtual display.
C-

      TEXT(1) = 'This virtual display has 7'
      TEXT(2) = 'rows and 70 columns.'
      TEXT(3) = 'Text entered by SMG$PUT_LINE_WIDE.'

C+
C After the first line of text is printed, advance two lines.
C-

      STATUS = SMG$PUT_LINE_WIDE ( DISPLAY1, TEXT(1), 2 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Underline the next line of text. Notice that 34 characters are being
C underlined. Advance 1 line of text after displaying the line.
C-

      STATUS = SMG$PUT_LINE_WIDE ( DISPLAY1, TEXT(2), 1,
1      SMG$M_UNDERLINE )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Display the third line of text.
C-

      STATUS = SMG$PUT_LINE_WIDE ( DISPLAY1, TEXT(3) )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

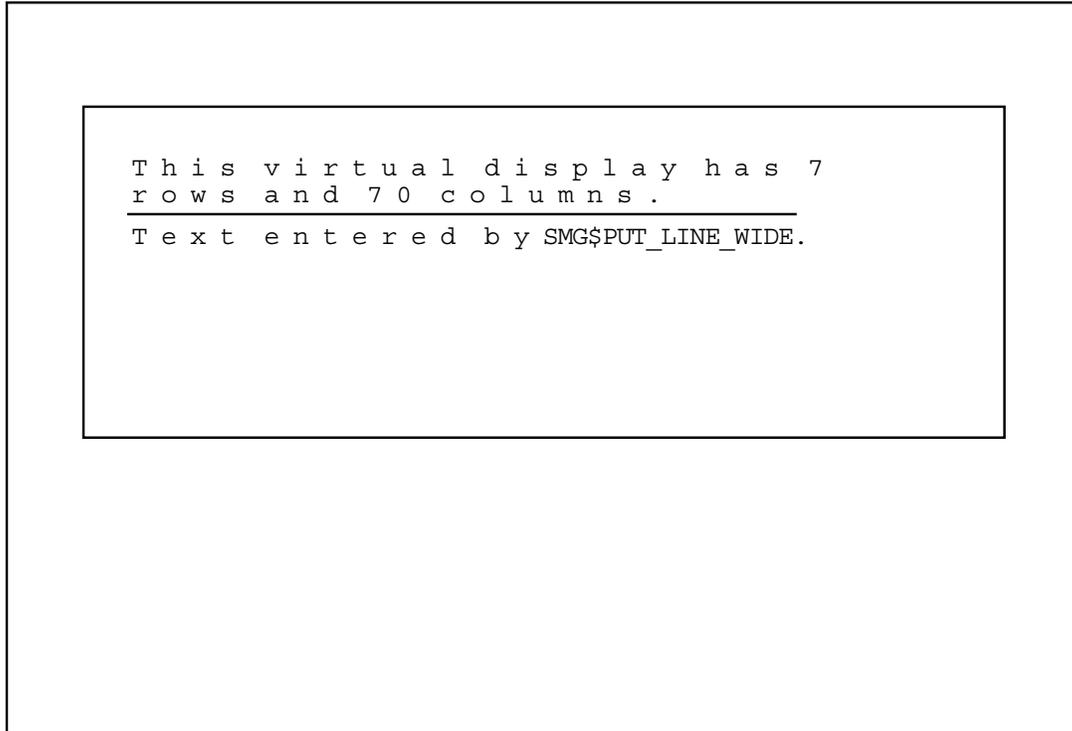
C+
C Paste the virtual display using SMG$PASTE_VIRTUAL_DISPLAY.
C-

      STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 5 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
```

END

この FORTRAN プログラムによって生成される出力は、図 8-32 に示すとおりです。

図 8-32 SMG\$PUT\_LINE\_WIDE を呼び出す FORTRAN プログラムによって生成される出力



ZK-4143-GE

---

## SMG\$PUT\_STATUS\_LINE

### Output Line of Text to Hardware Status Line

Output Line of Text to Hardware Status Line ルーチンは、1 行のテキストをハードウェア・ステータス行に出力します。

---

#### 形式

SMG\$PUT\_STATUS\_LINE *pasteboard-id* ,*text* [,*character-set*]

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

ハードウェア・ステータス行をもつペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

text

|         |                  |
|---------|------------------|
| VMS 用法: | char_string      |
| データ型:   | character string |
| アクセス:   | read only        |
| 受け渡し方:  | by descriptor    |

ハードウェア・ステータス行に書き込まれる文字。text 引数はテキストを指すディスクリプタのアドレスです。

character-set

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | longword_unsigned   |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

文字集合指定子。character-set 引数は省略可能で、テキストをハードウェア・ステータス行に書き込むときに使用する文字集合を指定する、符号なしロングワードのアドレスです。

| 値                    | 文字集合 (コードセット)       |
|----------------------|---------------------|
| SMG\$C_ASCII         | ASCII 文字集合          |
| SMG\$C_SPEC_GRAPHICS | DEC 特殊文字集合          |
| SMG\$C_SUPPLEMENTAL  | 補助文字集合 <sup>1</sup> |
| SMG\$C_KANJI         | DEC 漢字文字集合          |
| SMG\$C_JIS_KANA      | JIS カタカナ文字集合        |
| SMG\$C_SDK           | Super DEC 漢字コードセット  |

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$PUT\_STATUS\_LINE ルーチンは、1 行のテキストをターミナルのハードウェア・ステータス行に出力します。一部のターミナルでは、スクリーンの一番下 (25 行目) にハードウェア・ステータス行があります。このステータス行が“ホストから書き込み可能”として設定されている場合には、このルーチンを使用して 1 行のテキストをステータス行に出力できます (ハードウェア・ステータス行を使用できない場合には、エラーとして SMG\$\_OPNOTSUP が戻されます)。テキストは反転表示されません。

## 戻される状態値

|                 |                           |
|-----------------|---------------------------|
| SS\$_NORMAL     | 正常終了。                     |
| LIB\$_INVARG    | 引数に誤りがあります。               |
| SMG\$_INVPAS_ID | pasteboard-id 引数に誤りがあります。 |
| SMG\$_OPNOTSUP  | ハードウェア・ステータス行を使用できません。    |
| SMG\$_WRONUMARG | 引数の数が誤っています。              |

---

## SMG\$READ\_COMPOSED\_LINE

### Read Composed Line

Read Composed Line ルーチンは、通常のキー・ストロークと同値文字列で構成される 1 行の入力を読み込みます。

---

#### 形式

```
SMG$READ_COMPOSED_LINE keyboard-id [,key-table-id] ,resultant-string [,prompt-string]
                        [,resultant-length] [,display-id] [,flags] [,initial-string] [,timeout]
                        [,rendition-set] [,rendition-complement] [,word-terminator-code]
                        [,character-set]
```

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

keyboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

入力を読み込む対象となる仮想キーボードを指定します。keyboard-id 引数は、キーボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンによって与えられません。

key-table-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

キー・ストロークを変更するために使用されるキー・テーブルを指定します。key-table-id 引数は、キー・テーブル識別子を格納した符号なしのロングワードのアドレスです。

この引数は SMG\$CREATE\_KEY\_TABLE ルーチンによって与えられます。

#### resultant-string

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | write only       |
| 受け渡し方      | by descriptor    |

SMG\$READ\_COMPOSED\_LINE ルーチンが最終的に作成した行を書き込む文字列。resultant-string 引数は、生成される行が書き込まれる文字列を示すディスクリプタのアドレスです。

#### prompt-string

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | read only        |
| 受け渡し方      | by descriptor    |

読み込み操作に対するプロンプトとして使用される文字列。prompt-string 引数は、プロンプト文字列を示すディスクリプタのアドレスです。

#### resultant-length

|            |                 |
|------------|-----------------|
| OpenVMS 用法 | word_unsigned   |
| データ型       | word (unsigned) |
| アクセス       | write only      |
| 受け渡し方      | by reference    |

読み込んだバイト数と、resultant-string 引数の最大長のどちらか小さい方の値が書き込まれます。resultant-length 引数は、SMG\$READ\_COMPOSED\_LINE ルーチンが読み込んだバイト数が書き込まれる符号なしワードのアドレスです。

#### display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

ディスプレイ識別子。display-id 引数はディスプレイ識別子を格納した符号なしロングワードのアドレスです。画面管理機能の出カルーチンを使用しない場合のみこの引数は省略可能です。

画面管理機能の入カルーチンと出カルーチンを使用する場合には、この引数は、入力が行われる仮想ディスプレイを指定します。指定される仮想ディスプレイは、keyboard-id 引数によって指定されるペーストボードと同じペーストボードにペーストされていなければならない、他のディスプレイによって隠されていない状態であればなりません。仮想ディスプレイの隠された領域から入力を受け付けることはできません。

複数の仮想ディスプレイが存在する場合には、各仮想ディスプレイに対して仮想カーソルの位置が割り当てられています。同時に、物理カーソルの現在の位置に対応する物理カーソル位置が1つだけあります。display-id 引数を指定した場合には、読み込み操作は、指定された仮想ディスプレイ内の現在の仮想カーソルの位置から開始されます。ディスプレイ識別子を省略した場合には、読み込み操作は現在の物理カーソルの位置から開始されます。prompt-string 引数の長さを入力されるキーの合計は、ディスプレイ内に表示されるカラム数に制限されます。

---

### 注意

---

この仮想ディスプレイはカラム1にペーストしなければならず、その右側に他の仮想ディスプレイを配置することはできません。この制限事項が必要なのは、この条件が満足されない場合、`[Ctrl/R]`または`[Ctrl/U]`の発生によって、右側にペーストされた出力も含めて行全体が消去されるからです。

---

#### flags

OpenVMS 用法    mask\_longword  
 データ型        longword (unsigned)  
 アクセス        read only  
 受け渡し方      by reference

使用可能なキーを指定するビット・マスク (省略可能)。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。flags 引数に対して指定できる値は次のとおりです。

|                  |                                                                           |
|------------------|---------------------------------------------------------------------------|
| 0                | 行編集は許可されるが、ファンクション・キー ( <code>[F6]</code> ~ <code>[F14]</code> ) は使用できない。 |
| SMG\$M_FUNC_KEYS | ファンクション・キー ( <code>[F6]</code> ~ <code>[F14]</code> ) は使用できるが、行編集は禁止される。  |
| SMG\$M_NOKEEP    | 再呼び出しバッファに入力された行は保存されない。                                                  |
| SMG\$M_NORECALL  | この入出力に対してのみ、行の再呼び出しは禁止される。                                                |

OpenVMS ターミナル・ドライバは一部の漢字ターミナルで行編集のために `[F6]` ~ `[F14]` ファンクション・キーを使用するため、ファンクション・キーと、行編集を同時に許可することはできません。

これらのフラグが正しく機能するためには、端末に対して端末属性/LINE\_EDITING を設定しなければなりません。/LINE\_EDITING は省略時の設定です。

かな漢字変換キーの定義は、SMG\$DEFINE\_KEY ルーチンや SMG\$ADD\_KEY\_DEF ルーチンによるキー定義に優先されます。character-set 引数が SMG\$C\_KANJI または SMG\$C\_SDK の場合には、flags 引数に SMG\$M\_FUNC\_KEYS を指定しても行編集は禁止されません。

#### initial-string

OpenVMS 用法    char\_string  
 データ型        character string

アクセス read only  
受け渡し方 by descriptor

初期文字列 (省略可能)。initial-string 引数は、フィールドの初期文字列を格納した省略可能な文字列を示すディスクリプタのアドレスです。文字列はキーボードから入力されたかのように、ディスプレイの入力領域に書き込まれます。文字列は通常の方法で編集できます (function-keys-flag 引数が設定されていない場合)。

#### timeout

OpenVMS 用法 longword\_signed  
データ型 longword (signed)  
アクセス read only  
受け渡し方 by reference

時間切れの値 (省略可能)。timeout 引数は、時間切れの値を格納した符号付きロングワードのアドレスです。timeout 引数を指定した場合には、時間切れの前に入力されたすべての文字はバッファに書き込まれます。timeout 引数を省略した場合には、終了文字が入力されるまで文字はバッファに書き込まれます。

#### rendition-set

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

|                                 |                                                             |
|---------------------------------|-------------------------------------------------------------|
| SMG\$M_BLINK                    | 点滅する文字を表示します。                                               |
| SMG\$M_BOLD                     | 通常の輝度より高い輝度で文字を表示します。                                       |
| SMG\$M_REVERSE                  | 文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。              |
| SMG\$M_UNDERLINE                | 下線を付けた文字を表示します。                                             |
| SMG\$M_INVISIBLE                | 表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。 |
| SMG\$M_USER1 から<br>SMG\$M_USER8 | ユーザ定義属性を表示します。                                              |

rendition-set 引数を使用する場合は、display-id 引数を指定しなければなりません。

#### rendition-complement

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる属性はすべて、rendition-complement 引数で補足指定することができます。

rendition-complement 引数を使用する場合は、display-id 引数を指定しなければなりません。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御することができます。各属性で次の変換が可能です。

| 設定 | 補足 | 動作                |
|----|----|-------------------|
| 0  | 0  | 属性を省略時の状態に設定する。   |
| 1  | 0  | 属性を有効にする。         |
| 0  | 1  | 属性を省略時の逆の状態に設定する。 |
| 1  | 1  | 属性を無効にする。         |

#### word-terminator-code

OpenVMS 用法 word\_unsigned  
 データ型 word (unsigned)  
 アクセス write only  
 受け渡し方 by reference

キー終了文字コード。word-terminator-code 引数は、どの文字またはどのキーが読み込み操作を終了したかを示すコードが書き込まれる、符号なしワードのアドレスです。キー終了文字コードは SMG\$K\_TRM\_keyname という形式です。キー名については第 3 章の表 3-1 を参照してください。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

入力されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。この引数を省略し、かつ仮想ディスプレイが指定されていない場合には、SMG\$DEFAULT\_CHARACTER\_SET 論理名で示される文字集合が使用されます。

| 値                    | 文字集合 (コードセット)       |
|----------------------|---------------------|
| SMG\$C_ASCII         | ASCII 文字集合          |
| SMG\$C_SPEC_GRAPHICS | DEC 特殊文字集合          |
| SMG\$C_SUPPLEMENTAL  | 補助文字集合 <sup>1</sup> |
| SMG\$C_KANJI         | DEC 漢字文字集合          |
| SMG\$C_JIS_KANA      | JIS カタカナ文字集合        |
| SMG\$C_SDK           | Super DEC 漢字コードセット  |

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

SMG\$DEFAULT\_CHARACTER\_SET 論理名に定義できる値については、第 2.2.14 項の表 2-2 を参照してください。

## 説明

SMG\$READ\_COMPOSED\_LINE ルーチンは、指定されたキー・テーブルに定義されている方法で、通常のキー・ストロークとキー同値文字列で構成される 1 行を読み込みます。キー定義の属性は、同値文字列をエコー表示するかどうかと、定義されているキー・ストロークによって読み込み操作を終了するかどうかを制御します。通常のキー・ストロークは常にエコー表示されます。

キャリッジ・リターンは常に読み込み操作を終了します。[Ctrl/Z]が入力されたときに、キー定義テーブルに[Ctrl/Z]に対する定義が登録されていない場合には、"EXIT"がエコー表示され、読み込み操作は終了します。[Ctrl/Z]が行の最初の文字として入力された場合には、SMG\$\_EOF が戻されます。それ以外の場合には、次の読み込み操作で SMG\$\_EOF が戻されます。キー定義に属性として指定されていない限り、他の終了文字は認識されません。

矢印キー (□, □) と [Ctrl/B] が定義されていない場合には、SMG\$READ\_xxxx ルーチンを使用して読み込んだ前の行は、矢印キー (□, □) を使用して再呼び出しすることができます。後で再呼び出しするために保存される行数は、SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンの recall-size 引数に応じて異なります。省略時の設定は 20 行です。

SMG\$READ\_COMPOSED\_LINE ルーチンは入力操作を実行する前に、SMG\$FLUSH\_BUFFER ルーチンを呼び出します。このため、入力操作を開始するときに、画面イメージは常に最新の状態に更新されます。SMG\$READ\_COMPOSED\_LINE ルーチンを使用する場合には、ペーストボードと仮想ディスプレイの両方に対して、ディスプレイ・バッチ処理を禁止しておかなければなりません。

入力文字の文字集合が DEC 漢字または Super DEC 漢字文字集合であった場合には、SMG\$READ\_COMPOSED\_LINE ルーチンが、かな漢字変換の処理を行います。

す。この場合、変換に使用されるキーは、SMG\$DEFINE\_KEY ルーチンおよび SMG\$ADD\_KEY\_DEF ルーチンによるキー定義に優先されます。

---

## 戻される状態値

|                 |                                                         |
|-----------------|---------------------------------------------------------|
| SS\$_NORMAL     | 正常終了。                                                   |
| SS\$_ABORT      | 実行中に入出力操作が強制終了されました (SMG\$CANCEL_INPUT ルーチンにより)。        |
| SS\$_CANCEL     | キューに登録している間に入出力操作が取り消されました (SMG\$CANCEL_INPUT ルーチンにより)。 |
| SMG\$_EOF       | ファイルの最後。                                                |
| SMG\$_ILLBATFNC | バッチ処理されているディスプレイからの入力は認められません。                          |
| SMG\$_INVARG    | 引数が誤っています。                                              |
| SMG\$_INVCOL    | カラム番号に誤りがあります。読み込み操作が仮想ディスプレイの外側のカラムを使用しようとした。          |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。                                  |
| SMG\$_INVKBD_ID | keyboard-id 引数に誤りがあります。                                 |
| SMG\$_INVKTB_ID | key-table-id 引数に誤りがあります。                                |
| SMG\$_WRONUMARG | 引数の数が誤っています。                                            |
| SS\$_xxx        | \$QIOW から戻された状態値。                                       |
| JSYS\$_xxx      | JSYS\$_xxx から戻された状態値。                                   |
| LIB\$_xxx       | LIB\$COPY_R_DX から戻された状態値。                               |
| RMS\$_xxx       | \$GET から戻された状態値 (RMS\$_EOF を除く)。                        |
| IM\$_xxx        | IMLIB から戻された状態値。                                        |

---

## SMG\$READ\_FROM\_DISPLAY

### Read Text from Display

Read Text from Display ルーチンは、仮想ディスプレイから 1 行のテキストを読み込みます。

---

#### 形式

SMG\$READ\_FROM\_DISPLAY *display-id ,resultant-string [,terminator-string] [,start-row] [,rendition-string] [,character-set-string]*

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

テキストが読み込まれる仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

resultant-string

|         |                  |
|---------|------------------|
| VMS 用法: | char_string      |
| データ型:   | character string |
| アクセス:   | write only       |
| 受け渡し方:  | by descriptor    |

SMG\$READ\_FROM\_DISPLAY ルーチンが仮想ディスプレイから読み込んだ情報を書き込む文字列。resultant-string 引数は、文字列が書き込まれる文字列を示すデスク립タのアドレスです。

## terminator-string

VMS 用法: char\_string  
 データ型: character string  
 アクセス: read only  
 受け渡し方: by descriptor

逆向き検索を終了する終了文字を格納した文字列。戻される文字列の開始位置を決定するために用いられます。terminator-string 引数は、終了文字の文字列を示すディスクリプタのアドレスです。terminator-string 引数を省略した場合には、逆向き検索は実行されず、戻される文字列はカーソルの現在の位置にある文字から始まります。

## start-row

VMS 用法: longword\_signed  
 データ型: longword (signed)  
 アクセス: read only  
 受け渡し方: by reference

start-row 引数は、読み込まれる仮想ディスプレイの行番号を格納した、符号付きロングワードのアドレスです。この引数は省略可能です。

## rendition-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス write only  
 受け渡し方 by descriptor

ビデオ属性を制御するビット・マスク文字列 (省略可能)。rendition-string 引数は、ビット・マスク文字列を示すディスクリプタのアドレスです。ディスプレイ上のテキスト文字列のカラムに対応する属性を指定することができます。この rendition-string 引数で指定できる属性は次のとおりです。

|                  |                                                            |
|------------------|------------------------------------------------------------|
| SMG\$M_BLINK     | 点滅文字を指示します。                                                |
| SMG\$M_BOLD      | 通常より明るい輝度の文字を指定します。                                        |
| SMG\$M_REVERSE   | 反転表示文字を指示します。つまり、仮想ディスプレイの現在の属性の反対の状態では文字を表示しています。         |
| SMG\$M_UNDERLINE | 下線付き文字を指示します。                                              |
| SMG\$M_INVISIBLE | 表示されない文字を指定します。つまり、文字は仮想ディスプレイ内に存在しますが、ペーストボードには表示されていません。 |

## character-set-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス write only  
 受け渡し方 by descriptor

文字集合を示すバイト文字列 (省略可能)。character-set-string 引数は文字集合の文字列を示すディスクリプタのアドレスです。ディスクリプタ中の各バイトの値が、ディスプレイ中のテキスト文字列のバイトに対応する文字集合を示します。この character-set-string 引数に返される値は次の値の組み合わせからなります。

| 値                    | 文字集合 (コードセット)       |
|----------------------|---------------------|
| SMG\$C_ASCII         | ASCII 文字集合          |
| SMG\$C_SPEC_GRAPHICS | DEC 特殊文字集合          |
| SMG\$C_SUPPLEMENTAL  | 補助文字集合 <sup>1</sup> |
| SMG\$C_KANJI         | DEC 漢字文字集合          |
| SMG\$C_JIS_KANA      | JIS カタカナ文字集合        |
| SMG\$C_UNKNOWN       | 未定義文字集合             |
| SMG\$C_SDK           | Super DEC 漢字コードセット  |

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$READ\_FROM\_DISPLAY ルーチンは、指定された仮想ディスプレイの現在の行のすべてのテキストまたは一部のテキストを格納する文字列を戻します。terminator-string 引数を省略した場合には、現在の行の内容 (現在のカラム位置から右端のカラム位置まで) が戻されます。

現在のカラム位置が、全角文字の右半分である場合には、その位置は未定義文字として扱われます。start-row 引数を指定した場合には、start-row 引数によって指定される行の 1 カラム目から右端のカラムまでの内容が resultant-string 引数に戻されます。start-row 引数を指定した場合には、terminator-string 引数は無視されます。

terminator-string 引数を指定した場合には、その文字列内の各文字は逆向き検索の終了文字として機能します。つまり、戻される最初の文字列を判断するために使用されます。指定された終了文字がまったく検出されない場合には、検索操作はその行の最初の文字位置で終了します。

線画を読みこむ場合には、装置に依存しない文字(+, -, |)で戻されます。

## 戻される状態値

|                  |                                                                 |
|------------------|-----------------------------------------------------------------|
| SS\$ _NORMAL     | 正常終了。                                                           |
| SMG\$ _INVDIS_ID | display-id 引数に誤りがあります。                                          |
| SMG\$ _DSPIN_USE | 同じディスプレイへの日本語 SMG ルーチンコールを中断させた AST ルーチンから、日本語 SMG ルーチンが呼ばれました。 |
| SMG\$ _KBDIN_USE | 2 回目または次の読み込み操作で、複数の QIO が同じチャネルを使用しようとしていました。                  |
| LIB\$ _INSVIRMEM | 仮想メモリが不足しています。                                                  |

## 例

```

1. C+
   C This FORTRAN example demonstrates the use of SMG$READ_FROM_DISPLAY.
   C-

   C+
   C Include the SMG definitions. In particular, we want SMG$M_BORDER.
   C-

       IMPLICIT INTEGER (A-Z)
       INCLUDE '($SMGDEF)'
       CHARACTER*80 TEXT

   C+
   C Use SMG$CREATE_VIRTUAL_DISPLAY to create the virtual display
   C and give it a border.
   C-

       ROWS = 5
       COLUMNS = 60

       STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Create the pasteboard by calling SMG$CREATE_PASTEBOARD.
   C-

       STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Call SMG$PASTE_VIRTUAL_DISPLAY and SMG$PUT_LINE to paste
   C the virtual display and put some text on line 2.
   C-

       STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 2, 10 )
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

       STATUS = SMG$PUT_LINE ( DISPLAY1, ' ' )
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

       STATUS = SMG$PUT_LINE ( DISPLAY1,
1          'This is an example of using SMG$READ_FROM_DISPLAY.' )
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Use SMG$READ_FROM_DISPLAY to read line 2 from the virtual
   C display, starting at column 22.
   C-

       STATUS = SMG$SET_CURSOR_ABS ( DISPLAY1, 2, 22 )
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

```

```

C+
C Search line 2 from column 22 to column 1 for the null string.
C Since no terminator will be supplied, no "back-searching" will take
C place. TEXT will be assigned the "value" of the line from
C column 22 to the rightmost column.
C-

      STATUS = SMG$READ_FROM_DISPLAY ( DISPLAY1, TEXT )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Put the line of text found into the virtual display at row 4,
C column 10 by calling SMG$SET_CURSOR_ABS and SMG$PUT_LINE.
C-

      STATUS = SMG$SET_CURSOR_ABS ( DISPLAY1, 4, 10 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_LINE ( DISPLAY1, TEXT )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$SET_CURSOR_ABS to set the cursor back to line 2, column 22.
C-

      STATUS = SMG$SET_CURSOR_ABS ( DISPLAY1, 2, 22 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$READ_FROM_DISPLAY to search line 2 from column 22 to
C column 1 for an "f". Now, "back-searching" will take place.
C Starting at column 22, "back-track" to column 1 looking for "f".
C Text will then be assigned the "value" of the line from the
C present cursor position (where the "f" is, to the rightmost
C column.
C-

      STATUS = SMG$READ_FROM_DISPLAY ( DISPLAY1, TEXT, 'f' )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Put the line of text found into the virtual display at row 4, column 10.
C-

      STATUS = SMG$SET_CURSOR_ABS ( DISPLAY1, 5, 10 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

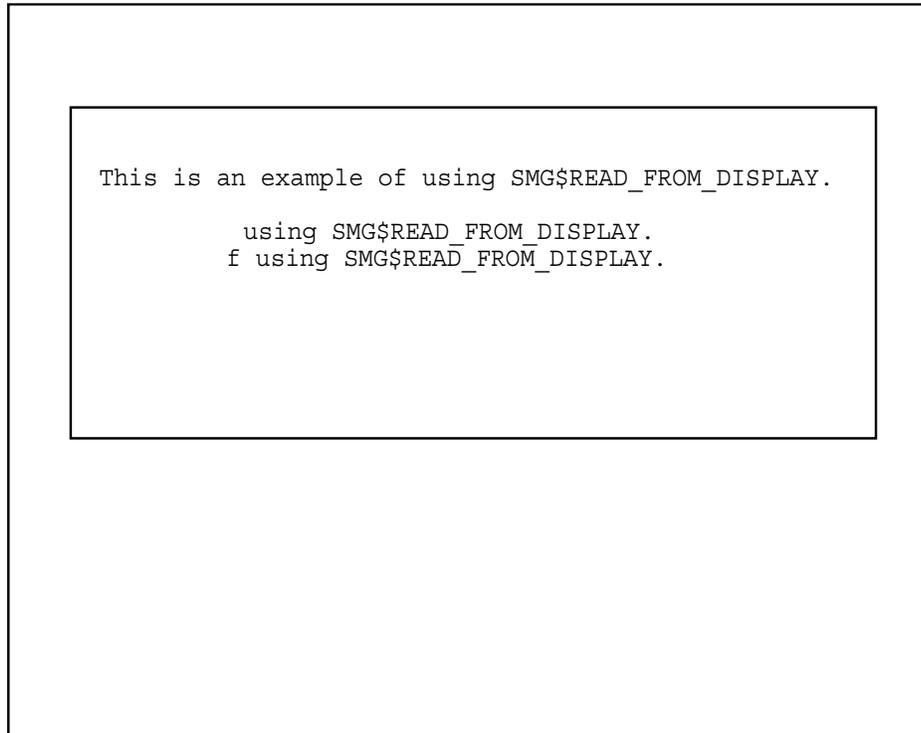
      STATUS = SMG$PUT_LINE ( DISPLAY1, TEXT )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      END

```

この FORTRAN プログラムによって生成される出力は、図 8-33 に示すとおりです。

図 8-33 SMG\$READ\_FROM\_DISPLAY ルーチン呼び出す出力



ZK-4134-GE

---

# SMG\$READ\_KEYSTROKE

## Read a Single Character

Read a Single Character ルーチンは、キーストロークを読み込み、そのキーストロークの終了文字コードを戻します。

---

### 形式

SMG\$READ\_KEYSTROKE *keyboard-id* ,*word-terminator-code* [*prompt-string*] [*timeout*] [*display-id*] [*rendition-set*] [*rendition-complement*]

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

keyboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

キーボード識別子。keyboard-id 引数は、キーストロークが読み込まれる仮想キーボードの識別子を格納した符号なしロングワードです。

この引数は SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンによって与えられません。

word-terminator-code

|            |                 |
|------------|-----------------|
| OpenVMS 用法 | word_unsigned   |
| データ型       | word (unsigned) |
| アクセス       | write only      |
| 受け渡し方      | by reference    |

キー終了文字コード。word-terminator-code 引数は、どの文字またはどのキーが読み込み操作を終了したかを示すコードが書き込まれる符号なしワードです。キー終了文字コードは SMG\$K\_TRM\_keyname の形式です。キー名は 第 3 章の表 3-1 に示すとおりです。

## prompt-string

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | read only        |
| 受け渡し方      | by descriptor    |

プロンプト文字列。prompt-string 引数は省略可能な文字列であり、読み込み操作に対するプロンプトとして使用されます。

## timeout

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

時間切れの値。timeout 引数は省略可能です。この引数を指定した場合には、時間切れの前に入力されたキーストロークは読み込まれます。

## display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

仮想ディスプレイ識別子。display-id 引数は省略可能であり、読み込み操作が実行される仮想ディスプレイの識別子を格納した、符号なしロングワードのアドレスです。

複数の仮想ディスプレイがペーストされているときに、prompt-string 引数を指定する場合には、プロンプト文字列をどの仮想ディスプレイに書き込むかを指定するために、display-id 引数も指定しなければなりません。prompt-string 引数を指定しない場合には、display-id 引数も指定しないでください。

複数の仮想ディスプレイが存在する場合には、各仮想ディスプレイにはそれぞれ仮想カーソルが対応づけられます。同時に、物理カーソルの現在の位置に対応する物理カーソルが 1 つだけ設定されます。

display-id 引数を指定した場合には、読み込み操作は指定された仮想ディスプレイ内の現在の仮想カーソルの位置から開始されます。display-id 引数を省略した場合には、読み込み操作は現在の物理カーソルの位置から開始されます。prompt-string 引数入力されるキーの合計の長さは、仮想ディスプレイに表示可能なカラム数に制限されます。

## rendition-set

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

プロンプトに対する属性指定子。rendition-set 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このビット・マスク内で設定されている各属性

に対して、ディスプレイ内の対応する属性が設定されます。rendition-set 引数を使用して指定できる属性は次のとおりです。

|                                 |                                                          |
|---------------------------------|----------------------------------------------------------|
| SMG\$M_BLINK                    | 点滅文字を指定します。                                              |
| SMG\$M_BOLD                     | 通常より明るい輝度の文字を表示します。                                      |
| SMG\$M_REVERSE                  | 文字を反転表示することを指定します。つまり仮想ディスプレイの現在の属性の反対の状態に文字を指定します。      |
| SMG\$M_UNDERLINE                | 下線付き文字を指定します。                                            |
| SMG\$M_INVISIBLE                | 表示されない文字を指定します。つまり、文字は仮想ディスプレイ内に存在しますが、ペーストボードには表示されません。 |
| SMG\$M_USER1 から<br>SMG\$M_USER8 | ユーザ定義属性を指定します。                                           |

rendition-set 引数を使用する場合には、display-id 引数を指定しなければなりません。

#### rendition-complement

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

プロンプトに対する属性補足指定子。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。ビット・マスク内で属性が設定されている場合には、ディスプレイ内で対応する属性が補足されます。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数を使用して補足できます。rendition-complement 引数を使用する場合には、display-id 引数を指定しなければなりません。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの 2 つの引数を組み合わせて使用すれば、1 回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

| 設定 | 補足 | 動作                |
|----|----|-------------------|
| 0  | 0  | 属性を省略時の状態に設定する。   |
| 1  | 0  | 属性をオンに設定する。       |
| 0  | 1  | 属性を省略時の逆の状態に設定する。 |
| 1  | 1  | 属性をオフに設定する。       |

---

## 説明

SMG\$READ\_KEYSTROKE ルーチンは、指定された仮想キーボードからキーストロークを読み込み、そのキーストロークの終了文字コードを SMG\$K\_TRM\_keyname の形式で戻します。読み込まれるキーストロークは画面にエコー表示されません。このキーストロークは標準的な英字、キーパッド・キーまたはファンクション・キー、または矢印キー (◀, ▶, ▲, ▼) のいずれでもかまいません。

SMG\$READ\_KEYSTROKE ルーチンを使用する場合には、ペーストボードと仮想ディスプレイの両方に対してディスプレイのバッチ処理を禁止しておかなければなりません。

VT100、VT200 シリーズ、および VT300 シリーズのキーボードから、すべてのキーを読み込むことができますが、以下は例外です。

- VT200 シリーズおよび VT300 シリーズ・キーボードの Compose Character キー
- VT200 シリーズおよび VT300 シリーズ・キーボードのカナキー
- ESCAPE キー
- SHIFT キー
- VT200 シリーズおよび VT300 シリーズ・キーボードの F1 ~ F5 キー

一部のキーおよびキー定義はユーザが定義可能ですが、なるべく定義しないようにしてください。これらのキーとキー定義を使用した場合、日本語 SMG はエラーを戻すわけではありませんが、DCL レベルで次の特殊な方法で漢字ターミナルを設定していない限り、これらのキーの組み合わせに割り当てた定義は実行されません。

- `Ctrl/C`, `Ctrl/O`, `Ctrl/X`, および `F6`

これらのキーを読み込むには、最初に DCL の SET TERMINAL/PASTHRU コマンドを入力しなければならない。

- `Ctrl/T`, `Ctrl/Y`

これらのキーを読み込むには、まず、DCL の SET TERMINAL/PASTHRU コマンドと SET NOCONTROL コマンドのどちらか一方または両方を入力しなければならない。

- `Ctrl/S`, `Ctrl/Q`

これらキーを読み込むには、まず、DCL の SET TERMINAL/NOTTSYNC コマンドを入力しなければならない。

これらの特殊なターミナル設定を使用することは望ましくありません。ターミナル・ドライバに制御を与えるために省略時の設定を変更した場合、その結果がどのようになるかを十分理解していない限り、これらの特殊な設定を使用すると、予測できない結果が発生することがあります。

---

 戻される状態値

|                 |                                                 |
|-----------------|-------------------------------------------------|
| SS\$_NORMAL     | 正常終了。                                           |
| SS\$_ABORT      | 実行中に入出力操作が強制終了されました (SMG\$CANCEL_INPUT によって)。   |
| SS\$_CANCEL     | キューに登録中に入力操作が取り消されました (SMG\$CANCEL_INPUT によって)。 |
| SMG\$_EOF       | ファイルの最後。                                        |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。                          |
| SMG\$_INVKBD_ID | keyboard-id 引数に誤りがあります。                         |
| SMG\$_WRONUMARG | 引数の数が誤っています。                                    |
| SS\$_xxx        | \$QIOW から戻されたエラー。                               |
| LIB\$_xxx       | LIB\$SCOPY_R_DX から戻されたエラー。                      |
| RMS\$_xxx       | \$GET から戻されたエラー (RMS\$_EOF を除く)。                |

---

 例

```

1. C+
   C This FORTRAN example program demonstrates the use of
   C SMG$READ_KEYSTROKE.
   C-

   C+
   C This routine creates a virtual display and writes it to the pasteboard.
   C Data is placed in the virtual display using SMG$PUT_CHARS.
   C
   C First, include the SMG definitions. In particular, we want SMG$_BORDER.
   C-

       IMPLICIT INTEGER (A-Z)
       INCLUDE '($SMGDEF)'
       CHARACTER*3 TEXT
       CHARACTER*27 TEXT_OUTPUT

   C+
   C Use SMG$CREATE_VIRTUAL_DISPLAY to create a virtual
   C display with a border.
   C-

       ROWS = 7
       COLUMNS = 60

       STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$_BORDER)
       IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

   C+
   C Create the pasteboard using SMG$CREATE_PASTEBOARD.
   C-
  
```

## SMG\$READ\_KEYSTROKE

```
STATUS = SMG$CREATE_PASTEBOARD ( PASTE1 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$CREATE_VIRTUAL_KEYBOARD to create a virtual keyboard.
C-

STATUS = SMG$CREATE_VIRTUAL_KEYBOARD ( KEYBOARD1 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Using SMG$PASTE_VIRTUAL_DISPLAY, paste the virtual display
C at row 3, column 9.
C-

STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 3, 9 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_LINE ( DISPLAY1,
1 'Enter the character K after the >> prompt.' )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_LINE ( DISPLAY1,
1 'This character will not be echoed as you type it.' )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_LINE ( DISPLAY1,
1 'The terminal character equivalent of K is displayed.' )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_LINE ( DISPLAY1, ' ' )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$READ_KEYSTROKE to read a keystroke from the virtual
C pasteboard.
C-

STATUS = SMG$READ_KEYSTROKE ( KEYBOARD1, TERM_CHAR, '>>', ,
1 DISPLAY1 )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

STATUS = SMG$PUT_LINE ( DISPLAY1, ' ' )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use OTS$CVT_L_TI to convert the decimal value of TERM_CHAR to
C a decimal ASCII text string.
C-

STATUS = OTS$CVT_L_TI ( TERM_CHAR, TEXT )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

TEXT_OUTPUT = ' TERMINAL CHARACTER IS: ' // TEXT

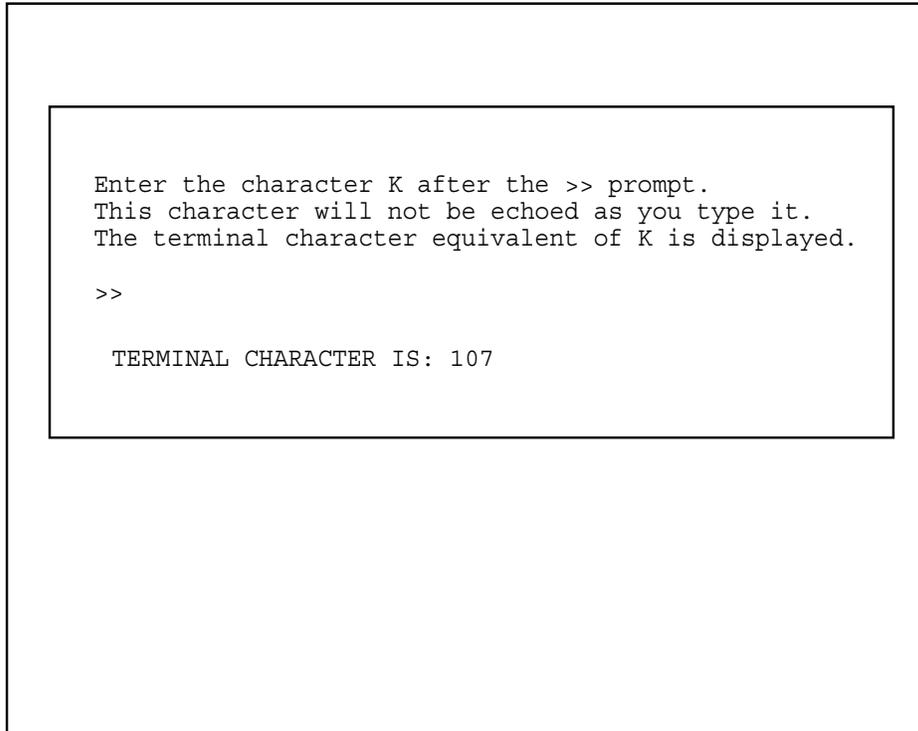
C+
C Call SMG$PUT_LINE and SMG$PUT_CHARS to print the decimal
C ASCII text string.
C-

STATUS = SMG$PUT_LINE ( DISPLAY1, TEXT_OUTPUT )
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
```

```
STATUS = SMG$PUT_CHARS ( DISPLAY1, TEXT, 7, 25 )  
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))  
  
END
```

このFORTRAN プログラムによって生成される出力は図 8-34 に示すとおりです。

図 8-34 SMG\$READ\_KEYSTROKE ルーチン呼び出す FORTRAN プログラムが生成する出力



The image shows a terminal window with the following text:

```
Enter the character K after the >> prompt.  
This character will not be echoed as you type it.  
The terminal character equivalent of K is displayed.  
  
>>  
  
TERMINAL CHARACTER IS: 107
```

ZK-4129-GE

---

## SMG\$READ\_STRING

### Read String

Read String ルーチンは、仮想キーボードから文字列を読み込みます。

---

#### 形式

SMG\$READ\_STRING *keyboard-id* ,*resultant-string* [*prompt-string*] [*maximum-length*] [*modifiers*]  
 [*timeout*] [*terminator-set*] [*resultant-length*] [*word-terminator-code*] [*display-id*]  
 [*initial-string*] [*rendition-set*] [*rendition-complement*] [*terminator-string*]  
 [*character-set*]

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

keyboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

入力を読み込む仮想キーボードを指定します。

keyboard-id 引数は、キーボード識別子を格納した符号なしロングワードのアドレスです。

この引数は、SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンによって与えられます。

resultant-string

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | write only       |
| 受け渡し方      | by descriptor    |

入力行が書き込まれる文字列。resultant-string 引数は、テキストが書き込まれる文字列を示すディスクリプタのアドレスです。

## prompt-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

読み込み操作に対するプロンプトとして使用される文字列。prompt-string 引数は、プロンプト文字列を示すディスクリプタのアドレスです。

## maximum-length

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

読み込む最大バイト数を指定します。maximum-length 引数は、読み込む最大バイト数を格納した符号付きロングワードのアドレスです。この引数に対して指定できる最大値は 512 です。省略時の設定は 512 になります。

かな漢字変換を使用する場合には、変換中ではなく、変換後に maximum-length 引数に指定されたバイト数で変換された文字列は、切り捨てられます。

ただし、変換バッファの最大長が 512 バイトであるため、512 バイトを超える文字列は変換できません。

得られた文字列の最後が全角文字の左半分であった場合には、その文字の左半分は切り捨てられ、maximum-length 引数よりも 1 バイト短い文字列が返されます。

## modifiers

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

オプションとして実行される動作を指定するビット・マスク (省略可能)。modifiers 引数は、フラグを格納した符号なしロングワードのアドレスです。

modifiers 引数に対して指定できる値は次のとおりです。

|                    |                                                                                |
|--------------------|--------------------------------------------------------------------------------|
| TRMSM_TM_CVTLOW    | 小文字を大文字に変換します。ASCII 文字および補助文字に対してのみ有効です。                                       |
| TRMSM_TM_NOECHO    | 入力された文字を画面にエコー表示しません。入力文字列の文字集合として Super DEC 漢字または DEC 漢字文字集合が指定された場合には無視されます。 |
| TRMSM_TM_PURGE     | 読み込み操作を実行する前に、先読みを可能にするバッファを消去します。                                             |
| TRMSM_TM_TRMNOECHO | 終了文字は入力されません。                                                                  |
| TRMSM_TM_NOEDIT    | 拡張編集機能が禁止されます。入力文字列の文字集合として Super DEC 漢字または DEC 漢字文字集合が指定された場合には、行編集は禁止されません。  |
| TRMSM_TM_NORECALL  | 行再呼び出し機能が禁止されます。                                                               |

modifiers 引数についての詳しい説明は、『OpenVMS I/O User's Reference Manual』のターミナル・ドライバの節を参照してください。TRMS シンボルは、HP が提供するシステム・シンボル・ライブラリ内の \$STRMDEF マクロ/モジュールによって定義されています。

#### timeout

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

プロンプトを表示した後、入力操作が終了するまでに認められる秒数を指定します。timeout 引数は、時間切れの値を格納した符号付きロングワードのアドレスです。

timeout 引数を指定した場合には、満了時間の前に入力されたすべての文字、または終了キーが入力されるまでに入力されたすべての文字は resultant-string 引数に戻されます。この引数を省略した場合には、終了文字が入力されるまで、入力操作は有効です。

#### terminator-set

VMS 用法: unspecified  
 データ型: unspecified  
 アクセス: read only  
 受け渡し方: by descriptor, fixed length

どの文字が終了文字として取り扱われるかを指定するマスク (短い形式)、またはこのようなマスクを示すディスクリプタ (長い形式)。terminator-set 引数は、マスクを示すディスクリプタのアドレスです。

0 ~ 31 の範囲の ASCII 値を持つ終了文字を使用する場合には、2 ロングワードで構成される短い形式を使用してください。最初のロングワードは 0 であり、2 番目のロングワードは終了文字ビット・マスクです。このマスクを作成するには、適切な終了文字の ASCII 値に対応するビットを設定します。たとえば、`Ctrl/A` (ASCII の 1) が終了文字であることを指定するには、terminator-set マスクのビット 1 を設定します。

0 ~ 31 以外の ASCII 値を持つ終了文字を使用する場合には、長い形式を使用します。まず、この形式のディスクリプタを作成してください。

|                 |                    |   |
|-----------------|--------------------|---|
| 31              | 16 15              | 0 |
| Not Used        | Mask Size in Bytes |   |
| Address of Mask |                    |   |

マスク自体は短い形式の場合と同じ形式です。しかし、長い形式では、短い形式の場合より包括的な終了文字集合を使用できます。たとえば、16 バイトのマスク・サイズの場合には、7 ビットの ASCII 文字を終了文字として設定できますが、マスク・サイズが 32 バイトの場合には、8 ビット文字を終了文字として設定できます。1 ~ 32 バイトのマスク・サイズを使用できます。

終了文字マスクがすべて 0 の場合には、指定された終了文字は存在しません。その場合、読み込み操作は、maximum-length 引数に指定された文字数が転送されたとき、または時間切れが満了したときに、読み込み操作を終了します。

terminator-set 引数を省略した場合には、終了文字集合は OpenVMS の省略時の終了文字集合になります。詳しい説明は、『OpenVMS I/O User's Reference Manual』を参照してください。

character-set 引数に SMG\$C\_KANJI または SMG\$C\_SDK が指定された場合には、かな漢字変換キーに設定されているキーを終了文字として指定しても、終了文字として認識されません。

#### resultant-length

VMS 用法: word\_unsigned  
 データ型: word (unsigned)  
 アクセス: write only  
 受け渡し方: by reference

読み込まれたバイト数と、resultant-string 引数の最大サイズのどちらか小さい方が書き込まれます。resultant-length 引数は、読み込まれたバイト数または最大サイズが書き込まれる符号なしワードのアドレスです。

#### word-terminator-code

VMS 用法: word\_unsigned  
 データ型: word (unsigned)  
 アクセス: write only  
 受け渡し方: by reference

キー終了文字コード。word-terminator-code 引数は、どの文字またはどのキーが読み込み操作を終了したかを示すコードが書き込まれる符号なしワードのアドレスです。キー終了文字コードは SMG\$K\_TRM\_keyname の形式です。キー名については第 3 章の表 3-1 を参照してください。

#### display-id

VMS 用法: identifier  
 データ型: longword (unsigned)  
 アクセス: read only  
 受け渡し方: by reference

ディスプレイ識別子。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。画面管理機能の出力ルーチンを使用しない場合のみ、この引数は省略可能です。

画面管理機能の入力ルーチンと出力ルーチンを使用する場合には、この引数は入力が行われる仮想ディスプレイを指定します。指定される仮想ディスプレイは、keyboard-id によって指定されるペーストボードと同じペーストボードにペーストされていなければならない、他のディスプレイによって隠されていない状態でなければなりません。仮想ディスプレイの隠された領域から入力を受け付けることはできません。

複数の仮想ディスプレイが存在する場合には、各仮想ディスプレイに対して仮想カーソルの位置が割り当てられています。同時に、物理カーソルの現在の位置に対応する物理カーソル位置が1つだけあります。display-id 引数を指定した場合には、読み込み操作は指定された仮想ディスプレイ内の現在の仮想カーソルの位置から開始されます。ディスプレイ識別子を省略した場合には、読み込み操作は現在の物理カーソルの位置から開始されます。prompt-string 引数の長さを入力されるキーの合計は、ディスプレイ内に表示されるカラム数に制限されます。

---

#### 注意

---

この仮想ディスプレイはカラム 1 にペーストしなければならない、その右側に他の仮想ディスプレイを配置することはできません。この制約事項が必要なのは、この条件が満足されない場合、`[Ctrl/R]`または`[Ctrl/U]`の発生によって、右側にペーストされた出力も含めて行全体が消去されるからです。

---

#### initial-string

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | read only        |
| 受け渡し方      | by descriptor    |

初期文字列 (省略可能)。initial-string 引数は、フィールドの初期文字列を格納した省略可能な文字列を示すディスクリプタのアドレスです。文字列はキーボードから入力されたかのように、ディスプレイの入力領域に書き込まれます。文字列は通常の方法で編集できます (TRMSM\_TM\_NOEDIT が設定されていない場合)。文字列長は maximum-length 引数より短くなければなりません。短くない場合には、SMG\$READ\_STRING ルーチンはただちに終了し、キーボード入力は受け付けられません。

#### rendition-set

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する属性が仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

|                                 |                                                             |
|---------------------------------|-------------------------------------------------------------|
| SMG\$M_BLINK                    | 点滅する文字を表示します。                                               |
| SMG\$M_BOLD                     | 通常の輝度より高い輝度で文字を表示します。                                       |
| SMG\$M_REVERSE                  | 文字を反転表示します。つまり、仮想表示の現在のシステム設定値の属性とは逆の状態を表示します。              |
| SMG\$M_UNDERLINE                | 下線を付けた文字を表示します。                                             |
| SMG\$M_INVISIBLE                | 表示されない文字を指定します。つまり、仮想ディスプレイに存在していても、ペーストボードに表示されない文字を指定します。 |
| SMG\$M_USER1 から<br>SMG\$M_USER8 | ユーザ定義属性を表示します。                                              |

rendition-set 引数を使用する場合は、display-id 引数を指定しなければなりません。

#### rendition-complement

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が仮想ディスプレイ上に設定されます。rendition-set 引数で指定できる属性はすべて、rendition-complement 引数で補足指定することができます。

rendition-complement 引数を使用する場合は、display-id 引数を指定しなければなりません。

rendition-set 引数と rendition-complement 引数は省略可能で、これらの引数を使用して仮想ディスプレイの各属性を制御することができます。rendition-set 引数は仮想ディスプレイの特定の属性を設定するのに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットを指定した場合は、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これら 2 つの引数を組み合わせて使用することにより、ユーザは 1 回のプロシージャ呼び出しで仮想ディスプレイの各属性を制御することができます。各属性で次の変換が可能です。

| 設定 | 補足 | 動作                |
|----|----|-------------------|
| 0  | 0  | 属性を省略時の状態に設定する。   |
| 1  | 0  | 属性を有効にする。         |
| 0  | 1  | 属性を省略時の逆の状態に設定する。 |
| 1  | 1  | 属性を無効にする。         |

#### terminator-string

VMS 用法: char\_string  
データ型: character string  
アクセス: write only  
受け渡し方: by descriptor

入出力を終了した文字。terminator-string 引数は省略可能であり、終了文字が書き込まれるディスクリプタのアドレスです。terminator-string 引数は、入出力を終了するために押されたキーではなく、実際の終了文字を戻します。

character-set

VMS 用法: unsigned\_longword  
 データ型: longword (unsigned)  
 アクセス: read only  
 受け渡し方: by reference

入力される文字の文字集合を指定します (省略可能)。character-set 引数は入力される文字集合を格納する符号なしロングワードのアドレスです。この引数が省略された場合には、指定された仮想ディスプレイの省略時の文字集合が使用されます。この引数を省略し、かつ仮想ディスプレイが指定されていない場合には、SMG\$DEFAULT\_CHARACTER\_SET 論理名で示される文字集合が使用されます。

character-set 引数として次の値を指定できます。

| 値                    | 文字集合 (コードセット)       |
|----------------------|---------------------|
| SMG\$C_ASCII         | ASCII 文字集合          |
| SMG\$C_SPEC_GRAPHICS | DEC 特殊文字集合          |
| SMG\$C_SUPPLEMENTAL  | 補助文字集合 <sup>1</sup> |
| SMG\$C_KANJI         | DEC 漢字文字集合          |
| SMG\$C_JIS_KANA      | JIS カタカナ文字集合        |
| SMG\$C_SDK           | Super DEC 漢字コードセット  |

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか

SMG\$DEFAULT\_CHARACTER\_SET 論理名に定義できる値については、第 2.2.14 項の表 2-2 を参照してください。

## 説明

SMG\$READ\_STRING ルーチンは、仮想キーボードから読み込んだ文字列を戻します。SMG\$READ\_STRING ルーチンを使用する場合には、ペーストボードと仮想ディスプレイの両方に対して、ディスプレイのバッチ処理を禁止しておかなければなりません。

SMG\$READ\_STRING が読み込んだテキストは、SMG\$READ\_COMPOSED\_LINE を使用して後で再呼び出しできるように保存されます。表示属性の指定はプロンプト (プロンプト文字列が存在する場合) と入力文字のエコー表示の両方に対して適用されます。

入力文字の文字集合が DEC 漢字または Super DEC 漢字文字集合の場合には、日本語 SMG は、かな漢字変換を行います。

---

 戻される状態値

|                 |                                                      |
|-----------------|------------------------------------------------------|
| SS\$_NORMAL     | 正常終了。                                                |
| SS\$_ABORT      | 実行中に入出力操作が強制終了されました (SMG\$CANCEL_INPUT ルーチンによって)。    |
| SS\$_CANCEL     | キューに登録中に入出力操作が取り消されました (SMG\$CANCEL_INPUT ルーチンによって)。 |
| SMG\$_EOF       | ファイルの最後。                                             |
| SMG\$_ILLBATFNC | バッチ処理されるディスプレイからの入力は認められません。                         |
| SMG\$_INVARG    | 引数が誤っています。                                           |
| SMG\$_INVCOL    | カラム番号に誤りがあります。仮想ディスプレイの外部から入力を読み込もうとしています。           |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。                               |
| SMG\$_INVKBD_ID | keyboard-id 引数に誤りがあります。                              |
| SMG\$_INVKTB_ID | key-table-id 引数に誤りがあります。                             |
| SMG\$_INVMAXLEN | 512 より大きい値が最大バイト数として指定されました。                         |
| SMG\$_WRONUMARG | 引数の数が誤っています。                                         |
| SS\$_xxx        | \$QIOW から戻されたエラー。                                    |
| JSYS\$_xxx      | JSYS\$_xxx から戻されたエラー。                                |
| LIB\$_xxx       | LIB\$SCOPY_R_DX から戻されたエラー。                           |
| RMS\$_xxx       | \$GET から戻されたエラー (RMS\$_EOF を除く)。                     |
| IMS\$_xxx       | IMLIB から戻されたエラー。                                     |

---

 例

```

1.
C+
C This FORTRAN example program demonstrates how to use
C SMG$READ_STRING.
C
C This routine creates a virtual display and writes it to the pasteboard.
C Data is placed in the virtual display using SMG$PUT_CHARS.
C-

C+
C Include the SMG definitions. In particular, we want SMG$M_BORDER.
C-

      IMPLICIT INTEGER (A-Z)
      INCLUDE '($SMGDEF)'
      CHARACTER*20 TEXT

C+
C Create a virtual display with a border using SMG$CREATE_VIRTUAL_DISPLAY.
C-

```

## SMG\$READ\_STRING

```
      ROWS = 7
      COLUMNS = 50
      STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Use SMG$CREATE_PASTEBOARD to create the pasteboard.
C-
      STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Create a virtual keyboard by calling SMG$CREATE_VIRTUAL_KEYBOARD.
C-
      STATUS = SMG$CREATE_VIRTUAL_KEYBOARD ( KEYBOARD1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

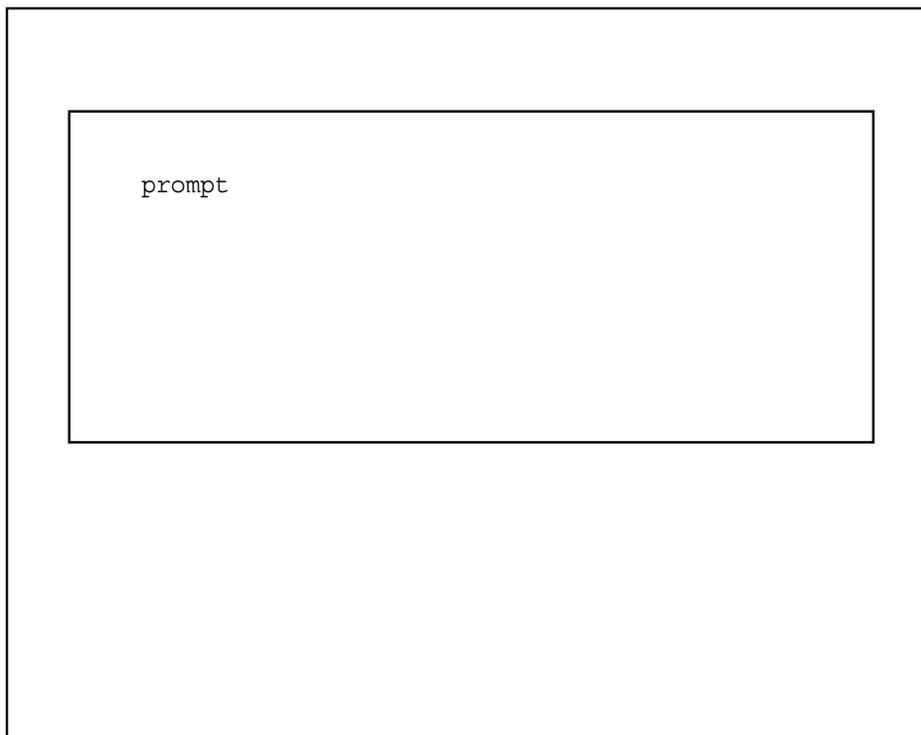
C+
C Use SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual display
C at row 3, column 9.
C-
      STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 3, 9 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Read a string from the virtual pasteboard using SMG$READ_STRING.
C-
      STATUS = SMG$READ_STRING ( KEYBOARD1,
1      TEXT, 'prompt', 20, , , , , DISPLAY1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      END
```

SMG\$READ\_STRING ルーチン呼び出す前に、この FORTRAN プログラムによって生成される出力は図 8-35 に示すとおりです。このプログラムは入力を待ちます。カーソルは "prompt" という単語のすぐ後に移動します。

図 8-35 SMG\$READ\_STRING ルーチン呼び出す前に生成される出力



ZK-4138-GE

SMG\$READ\_STRING ルーチン呼び出した後で生成される出力は、図 8-36 に示すとおりです。

SMG\$READ\_STRING

図 8-36 SMG\$READ\_STRING ルーチン呼び出し後に生成される出力



ZK-4140-GE

---

## SMG\$READ\_VERIFY

### Read and Verify a String

Read and Verify a String ルーチンは、一連の文字列を読み込み、検査します。

---

#### 形式

```
SMG$READ_VERIFY keyboard-id ,resultant-string ,initial-string ,pricture-string ,fill-character
,clear-character [,prompt-string] [,modifiers] [,timeout] [,placeholder-arg]
[,initial-offset] [,word-terminator-code] [,display-id] [,alternate-echo-string]
[,alternate-display-id] [,rendition-set] [,rendition-complement] [,input-length]
[,character-set]
```

---

#### 戻り値

VMS 用法: cond\_value  
 データ型: longword (unsigned)  
 アクセス: write only  
 受け渡し方: by value

---

#### 引数

keyboard-id

VMS 用法: identifier  
 データ型: longword (unsigned)  
 アクセス: write only  
 受け渡し方: by value

キーボード識別子。keyboard-id 引数は、読み込まれる仮想キーボードの識別子を格納した符号なしロングワードのアドレスです。

仮想キーボードは SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンを呼び出すことにより作成されます。

resultant-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス write only  
 受け渡し方 by descriptor

SMG\$READ\_VERIFY ルーチンが読み込んだ文字列が書き込まれる出力文字列。resultant-string 引数は、この出力文字列を示すディスクリプタのアドレスです。

## initial-string

VMS 用法: char\_string  
 データ型: character string  
 アクセス: read only  
 受け渡し方: by descriptor

入力フィールドの初期文字を格納した入力文字列。initial-string 引数は、入力文字列を示すディスクリプタのアドレスです。

## picture-string

VMS 用法: char\_string  
 データ型: character string  
 アクセス: read only  
 受け渡し方: by descriptor

入力フィールドがどのようになるかを示す文字列。picture-string 引数は、ピクチャ文字列を示すディスクリプタのアドレスです。

picture-string 引数に対して指定できる値については、『OpenVMS I/O User's Reference Manual』を参照してください。

character-set に SMG\$C\_KANJI または SMG\$C\_SDK を指定する場合には、picture-string には TRASV\_CV\_ANY 以外の値は指定できません。

## fill-character

VMS 用法: char\_string  
 データ型: character string  
 アクセス: read only  
 受け渡し方: by descriptor

フィル文字。fill-character 引数は、initial-string 引数でフィル文字として使用される文字を格納した文字列を示すディスクリプタのアドレスです。

fill-character 引数についての詳しい説明は、『OpenVMS I/O User's Reference Manual』を参照してください。

## clear-character

VMS 用法: char\_string  
 データ型: character string  
 アクセス: read only  
 受け渡し方: by descriptor

クリア文字。clear-character 引数は、initial-string 引数内の、各 fill-character 引数に対して表示される文字を格納した文字列を示すディスクリプタのアドレスです。

clear-character 引数についての詳しい説明は、『OpenVMS I/O User's Reference Manual』を参照してください。

## prompt-string

VMS 用法: char\_string  
 データ型: character string

アクセス: read only  
受け渡し方: by descriptor

プロンプト文字列。prompt-string 引数は、SMG\$READ\_VERIFY ルーチンの読み込み操作に対するプロンプトとして使用される文字列を示すディスクリプタのアドレスです。これは省略可能な引数です。

#### modifiers

OpenVMS 用法 mask\_longword  
データ型 longowrd (unsigned)  
アクセス read only  
受け渡し方 by reference

修飾子。modifiers 引数は、省略可能な動作を指定するロングワード・ビット・マスクです。定義されているビットは、\$QIO 項目リスト・エントリ TRMS\_MODIFIERS に対するビットと同じです。これは省略可能な引数です。

modifier 引数に対して指定できる値は次のとおりです。

|                    |                                                                                      |
|--------------------|--------------------------------------------------------------------------------------|
| TRMSM_TM_CVTLOW    | 小文字を大文字に変換します。ASCII 文字および補助文字集合についてのみ有効です。                                           |
| TRMSM_TM_NOECHO    | 入力された文字は画面に表示されません。入力文字列の文字集合として DEC 漢字文字集合または Super DEC 漢字が指定された場合には無視されます。         |
| TRMSM_TM_PURGE     | 読み込み操作を実行する前に、先読みを可能にするバッファを消去します。                                                   |
| TRMSM_TM_TRMNOECHO | 終了文字は入力されません。                                                                        |
| TRMSM_TM_NOEDIT    | 拡張編集機能は禁止されます。入力文字列の文字集合として DEC 漢字文字集合または Super DEC 漢字が指定された場合には、行編集は禁止されません。        |
| TRMSM_TM_NORECALL  | 行再呼び出し機能は禁止されます。                                                                     |
| TRMSM_TM_AUTO_TAB  | 入力フィールドが満杯になった時点で、入力が終了します。入力文字列の文字集合として DEC 漢字または Super DEC 漢字文字集合が指定された場合には無視されます。 |
| TRMSM_TM_R_JUST    | 入力は右詰めされます。入力文字列の文字集合として DEC 漢字文字集合または Super DEC 漢字が指定された場合には無視されます。                 |

修飾子についての詳しい説明は、『OpenVMS I/O User's Reference Manual』のターミナル・ドライバの節を参照してください。TRMS シンボルは、HP が提供するシステム・シンボル・ライブラリ内の \$TRMDEF マクロ/モジュールによって定義されています。

#### timeout

OpenVMS 用法 longword\_signed  
データ型 longword (signed)  
アクセス read only  
受け渡し方 by reference

時間切れの値。timeout 引数は、時間切れの値を格納した符号付きロングワードのアドレスです。timeout 引数は省略可能です。timeout 引数を指定した場合には、時間切れの前に入力されたすべての文字、または終了文字が入力される前に入力されたすべ

ての文字はバッファに書き込まれます。timeout 引数を省略した場合には、終了文字が入力されるまで、文字はバッファに書き込まれます。

#### placeholder-arg

VMS 用法: unspecified  
 データ型: unspecified  
 アクセス: read only  
 受け渡し方: unspecified

プレースホルダ引数。OpenVMS ターミナル・ドライバは読み込み/検査操作を実行するときに、この引数を無視します。

#### initial-offset

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

入力文字列オフセット。initial-offset 引数は、プロンプトを表示した後、入力を待つまでの間に出力されるバイト数 (initial-string 引数からのバイト数) を格納した符号付きロングワードのアドレスです。

#### word-terminator-code

VMS 用法: word\_unsigned  
 データ型: word (unsigned)  
 アクセス: write only  
 受け渡し方: by reference

キー終了文字コード。word-terminator-code 引数は、どの文字またはどのキーが読み込み操作を終了するかを示すコードを SMG\$READ\_VERIFY ルーチンが書き込む符号なしワードです。キー終了文字コードは SMG\$K\_TRM\_keyname の形式です。キー名については第 3 章の表 3-1 を参照してください。

#### display-id

VMS 用法: identifier  
 データ型: longword (unsigned)  
 アクセス: read only  
 受け渡し方: by reference

ディスプレイ識別子。display-id 引数は省略可能であり、読み込み操作が実行される仮想ディスプレイの識別子を格納した符号なしロングワードです。

複数の仮想ディスプレイが存在する場合には、各仮想ディスプレイにはそれぞれ仮想カーソルの位置が割り当てられています。同時に、現在の物理カーソルの位置に対応する物理カーソル位置が 1 つだけ存在します。display-id 引数を指定した場合には、読み込み操作は、指定された仮想ディスプレイ内の現在の仮想カーソルの位置から開始されます。prompt-string 引数の長さ、initial-offset 引数、および入力された文字列の合計は、ディスプレイ内の表示可能なカラム数に制限されます。

## alternate-echo-string

VMS 用法: char\_string  
 データ型: character string  
 アクセス: read only  
 受け渡し方: by descriptor

代替エコー表示文字列。alternate-echo-string 引数は、読み込み操作で最初の文字が入力された後にプリントされる文字列です。これは省略可能な引数です。

## alternate-display-id

VMS 用法: identifier  
 データ型: longword (signed)  
 アクセス: read only  
 受け渡し方: by reference

代替ディスプレイ識別子。alternate-display-id 引数は、alternate-echo-string 引数がプリントされる仮想ディスプレイの識別子を格納した符号付きロングワードです。これは省略可能な引数です。この引数を指定した場合には、出力はその仮想ディスプレイの現在の仮想カーソルの位置から開始されます。この引数を省略した場合には、display-id 引数の値が省略時の設定として使用されます。display-id 引数を指定しなかった場合には、出力は現在の物理カーソルの位置から開始されます。

## rendition-set

VMS 用法: mask\_longword  
 データ型: longword (unsigned)  
 アクセス: read only  
 受け渡し方: by reference

属性指定子。rendition-set 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。このビット・マスク内の各属性が設定されている場合には、ディスプレイ内の対応する属性が設定されます。rendition-set 引数を使用して設定できる属性は次のとおりです。

|                              |                                                              |
|------------------------------|--------------------------------------------------------------|
| SMG\$M_BLINK                 | 点滅する文字を表示します。                                                |
| SMG\$M_BOLD                  | 通常の輝度より明るい輝度で文字を表示します。                                       |
| SMG\$M_REVERSE               | 文字を反転表示します。つまり、仮想ディスプレイの省略時の属性と逆の属性を使用して文字を表示します。            |
| SMG\$M_UNDERLINE             | 下線付き文字を表示します。                                                |
| SMG\$M_INVISIBLE             | 表示されない文字を指定します。つまり、仮想ディスプレイ内に存在していても、ペーストボードに表示されない文字を指定します。 |
| SMG\$M_USER1 から SMG\$M_USER8 | ユーザ定義属性を表示します。                                               |

rendition-set 引数を使用する場合には、display-id 引数を指定しなければなりません。

## rendition-complement

VMS 用法: mask\_longword  
 データ型: longword (unsigned)  
 アクセス: read only  
 受け渡し方: by reference

属性補足指定子。rendition-complement 引数は省略可能であり、ロングワード・ビット・マスクのアドレスです。ビット・マスク内に属性が設定されている場合には、ディスプレイ内で対応する属性が補足されます。rendition-set 引数に指定できる属性はすべて、rendition-complement 引数を使用して補足できます。rendition-complement 引数を使用する場合には、display-id 引数を指定しなければなりません。

rendition-set 引数と rendition-complement 引数はどちらも省略可能であり、これらの引数を組み合わせて指定した場合には、ユーザは読み込み操作が実行される仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ特性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。同じビットが rendition-set 引数と rendition-complement 引数の両方に指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの2つの引数を組み合わせて使用すれば、1回のプロシージャ呼び出しで各仮想ディスプレイ属性を制御できます。各属性で次の変換が可能です。

| 設定 | 補足 | 動作                  |
|----|----|---------------------|
| 0  | 0  | 属性を省略時の状態に設定する。     |
| 1  | 0  | 属性をオンに設定する。         |
| 0  | 1  | 属性を現在の設定の逆の状態に設定する。 |
| 1  | 1  | 属性をオフに設定する。         |

#### input-length

OpenVMS 用法 word\_unsigned  
 データ型 word (unsigned)  
 アクセス write only  
 受け渡し方 by reference

読み込んだバイト数が書き込まれます。input-length 引数は、SMG\$READ\_VERIFY ルーチンが読み込んだバイト数が書き込まれる、符号なしワードのアドレスです。

#### character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

入力されるテキストの文字集合。character-set 引数は、文字集合コードを格納した符号なしマスク・ロングワードのアドレスです。この引数を省略した場合は、SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンまたは SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチンで設定された値が使用されます。この引数を省略し、かつ仮想ディスプレイが指定されていない場合には、SMG\$DEFAULT\_CHARACTER\_SET 論理名で示される文字集合が使用されます。

character-set 引数として次の値を指定できます。

| 値                    | 文字集合 (コードセット)       |
|----------------------|---------------------|
| SMG\$C_ASCII         | ASCII 文字集合          |
| SMG\$C_SPEC_GRAPHICS | DEC 特殊文字集合          |
| SMG\$C_SUPPLEMENTAL  | 補助文字集合 <sup>1</sup> |
| SMG\$C_KANJI         | DEC 漢字文字集合          |
| SMG\$C_JIS_KANA      | JIS カタカナ文字集合        |
| SMG\$C_SDK           | Super DEC 漢字コードセット  |

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか

SMG\$DEFAULT\_CHARACTER\_SET 論理名に定義できる値については、第 2.2.14 項の表 2-2 を参照してください。

## 説明

このルーチンは、指定された仮想キーボードから一連の文字を読み込み、そのシーケンスをピクチャ文字列に対して検査します。その後、読み込んだ文字を呼び出し側に戻します。呼び出し側は、終了文字を示すコードを戻すことも指定できます。表示属性の指定はプロンプト (プロンプトが存在する場合) と入力文字列のエコー表示の両方に適用されます。

読み込み/検査操作についての詳しい説明は、『OpenVMS I/O User's Reference Manual』を参照してください。SMG\$READ\_VERIFY ルーチンを使用する場合には、ペーストボードと仮想ディスプレイの両方に対して、ディスプレイのバッチ処理を禁止しておかなければなりません。

入力文字の文字集合が、DEC 漢字または Super DEC 漢字文字集合であった場合には、日本語 SMG はかな漢字変換を行います。

## 戻される状態値

|              |                                                     |
|--------------|-----------------------------------------------------|
| SS\$ _NORMAL | 正常終了。                                               |
| SS\$ _ABORT  | 実行中に入出力操作が強制終了されました (SMG\$CANDEL_INPUT ルーチンによって)。   |
| SS\$ _CANCEL | キューに登録中に入力操作が取り消されました (SMG\$CANCEL_INPUT ルーチンによって)。 |

## SMG\$READ\_VERIFY

|                 |                                                                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SMG\$_DISREQ    | SMG\$READ_VERIFY ルーチンを呼び出したときに、右詰めが指定されました。display-id 引数は、指定されませんでした。TERMTABLE.EXE からこの漢字ターミナルに対して SCROLL_REVERSE シーケンスが検索されませんでした。display-id 引数を SMG\$READ_VERIFY ルーチンの呼び出しに追加するか、または SCROLL_REVERSE シーケンスを TERMTABLE.EXE に追加してください。 |
| SMG\$_EOF       | ファイルの最後。                                                                                                                                                                                                                              |
| SMG\$_INVCOL    | カラム番号の指定に誤りがあります。                                                                                                                                                                                                                     |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。                                                                                                                                                                                                                |
| SMG\$_INVKBD_ID | keyboard-id 引数に誤りがあります。                                                                                                                                                                                                               |
| SMG\$_LENNOTEQL | picture-string 引数と initial-string 引数の長さが等しくありません。                                                                                                                                                                                     |
| SMG\$_LENMUSONE | fill-character 引数と clear-character 引数の長さは 1 でなければなりません。                                                                                                                                                                               |
| SMG\$_WRONUMARG | 引数の数が誤っています。                                                                                                                                                                                                                          |
| SS\$_xxx        | \$QIOW から戻されたエラー。                                                                                                                                                                                                                     |
| JSY\$_xxx       | JSY\$から戻されたエラー。                                                                                                                                                                                                                       |
| LIB\$_xxx       | LIB\$SCOPY_R_DX から戻されたエラー。                                                                                                                                                                                                            |
| RMS\$_xxx       | \$GET から戻されたエラー (RMS\$_EOF を除く)。                                                                                                                                                                                                      |
| IM\$_xxx        | IMLIB から戻されたエラー。                                                                                                                                                                                                                      |

---

# SMG\$REMOVE\_LINE

## Remove a Line from a Virtual Display

Remove a Line from a Virtual Display ルーチンは、SMG\$DRAW\_LINE または SMG\$DRAW\_RECTANGLE ルーチンを使用して作成したラインを指定された仮想ディスプレイから削除します。

---

### 形式

SMG\$REMOVE\_LINE *display-id ,start-row ,start-column ,end-row ,end-column*

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

ラインを削除する仮想ディスプレイを指定します。display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンで与えられます。

start-row

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

ラインの削除を開始する行を指定します。start-row 引数は、行番号を格納した符号付きロングワードのアドレスです。

start-column

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

ラインの削除を開始するカラムを指定します。start-column 引数は、カラム番号を格納した符号付きロングワードのアドレスです。

水平線削除の場合で start-column 引数が全角文字の右半分上であった場合、残る左半分の部分は未定義文字になります。垂直線削除の場合で、start-column 引数が全角文字上にあった場合には、それぞれの残る部分は未定義文字になります。

end-row

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

ラインの削除を終了する行を指定します。end-row 引数は、行番号を格納した符号付きロングワードのアドレスです。

end-column

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

ラインの削除を終了するカラムを指定します。end-column 引数は、カラム番号を格納した符号付きロングワードのアドレスです。

水平線削除の場合で end-column 引数が全角文字の左半分上であった場合、残る右半分の部分は未定義文字になります。垂直線削除の場合で、end-column 引数が全角文字上にあった場合には、それぞれの残る部分は未定義文字になります。

---

## 説明

SMG\$REMOVE\_LINE ルーチンは、SMG\$DRAW\_LINE または SMG\$DRAW\_RECTANGLE ルーチンを使用して作成したラインを、指定された開始行/カラムから指定された終了行/カラムまで削除します。

このルーチンはユーザが指定したラインを消去しますが、ラインの交点にあるライン描画文字は残されます (ライン描画文字は漢字ターミナルのライン描画文字集合です。この文字集合を使用できない場合には、+、-、および | が使用されます)。

DEC 漢字または Super DEC 漢字文字集合が使用されており、描画されたラインが全角文字の上にあった場合には、その全角文字の残りの部分は未定義文字になります。

---

戻される状態値

|                 |                                      |
|-----------------|--------------------------------------|
| SS\$_NORMAL     | 正常終了。                                |
| SMG\$_DIALINNOT | 斜めのラインは削除できません。                      |
| SMG\$_INVCOL    | カラム番号が誤っています。指定されたカラムは仮想ディスプレイの外部です。 |
| SMG\$_INVROW    | 行番号が誤っています。指定された行は仮想ディスプレイの外部です。     |
| SMG\$_WRONUMARG | 引数の数が誤っています。                         |

---

# SMG\$REPAINT\_SCREEN

## Repaint Current Pasteboard

Repaint Current Pasteboard ルーチンは、日本語 SMG ルーチンによるもの以外の入出力が実行された後、指定されたペーストボードを描き直します。

---

### 形式

SMG\$REPAINT\_SCREEN *pasteboard-id*

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

*pasteboard-id*

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

リペイントされるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

---

### 説明

SMG\$REPAINT\_SCREEN ルーチンは、指定されたペーストボードを描き直します。このルーチンは、外部エージェント (たとえば、ブロードキャスト・メッセージ) がペーストボードを乱したときに使用します。

---

 戻される状態値

|                 |                           |
|-----------------|---------------------------|
| SS\$_NORMAL     | 正常終了。                     |
| SMG\$_INVPAS_ID | pasteboard-id 引数に誤りがあります。 |
| SMG\$_WRONUMARG | 引数の数が誤っています。              |

---

 例

```

1.
C+
C This FORTRAN example program demonstrates
C the use of SMG$REPAINT_SCREEN.
C-

      IMPLICIT INTEGER (A-Z)

C+
C Create the virtual display by calling
C SMG$CREATE_VIRTUAL_DISPLAY. To create
C a border, we set BORDER = 1. No border
C would be BORDER = 0.
C-

      INCLUDE '($SMGDEF)'
      ROWS = 3
      COLUMNS = 50
      BORDER = 1

      STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, BORDER)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Create the pasteboard using SMG$CREATE_PASTEBOARD.
C-

      STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Put data in the virtual display by calling SMG$PUT_CHARS.
C-

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This virtual display has 3 rows and 50 columns.', 1, 1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This is a bordered virtual display.', 2, 1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' SMG$PUT_CHARS puts data in this virtual display.', 3, 1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
  
```

## SMG\$REPAINT\_SCREEN

```
C+
C Call SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual display.
C-
      STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

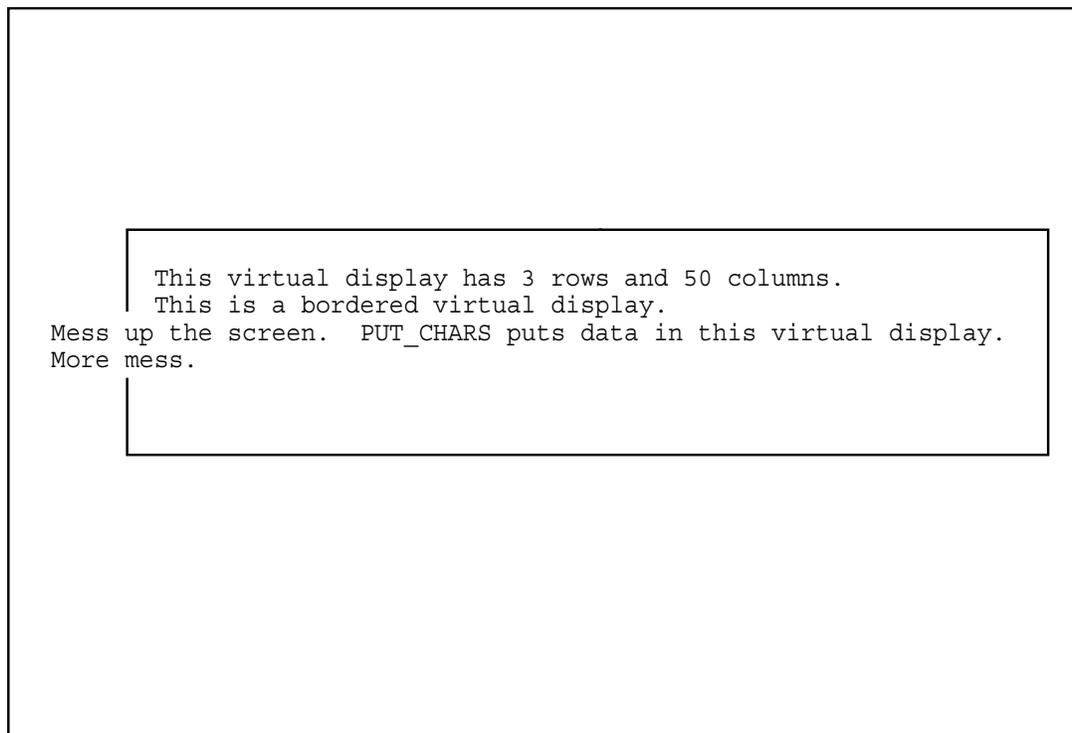
C+
C Mess up the screen with some FORTRAN output.
C-
      WRITE (6,*) 'Mess up the screen.'
      WRITE (6,*) 'More mess.'

C+
C Call SMG$REPAINT_SCREEN to repaint the screen.
C-
      STATUS = SMG$REPAINT_SCREEN ( PASTE1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      END
```

SMG\$REPAINT\_SCREEN ルーチン呼び出す前にこの FORTRAN プログラムが生成する出力は 図 8-37 に示すとおりです。

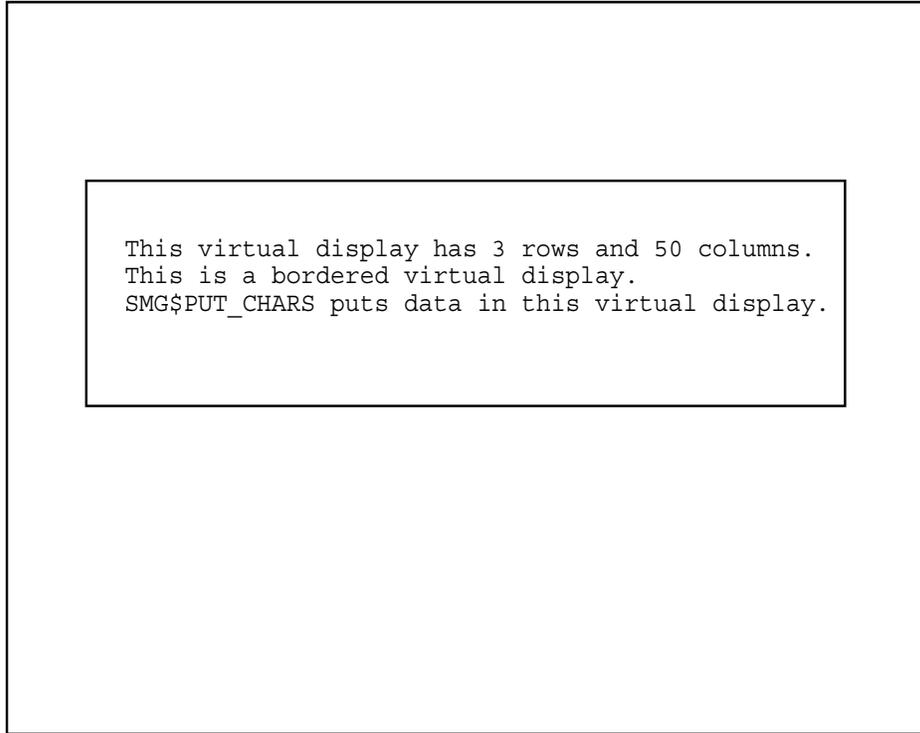
図 8-37 SMG\$REPAINT\_SCREEN ルーチン呼び出す FORTRAN プログラムが作成する出力



ZK-4136-GE

SMG\$REPAINT\_SCREEN ルーチン呼び出し後で生成される出力は 図 8-38 に示すとおりです。

図 8-38 SMG\$REPAINT\_SCREEN ルーチン呼び出す FORTRAN プログラムが生成する出力



ZK-4137-GE

---

# SMG\$REPASTE\_VIRTUAL\_DISPLAY

## Repaste Virtual Display

Repaste Virtual Display ルーチンは、仮想ディスプレイをペーストボードの新しい位置に移動します。ペースト順序も変更されます。

---

### 形式

SMG\$REPASTE\_VIRTUAL\_DISPLAY

*display-id ,pasteboard-id ,pasteboard-row ,pasteboard-column  
[,top-display-id]*

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

再ペーストする仮想ディスプレイを指定します。display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

仮想ディスプレイを再ペーストするペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

## pasteboard-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

指定された仮想ディスプレイの位置を示すペーストボードの行番号を指定します。pasteboard-row 引数は、ペーストボード行番号を格納した符号付きロングワードのアドレスです。

## pasteboard-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

指定された仮想ディスプレイの位置を示すペーストボード・カラム番号を指定します。pasteboard-column 引数は、ペーストボードのカラム番号を格納した符号付きロングワードのアドレスです。

## top-display-id

OpenVMS 用法 identifier  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

display-id 引数によって指定される仮想ディスプレイがペーストされるときに、その上になる仮想ディスプレイの識別子(省略可能)。top-display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。top-display-id 引数によって指定される仮想ディスプレイはすでにペーストされていなければなりません。

---

 説明

SMG\$REPASTE\_VIRTUAL\_DISPLAY ルーチンを使用すれば、仮想ディスプレイをペーストボードの新しい位置に移動できます。このルーチンは SMG\$UNPASTE\_VIRTUAL\_DISPLAY ルーチンと SMG\$PASTE\_VIRTUAL\_DISPLAY ルーチンを呼び出します。このルーチンを実行すると、ペースト順序も変更されます。再ペースト操作が終了するまで、画面は変更されません。

このルーチンを使用すると、仮想ディスプレイはペースト順序の一番上になります。ペースト順序を変更せずに、仮想ディスプレイを移動する場合には、SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチンを使用します。top-display-id 引数は省略可能であり、この引数を指定した場合には、SMG\$REPASTE\_VIRTUAL\_DISPLAY ルーチンは、top-display-id 引数によって指定される仮想ディスプレイの下に、display-id 引数によって指定される仮想ディスプレイをペーストします。この場合、top-display-id 引数によって指定される仮想ディスプレイは、すでにペーストされていなければなりません。

---

 戻される状態値

|                  |                           |
|------------------|---------------------------|
| SS\$ _NORMAL     | 正常終了。                     |
| SMG\$ _INVDIS_ID | display-id 引数に誤りがあります。    |
| SMG\$ _INVPAS_ID | pasteboard-id 引数に誤りがあります。 |
| SMG\$ _WRONUMARG | 引数の数が誤っています。              |

---

## 例

```

1.
C+
C This FORTRAN example program demonstrates the use of
C SMG$REPASTE_VIRTUAL_DISPLAY and SMG$MOVE_VIRTUAL_DISPLAY.
C-

      IMPLICIT INTEGER (A-Z)

C+
C Include the SMG definitions. In particular, we want SMG$M_BORDER.
C-

      INCLUDE '($SMGDEF)'

C+
C Create a virtual display with a border by calling
C SMG$CREATE_VIRTUAL_DISPLAY.
C-

      ROWS = 3
      COLUMNS = 50

      STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$CREATE_PASTEBOARD to create the pasteboard.
C-

      STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Put data in the virtual display using SMG$PUT_CHARS.
C-

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This virtual display has 3 rows and 50 columns.', 1, 1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' This is a bordered virtual display.', 2, 1 )
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

```

```

        STATUS = SMG$PUT_CHARS ( DISPLAY1,
1      ' SMG$PUT_CHARS puts data in this virtual display.', 3, 1 )
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$PASTE_VIRTUAL_DISPLAY to paste the virtual display.
C-

        STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15 )
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Move the virtual display by calling SMG$MOVE_VIRTUAL_DISPLAY.
C-

        STATUS = SMG$MOVE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 10, 5 )
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C+
C Call SMG$REPASTE_VIRTUAL_DISPLAY to repaste the
C original virtual display as it was.
C-

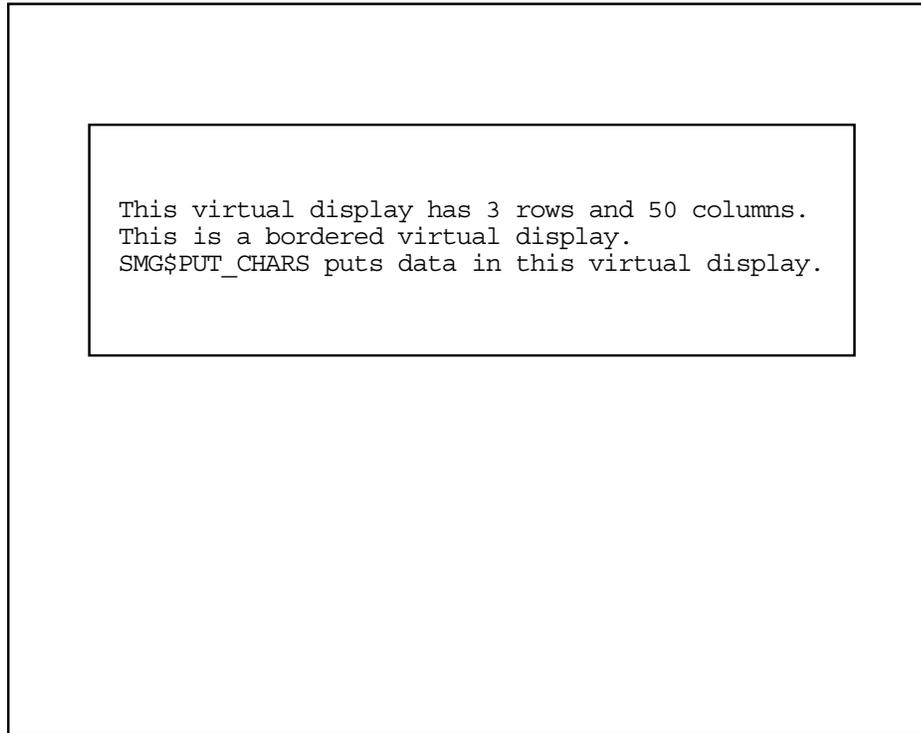
        STATUS = SMG$REPASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 15 )
        IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

        END

```

SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチンを呼び出す前に、この FORTRAN プログラムが生成する出力は、図 8-39 に示すとおりです。

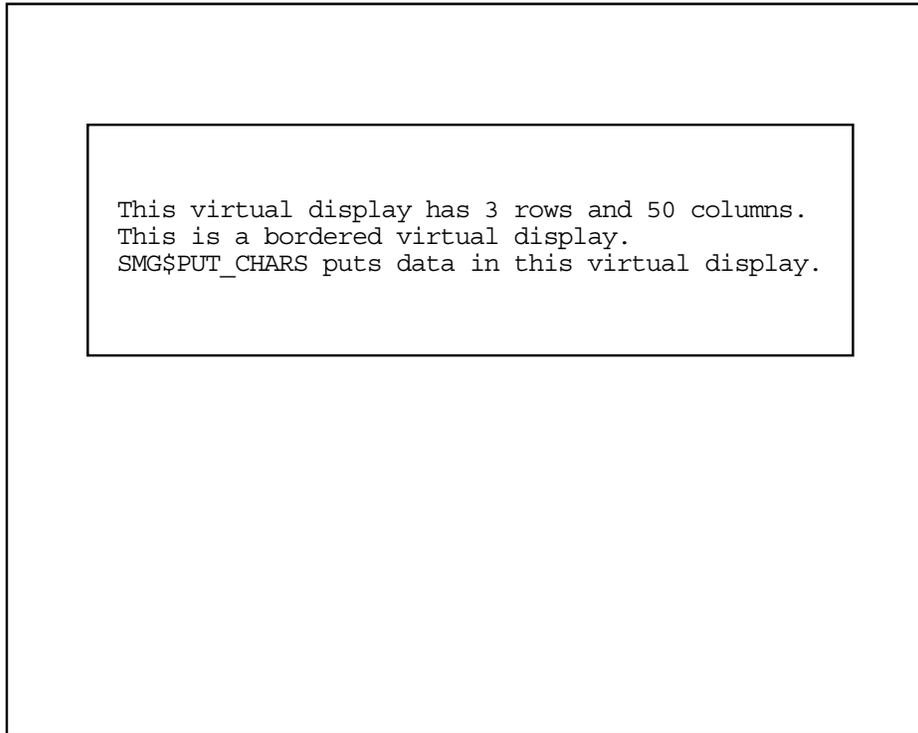
図 8-39 SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチン呼び出す前の出力



ZK-4139-GE

SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチン呼び出した後の出力は、図 8-40 に示すとおりです。

図 8-40 SMG\$MOVE\_VIRTUAL\_DISPLAY ルーチン呼び出した後に表示される出力

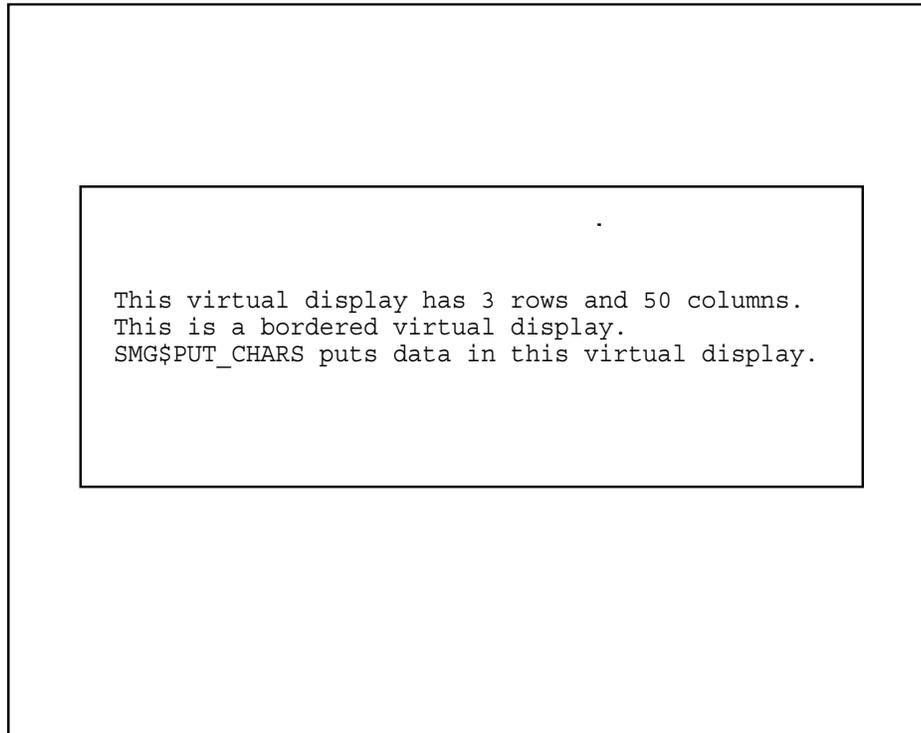


ZK-4141-GE

図 8-41 は、SMG\$REPASTE\_VIRTUAL\_DISPLAY ルーチン呼び出した後に表示される最終的な出力を示しています。

# SMG\$REPASTE\_VIRTUAL\_DISPLAY

図 8-41 SMG\$REPASTE\_VIRTUAL\_DISPLAY ルーチン呼び出し後に表示される出力



ZK-4130-GE

---

# SMG\$REPLACE\_INPUT\_LINE

## Replace Input Line

Replace Input Line ルーチンは、再呼び出しバッファ内の指定された行を指定された文字列に置き換えます。

---

### 形式

SMG\$REPLACE\_INPUT\_LINE *keyboard-id* [,*replace-string*] [,*line-count*] [,*flags*] [,*character-set*]

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

keyboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

キーボード識別子。keyboard-id 引数は、読み取る対象となる仮想キーボードの識別子を格納した符号なしロングワードのアドレスです。

この引数は、SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンによって与えられません。

replace-string

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | read only        |
| 受け渡し方      | by descriptor    |

再呼び出しバッファに入力される行を格納した文字列。replace-string 引数は、この文字列を指すディスクリプタのアドレスです。省略時の設定は空文字列であり、最後に入力された行を削除します。

## line-count

OpenVMS 用法 byte\_unsigned  
 データ型 byte (unsigned)  
 アクセス read only  
 受け渡し方 by reference

置換する行数。line-count 引数は、replace-string 引数に置換する行数を格納した符号なしバイトのアドレスです。line-count 引数の省略時の値は 1 (最後に入力した行) です。

## flags

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

任意に実行できる動作を指定します。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。指定できる値は以下のとおりです。

SMG\$M\_KEEP\_CONTENTS replace-string 引数によって指定された文字列は、line-count 引数によって指定された既存の文字列と置き換えられます。このフラグをループ内で使用すれば、アプリケーションから再呼び出しバッファに情報をプリロードすることができます。

## character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

文字集合指定子。再呼び出しバッファに入力される行、replace-string 引数の文字集合。character-set 引数は、文字集合コードを格納した符号なしロングワードのアドレスです。この引数を省略した場合のシステム設定値は、UNKNOWN です。

| 値                    | 文字集合 (コードセット)       |
|----------------------|---------------------|
| SMG\$C_ASCII         | ASCII 文字集合          |
| SMG\$C_SPEC_GRAPHICS | DEC 特殊文字集合          |
| SMG\$C_SUPPLEMENTAL  | 補助文字集合 <sup>1</sup> |
| SMG\$C_KANJI         | DEC 漢字文字集合          |
| SMG\$C_JIS_KANA      | JIS カタカナ文字集合        |
| SMG\$C_SDK           | Super DEC 漢字コードセット  |

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

## 説明

SMG\$REPLACE\_INPUT\_LINE ルーチンは、再呼び出しバッファ内の要求された行

を指定された文字列に置き換えます。残りの行 (line-count-1) は削除されます。このルーチンは行継続の処理を補助するためのものです。

---

## 戻される状態値

|                |                         |
|----------------|-------------------------|
| SS\$NORMAL     | 正常終了。                   |
| LIB\$INSVIRMEM | 仮想メモリが不足しています。          |
| SMG\$INVKBD_ID | keyboard-id 引数に誤りがあります。 |
| SMG\$WRONUMARG | 引数の数が誤っています。            |

---

## 例

```

1.
!+
! This FORTRAN example uses the routine
! SMG$REPLACE_INPUT_LINE to concatenate
! the last 2 lines in the recall buffer.
!-

IMPLICIT INTEGER (A-Z)
INCLUDE '($SMGDEF)'
INCLUDE '($SMGMSG)'
CHARACTER*20 TEXT, TEXT1

WRITE (5,*) 'Enter number of lines to save.'
READ (5,*) R

S = SMG$CREATE_PASTEBOARD(PBID)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$CREATE_VIRTUAL_DISPLAY(22,70,DID,SMG$M_BORDER)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$CREATE_VIRTUAL_KEYBOARD(KBID,,,R)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$PUT_LINE(DID,'Enter lines of text:')
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$PASTE_VIRTUAL_DISPLAY(DID,PBID,2,2)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))

!+
! Read in lines of text.
!-
DO 10 I = 1,R
    S = SMG$READ_COMPOSED_LINE(KBID,,TEXT,'Example>',,DID)
    IF (.NOT. S) CALL LIB$STOP(%VAL(S))
10 CONTINUE

!+
! Recall last 2 lines in the buffer.
!-

```

## SMG\$REPLACE\_INPUT\_LINE

```
S = SMG$RETURN_INPUT_LINE(KBID,TEXT,,1,LEN)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$RETURN_INPUT_LINE(KBID,TEXT1,,2,LEN1)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
TEXT(LEN:20) = TEXT1(1:LEN1)

!+
! Concatenate them and replace the last 2
! lines in the buffer with the new line.
!-

S = SMG$REPLACE_INPUT_LINE(KBID,TEXT,2)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))

!+
! Recall the last line which is now the
! concatenated line.
!-

S = SMG$RETURN_INPUT_LINE(KBID,TEXT,,1)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))

S = SMG$PUT_LINE(DID,'**** The last line of text is:')
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$PUT_LINE(DID,TEXT)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))

END
```

以下の例はこの FORTRAN プログラムによって生成される出力を示しています。

```
$ RUN REPLACE
Enter number of lines to save.
  3
Enter lines of text:
Example> PASTEBOARD
Example> DISPLAY
Example> KEYBOARD
****The last line of text is:
KEYBOARDDISPLAY
```

---

# SMG\$RESTORE\_PHYSICAL\_SCREEN

## Restore Physical Pasteboard

Restore Physical Pasteboard ルーチンは、SMG\$SAVE\_PHYSICAL\_SCREEN ルーチン呼び出し時の状態にペーストボード・イメージを復元します。

---

### 形式

SMG\$RESTORE\_PHYSICAL\_SCREEN *pasteboard-id* ,*display-id*

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

復元するペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

SMG\$SAVE\_PHYSICAL\_SCREEN ルーチンによって作成された仮想ディスプレイを指定します。display-id 引数は、この仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

---

## 説明

SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチンは、SMG\$SAVE\_PHYSICAL\_SCREEN ルーチンによって保存されたペーストボード・イメージを再作成します。SMG\$SAVE\_PHYSICAL\_SCREEN ルーチンから戻された display-id 引数を SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチンに渡さなければなりません。SMG\$SAVE\_PHYSICAL\_SCREEN ルーチンと SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチンを 2 回以上呼び出す場合には、呼び出しはネスティング方式で実行しなければなりません。つまり、最後に保存したペーストボードは最初に復元しなければなりません。

---

## 戻される状態値

|                 |                           |
|-----------------|---------------------------|
| SS\$_NORMAL     | 正常終了。                     |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。    |
| SMG\$_INVPA\$ID | pasteboard-id 引数に誤りがあります。 |

---

## SMG\$RETURN\_CURSOR\_POS

### Return Cursor Position

Return Cursor Position ルーチンは、指定された仮想ディスプレイの仮想カーソルの現在の位置を戻します。

---

#### 形式

SMG\$RETURN\_CURSOR\_POS *display-id ,start-row ,start-column*

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

現在の仮想カーソルの位置を得たい仮想ディスプレイを指定します。display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | write only        |
| 受け渡し方      | by reference      |

指定された仮想ディスプレイ内の仮想カーソルの現在の行位置が書き込まれます。start-row 引数は、現在の行位置が書き込まれる符号付きロングワードのアドレスです。

## SMG\$RETURN\_CURSOR\_POS

start-column

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | write only        |
| 受け渡し方      | by reference      |

指定された仮想ディスプレイ内の仮想カーソルの現在のコラム位置が書き込まれます。start-column 引数は、現在のコラム位置が書き込まれる符号付きロングワードのアドレスです。

---

### 説明

SMG\$RETURN\_CURSOR\_POS ルーチンは、指定された仮想ディスプレイ内の仮想カーソルの現在の行位置とコラム位置を戻します。

---

### 戻される状態値

|                 |                        |
|-----------------|------------------------|
| SS\$_NORMAL     | 正常終了。                  |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。 |
| SMG\$_WRONUMARG | 引数の数が誤っています。           |

---

## SMG\$RETURN\_INPUT\_LINE

### Return Input Line

The Return Input Line ルーチンは、再呼び出しバッファ内の要求された行を、呼び出し側に戻します。この行は、指定された文字列に適合させる、あるいは適切な行数を指定することによって回復します。

---

#### 形式

```
SMG$RETURN_INPUT_LINE keyboard-id ,resultant-string [,match-string] [,byte-integer-line-number]
                        [,resultant-length] [,character-set]
```

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

keyboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

仮想キーボードの識別子。keyboard-id 引数は、読み取る対象となる仮想キーボードの識別子を格納した符号なしロングワードのアドレスです。

この引数は、SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンによって与えられます。

resultant-string

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | write only       |
| 受け渡し方      | by descriptor    |

SMG\$RETURN\_INPUT\_LINE ルーチンが、最終的に作成した行を書き込む文字列。resultant-string 引数は、生成される行が書き込まれる文字列を示すディスクリプタのアドレスです。

## match-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス read only  
 受け渡し方 by descriptor

再呼び出しする行を検索するときを使用される照合文字列。match-string 引数は省略可能であり、この照合文字列を指すディスクリプタのアドレスです。検索操作は最後に入力した行から開始されます。

## byte-integer-line-number

OpenVMS 用法 byte\_unsigned  
 データ型 byte (unsigned)  
 アクセス read only  
 受け渡し方 by reference

呼び出す行を検索するときを使用される行番号。byte-integer-line-number 引数は省略可能であり、再呼び出しする行の行番号を格納した符号なしバイトのアドレスです。最後に入力した行の行番号は 1 です。

## resultant-length

OpenVMS 用法 word\_unsigned  
 データ型 word (unsigned)  
 アクセス write only  
 受け渡し方 by reference

resultant-string 文字列の長さ。読み込まれたバイト数と、resultant-string 引数の最大サイズのどちらか小さい方が書き込まれます。resultant-length 引数は、読み込まれたバイト数または最大サイズが書き込まれる符号なしワードのアドレスです。

## character-set

OpenVMS 用法 longword\_unsigned  
 データ型 longword (unsigned)  
 アクセス write only  
 受け渡し方 by reference

再呼び出しバッファから戻された行の文字集合。character-set 引数は、文字集合コードを格納する符号なしロングワードのアドレスです (省略可能)。文字集合が未定義の場合には、SMGSC\_UNKOWN が戻されます。

| 値                   | 文字集合 (コードセット)       |
|---------------------|---------------------|
| SMGSC_ASCII         | ASCII 文字集合          |
| SMGSC_SPEC_GRAPHICS | DEC 特殊文字集合          |
| SMGSC_SUPPLEMENTAL  | 補助文字集合 <sup>1</sup> |
| SMGSC_KANJI         | DEC 漢字文字集合          |
| SMGSC_JIS_KANA      | JIS カタカナ文字集合        |
| SMGSC_UNKNOWN       | 未定義文字集合             |
| SMGSC_SDK           | Super DEC 漢字コードセット  |

<sup>1</sup>DEC 補助文字集合または Latin-1 文字集合のいずれか。

---

 説明

SMG\$RETURN\_INPUT\_LINE ルーチンは、再呼び出しバッファ内の指定された行を、呼び出し側に戻します。このルーチンは、DCL 形式の "RECALL" コマンドを実現するための補助として使用されます。

match-string 引数を指定した場合には、SMG\$RETURN\_INPUT\_LINE ルーチンは、指定された文字列と一致する行を検索し、その行に戻します。byte-integer-line-number 引数を指定した場合には、SMG\$RETURN\_INPUT\_LINE ルーチンは、指定された行番号に対応する行に戻します。match-string 引数と byte-integer-line-number 引数をどちらも指定した場合には、SMG\$\_INVARG が戻されません。match-string 引数を指定したときに、一致する文字列を検索できない場合には、SMG\$\_LINNOTFND が戻されます。

---

 戻される状態値

|                 |                           |
|-----------------|---------------------------|
| SS\$_NORMAL     | 正常終了。                     |
| SMG\$_INVARG    | 引数に誤りがあります。               |
| SMG\$_INVKBD_ID | keyboard-id 引数に誤りがあります。   |
| SMG\$_LINNOTFND | 一致する行を検索できませんでした。         |
| SMG\$_WRONUMARG | 引数の数が誤っています。              |
| LIB\$_xxx       | LIB\$COPY_R_DX から戻されたエラー。 |

---

 例

```

1.
  !+
  ! This FORTRAN example uses the routine
  ! SMG$RETURN_INPUT_LINE to implement a
  ! RECALL/ALL command.
  !-

  IMPLICIT INTEGER (A-Z)
  INCLUDE '($SMGDEF)'
  INCLUDE '($SMGMSG)'
  CHARACTER*20 TEXT

  WRITE (5,*) 'Enter number of lines to save.'
  READ  (5,*) R
  
```

## SMG\$RETURN\_INPUT\_LINE

```

S = SMG$CREATE_PASTEBOARD(PBID)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$CREATE_VIRTUAL_DISPLAY(22,70,DID,SMG$M_BORDER)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$CREATE_VIRTUAL_KEYBOARD(KBID,,,R)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$PUT_LINE(DID,'Enter lines of text:')
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$PASTE_VIRTUAL_DISPLAY(DID,PBID,2,2)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))

!+
! Read in lines of text.
!-
DO 10 I = 1,R
    S = SMG$READ_COMPOSED_LINE(KBID,,TEXT,'Example>',,DID)
    IF (.NOT. S) CALL LIB$STOP(%VAL(S))
    10 CONTINUE

S = SMG$PUT_LINE(DID,'**** The lines of text are:')
IF (.NOT. S) CALL LIB$STOP(%VAL(S))

!+
! Recall all lines in the buffer.
!-
DO 30 N = 1,R
    S = SMG$RETURN_INPUT_LINE(KBID,TEXT,,N)
    IF (.NOT. S) CALL LIB$STOP(%VAL(S))
    S = SMG$PUT_LINE(DID,TEXT)
    IF (.NOT. S) CALL LIB$STOP(%VAL(S))
    30 CONTINUE

!+
! Recall the line containing 'fox'
!-
S = SMG$PUT_LINE(DID,'**** The line containing "fox" is:',2)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))
S = SMG$RETURN_INPUT_LINE(KBID,TEXT,'FOX')
IF (S .EQ. SMG$_LNNOTFND) TEXT = 'None found!'
S = SMG$PUT_LINE(DID,TEXT)
IF (.NOT. S) CALL LIB$STOP(%VAL(S))

END

```

以下の例は、この FORTRAN プログラムによって生成される出力を示しています。

```
$ RUN RETURN
Enter number of lines to save.
  3
Enter lines of text:
Example> PASTEBOARD
Example> DISPLAY
Example> KEYBOARD
****The lines of text are:
KEYBOARD
DISPLAY
PASTEBOARD
****The line containing "fox" is:

None found!
```

---

## SMG\$RING\_BELL

### Ring the Terminal Bell or Buzzer

Ring the Terminal Bell or Buzzer ルーチンは、漢字ターミナルのベルまたはブザーを鳴らします。

---

#### 形式

SMG\$RING\_BELL *display-id* [*number-of-times*]

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

ベルまたはブザーを鳴らす仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

number-of-times

|            |                           |
|------------|---------------------------|
| OpenVMS 用法 | longword_signed           |
| データ型       | longword integer (signed) |
| アクセス       | read only                 |
| 受け渡し方      | by reference              |

ベルまたはブザーを鳴らす回数を指定します。number-of-times 引数は、ベルまたはブザーを鳴らす回数を格納した符号付きロングワード整数のアドレスです。number-of-times 引数を省略した場合には、値は 1 になります。

---

## 説明

SMG\$RING\_BELL ルーチンは、指定された仮想ディスプレイがペーストされている各ペーストボードでベルまたはブザーを鳴らします。ベルまたはブザーは、指定された回数だけ鳴ります。省略時には 1 回だけ鳴ります。

---

## 戻される状態値

|                |                                              |
|----------------|----------------------------------------------|
| SS\$NORMAL     | 正常終了。                                        |
| SMG\$INVDIS_ID | display-id 引数に誤りがあります。                       |
| その他            | \$QIOW, LIB\$GET_VM, LIB\$FREE_VM から戻される状態値。 |

---

# SMG\$SAVE\_PHYSICAL\_SCREEN

## Save Physical Screen

Save Physical Screen ルーチンは、後で SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチン呼び出すことにより復元できるように、ペーストボードの内容を保存します。

---

### 形式

SMG\$SAVE\_PHYSICAL\_SCREEN *pasteboard-id* *display-id* [*desired-start-row*] [*desired-end-row*]

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

内容が保存されるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by reference        |

指定されたペーストボードの内容を格納するために作成された仮想ディスプレイの仮想ディスプレイ識別子を受け付けます。display-id 引数は、仮想ディスプレイ識別子を書き込まれる符号なしロングワードのアドレスです。

display-id 引数は、保存されている情報を回復するときに SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチンに渡さなければなりません。

desired-start-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

保存する最初の行を指定します。desired-start-row 引数は、行番号を格納した符号付きロングワードのアドレスです。desired-start-row 引数を省略した場合には、ペーストボードの 1 行目が使用されます。

desired-end-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

保存する最後の行を指定します。desired-end-row 引数は、行番号を格納した符号付きロングワードのアドレスです。desired-end-row 引数を省略した場合には、ペーストボードの最後の行が使用されます。

---

## 説明

SMG\$SAVE\_PHYSICAL\_SCREEN ルーチンは、指定されたペーストボードと同じ幅で、desired-start-row 引数と desired-end-row 引数によって指定された高さの仮想ディスプレイを作成することにより、画面を消去します。これらの 2 つの引数を省略した場合には、作成される仮想ディスプレイの高さは、指定されたペーストボードの高さと同じになります。

作成された仮想ディスプレイは省略時の文字集合を持たないので、仮想ディスプレイへの出力表示は、文字集合について考慮しなければなりません。

保存される情報、つまり、ペーストボード・イメージは、SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチンを呼び出すことにより復元できます。SMG\$SAVE\_PHYSICAL\_SCREEN ルーチンと SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチンを 2 回以上呼び出す場合には、呼び出しはネスティング方式で実行しなければなりません。つまり、最後に保存したペーストボードは最初に復元しなければなりません。

これらのルーチンは、日本語 SMG 機能を使用せずに出力を画面に送信する可能性のあるプロシージャを呼び出すときに便利です。このようなプロシージャを呼び出す場合には、その前に SMG\$SAVE\_PHYSICAL\_SCREEN ルーチンを使用してペーストボード・イメージを保存します。プロシージャを実行した後、SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチンを使用してペーストボード・イメージを復元します。

## SMG\$SAVE\_PHYSICAL\_SCREEN

保存する領域は少なくとも 2 行以上でなければなりません。スクロール領域をサポートしないターミナルで SMG\$SAVE\_PHYSICAL\_SCREEN ルーチンを使用する場合には、ペーストボード全体を保存し、復元しなければなりません。

---

### 戻される状態値

|                 |                                      |
|-----------------|--------------------------------------|
| SS\$_NORMAL     | 正常終了。                                |
| LIB\$_INSVIRMEM | 仮想メモリが不足しています。                       |
| SMG\$_INVPAS_ID | pasteboard-id 引数に誤りがあります。            |
| SMG\$_INVROW    | 指定された行番号が誤っているか、または指定された範囲が 1 行以下です。 |
| SMG\$_WRONUMARG | 引数の数が誤っています。                         |

---

# SMG\$SAVE\_VIRTUAL\_DISPLAY

## Save the Virtual Display to a File

Save the Virtual Display to a File ルーチンは、仮想ディスプレイの内容をファイルに格納します。

---

### 形式

SMG\$SAVE\_VIRTUAL\_DISPLAY *display-id* [*filespec*]

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

保存する仮想ディスプレイを指定します。display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

filespec

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | read only        |
| 受け渡し方      | by descriptor    |

指定された仮想ディスプレイを保存するファイルのファイル指定を格納した文字列。filespec 引数は、ファイル指定を格納した文字列を指すディスクリプタのアドレスです。

このルーチンを呼び出すたびに、新しいファイルが作成されます。filespec 引数を省略した場合のファイル指定は SMGDISPLY.DAT です。

---

## 説明

SMG\$SAVE\_VIRTUAL\_DISPLAY ルーチンは仮想ディスプレイの内容を保存し、それを印字不可能なファイルに格納します。仮想ディスプレイを再作成するのに必要なテキスト、ビデオ属性、文字集合、およびすべてのディスプレイ属性は保存されます。メニュー、ビューポート、およびサブプロセスの内容は保存されません。仮想ディスプレイを復元するには、SMG\$LOAD\_VIRTUAL\_DISPLAY ルーチンを使用します。

---

## 戻される状態値

|                 |                                                  |
|-----------------|--------------------------------------------------|
| SS\$_NORMAL     | 正常終了。                                            |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。                           |
| RMS\$_xxxx      | \$OPEN, \$CONNECT, \$PUT, および \$CLOSE から戻されたエラー。 |

---

## SMG\$SCROLL\_DISPLAY\_AREA

### Scroll Display Area

Scroll Display Area ルーチンは、仮想ディスプレイの長方形領域をスクロールします。

---

#### 形式

```
SMG$SCROLL_DISPLAY_AREA display-id [,start-row] [,start-column] [,height] [,width] [,direction]
                        [,count]
```

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

スクロールが実行される仮想ディスプレイを指定します。display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

スクロール領域の 1 行目を指定します。start-row 引数は、開始行の行番号を格納した符号付きロングワードのアドレスです。

start-row 引数を省略した場合には、指定した仮想ディスプレイの 1 行目が使用されます。start-row 引数または start-column 引数を省略した場合には、省略時の設定 (行 1 とカラム 1) が使用されます。

start-column

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

スクロール領域の最初のカラムを指定します。start-column 引数は、開始カラムを格納した符号付きロングワードのアドレスです。

この引数を省略した場合には、指定した仮想ディスプレイのカラム 1 が使用されます。start-row 引数または start-column 引数を省略した場合には、省略時の値として行 1 とカラム 1 が使用されます。start-column 引数が全角文字の右半分から始まった場合、その文字は 2 つの未定義文字になります。

height

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

スクロール領域の行数を指定します。height 引数は、行数を格納した符号付きロングワードのアドレスです。

この引数を省略した場合には、省略時の値として仮想スクロール領域の高さ (SMG\$SET\_DISPLAY\_SCROLL\_REGION ルーチンによって設定されている場合)、または指定された仮想ディスプレイの高さが使用されます。

width

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

スクロール領域のカラム数を指定します。width 引数は、カラム数を格納した符号付きロングワードのアドレスです。

この引数を省略した場合には、省略時の値として、指定した仮想ディスプレイの幅が使用されます。

direction

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

スクロールの方向を指定します。direction 引数は、方向コードを格納したロングワード・ビット・マスクのアドレスです。

指定できる値は SMG\$M\_UP, SMG\$M\_DOWN, SMG\$M\_RIGHT, および SMG\$M\_LEFT です。省略時の設定は SMG\$M\_UP です。

count

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

スクロールする行数またはカラム数を指定します。count 引数は、スクロールの量 (行数またはカラム数) を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、1 行または 1 カラムがスクロールされます。

---

## 説明

SMG\$SCROLL\_DISPLAY\_AREA ルーチンは、指定された仮想ディスプレイの長方形領域をスクロールします。このルーチンは指定された行数または指定されたカラム数だけ、指定された方向に領域をスクロールします。スクロールの後、カーソルは start-row 引数、start-column 引数の位置に設定されます。

DEC 漢字または Super DEC 漢字文字集合が指定され、start-column 引数が全角文字の右半分から始まる場合は、全角文字は 2 つの未定義文字になります。長方形領域の右端のカラムが、全角文字の左半分を含む場合には、全角文字は 2 つの未定義文字になります。

また、長方形領域が左方向にスクロールし、左端のカラムが全角文字の右半分を含んでいる場合は、全角文字の右半分は未定義文字になります。長方形領域が右方向にスクロールし、右端のカラムが全角文字の左半分を含んでいる場合は、全角文字の左半分は未定義文字になります。

---

## 戻される状態値

|                  |                         |
|------------------|-------------------------|
| SS\$ _NORMAL     | 正常終了。                   |
| SMG\$ _INVARG    | 定義された長方形は仮想ディスプレイの外部です。 |
| SMG\$ _INVCOL    | カラムが誤っています。             |
| SMG\$ _INVDIS_ID | display-id 引数に誤りがあります。  |
| SMG\$ _INVROW    | 行が誤っています。               |
| SMG\$ _WRONUMARG | 引数の数が誤っています。            |

---

## SMG\$SCROLL\_VIEWPORT

### Scroll a Display Under a Viewport

Scroll a Display Under a Viewport ルーチンは、対応するビューポートで表示されている仮想ディスプレイをスクロールします。

---

#### 形式

SMG\$SCROLL\_VIEWPORT *display-id* [,*direction*] [,*count*]

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

スクロールされる仮想ディスプレイのディスプレイ識別子。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

direction

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

方向指定子 (省略可能)。direction 引数は、スクロールの方向を指定する方向コードを格納したロングワード・ビット・マスクのアドレスです。count 引数 (省略可能) を組み合わせて使用することにより、指定した方向にスクロールする行数を指定できます。direction 引数に対して指定できる値は以下のとおりです。

|           |                      |
|-----------|----------------------|
| SMG\$M_UP | count 行だけ上にスクロールします。 |
|-----------|----------------------|

SMG\$M\_DOWN           count 行だけ下にスクロールします。  
 SMG\$M\_RIGHT         count カラムだけ右にスクロールします。  
 SMG\$M\_LEFT          count カラムだけ左にスクロールします。

省略時の設定は SMG\$M\_UP です。

count

OpenVMS 用法   longword\_signed  
 データ型       longword (signed)  
 アクセス       read only  
 受け渡し方     by reference

スクロールする行数またはカラム数 (省略可能)。省略時の値は 1 です。

---

## 説明

SMG\$SCROLL\_VIEWPORT ルーチンは、対応するビューポートで表示されている仮想ディスプレイをスクロールします。実際には、スクロールをシミュレートするために、仮想ディスプレイの上をビューポートが移動するように、ビューポートはその座標を変更します。しかし、画面上で物理的な位置が変化するわけではありません。ビューポートが仮想ディスプレイの外部に移動する場合には、ビューポートのサイズが変化する可能性があります。ビューポートのサイズを元に戻すには、SMG\$CHANGE\_VIEWPORT ルーチンを使用します。

ビューポートの開始カラムが、全角文字の右半分であった場合には、そのカラムは表示されません。また、ビューポートが全角文字の左半分で終了した場合にも、そのカラムは表示されません。

---

## 戻される状態値

SS\$\_NORMAL           正常終了。  
 SMG\$\_INVARG          count 引数の値が 0 より小さい値です。  
 SMG\$\_INVDIS\_ID       display-id 引数に誤りがあります。  
 SMG\$\_NO\_WINASSOC     指定された仮想ディスプレイにビューポートは対応づけられていません。  
 SMG\$\_WRONUMARG       引数の数が誤っています。  
 SMG\$\_WINTRUFIT       正常終了。ただし、仮想ディスプレイに対応づけられているビューポートはサイズが変更されました。

## 例

```

1.
C This FORTRAN example demonstrates SMG$SCROLL_VIEWPORT.
C Include the SMG definitions. In particular, we want SMG$M_BORDER.

      IMPLICIT INTEGER (A-Z)
      INCLUDE '($SMGDEF)'

C Create the Virtual Displays. Give them borders.

      ROWS = 10
      COLUMNS = 22

      STATUS = SMG$CREATE_VIRTUAL_DISPLAY
1      (ROWS, COLUMNS, DISPLAY1, SMG$M_BORDER)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C Create viewport

      STATUS = SMG$CREATE_VIEWPORT (DISPLAY1, 2, 2, 3, 10)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C Create the Pasteboard

      STATUS = SMG$CREATE_PASTEBOARD (PASTE1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C Put data into the Virtual Displays

      STATUS = SMG$PUT_CHARS ( DISPLAY1, '11111111111111', 1, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      STATUS = SMG$PUT_CHARS ( DISPLAY1, 'This is row 2.', 2, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      STATUS = SMG$PUT_CHARS ( DISPLAY1, '33333333333333', 3, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      STATUS = SMG$PUT_CHARS ( DISPLAY1, 'This is row 4.', 4, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = SMG$PUT_CHARS ( DISPLAY1, '55555555555555', 5, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      STATUS = SMG$PUT_CHARS ( DISPLAY1, 'This is row 6.', 6, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      STATUS = SMG$PUT_CHARS ( DISPLAY1, '77777777777777', 7, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      STATUS = SMG$PUT_CHARS ( DISPLAY1, 'This is row 8.', 8, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      STATUS = SMG$PUT_CHARS ( DISPLAY1, '99999999999999', 9, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
      STATUS = SMG$PUT_CHARS ( DISPLAY1, 'This is row 10.', 10, 1)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

C Paste the Virtual Display

      STATUS = SMG$PASTE_VIRTUAL_DISPLAY ( DISPLAY1, PASTE1, 4, 3)
      IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))

      STATUS = LIB$WAIT (2.0)

```

```

DO 1 I = 1, 3
STATUS = SMG$SCROLL_VIEWPORT(DISPLAY1,SMG$M_UP,1)
IF (.NOT. STATUS) CALL LIB$SIGNAL(%VAL(STATUS))
CALL LIB$WAIT (2.0)
1 CONTINUE

STATUS = LIB$WAIT (2.0)

END

```

この例では、1つの仮想ディスプレイが作成され、ビューポートが対応づけられています。仮想ディスプレイはビューポートに対応づけられた後、ペーストボードにペーストされたため、仮想ディスプレイの中でビューポートの内部だけが表示されます。この状態は図 8-42 に示すとおりです。

図 8-42 仮想ディスプレイをペーストすることにより生成される出力

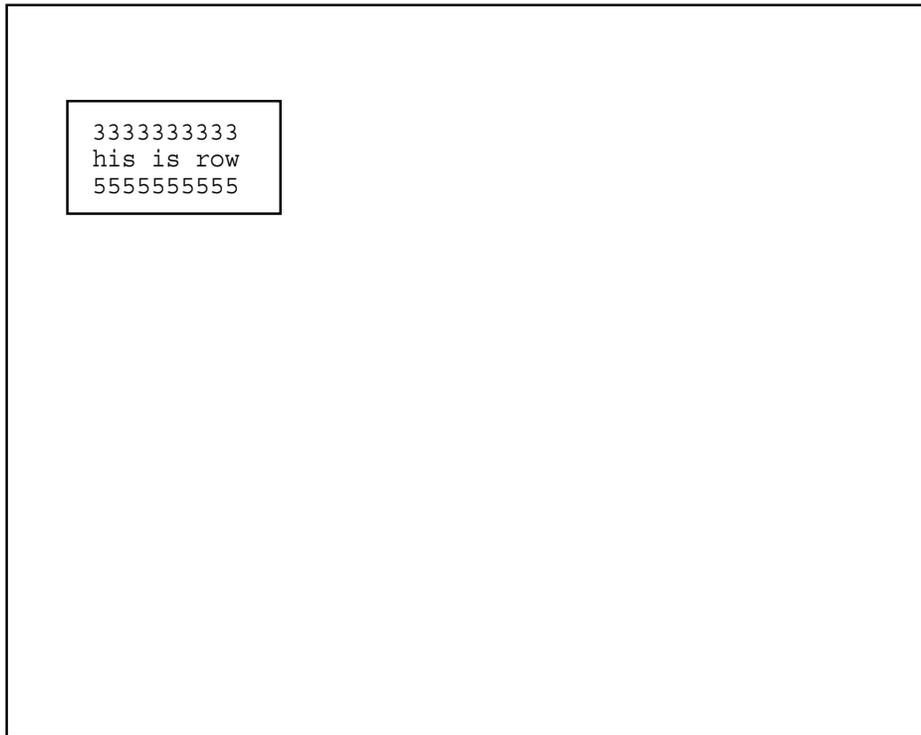


ZK-6426/1-GE

SMG\$SCROLL\_VIEWPORT ルーチンを全部で 3 回呼び出しています。図 8-43 は、SMG\$SCROLL\_VIEWPORT ルーチンを最初に呼び出した後のビューポートを示しています。

# SMG\$SCROLL\_VIEWPORT

図 8-43 SMG\$SCROLL\_VIEWPORT ルーチンを最初に呼び出した後に生成される出力



ZK-6426/2-GE

図 8-44 は、SMG\$SCROLL\_VIEWPORT ルーチンを 2 回目に呼び出した後のビューポートの内容を示しています。

図 8-44 SMG\$SCROLL\_VIEWPORT ルーチンを 2 回目に呼び出した後生成される出力

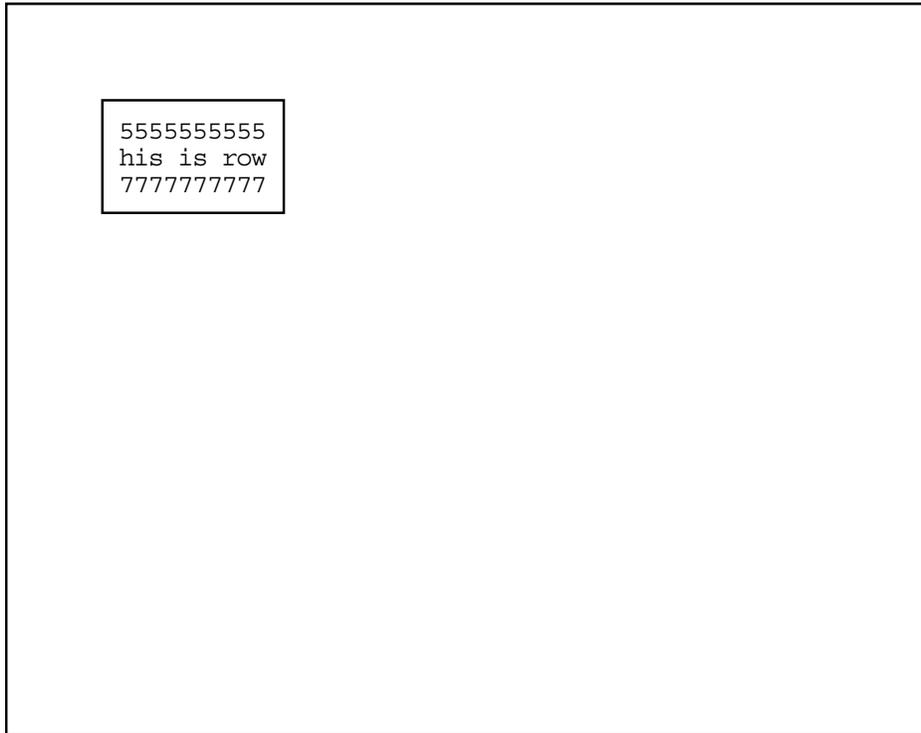


ZK-6426/3-GE

図 8-45 は、SMG\$SCROLL\_VIEWPORT ルーチンを最後に呼び出した後のビューポートの内容を示しています。

# SMG\$SCROLL\_VIEWPORT

図 8-45 SMG\$SCROLL\_VIEWPORT ルーチンを最後に呼び出した後に生成される出力



ZK-6426/4-GE

---

## SMG\$SELECT\_FROM\_MENU

### Make a Selection from the Menu

Make a Selection from the Menu ルーチンは、矢印キーを使って選択メニュー間の移動をしたり、`[Return]`キーを使ってメニューの選択を確定することを可能にします。

---

#### 形式

```
SMG$SELECT_FROM_MENU keyboard-id ,display-id ,selected-choice-number [,default-choice-number]
                    [,flags] [,help-library] [,timeout] [,word-terminator-code]
                    [,selected-choice-string] [,rendition-set] [,rendition-complement]
```

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

keyboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

入力を読み込む対象となる仮想キーボードを指定します。keyboard-id 引数は、キーボード識別子を格納した符号なしロングワードのアドレスです。

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

選択した項目が表示される仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。仮想ディスプレイは、ペーストボードにペーストされ、バッチ処理されたり重ね書きされたりすることはできません。

## SMG\$SELECT\_FROM\_MENU

### selected-choice-number

OpenVMS 用法 word\_unsigned  
データ型 word (unsigned)  
アクセス write only  
受け渡し方 by reference

選択されるメニュー項目の指定番号。selected-choice-number 引数はこの番号を受け付ける符号なしワードのアドレスです。selected-choice-number 引数は、SMG\$CREATE\_MENU ルーチン内の静的配列内のメニュー項目の索引に対応します。

### default-choice-number

OpenVMS 用法 word\_unsigned  
データ型 word (unsigned)  
アクセス read only  
受け渡し方 by reference

省略時のメニュー項目の指定番号 (省略可能)。default-choice-number 引数は省略時のメニュー項目の番号を格納する符号なしワードのアドレスです。default-choice-number 引数は、SMG\$CREATE\_MENU ルーチン内の静的配列内のメニュー項目の索引に対応します。この引数が省略された場合の選択は、すでに選択された項目中の最後のメニュー項目になります。何も選択されていなかった場合には、メニューの中の最初の項目が選択されます。

### flags

OpenVMS 用法 mask\_longword  
データ型 longword (unsigned)  
アクセス read only  
受け渡し方 by reference

選択された時の動作を指示するビットマスク (省略可能)。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。この引数に対して指定できる値は次のとおりです。

SMG\$M\_RETURN\_IMMED

矢印キー以外のキーが入力されたときに、コントロールをユーザに戻します。

SMG\$M\_REMOVE\_ITEM

SMG\$SELECT\_FROM\_MENU ルーチンによって各々のメニュー項目が一度のみ選択されるようにします。

### help-library

OpenVMS 用法 char\_string  
データ型 character string  
アクセス read only  
受け渡し方 by descriptor

[PF2] / [HELP]が入力された場合のヘルプ・ライブラリ (省略可能)。help-library 引数は、このヘルプ・ライブラリを示すディスクリプタのアドレスです。この引数は、flags 引数が SMG\$M\_RETURN\_IMMED を指定した場合には、無視されるので注意してください。省略時の設定値は、SYS\$HELP:HELPLIB.HLB です。

## timeout

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

時間切れの値 (省略可能)。timeout 引数は、選択が決定されるまでにかかる秒数を指定する符号付きロングワードのアドレスです。

## word-terminator-code

OpenVMS 用法 word\_unsigned  
 データ型 word (unsigned)  
 アクセス write only  
 受け渡し方 by reference

word-terminator-code 引数は、どの文字またはどのキーが読み込み操作を終了したかを示すコードが書き込まれる符号なしワードのアドレスです (省略可能)。

## selected-choice-string

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス write only  
 受け渡し方 by descriptor

選択されたメニュー項目のテキストを受け取る文字列 (省略可能)。selected-choice-string 引数は、この文字列を示すディスクリプタのアドレスです。

## rendition-set

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性指定子。rendition-set 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する属性が、仮想ディスプレイ上に設定されます。この引数に対して指定できる値は次のとおりです。

|                                 |                                              |
|---------------------------------|----------------------------------------------|
| SMG\$M_BLINK                    | 点滅する文字を表示します。                                |
| SMG\$M_BOLD                     | 通常より明るい輝度の文字を表示します。                          |
| SMG\$M_REVERSE                  | 文字を反転表示します。つまり仮想ディスプレイの現在の省略時属性とは逆の状態を表示します。 |
| SMG\$M_UNDERLINE                | 下線を付けた文字を表示します。                              |
| SMG\$M_USER1 から<br>SMG\$M_USER8 | ユーザ定義属性を表示します。                               |

## rendition-complement

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

属性補足指定子。rendition-complement 引数は省略可能で、ロングワード・ビット・マスクのアドレスです。この引数で指定する補足的な属性が、仮想ディスプレイ上に

設定されます。rendition-set 引数で指定できる引数はすべて、rendition-complement 引数で補足指定することができます。

rendition-set 引数と rendition-complement 引数はどちらも省略可能な引数であり、これらの引数を組み合わせて使用すれば、ユーザは仮想ディスプレイの属性を制御できます。rendition-set 引数は特定の仮想ディスプレイ属性を設定します。それに対し、rendition-complement 引数はこれらの属性を補足します。rendition-set 引数と rendition-complement 引数の両方に同じビットが指定されている場合には、最初に rendition-set 引数が評価され、その後、rendition-complement 引数が評価されます。これらの2つの引数を組み合わせて使用すれば、1回のプロシージャ呼び出しで複数の仮想ディスプレイ属性を制御できます。各属性に対して、ユーザは次の組み合わせを指定できます。

| 設定 | 補足 | 動作                   |
|----|----|----------------------|
| 0  | 0  | 属性を省略時の状態に設定する。      |
| 1  | 0  | 属性をオンに設定する。          |
| 0  | 1  | 属性を省略時の状態の逆の状態に設定する。 |
| 1  | 1  | 属性をオフに設定する。          |

## 説明

SMG\$SELECT\_FROM\_MENU ルーチンによって、メニューの中の項目を選択することができます。SMG\$CREATE\_MENU ルーチンは、SMG\$SELECT\_FROM\_MENU ルーチンの前に呼び出されなければなりません。

矢印キーを使って、メニュー項目内を移動し、`[Return]`キーを押すことによって選択を確定することができます。`[Ctrl/Z]`を入力すると、現在の選択を確定し、SMG\$\_EOF という値が戻されます。

選択メニュー数が1画面で表示しきれない場合は、適切な矢印キーが入力されると、仮想ディスプレイがスクロールされ、表示できなかった選択メニューが見えるようになります。

現在選択されているメニューは、反転表示され(または、指定された属性で)物理カーソルで示されます。選択されたメニューは、selected-choice-number 引数に戻されます。さらに、flag 引数に SMG\$M\_REMOVE\_ITEM が指定されている場合には、選択された項目は、もう一度選択することはできません。

SMG\$SELECT\_FROM\_MENU ルーチンには、flags 引数によって指定される2つの操作モードがあります。SMG\$M\_RETURN\_IMMED という値を設定しない場合(省略時設定)は、次のようになります。

- ユーザは、矢印キーを使って選択メニュー間を移動できます。

- 現在の項目を選択するのに使われるキーは、次のとおりです。

`RETURN` キー

`DO` キー

`SELECT` キー

`ENTER` キー

- `HELP` キーまたは `PF2` キーは現在選択した項目のヘルプを出力します。
- `CTRL/W` は、SMG\$REPAINT\_SCREEN を呼び出して画面をリフレッシュします。
- `CTRL/Z` は現在の項目を選択し、SMG\$\_EOF の値を戻します。
- `PF1/` はメニューの中の最初の項目を選択します。
- `PF1/` はメニューの中の最後の項目を選択します。
- `PF1/` は現在カーソルがある行の最初の項目を選択します。
- `PF1/` は現在カーソルがある行の最後の項目を選択します。
- 他のすべてのキーは無視されます。

SMG\$M\_RETURN\_IMMED の値を設定した場合は、次のようになります。

- ユーザは、矢印キーを使って選択メニュー間を移動できます。
- `CTRL/Z` は現在の項目を選択し、SMG\$\_EOF の値を戻します。
- その他のキーが入力された場合は、現在の項目を選択します。

(上記の) どちらの場合でも、次のキーで代用が可能です。

- `LF` キーは、`CR` キーの代わりに使用されます。
- `BACKSPACE` キーは、`CR` キーの代わりに使用されます。
- `TAB` キーは、`CR` キーの代わりに使用されます。

読み込み操作を終了したコード SMG\$R\_TRM\_key-name は、word-terminator-code 引数 (省略可能) に戻されます。複数の文字で構成されたメニュー項目も許可されています。メニュー項目が Super DEC 漢字、DEC 漢字文字または JIS カタカナ文字の場合、ヘルプ機能は起動されますが、メニュー項目は情報検索に使用されません。

---

## 戻される状態値

|                 |                                 |
|-----------------|---------------------------------|
| SS\$_NORMAL     | 正常終了。                           |
| SMG\$_EOF       | ファイルの最後。                        |
| SMG\$_ILLBATFNC | 仮想ディスプレイまたはペーストボードがバッチ処理されています。 |

## SMG\$SELECT\_FROM\_MENU

|                 |                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SMG\$_INVARG    | 引数に誤りがあります。選択できるメニュー項目がありません。                                                                                                                               |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。メニューが含まれていません。または、ビューポートが含まれています。                                                                                                     |
| SMG\$_NOTPASTED | 仮想ディスプレイがペーストされていません。                                                                                                                                       |
| LIB\$_xxx       | LIB\$SCOPY から戻された状態値。                                                                                                                                       |
| SMG\$_xxxx      | SMG\$CHANGE_RENDITION, SMG\$SET_CURSOR_ABS, SMG\$READ_KEYSTROKE, SMG\$PUT_HELP_TEXT, SMG\$SAVE_PHYSICAL_SCREEN, SMG\$RESTORE_PHYSICAL_SCREEN ルーチンから戻された状態値。 |

## 例

## 1.

```

C+
C This FORTRAN example program illustrates the use of
C SMG$CREATE_MENU, SMG$SELECT_FROM_MENU, and SMG$DELETE_MENU
C to create an application that lets a user make multiple
C selections from a horizontal menu.
C-

      IMPLICIT INTEGER (A-Z)
      INCLUDE '($SMGDEF)'

      CHARACTER*20 c
      CHARACTER*20 a(20) /'One','Two','Three','This is Four','Five',
1          'Six','Seven','Eight','Nine','I like ten',
2          'Eleven','Twelve','Thirteen','Fourteen',
3          'Fifteen','Sixteen','Seventeen','Eighteen',
4          'Nineteen','Exit this menu.' /

      s = SMG$CREATE_PASTEBOARD(p_id)
      if (.not. s) call LIB$SIGNAL(%VAL(s))
      s = SMG$CREATE_VIRTUAL_KEYBOARD(k_id)
      if (.not. s) call LIB$SIGNAL(%VAL(s))
      s = SMG$CREATE_VIRTUAL_DISPLAY(6,50, d_id2, SMG$M_BORDER)
      if (.not. s) call LIB$SIGNAL(%VAL(s))
      s = SMG$CREATE_VIRTUAL_DISPLAY(6,50, d_id, SMG$M_BORDER)
      if (.not. s) call LIB$SIGNAL(%VAL(s))

      s = SMG$PASTE_VIRTUAL_DISPLAY(d_id2, p_id, 2,2)
      if (.not. s) call LIB$SIGNAL(%VAL(s))
      s = SMG$PASTE_VIRTUAL_DISPLAY(d_id, p_id, 10,2)
      if (.not. s) call LIB$SIGNAL(%VAL(s))

      s = SMG$CREATE_MENU(d_id,a,SMG$K_HORIZONTAL,,2,SMG$M_REVERSE)
      if (.not. s) call LIB$SIGNAL(%VAL(s))

20     s = SMG$SELECT_FROM_MENU(k_id, d_id, n,6,,,,,C,SMG$M_BOLD,0)
      if (.not. s) call LIB$SIGNAL(%VAL(s))
      s = SMG$PUT_LINE(d_id2,c)
      if (.not. s) call LIB$SIGNAL(%VAL(s))

      if (n .ne. 20) goto 20

      s = SMG$DELETE_MENU(d_id)
      if (.not. s) call LIB$SIGNAL(%VAL(s))

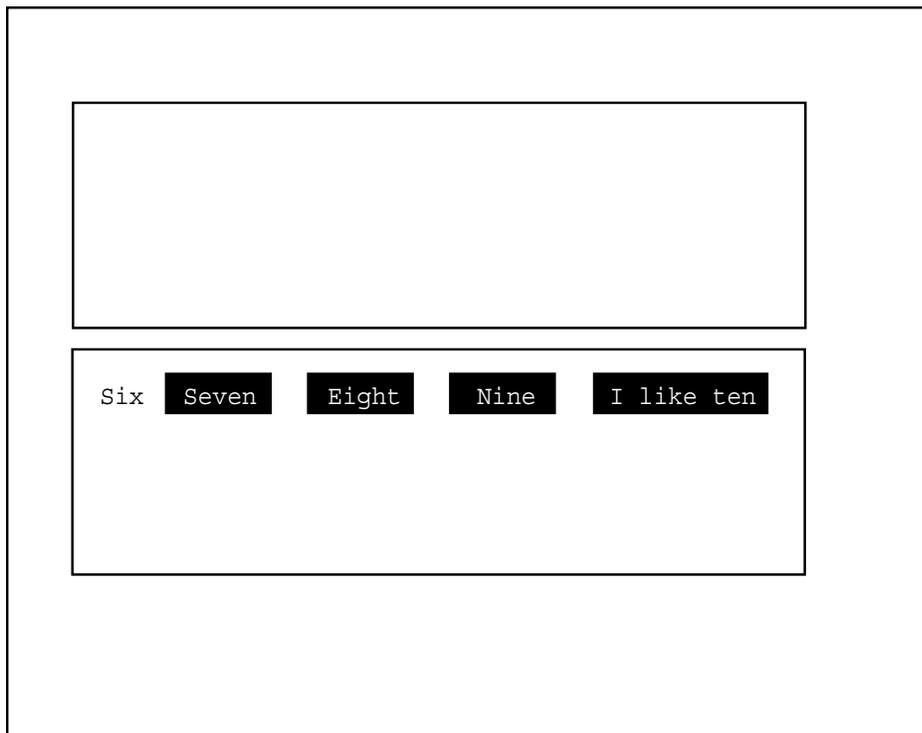
      END

```

以下の図は、この FORTRAN プログラムによって生成される水平方向のメニューです。

図 8-46 では、プログラムは、省略時で選択されるものを除くすべてのメニュー項目を、反転表示します。この時点で、プログラムはユーザが選択をするのを待っています。

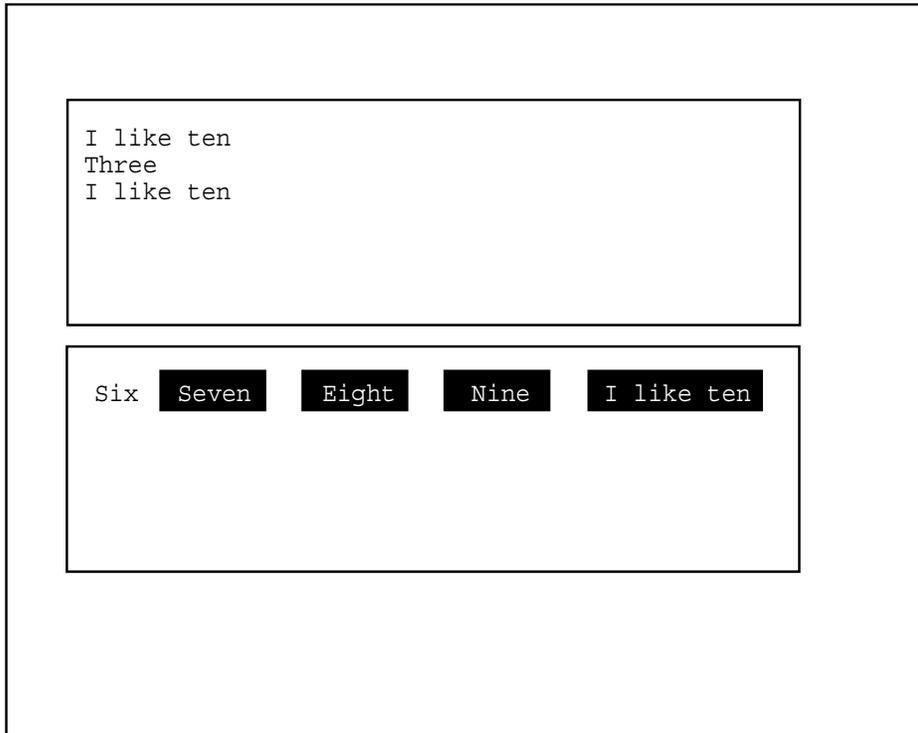
図 8-46 メニュー項目が選択される前に生成される出力



ZK-6429/1-GE

このメニューが作成されるときには、何も引数が指定されていなかったため、メニューの中の項目は再び選択されることができます。図 8-47 は、ユーザが3つの項目 (その中の2つは同じ) を選択した後の画面イメージを表しています。

図 8-47 3つのメニュー選択をした後に生成された出力



ZK-6429/2-GE

図 8-48 では、ユーザは“Exit this menu”を選択し、プログラムは実行を終了しました。

図 8-48 プログラムが終了した後に生成される出力

```
I like ten
Three
I like ten
Exit this menu.
$

Nineteen Exit this menu.
```

ZK-6429/3-GE

---

# SMG\$SET\_BROADCAST\_TRAPPING

## Enable Broadcast Trapping

Enable Broadcast Trapping ルーチンは、ブロードキャスト・メッセージのトラッピングを許可します。

---

### 形式

SMG\$SET\_BROADCAST\_TRAPPING *pasteboard-id* [*,AST-routine*] [*,AST-argument*]

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

ブロードキャスト・メッセージがトラップされるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

AST-routine

|            |                      |
|------------|----------------------|
| OpenVMS 用法 | ast_procedure        |
| データ型       | procedure entry mask |
| アクセス       | read only            |
| 受け渡し方      | by value             |

メッセージがペーストボードに受信されたときに呼び出される AST ルーチンのアドレス。AST-routine 引数はルーチンのプロシージャ・エントリ・マスクのアドレスです。つまり、ルーチン自体のアドレスです。

AST-routine 引数を省略した場合や、この引数の値を 0 として指定した場合には、BROADCAST モードは同期モードに設定されます。このモードでは、ブロードキャスト・メッセージが到着したかどうかを確認するために、SMG\$GET\_BROADCAST\_MESSAGE ルーチンを定期的呼び出さなければなりません。

AST ルーチンは 5 つの引数を使用して呼び出されます。それらは AST-argument 引数、R0、R1、PC、および PSL です。

| AST Argument |
|--------------|
| R0           |
| R1           |
| PC           |
| PSL          |

ZK-4803-GE

#### AST-argument

OpenVMS 用法 user\_arg  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by value

AST ルーチンに渡される値。AST-argument 引数は AST ルーチンに渡される値を格納した符号なしロングワードです。

---

## 説明

SMG\$SET\_BROADCAST\_TRAPPING ルーチンは、指定されたペーストボード (漢字ターミナル) に送信されたブロードキャスト・メッセージのトラッピングを許可します。ブロードキャスト・トラッピングを禁止した場合には、漢字ターミナルのキューに登録されていたブロードキャスト・メッセージは失われます。SMG\$SET\_BROADCAST\_TRAPPING ルーチンを使用してブロードキャスト・トラッピングを許可した後、イメージが終了する前に SMG\$DISABLE\_BROADCAST\_TRAPPING ルーチンを使用してブロードキャスト・トラッピングを禁止しなかった場合には、漢字ターミナルにブロードキャストされたメッセージは、イメージが終了するときに失われます。

SYSGEN パラメータである DEFMBXBUFQUO と DEFMBXMXMSG がブロードキャスト・メッセージを受信するメールボックスを作成する時に使用されます。

---

## 戻される状態値

|                 |                                                     |
|-----------------|-----------------------------------------------------|
| SS\$_NORMAL     | 正常終了。                                               |
| SMG\$_INVPAS_ID | pasteboard-id 引数に誤りがあります。                           |
| SMG\$_NOT_A_TRM | 情報メッセージ。ペーストボードは漢字ターミナルではありません。                     |
| SMG\$_WRONUMARG | 引数の数が誤っています。                                        |
| その他             | \$DASSGN, \$CANCEL, または LIB\$ASN_WTH_MBX から戻される状態値。 |

---

# SMG\$SET\_CURSOR\_ABS

## Set Absolute Cursor Position

Set Absolute Cursor Position ルーチンは、仮想ディスプレイの仮想カーソルを指定位置に移動させます。

---

### 形式

SMG\$SET\_CURSOR\_ABS *display-id* [*start-row*] [*start-column*]

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

仮想カーソルの位置を指定する仮想ディスプレイの識別子。display-id 引数は、仮想ディスプレイの識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

仮想カーソルが移動される行。start-row 引数は、行の位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、カーソルは現在の行に置かれます。

start-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

仮想カーソルが移動されるカラム。start-column 引数は、カラム位置を格納した符号付きロングワードのアドレスです。この引数を省略した場合は、カーソルは現在のカラムに置かれます。

---

## 説明

SMG\$SET\_CURSOR\_ABS ルーチンは、仮想カーソルを仮想ディスプレイの指定位置に移動させます。

---

## 戻される状態値

|                 |                       |
|-----------------|-----------------------|
| SS\$_NORMAL     | 正常終了。                 |
| SMG\$_INVCOL    | カラム位置が誤っています。         |
| SMG\$_INVDIS_ID | display-id 引数が誤っています。 |
| SMG\$_INVROW    | 行の位置が誤っています。          |
| SMG\$_WRONUMARG | 引数の数が誤っています。          |

---

# SMG\$SET\_CURSOR\_MODE

## Set the Cursor Mode

Set the Cursor Mode ルーチンは、物理カーソルをオンまたはオフに設定し、ジャンプ・スクロールまたは連続スクロールを選択します。

---

### 形式

SMG\$SET\_CURSOR\_MODE *pasteboard-id* [*flags*]

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

ペーストボード識別子。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

flags

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

スクロール属性とカーソル属性を指定するビット・マスク (省略可能)。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。flags 引数に対して指定できる値は次のとおりです。

|                      |              |
|----------------------|--------------|
| SMG\$M_CURSOR_OFF    | 物理カーソルを消去する。 |
| SMG\$M_CURSOR_ON     | 物理カーソルを表示する。 |
| SMG\$M_SCROLL_JUMP   | ジャンプ・スクロール。  |
| SMG\$M_SCROLL_SMOOTH | 連続スクロール。     |

---

## 説明

SMG\$SET\_CURSOR\_MODE ルーチンはカーソルを表示または消去し、ジャンプ・スクロールと連続スクロールのいずれかを選択します。漢字ターミナルがこれらの機能を備えていない場合には、このルーチンを実行しても無効です。

---

## 戻される状態値

|                 |                           |
|-----------------|---------------------------|
| SS\$_NORMAL     | 正常終了。                     |
| SMG\$_INVARG    | 引数に誤りがあります。               |
| SMG\$_INVPAS_ID | pasteboard-id 引数に誤りがあります。 |
| SMG\$_WRONUMARG | 引数の数が誤っています。              |

---

## SMG\$SET\_CURSOR\_REL

### Move Cursor Relative to Current Position

Move Cursor Relative to Current Position ルーチンは、仮想ディスプレイ内の現在の仮想カーソルの位置から指定された行数およびカラム数だけ、仮想カーソルを移動します。

---

#### 形式

```
SMG$SET_CURSOR_REL display-id [delta-row] [delta-column]
```

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

##### display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

仮想カーソルを移動する仮想ディスプレイを指定します。display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

##### delta-row

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

仮想カーソルを移動する行数を指定します。delta-row 引数は、移動する行数を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、仮想カーソルは現在の行位置から移動しません。delta-row 引数が正の値の場合には、仮想カ

カーソルは指定された行数だけ下に移動します。delta-row 引数が負の値の場合には、仮想カーソルは指定された行数だけ上に移動します。

delta-column

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

仮想カーソルを移動するカラム数を指定します。delta-column 引数は、移動するカラム数を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、仮想カーソルは現在のカラム位置から移動しません。delta-column 引数が正の値の場合には、仮想カーソルは指定されたカラム数だけ右に移動します。delta-column 引数が負の値の場合には、仮想カーソルは指定されたカラム数だけ左に移動します。

---

## 説明

SMG\$SET\_CURSOR\_REL ルーチンは、現在の仮想カーソルの位置をもとに、指定された行数およびカラム数だけ仮想カーソルを移動します。delta-row 引数および delta-column 引数の指定に従ってカーソルを移動した結果、カーソルが仮想ディスプレイの外部に移動する場合には、SMG\$\_INVROW または SMG\$\_INVCOL が戻されます。

---

## 戻される状態値

|                 |                                                      |
|-----------------|------------------------------------------------------|
| SS\$_NORMAL     | 正常終了。                                                |
| SMG\$_INVARG    | 引数に誤りがあります。                                          |
| SMG\$_INVCOL    | delta-column 引数の値が誤っているため、カーソルを仮想ディスプレイの外部に移動しようとした。 |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。                               |
| SMG\$_INVROW    | delta-row 引数の値が誤っているため、カーソルを仮想ディスプレイの外部に移動しようとした。    |
| SMG\$_WRONUMARG | 引数の数が誤っています。                                         |

---

## SMG\$SET\_DEFAULT\_STATE

### Set Default State

Set Default State ルーチンは、キー・テーブルに対して現在の省略時の状態を設定するか、または現在の省略時の状態を戻します。

---

#### 形式

```
SMG$SET_DEFAULT_STATE key-table-id [new-state] [old-state]
```

---

#### 戻り値

|            |                    |
|------------|--------------------|
| OpenVMS 用法 | cond_value         |
| データ型       | longword(unsigned) |
| アクセス       | write only         |
| 受け渡し方      | by value           |

---

#### 引数

key-table-id

|            |                    |
|------------|--------------------|
| OpenVMS 用法 | identifier         |
| データ型       | longword(unsigned) |
| アクセス       | read only          |
| 受け渡し方      | by reference       |

省略時の状態を設定または検索するキー・テーブルを指定します。key-table-id 引数は、キー・テーブル識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_KEY\_TABLE ルーチンによって与えられます。

new-state

|            |                  |
|------------|------------------|
| OpenVMS 用法 | char_string      |
| データ型       | character string |
| アクセス       | read only        |
| 受け渡し方      | by descriptor    |

キー・テーブル全体に対する新しい省略時の状態を指定します。new-state 引数は、新しい状態文字列を示すディスクリプタのアドレスです。指定された状態名は、使用する前に大文字に交換され、後続する空白は削除されます。

old-state

OpenVMS 用法 char\_string  
 データ型 character string  
 アクセス write only  
 受け渡し方 by descriptor

指定されたキー定義テーブルの既存の省略時状態名が書き込まれます。old-state 引数は、既存の状態文字列が書き込まれる文字列を示すディスクリプタのアドレスです。

---

## 説明

SMG\$SET\_DEFAULT\_STATE ルーチンは、キー定義テーブル全体に対して省略時の状態名を設定するか、または現在の省略時の状態名を戻します。キー定義テーブル全体に対する省略時の状態を変更すれば、新しい機能集合に対してキーパッド・キーを使用できます。キー定義テーブルは SMG\$READ\_COMPOSED\_LINE ルーチンで使用できます。

---

## 戻される状態値

|                 |                          |
|-----------------|--------------------------|
| SS\$_NORMAL     | 正常終了。                    |
| SMG\$_INVKTB_ID | key-table-id 引数に誤りがあります。 |
| SMG\$_INVSTANAM | 状態名に誤りがあります。             |
| LIB\$_INVSTRDES | 文字列ディスクリプタに誤りがあります。      |

---

## SMG\$SET\_DISPLAY\_SCROLL\_REGION

### Create Display Scrolling Region

Create Display Scrolling Region ルーチンは、仮想ディスプレイ内に仮想スクロール領域を作成します。

---

#### 形式

SMG\$SET\_DISPLAY\_SCROLL\_REGION *display-id* [*start-row*][*end-row*]

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

スクロールが実行される仮想ディスプレイを指定します。display-id 引数は、仮想ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

start-row

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

仮想スクロール領域の 1 行目を指定します。start-row 引数は、開始行番号を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、ディスプレイの 1 行目が使用されます。

end-row

OpenVMS 用法 longword\_signed  
 データ型 longword (signed)  
 アクセス read only  
 受け渡し方 by reference

仮想スクロール領域の最後の行を指定します。end-row 引数は、終了行番号を格納した符号付きロングワードのアドレスです。この引数を省略した場合には、仮想ディスプレイの最後の行が使用されます。

---

## 説明

SMG\$SET\_DISPLAY\_SCROLL\_REGION ルーチンは、指定された開始行と終了行を使用して、指定された仮想ディスプレイ内に仮想スクロール領域を作成します。start-row 引数と end-row 引数を省略した場合には、仮想ディスプレイ全体がスクロール領域になります。このルーチンはペーストボードの表示を変更せず、また仮想カーソルの位置も変更しません。

SMG\$SET\_DISPLAY\_SCROLL\_REGION ルーチンを使用すれば、仮想ディスプレイの一部をスクロール領域として定義できます。日本語 SMG では、スクロール操作は以下の方法で実行されます。スクロール領域の一番下の行に書き込むために SMG\$PUT\_LINE ルーチンを呼び出した場合には、仮想ディスプレイはただちに上にスクロールされるわけではありません。仮想ディスプレイをただちに上にスクロールした場合には、スクロール領域の一番下の行は絶対に使用できなくなります。このため、SMG\$PUT\_LINE ルーチンを次に呼び出すまで、スクロール要求は“格納”されます (スクロール要求を格納しなければならないのは、行全体を書き込むルーチン、たとえば、SMG\$PUT\_LINE ルーチンなどを次に呼び出すまでです)。

SMG\$PUT\_LINE ルーチンを呼び出した後、SMG\$SET\_CURSOR\_ABS ルーチンや SMG\$PUT\_CHARS ルーチンなどの別のルーチンを呼び出した場合には、格納されているスクロール要求は破棄され、スクロール領域の一番下の行に書き込まれるテキストは既存の行の上に重ね書きされます。

---

## 戻される状態値

|                |                                           |
|----------------|-------------------------------------------|
| SS\$NORMAL     | 正常終了。                                     |
| SMG\$INVDIS_ID | display-id 引数に誤りがあります。                    |
| SMG\$INVARG    | end-row 引数が start-row 引数より小さいか、または等しい値です。 |
| SMG\$INVROW    | 行番号が誤っています。                               |
| SMG\$WRONUMARG | 引数の数が誤っています。                              |

---

## SMG\$SET\_KEYPAD\_MODE

### Set Keypad Mode

Set Keypad Mode ルーチンは、漢字ターミナルの数値キーパッドを数値モードまたはアプリケーション・モードに設定します。

---

#### 形式

SMG\$SET\_KEYPAD\_MODE *keyboard-id* ,*flags*

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

keyboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

モードを変更する仮想キーボードを指定します。keyboard-id 引数は、キーボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチンから戻されます。

flags

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

省略可能なビット・マスクであり、キーパッドをアプリケーション・モードに設定するのか、数値モードに設定するのかを指定します。flags 引数は、フラグを格納した符号なしロングワードのアドレスです。flags 引数に対して指定できる値は次のとおりです。

|                           |                           |
|---------------------------|---------------------------|
| 0                         | キーパッドを数値モードに設定します。        |
| SMG\$M_KEYPAD_APPLICATION | キーパッドをアプリケーション・モードに設定します。 |

---

## 説明

SMG\$SET\_KEYPAD\_MODE ルーチンは、漢字ターミナルの数値キーパッドを数値モードまたはアプリケーション・モードに設定します。アプリケーション・モードでは、数値キーパッドのキーはファンクション・キーであると解釈され、終了文字としても使用できます。数値モードでは、これらのキーはメイン・キーボードの対応するキーと同じになります。

漢字ターミナルがアプリケーション・キーパッド・モードをサポートしない場合には、このルーチンはまったく効果がありません。

---

### 注意

キーパッド・モードを変更すると、物理端末の設定も変更されます。これは、keyboard-id 引数によって設定される仮想キーボードだけでなく、すべての仮想キーボードに対するグローバルな変更です。

---



---

## 戻される状態値

|                 |                         |
|-----------------|-------------------------|
| SS\$_NORMAL     | 正常終了。                   |
| SMG\$_INVKBD_ID | keyboard-id 引数に誤りがあります。 |
| SMG\$_WRONUMARG | 引数の数が誤っています。            |

---

## SMG\$SET\_OUT\_OF\_BAND\_ASTS

### Set Out-of-Band ASTs

Set Out-of-Band ASTs ルーチンは、アウト・オブ・バンド制御文字のトラッピングを許可または禁止します。

---

#### 形式

```
SMG$SET_OUT_OF_BAND_ASTS
                                pasteboard-id
                                ,control-character-mask
                                ,AST-routine
                                [,AST-argument]
```

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

アウト・オブ・バンド文字が許可または禁止されるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

control-character-mask

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

どの制御文字が新しいアウト・オブ・バンド制御文字になるかを指定します。control-character-mask 引数は、マスクを格納した符号なしロングワードのアドレスです。このマスクを作成するには、適切な文字の ASCII 値に対応するビットを設定します。たとえば、`Ctrl/C`(ASCII の 3) がアウト・オブ・バンド制御文字であることを指定する場合には、control-character-mask 引数のビット 3 (8 という値) を設定します。このマスクにビットが設定されていない場合には、アウト・オブ・バンド AST は発生しません。詳しい説明は、『OpenVMS I/O User's Reference Manual』を参照してください。

#### AST-routine

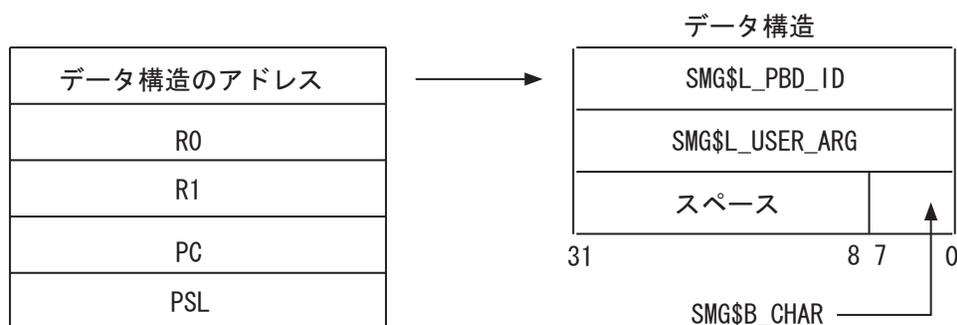
OpenVMS 用法 ast\_procedure  
 データ型 procedure entry mask  
 アクセス read only  
 受け渡し方 by value

アウト・オブ・バンド制御文字が漢字ターミナルから入力されたときに呼び出される AST ルーチンのアドレス。AST-routine 引数はルーチンのプロシージャ・エントリ・マスクのアドレスです。つまり、ルーチン自体のアドレスです。

#### AST-argument

OpenVMS 用法 user\_arg  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by value

AST に渡される引数。AST-argument 引数は、AST ルーチンに渡される値を格納した符号なしロングワードです。しかし、AST ルーチンは入力されたアウト・オブ・バンド文字と、それが入力された pasteboard-id 引数も判断する必要があります。したがって、日本語 SMG はこの情報を格納したために 3 ロングワードの構造を作成し、この構造のアドレスを AST ルーチンに対する最初の引数として渡します。残りの 4 つの引数は R0, R1, PC, および PSL です。日本語 SMG は、ユーザが指定した引数を次のデータ構造に格納します。



最初のロングワードには pasteboard-id 引数が格納され、このロングワードのシンボル名は SMG\$L\_PBD\_ID です。2 番目のロングワードには AST-argument 引数が

格納され、シンボル名は SMG\$USER\_ARG です。3 番目のロングワードには入力されたアウト・オブ・バンド文字の ASCII 値が格納され、2 つのシンボル名によってアクセスできます。それは SMG\$B\_CHAR (ASCII 値を格納する下位バイト) と SMG\$L\_CHAR (下位バイトに ASCII 値を格納し、上位バイトにスペースを格納するロングワード) です。

---

## 説明

SMG\$SET\_OUT\_OF\_BAND\_ASTS ルーチンは、指定された漢字ターミナルからアウト・オブ・バンド制御文字を受け付けることを許可または禁止します。これらの文字が漢字ターミナルから入力されると、AST ルーチンが呼び出されます。

このルーチンを使用すれば、`Ctrl/C` や `Ctrl/Y`、`Ctrl/O` などのアウト・オブ・バンド文字をトラップできます。

---

## 戻される状態値

|                |                           |
|----------------|---------------------------|
| SS\$NORMAL     | 正常終了。                     |
| SMG\$INVPAS_ID | pasteboard-id 引数に誤りがあります。 |
| SMG\$WRONUMARG | 引数の数が誤っています。              |

---

## SMG\$SET\_PHYSICAL\_CURSOR

### Set Cursor on Physical Screen

Set Cursor on Physical Screen ルーチンは、物理カーソルをペーストボードの指定された位置に移動します。

---

#### 形式

SMG\$SET\_PHYSICAL\_CURSOR *pasteboard-id* ,*pasteboard-row* ,*pasteboard-column*

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

物理カーソルを移動するペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

pasteboard-row

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

物理カーソルの移動先となる行を指定します。pasteboard-row 引数は、行番号を格納した符号付きロングワードのアドレスです。

## SMG\$SET\_PHYSICAL\_CURSOR

pasteboard-column

|            |                   |
|------------|-------------------|
| OpenVMS 用法 | longword_signed   |
| データ型       | longword (signed) |
| アクセス       | read only         |
| 受け渡し方      | by reference      |

物理カーソルの移動先となるカラムを指定します。pasteboard-column 引数は、カラム番号を格納した符号付きロングワードのアドレスです。

---

### 説明

SMG\$SET\_PHYSICAL\_CURSOR ルーチンは、物理カーソルを指定されたペーストボードの指定された行およびカラムに移動します。ペーストボードがバッチ処理されている場合には、このルーチンを使用することはできません。

---

### 戻される状態値

|                |                           |
|----------------|---------------------------|
| SS\$NORMAL     | 正常終了。                     |
| SMG\$INVARG    | カラムが誤っています。               |
| SMG\$INVPAS_ID | pasteboard-id 引数に誤りがあります。 |
| SMG\$WRONUMARG | 引数の数が誤っています。              |

---

# SMG\$SET\_TERM\_CHARACTERISTICS

## Change Terminal Characteristics

Change Terminal Characteristics は、指定されたペーストボードに対してターミナル属性を変更するか、または現在のターミナル属性を検索します。

---

### 形式

SMG\$SET\_TERM\_CHARACTERISTICS

*pasteboard-id* [*on-characteristics1*] [*on-characteristics2*]  
 [*off-characteristics1*] [*off-characteristics2*] [*old-characteristics1*]  
 [*old-characteristics2*]

---

### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

### 引数

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

属性を変更または検索するペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

on-characteristics1

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | mask_longword       |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

\$TTDEF で設定されるターミナル属性を指定するビット・マスク。

on-characteristics1 引数は、ビット・マスクを格納した符号なしロングワードのアドレスです。

## on-characteristics2

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

\$TT2DEF で設定されるターミナル属性を指定するビット・マスク。

on-characteristics2 引数は、ビット・マスクを格納した符号なしロングワードのアドレスです。

## off-characteristics1

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

\$TTDEF で設定解除されるターミナル属性を指定するビット・マスク。

off-characteristics1 引数は、ビット・マスクを格納した符号なしロングワードのアドレスです。

## off-characteristics2

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス read only  
 受け渡し方 by reference

\$TT2DEF で設定解除されるターミナル属性を指定するビット・マスク。

off-characteristics2 引数は、ビット・マスクを格納した符号なしロングワードのアドレスです。

## old-characteristics1

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス write only  
 受け渡し方 by reference

第 1 グループ内の現在のターミナル属性が検索されます。

old-characteristics1 引数は、ビット・マスクが書き込まれる符号なしロングワードのアドレスです。

## old-characteristics2

OpenVMS 用法 mask\_longword  
 データ型 longword (unsigned)  
 アクセス write only  
 受け渡し方 by reference

第 2 グループ内の現在のターミナル属性が検索されます。

old-characteristics2 引数は、ビット・マスクが書き込まれる符号なしロングワードのアドレスです。

---

## 説明

SMG\$SET\_TERM\_CHARACTERISTICS ルーチンは、指定されたペーストボードに対してターミナル属性を変更するか、または現在のターミナル属性を検索します。属性は、HP が提供するシステム・シンボル・ライブラリ内の\$TTDEF および\$TT2DEF マクロ・モジュールによって定義されます。このルーチンを使用すれば、1 回のルーチン呼び出しで複数のターミナル属性を制御できます。

---

## 戻される状態値

|                 |                         |
|-----------------|-------------------------|
| SS\$_NORMAL     | 正常終了。                   |
| SMG\$_NOT_A_TRM | ペーストボードは漢字ターミナルではありません。 |
| SS\$_xyz        | LIB\$QIOW から戻されたエラー。    |

---

## SMG\$UNPASTE\_VIRTUAL\_DISPLAY

### Remove Virtual Display

Remove Virtual Display ルーチンは、ペーストボードから仮想ディスプレイを削除します。

---

#### 形式

SMG\$UNPASTE\_VIRTUAL\_DISPLAY *display-id ,pasteboard-id*

---

#### 戻り値

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | cond_value          |
| データ型       | longword (unsigned) |
| アクセス       | write only          |
| 受け渡し方      | by value            |

---

#### 引数

display-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

ペーストボードから削除する仮想ディスプレイを指定します。display-id 引数は、ディスプレイ識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチンによって与えられます。

pasteboard-id

|            |                     |
|------------|---------------------|
| OpenVMS 用法 | identifier          |
| データ型       | longword (unsigned) |
| アクセス       | read only           |
| 受け渡し方      | by reference        |

仮想ディスプレイが削除されるペーストボードを指定します。pasteboard-id 引数は、ペーストボード識別子を格納した符号なしロングワードのアドレスです。

この引数は SMG\$CREATE\_PASTEBOARD ルーチンによって与えられます。

---

## 説明

SMG\$UNPASTE\_VIRTUAL\_DISPLAY ルーチンは、指定された仮想ディスプレイを指定されたペーストボードから削除し、ペーストボードに対応する画面からディスプレイを削除します。このルーチンは仮想ディスプレイやその内容を破壊するわけではなく、特定のペーストボードとの関係を解除することにより、仮想ディスプレイが画面に表示されないようにします。指定された仮想ディスプレイによってそれまで覆われていたテキストは再度表示されるようになります。

---

## 戻される状態値

|                 |                                                    |
|-----------------|----------------------------------------------------|
| SS\$_NORMAL     | 正常終了。                                              |
| SMG\$_INVARG    | 引数に誤りがあります。指定された仮想ディスプレイは指定されたペーストボードにペーストされていません。 |
| SMG\$_INVDIS_ID | display-id 引数に誤りがあります。                             |
| SMG\$_INVPAS_ID | pasteboard-id 引数に誤りがあります。                          |
| SMG\$_NOTPASTED | 指定された仮想ディスプレイは指定されたペーストボードにペーストされていません。            |
| SMG\$_WRONUMARG | 引数の数が誤っています。                                       |

## A

ASCII 文字集合 . . . . . 2-19, SMG-30, SMG-31,  
SMG-69, SMG-169, SMG-200, SMG-208,  
SMG-209, SMG-220, SMG-252, SMG-257,  
SMG-262, SMG-267, SMG-272, SMG-276,  
SMG-277, SMG-284, SMG-288, SMG-292,  
SMG-297, SMG-303, SMG-307, SMG-324,  
SMG-335, SMG-352, SMG-360

## AST

リエントラント . . . . . 4-1  
ルーチン . . . . . 4-2

## C

CLISDCL\_PARSE . . . . . SMG-83  
CREATE\_VIRTUAL\_DISPLAY ルーチン  
. . . . . SMG-255

## D

DEC 漢字文字集合 . . . . . 2-9, 2-10, 2-11,  
2-19, SMG-30, SMG-69, SMG-86, SMG-120,  
SMG-126, SMG-137, SMG-142, SMG-145,  
SMG-150, SMG-169, SMG-200, SMG-201,  
SMG-208, SMG-209, SMG-220, SMG-237,  
SMG-252, SMG-257, SMG-262, SMG-267,  
SMG-272, SMG-276, SMG-277, SMG-284,  
SMG-288, SMG-292, SMG-297, SMG-303,  
SMG-307, SMG-324, SMG-335, SMG-338,  
SMG-352, SMG-360, SMG-373

DEC 特殊文字集合 . . . . 2-19, SMG-30, SMG-31,  
SMG-69, SMG-169, SMG-200, SMG-208,  
SMG-209, SMG-220, SMG-252, SMG-257,  
SMG-262, SMG-267, SMG-272, SMG-276,  
SMG-277, SMG-284, SMG-288, SMG-292,  
SMG-297, SMG-303, SMG-307, SMG-324,  
SMG-335, SMG-352, SMG-360

DEC 補助文字集合 . . . . 2-19, SMG-30, SMG-69,  
SMG-169, SMG-200, SMG-208, SMG-220,  
SMG-252, SMG-257, SMG-262, SMG-267,  
SMG-272, SMG-276, SMG-284, SMG-288,  
SMG-292, SMG-297, SMG-303, SMG-307,  
SMG-324, SMG-335, SMG-352, SMG-360

DEFINE/KEY コマンド . . . . . 3-4

DIGITAL GKS . . . . . 1-5

## E

ECHO 属性 . . . . . 3-4  
ERASE\_TO\_END\_DISPLAY ルーチン . . . . . 5-23  
ERASE\_TO\_END\_LINE ルーチン . . . . . 5-23  
ERASE\_WHOLE\_DISPLAY シーケンス . . . . . 5-21

## F

FORTRAN . . . . . 7-1

## I

IF\_STATE 属性 . . . . . 3-4  
If state . . . . . 3-4  
ISO Latin-1 補助文字集合 . . . . . 2-19, SMG-30,  
SMG-69, SMG-169, SMG-200, SMG-208,  
SMG-220, SMG-252, SMG-257, SMG-262,  
SMG-267, SMG-272, SMG-276, SMG-284,  
SMG-288, SMG-292, SMG-297, SMG-303,  
SMG-307, SMG-324, SMG-335, SMG-352,  
SMG-360

## J

JIS カタカナ文字集合 . . . . . 2-19, SMG-30,  
SMG-69, SMG-169, SMG-200, SMG-208,  
SMG-220, SMG-252, SMG-257, SMG-262,  
SMG-267, SMG-272, SMG-276, SMG-284,  
SMG-288, SMG-292, SMG-297, SMG-303,  
SMG-307, SMG-324, SMG-335, SMG-352,  
SMG-360

## L

LIBSSPAWN . . . . . SMG-110, SMG-112  
LOCK\_STATE 属性 . . . . . 3-4

## N

NOTERMINATE 属性 . . . . . 3-4

## O

OpenVMS ターミナル機能ファイルの作成 . . . 5-19  
OpenVMS ターミナル・ドライバ . . . . . 5-24

## P

PROTECT 属性 ..... 3-4

## S

SET\_CURSOR\_ABS ルーチン ..... 5-18  
SET\_SCROLLING\_REGION ルーチン ..... 5-23  
SET TERMINAL コマンド ..... 5-25  
SMG\$ADD\_KEY\_DEF ルーチン ... 3-3, SMG-2,  
SMG-48, SMG-83, SMG-172, SMG-300,  
SMG-304  
SMG\$BEGIN\_DISPLAY\_UPDATE ルーチ  
ン ..... SMG-6  
SMG\$BEGIN\_PASTEBOARD\_UPDATE ルーチ  
ン ..... 2-21, SMG-8, SMG-135  
SMG\$COLOR\_UNKNOWN ルーチン  
..... SMG-14  
SMG\$C\_JIS\_KANA ..... SMG-31, SMG-200  
SMG\$C\_KANJI ..... SMG-31, SMG-200  
SMG\$C\_SDK ..... SMG-31, SMG-200  
SMG\$C\_SUPPLEMENTAL ..... SMG-31,  
SMG-200  
SMG\$CANCEL\_INPUT ルーチン ..... 1-9, 3-2,  
SMG-10  
SMG\$CHANGE\_PBD\_CHARACTERISTICS ルーチ  
ン ..... 1-7, SMG-12  
SMG\$CHANGE\_RENDITION ルーチン  
..... SMG-16  
SMG\$CHANGE\_VIEWPORT ルーチン ... 2-16,  
2-17, SMG-21, SMG-26, SMG-375  
SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチ  
ン ..... 1-8, 2-12, SMG-28, SMG-200,  
SMG-208, SMG-219, SMG-276, SMG-288,  
SMG-292, SMG-302, SMG-334  
SMG\$CHANGE\_VIRTUAL\_DISPLAY ルーチ  
ン ..... 2-17  
SMG\$CHECK\_FOR\_OCCLUSION ルーチ  
ン ..... 2-6, 2-7, SMG-32  
SMG\$CONTROL\_MODE ルーチン ..... 2-20,  
SMG-37  
SMG\$COPY\_VIRTUAL\_DISPLAY ルーチ  
ン ..... SMG-41  
SMG\$CREATE\_KEY\_TABLE ルーチン .... 3-3,  
SMG-2, SMG-4, SMG-82, SMG-171,  
SMG-224, SMG-299, SMG-400  
SMG\$CREATE\_MENU ルーチン ..... 2-17,  
SMG-50, SMG-382, SMG-384  
SMG\$CREATE\_PASTEBOARD ルーチン ... 1-6,  
6-2, SMG-8, SMG-12, SMG-32, SMG-37,  
SMG-55, SMG-75, SMG-99, SMG-109,  
SMG-111, SMG-129, SMG-134, SMG-154,  
SMG-160, SMG-162, SMG-164, SMG-178,  
SMG-181, SMG-227, SMG-344, SMG-355,  
SMG-391, SMG-396, SMG-406, SMG-409,  
SMG-411, SMG-414

SMG\$CREATE\_SUBPROCESS ルーチン ... 4-3,  
SMG-58, SMG-101, SMG-158  
SMG\$CREATE\_VIEWPORT ルーチン .... 2-15,  
SMG-61  
SMG\$CREATE\_VIRTUAL\_DISPLAY ルーチ  
ン ..... 1-7, 2-13,  
2-19, SMG-16, SMG-28, SMG-32, SMG-58,  
SMG-67, SMG-78, SMG-80, SMG-85,  
SMG-92, SMG-101, SMG-105, SMG-118,  
SMG-124, SMG-136, SMG-143, SMG-149,  
SMG-157, SMG-167, SMG-181, SMG-191,  
SMG-198, SMG-200, SMG-205, SMG-208,  
SMG-217, SMG-219, SMG-233, SMG-234,  
SMG-235, SMG-238, SMG-244, SMG-247,  
SMG-249, SMG-274, SMG-276, SMG-288,  
SMG-290, SMG-292, SMG-302, SMG-305,  
SMG-334, SMG-337, SMG-344, SMG-357,  
SMG-364, SMG-369, SMG-371, SMG-394,  
SMG-398, SMG-402, SMG-414  
SMG\$CREATE\_VIRTUAL\_KEYBOARD ルーチ  
ン ... 1-9, 3-1, SMG-10, SMG-57, SMG-73,  
SMG-107, SMG-174, SMG-298, SMG-303,  
SMG-311, SMG-318, SMG-329, SMG-351,  
SMG-359, SMG-404  
SMG\$CURSOR\_COLUMN ルーチン ..... 2-8,  
SMG-78  
SMG\$CURSOR\_ROW ルーチン ... 2-8, SMG-80  
SMG\$DEFAULT\_CHARACTER\_SET 論理  
名 ..... 2-19, 3-2, SMG-69, SMG-302,  
SMG-303, SMG-324, SMG-334, SMG-335  
SMG\$DEFINE\_KEY ルーチン ... 3-3, SMG-48,  
SMG-82, SMG-231, SMG-300, SMG-304  
SMG\$DEL\_TERM\_TABLE ルーチン ..... 5-3,  
SMG-84  
SMG\$DELETE\_CHARS ルーチン ..... 2-9,  
SMG-85  
SMG\$DELETE\_KEY\_DEF ルーチン ..... 3-3,  
SMG-90  
SMG\$DELETE\_LINE ルーチン ... 2-9, SMG-92  
SMG\$DELETE\_MENU ルーチン ..... 2-18,  
SMG-97  
SMG\$DELETE\_PASTEBOARD ルーチン ... 1-7,  
SMG-99  
SMG\$DELETE\_SUBPROCESS ルーチン ... 4-3,  
SMG-60, SMG-101  
SMG\$DELETE\_VIEWPORT ルーチン .... 2-16,  
SMG-62, SMG-103  
SMG\$DELETE\_VIRTUAL\_DISPLAY ルーチ  
ン ..... 1-8, 2-6, 6-2, SMG-103, SMG-105  
SMG\$DELETE\_VIRTUAL\_KEYBOARD ルーチ  
ン ..... 1-9, 3-1, SMG-107  
SMG\$DISABLE\_BROADCAST\_TRAPPING ルーチ  
ン ..... SMG-109, SMG-112, SMG-392  
SMG\$DISABLE\_UNSOLICITED\_INPUT ルーチ  
ン ..... SMG-110, SMG-111  
SMG\$DRAW\_CHAR ルーチン ... 2-13, SMG-113

SMG\$DRAW\_LINE ルーチン ... 2-13, SMG-118, SMG-338  
 SMG\$DRAW\_RECTANGLE ルーチン .... 2-13, SMG-124, SMG-338  
 SMG\$ENABLE\_UNSOLICITED\_INPUT ルーチン ..... 4-2, SMG-112, SMG-129  
 SMG\$END\_DISPLAY\_UPDATE ルーチン ..... SMG-132  
 SMG\$END\_PASTEBOARD\_UPDATE ルーチン ..... 2-21, SMG-8, SMG-9, SMG-134  
 SMG\$ERASE\_CHARS ルーチン ..... 2-10, SMG-136  
 SMG\$ERASE\_COLUMN ルーチン ..... 2-10, SMG-141  
 SMG\$ERASE\_DISPLAY ルーチン ..... 2-10, SMG-143  
 SMG\$ERASE\_LINE ルーチン .. 2-10, SMG-149  
 SMG\$ERASE\_PASTEBOARD ルーチン .... 1-7, SMG-154  
 SMG\$EXECUTE\_COMMAND ルーチン .... 4-3, SMG-59, SMG-157  
 SMG\$FIND\_CURSOR\_DISPLAY ルーチン ..... SMG-160  
 SMG\$FLUSH\_BUFFER ルーチン ..... 2-21, SMG-39, SMG-162, SMG-303  
 SMG\$GET\_BROADCAST\_MESSAGE ルーチン ..... 4-2, SMG-164, SMG-392  
 SMG\$GET\_DISPLAY\_ATTR ルーチン ..... 1-8  
 SMG\$GET\_DISPLAY\_ATTR ルーチン ..... SMG-167  
 SMG\$GET\_KEY\_DEF ルーチン ..... SMG-171  
 SMG\$GET\_KEYBOARD\_ATTRIBUTES ルーチン ..... 3-2, SMG-174  
 SMG\$GET\_NUMERIC\_DATA ルーチン .... 5-3, SMG-176  
 SMG\$GET\_PASTEBOARD\_ATTRIBUTES ルーチン ..... 1-7, SMG-178  
 SMG\$GET\_PASTING\_INFO ルーチン ..... SMG-181  
 SMG\$GET\_TERM\_DATA ルーチン ..... 5-2, 5-3, 5-18, SMG-184, SMG-193, SMG-194, SMG-195, SMG-196  
 SMG\$GET\_VIEWPORT\_CHAR ルーチン .. 2-17, SMG-187  
 SMG\$HOME\_CURSOR ルーチン ..... 2-9, SMG-191  
 SMG\$INIT\_TERM\_TABLE\_BY\_TYPE ルーチン .... 5-3, SMG-176, SMG-177, SMG-184, SMG-194, SMG-195  
 SMG\$INIT\_TERM\_TABLE ルーチン ..... 5-3, SMG-176, SMG-177, SMG-184, SMG-193  
 SMG\$INSERT\_CHARS ルーチン ..... 2-10, SMG-198, SMG-200, SMG-262  
 SMG\$INSERT\_LINE ルーチン ..... 2-10, SMG-205, SMG-277  
 SMG\$INVALIDATE\_DISPLAY ルーチン ..... SMG-213  
 SMG\$SK\_TRM\_BUFFER\_FULL ルーチン .... 3-5  
 SMG\$SK\_TRM\_CANCELLED ルーチン ..... 3-5  
 SMG\$SK\_TRM\_TIMEOUT ルーチン ..... 3-5  
 SMG\$SK\_TRM\_UNKNOWN ルーチン ..... 3-5  
 SMG\$KEYCODE\_TO\_NAME ルーチン .... 3-5, SMG-215  
 SMG\$SM\_BUF\_ENABLED ルーチン ... SMG-38  
 SMG\$SM\_DISPLAY\_CONTROLS ルーチン ..... 2-21  
 SMG\$SM\_INVISIBLE ルーチン ..... 2-15  
 SMG\$SM\_SEND\_EOF ルーチン ..... 4-4  
 SMG\$SMOVE\_TEXT ルーチン .... 2-5, SMG-234  
 SMG\$SMOVE\_VIRTUAL\_DISPLAY ルーチン ..... 2-5, SMG-238  
 SMG\$PASTE\_VIRTUAL\_DISPLAY ルーチン ..... 2-2, 2-15, 2-16, SMG-42, SMG-62, SMG-103, SMG-244, SMG-345  
 SMG\$POP\_VIRTUAL\_DISPLAY ルーチン .. 2-6, 6-2, SMG-247  
 SMG\$PUT\_CHARS\_HIGHWIDE ルーチン ..... SMG-255  
 SMG\$PUT\_CHARS\_MULTI ルーチン ..... SMG-259  
 SMG\$PUT\_CHARS\_WIDE ルーチン .. SMG-264  
 SMG\$PUT\_CHARS ルーチン .... 2-11, SMG-29, SMG-249, SMG-403  
 SMG\$PUT\_HELP\_TEXT ルーチン ... SMG-269  
 SMG\$PUT\_LINE\_HIGHWIDE ルーチン ..... SMG-282, SMG-293  
 SMG\$PUT\_LINE\_MULTI ルーチン ... SMG-286  
 SMG\$PUT\_LINE\_WIDE ルーチン ... SMG-290  
 SMG\$PUT\_LINE ルーチン ..... 2-11, SMG-209, SMG-274, SMG-293, SMG-403  
 SMG\$PUT\_STATUS\_LINE ルーチン .. SMG-296  
 SMG\$PUT\_WITH\_SCROLL ルーチン ..... SMG-279  
 SMG\$READ\_COMPOSED\_LINE ルーチン ..... 1-9, 3-2, 3-3, 3-4, 3-5, 3-8, SMG-4, SMG-10, SMG-11, SMG-48, SMG-130, SMG-173, SMG-215, SMG-226, SMG-242, SMG-243, SMG-298, SMG-324, SMG-401  
 SMG\$READ\_FROM\_DISPLAY ルーチン .. 2-15, SMG-305  
 SMG\$READ\_KEYSTROKE ルーチン ... 1-9, 3-1, 3-2, 3-8, 7-1, SMG-10, SMG-11, SMG-130, SMG-215, SMG-216, SMG-242, SMG-243, SMG-311  
 SMG\$READ\_STRING ルーチン .. 1-9, 3-2, 3-5, 3-8, SMG-10, SMG-11, SMG-130, SMG-215, SMG-242, SMG-243, SMG-318  
 SMG\$READ\_VERIFY ルーチン .. 1-9, 3-2, 3-5, 3-8, SMG-10, SMG-11, SMG-130, SMG-215, SMG-242, SMG-243, SMG-329  
 SMG\$REMOVE\_LINE ルーチン ..... 2-13, SMG-337  
 SMG\$REPAINT\_SCREEN ルーチン .. SMG-340

SMG\$REPASTE\_VIRTUAL\_DISPLAY ルーチン ..... 2-4, SMG-344

SMG\$REPLACE\_INPUT\_LINE ルーチン ..... SMG-351

SMG\$RESTORE\_PHYSICAL\_SCREEN ルーチン ..... 6-3, SMG-355, SMG-366, SMG-367

SMG\$RETURN\_CURSOR\_POS ルーチン ... 2-8, SMG-357

SMG\$RETURN\_INPUT\_LINE ルーチン ..... SMG-359

SMG\$RING\_BELL ルーチン ..... SMG-364

SMG\$SAVE\_PHYSICAL\_SCREEN ルーチン ..... 6-3, SMG-355, SMG-366

SMG\$SAVE\_VIRTUAL\_DISPLAY ルーチン ..... 2-19, SMG-232, SMG-369

SMG\$SCROLL\_DISPLAY\_AREA ルーチン ..... SMG-371

SMG\$SCROLL\_VIEWPORT ルーチン ... 2-16, SMG-374

SMG\$SELECT\_FROM\_MENU ルーチン .. 2-18, SMG-54, SMG-381

SMG\$SET\_BROADCAST\_TRAPPING ルーチン ..... 4-2, SMG-110, SMG-391

SMG\$SET\_CURSOR\_ABS ルーチン ..... 2-9, SMG-394, SMG-403

SMG\$SET\_CURSOR\_MODE ルーチン ..... SMG-396

SMG\$SET\_CURSOR\_REL ルーチン ..... 2-9, SMG-398

SMG\$SET\_DEFAULT\_STATE ルーチン ..... SMG-400

SMG\$SET\_DISPLAY\_SCROLL\_REGION ルーチン ..... SMG-372, SMG-402

SMG\$SET\_KEYPAD\_MODE ルーチン ..... 3-3, SMG-404

SMG\$SET\_OUT\_OF\_BAND\_ASTS ルーチン ..... 4-2, SMG-406

SMG\$SET\_PHYSICAL\_CURSOR ルーチン ..... 4-4, SMG-409

SMG\$SET\_TERM\_CHARACTERISTICS ルーチン ..... 2-19, SMG-411

SMG\$SNAPSHOT ルーチン ..... SMG-57

SMG\$TERMTABLE ..... SMG-194, SMG-196

SMG\$UNPASTE\_VIRTUAL\_DISPLAY ルーチン ... 2-3, 2-16, 6-2, SMG-345, SMG-414

SMG\$LABEL\_BORDER ルーチン ... SMG-217

SMG\$NAME\_TO\_KEYCODE ルーチン ... 3-5, SMG-242

SMG\$LIST\_KEY\_DEFS ルーチン ... SMG-224

SMG\$LIST\_PASTING\_ORDER ルーチン ... 1-8, 2-7, SMG-227

SMG\$LOAD\_KEY\_DEFS ルーチン ..... 3-3, SMG-48, SMG-230

SMG\$LOAD\_VIRTUAL\_DISPLAY ルーチン ..... 2-19, SMG-232, SMG-370

SMGTERMS.TXT ..... 5-3

SS\$ABORT 状態コード ..... SMG-11

SS\$CANCEL 状態コード ..... SMG-11

STATE 属性 ..... 3-4

STRING\_2 機能 ..... 2-12

Super DEC 漢字コードセット ... 2-19, SMG-30, SMG-69, SMG-169, SMG-200, SMG-208, SMG-220, SMG-252, SMG-257, SMG-262, SMG-267, SMG-272, SMG-276, SMG-284, SMG-288, SMG-292, SMG-297, SMG-303, SMG-307, SMG-324, SMG-331, SMG-335, SMG-352, SMG-360

SYSS\$SYSTEM:SMGBLDTRM.EXE ..... 5-22

SYSS\$SYSTEM:SMGTERMS.TXT ..... 5-1

## T

TERM\$TABLOC 論理名 ..... 5-20

TERMINATE 属性 ..... 3-4

TERMTABLE  
 インターフェイス・ルーチン ..... 5-2, 5-18  
 エントリ ..... 5-4  
 サポート機能 ..... 5-1  
 定義 ..... 2-12  
 データベース ..... 5-1, SMG-193, SMG-195

TERMTABLE.EXE ..... 5-1, 5-20, SMG-84, SMG-184, SMG-194, SMG-196  
 作成 ..... 5-22  
 ファイル ..... 5-2

TERMTABLE.TXT ..... 5-3, 5-19, 5-22  
 ファイル ..... 2-12, 5-1

## V

VAX FMS ..... 1-5

## A

アウト・オブ・バンド AST ..... 4-2

アンペースト ..... 1-7  
 操作 ..... 2-3

## I

移動操作 ..... 2-5

## カ

外部テキストの表示 ..... 2-14

書き込み操作 ..... 2-11

重なり合い ..... 2-7  
 確認操作 ..... 2-6

仮想キーボード ..... 1-1, 1-9, 3-1  
 属性の設定と検索 ..... 3-2  
 設定 ..... 3-1  
 データの取り込み ..... 3-1  
 入力 ..... 3-1

仮想ディスプレイ ..... 1-1, 1-6, 1-7  
 識別子 (display-id) ..... 1-8  
 出力 ..... 2-7

|                                            |                                |
|--------------------------------------------|--------------------------------|
| 仮想ディスプレイ (続き)                              |                                |
| 属性の変更                                      | 2-12                           |
| 保存                                         | 2-19                           |
| 読み込み                                       | 2-15                           |
| カーソル                                       |                                |
| 位置                                         | 2-8                            |
| 移動                                         | 4-4                            |
| 表示と消去の切り替え                                 | SMG-397                        |
| かな漢字変換                                     | 3-2, SMG-304, SMG-324, SMG-335 |
| 画面                                         |                                |
| 最小限の更新                                     | 2-20                           |
| 画面管理                                       |                                |
| 使用する機能フィールド                                | 5-23                           |
| 設定                                         | 1-1                            |
| 画面消去                                       | SMG-39                         |
| 漢字ターミナル                                    |                                |
| SET TERMINAL コマンドと SHOW TERMINAL コマンドのサポート | 5-25                           |
| 間接呼び出し                                     | 6-1                            |

## キ

|                     |                       |
|---------------------|-----------------------|
| 機能フィールド             | 5-4                   |
| 画面管理が使用する           | 5-23                  |
| 区切り文字               | 5-4                   |
| 作成                  | 5-19                  |
| 数値                  | 5-7                   |
| 通常の ASCII 値による文字の使用 | 5-16                  |
| パッド文字の追加            | 5-17                  |
| プリントされない文字          | 5-15                  |
| 文字列                 | 5-8                   |
| ユーザ定義レンディション        | 5-14                  |
| 論理                  | 5-5                   |
| キーパッド・キーによる行作成機能    | 3-3                   |
| キーボード情報テーブル (KIT)   | 3-2, SMG-174, SMG-175 |
| 行単位の出力              | 2-11                  |
| 行の作成                | 3-3                   |

## ケ

|        |     |
|--------|-----|
| 現在の状態名 | 3-4 |
|--------|-----|

## コ

|      |     |
|------|-----|
| 構成操作 | 2-1 |
|------|-----|

## サ

|            |          |
|------------|----------|
| 最小限の画面更新   | SMG-39   |
| 再ペースト操作    | 2-4      |
| 削除する操作     | 2-13     |
| 削除操作       | 2-5, 2-9 |
| サブプロセス     | 4-3      |
| 削除         | 4-3      |
| 作成         | 4-3      |
| 内でのコマンドの実行 | 4-3      |

|         |      |
|---------|------|
| 算術演算    | 5-18 |
| で使用する文字 | 5-19 |

## シ

|                         |      |
|-------------------------|------|
| 終了ハンドラ                  | 4-4  |
| 終了文字                    | 3-5  |
| 値                       | 3-5  |
| シンボリック定義                | 3-7  |
| コード                     | 3-5  |
| 出力操作                    | 2-1  |
| 出力のバッチ処理                | 2-21 |
| 日本語 SMG 出力ルーチン          | 1-1  |
| 消去操作                    | 2-9  |
| 状態                      | 3-4  |
| 省略可能な引数 (character-set) | 2-19 |
| 省略時文字集合名                | 2-19 |

## ス

|       |                  |
|-------|------------------|
| 垂直線   | 2-13             |
| 水平線   | 2-13             |
| 数値機能  | 5-7, 5-8         |
| フィールド | 5-7              |
| スクロール | 2-16             |
| ジャンプ  | SMG-396, SMG-397 |
| 連続    | SMG-396, SMG-397 |

## セ

|        |         |
|--------|---------|
| 選択メニュー | SMG-381 |
| 終了     | SMG-97  |
| 表示     | SMG-50  |

## ソ

|       |      |
|-------|------|
| 属性を検索 | 2-17 |
| 操作制御  | 2-20 |
| 挿入操作  | 2-10 |

## タ

|                         |                  |
|-------------------------|------------------|
| タブ                      | 2-21             |
| タブ解除                    | SMG-39           |
| タブ・ストップ                 | 2-21             |
| タブ文字                    | 2-21             |
| ターミナル機能テーブル (TERMTABLE) | SMG-176, SMG-184 |
| ターミナル・コマンド・シーケンス        | 5-18             |
| ターミナル属性の変更              | 2-19             |
| 単一文字終了文字                | 3-5              |

## チ

|          |      |
|----------|------|
| 直接アクセス装置 | 2-11 |
| 直接呼び出し   | 6-1  |

## ト

|          |      |
|----------|------|
| 特殊な操作モード | 2-20 |
|----------|------|

## ニ

|                  |      |
|------------------|------|
| 日本語画面管理ライブラリ使用方法 | 1-10 |
| 入力操作             | 3-1  |
| 入力と出力の相互作用       | 3-8  |
| 日本語 SMG 入力ルーチン   | 1-4  |

## ハ

|           |        |
|-----------|--------|
| バッファ処理モード | 2-20   |
| バッファ操作    | 2-20   |
| バッファリング   | SMG-39 |

## ヒ

|         |            |
|---------|------------|
| 引数の置き換え | 5-18       |
| 非同期イベント | 4-1        |
| ビットの設定  | 2-12       |
| ビット・マスク | 2-12       |
| ビューポート  | 2-15, 2-17 |
| アンペースト  | 2-16       |
| 移動      | 2-16       |
| 削除      | 2-16       |
| 作成      | 2-15       |
| スクロール   | 2-16       |
| 属性の変更   | 2-17       |
| 操作      | 1-8, 2-15  |
| ペースト    | 2-16       |

## フ

|                |                |
|----------------|----------------|
| フォーリン・ターミナル    | 5-1, 5-4, 5-23 |
| サポート           | 5-1            |
| サポート・ルーチン      | 5-17           |
| 設定             | 5-1            |
| 入力のサポート        | 5-23           |
| ルーチン           | 5-1            |
| ブロードキャスト・メッセージ | 4-1            |

## ヘ

|                     |          |
|---------------------|----------|
| ペースト                | 1-7      |
| ペーストボード             | 1-5, 2-1 |
| 更新のバッチ処理            | 2-21     |
| 作成                  | SMG-55   |
| 座標系                 | 1-6      |
| 識別子 (pasteboard-id) | 1-6      |

## ペーストボード (続き)

|          |      |
|----------|------|
| バッチ処理レベル | 2-22 |
|----------|------|

## ホ

|        |                                                                                                                                                                                                                         |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 補助文字集合 | 2-19, SMG-30, SMG-69,<br>SMG-169, SMG-200, SMG-208, SMG-220,<br>SMG-252, SMG-257, SMG-262, SMG-267,<br>SMG-272, SMG-276, SMG-284, SMG-288,<br>SMG-292, SMG-297, SMG-303, SMG-307,<br>SMG-324, SMG-335, SMG-352, SMG-360 |
| ポップ操作  | 2-5                                                                                                                                                                                                                     |

## ミ

|         |                              |
|---------|------------------------------|
| 未定義文字集合 | SMG-169, SMG-307,<br>SMG-360 |
|---------|------------------------------|

## メ

|         |         |
|---------|---------|
| メニュー    | 2-17    |
| 削除      | 2-18    |
| 作成      | 2-17    |
| 選択      | 2-18    |
| メールボックス | SMG-131 |

## モ

|             |                                                                                                                                                                                                                                              |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 文字シーケンス終了文字 | 3-5                                                                                                                                                                                                                                          |
| 文字集合        |                                                                                                                                                                                                                                              |
| ASCII       | 2-19, SMG-30, SMG-69, SMG-169,<br>SMG-200, SMG-208, SMG-220, SMG-252,<br>SMG-257, SMG-262, SMG-267, SMG-272,<br>SMG-276, SMG-284, SMG-288, SMG-292,<br>SMG-297, SMG-303, SMG-307, SMG-324,<br>SMG-335, SMG-352, SMG-360                      |
| DEC 漢字      | 2-19, SMG-30, SMG-69,<br>SMG-142, SMG-169, SMG-200, SMG-208,<br>SMG-220, SMG-252, SMG-257, SMG-262,<br>SMG-267, SMG-272, SMG-276, SMG-284,<br>SMG-288, SMG-292, SMG-297, SMG-303,<br>SMG-307, SMG-324, SMG-335, SMG-352,<br>SMG-360, SMG-373 |
| DEC 特殊      | 2-19, SMG-30, SMG-69,<br>SMG-169, SMG-200, SMG-208, SMG-220,<br>SMG-252, SMG-257, SMG-262, SMG-267,<br>SMG-272, SMG-276, SMG-284, SMG-288,<br>SMG-292, SMG-297, SMG-303, SMG-307,<br>SMG-324, SMG-335, SMG-352, SMG-360                      |
| DEC 補助      | 2-19, SMG-30, SMG-69,<br>SMG-169, SMG-200, SMG-208, SMG-220,<br>SMG-252, SMG-257, SMG-262, SMG-267,<br>SMG-272, SMG-276, SMG-284, SMG-288,<br>SMG-292, SMG-297, SMG-303, SMG-307,<br>SMG-324, SMG-335, SMG-352, SMG-360                      |

文字集合 (続き)

ISO Latin-1 補助 . . . 2-19, SMG-30, SMG-69,  
SMG-169, SMG-200, SMG-208, SMG-220,  
SMG-252, SMG-257, SMG-262, SMG-267,  
SMG-272, SMG-276, SMG-284, SMG-288,  
SMG-292, SMG-297, SMG-303, SMG-307,  
SMG-324, SMG-335, SMG-352, SMG-360

JIS カタカナ . . . . . 2-19, SMG-30, SMG-69,  
SMG-169, SMG-200, SMG-208, SMG-220,  
SMG-252, SMG-257, SMG-262, SMG-267,  
SMG-272, SMG-276, SMG-284, SMG-288,  
SMG-292, SMG-297, SMG-303, SMG-307,  
SMG-324, SMG-335, SMG-352, SMG-360

SDK . . . . . 2-19

補助 . . . . 2-19, SMG-30, SMG-69, SMG-169,  
SMG-200, SMG-208, SMG-220, SMG-252,  
SMG-257, SMG-262, SMG-267, SMG-272,  
SMG-276, SMG-284, SMG-288, SMG-292,  
SMG-297, SMG-303, SMG-307, SMG-324,  
SMG-335, SMG-352, SMG-360

未定義 . . . . . SMG-169, SMG-307, SMG-360

文字単位の出力 . . . . . 2-11

文字列機能 . . . . . 5-9

    フィールド . . . . . 5-8

ラ

---

ライン描画文字 . . . . . 2-13

    セット . . . . . 5-17

レ

---

レンディション (属性)

    ビデオ属性 . . . . . 1-8, SMG-16

ロ

---

論理機能フィールド . . . . . 5-5

論理値 . . . . . 5-5

    機能 . . . . . 5-5



日本語 OpenVMS  
日本語画面管理ライブラリ 利用者の手引き

---

2004 年 2 月 発行

日本ヒューレット・パカード株式会社

〒102-0076 東京都千代田区五番町 7 番地

電話 (03)3512-5700 (大代表)

---

AA-PU8RF-TE