

# 4D View

---

ランゲージリファレンス  
Windows® and Mac™ OS



---

# 4D View ランゲージリファレンス

**Windows® and Mac™ OS**

Copyright© 1995 - 2003 4D SA

All rights reserved.

---

このマニュアルに記載されている事項は、将来予告なしに変更されることがあり、いかなる変更に関しても 4D SA は一切の責任を負いかねます。このマニュアルで説明されるソフトウェアは、本製品に同梱の License Agreement（使用許諾契約書）のもとでのみ使用することができます。

ソフトウェアおよびマニュアルの一部または全部を、ライセンス保持者がこの契約条件を許諾した上での個人使用目的以外に、いかなる目的であれ、電子的、機械的、またどのような形であっても、無断で複製、配布することはできません。

4th Dimension、4D Server、4D、4D ロゴ、4D ロゴ、およびその他の 4D 製品の名称は、4D SA の商標または登録商標です。

Microsoft と Windows は Microsoft Corporation 社の登録商標です。

Apple, Macintosh, Mac, Power Macintosh, Laser Writer, Image Writer, ResEdit, QuickTime は Apple Computer Inc. の登録商標または商標です。

その他、記載されている会社名、製品名は、各社の登録商標または商標です。

## 注意

このソフトウェアの使用に際し、本製品に同梱の License Agreement（使用許諾契約書）に同意する必要があります。ソフトウェアを使用する前に、License Agreement を注意深くお読みください。

<b>第 1 章</b>	<b>はじめに</b> .....	<b>1 1</b>
	4D View とは .....	11
	4D View のドキュメントについて .....	11
	表記方法について .....	12
	4D View のコマンドと定数 .....	13
	メソッドエディタ内のコマンド .....	13
	エクスプローラ上の定数 .....	14
	4D View エリアを使用する .....	15
	4D View エリアの参照 .....	15
	フォーム上の 4D View プラグインエリア .....	16
	4D View プラグインウィンドウ .....	17
	4D View オフスクリーンエリア .....	17
	4D View メニューコマンドの使用 .....	17
	クロスプラットフォームドキュメントの管理 .....	18
	MacOS と Windows ドキュメントの対応 .....	18
	4D View ドキュメント .....	18
<b>第 2 章</b>	<b>PV Current cell</b> .....	<b>2 1</b>
	PV Current cell、はじめに .....	21
	PV GET CURRENT CELL .....	22
	PV VALIDATE CURRENT CELL .....	23
	PV GET NEXT FREE CELL .....	24
	PV GOTO CELL .....	25
	PV GOTO NEXT CELL .....	26
	PV GET PREVIOUS ACTIVE CELL .....	27
<b>第 3 章</b>	<b>PV Area</b> .....	<b>2 9</b>
	PV Area、はじめに .....	29
	コールバックメソッド .....	29
	PV New offscreen area .....	30
	PV DELETE OFFSCREEN AREA .....	31

	PV SET AREA PROPERTY	32
	PV Get area property	45
	PV BLOB TO AREA	46
	PV Area to blob	47
	PV ON EVENT	48
	PV Get on event method	50
	PV ON COMMAND	51
	PV Get on command method	52
	PV ON ERROR	53
	PV Get on error method	54
	PV GET LAST ERROR	54
	PV EXECUTE COMMAND	56
	PV SET COMMAND STATUS	57
	PV GET COMMAND STATUS	58
	PV REDRAW	59
<b>第 4 章</b>	<b>PV Borders</b>	<b>61</b>
	PV Borders、はじめに	61
	PV SET RANGE BORDER	62
	PV SET BORDER STYLE	63
	PV GET BORDER STYLE	65
<b>第 5 章</b>	<b>PV Cell property</b>	<b>67</b>
	PV Cell property、はじめに	67
	PV SET CELL PROPERTY	68
	PV SET RANGE PROPERTY	77
	PV Get cell property	85
	PV Get range property	86
	PV SET CELL NAME	88
	PV Get cell name	89
	PV GET CELL NAME LIST	90
<b>第 6 章</b>	<b>PV Cell value</b>	<b>91</b>
	PV Cell value、はじめに	91
	PV Get cell value type	92
	PV SET CELL STRING VALUE	93
	PV Get cell string value	94
	PV SET CELL TEXT VALUE	95

PV Get cell text value	96
PV SET CELL NUM VALUE	97
PV Get cell num value	98
PV SET CELL DATE VALUE	99
PV Get cell date value	100
PV SET CELL TIME VALUE	101
PV Get cell time value	102
PV SET CELL DATE TIME VALUE	103
PV GET CELL DATE TIME VALUE	104
PV SET CELL BOOLEAN VALUE	105
PV Get cell boolean value	106
PV SET CELL PICTURE VALUE	107
PV Get cell picture value	108
PV SET CELL VARIABLE	109
PV Get cell variable	110
PV SET CELL FIELD	111
PV GET CELL FIELD	113
PV SET CELL FORMULA	114
PV Get cell formula	115
PV SET CELL CONTROL	116
PV GET CELL CONTROL	120
PV ARRAY TO CELLS	122
PV CELLS TO ARRAY	124
PV FIELD TO CELLS	126
PV FIELDS LIST TO CELLS	128
PV REPORT MANY	130
PV REPORT ONE	133
PV ADD DYNAMIC ARRAYS	136
PV ADD DYNAMIC FIELDS	138
PV CLEAR DYNAMIC COLUMNS	141

## 第 7 章 PV Cell manipulation..... 143

PV Cell manipulation、はじめに	143
PV FIND ONE	144
PV FIND ALL	146
PV REPLACE ONE	147
PV REPLACE ALL	149
PV SORT ONE	150

	PV SORT MANY	152
	PV SPECIAL CLEAR	153
	PV SPECIAL PASTE	154
	PV SPECIAL CUT	155
<b>第 8 章</b>	<b>PV Columns &amp; Rows</b>	<b>157</b>
	PV Columns & rows、はじめに	157
	行とカラムの名前	157
	PV INSERT COLUMNS	158
	PV INSERT ROWS	159
	PV INSERT CELLS	160
	PV DELETE COLUMNS	161
	PV DELETE ROWS	161
	PV DELETE CELLS	162
	PV SET COLUMNS WIDTH	163
	PV SET ROWS HEIGHT	164
	PV Get column width	165
	PV Get row height	165
	PV SET COLUMN HEADER	166
	PV GET COLUMN HEADER	167
	PV SET ROW HEADER	167
	PV Get row header	168
<b>第 9 章</b>	<b>PV Document</b>	<b>169</b>
	PV Document、はじめに	169
	PV OPEN DOCUMENT	170
	PV SAVE DOCUMENT	172
	PV EXPORT	174
	PV SET DOCUMENT PROPERTY	176
	PV Get document property	178
	PV SET DOCUMENT INFO	179
	PV GET DOCUMENT INFO	180
<b>第 10 章</b>	<b>PV Panes</b>	<b>181</b>
	PV Panes、はじめに	181
	PV ADD VERT SPLITTER	182
	PV ADD HOR SPLITTER	183
	PV REMOVE VERT SPLITTER	184

	PV REMOVE HOR SPLITTER	185
	PV SET VERT PANE PROPERTY	186
	PV SET HOR PANE PROPERTY	187
	PV Get vert pane property	188
	PV Get hor pane property	189
<b>第 11 章</b>	<b>PV Pictures</b>	<b>191</b>
	PV Pictures、はじめに	191
	PV Add picture	192
	PV REMOVE PICTURE	193
	PV Get picture	194
	PV Create picture	195
	PV SET PICTURE PROPERTY	196
	PV Get picture property	197
<b>第 12 章</b>	<b>PV Printing</b>	<b>199</b>
	PV Printing、はじめに	199
	PV PRINT	200
	PV PRINT FORMULAS	201
	PV SET PRINT HEADER	202
	PV Get print header	203
	PV SET PRINT PROPERTY	204
	PV Get print property	208
<b>第 13 章</b>	<b>PV Selection</b>	<b>209</b>
	PV Selection、はじめに	209
	選択範囲とセル範囲	209
	PV SELECT CELL	210
	PV Is cell selected	211
	PV SELECT RANGE	212
	PV Is range selected	213
	PV SELECT RANGES LIST	214
	PV GET SELECTED RANGES LIST	215
	PV SELECT COLUMNS	216
	PV Is column selected	217
	PV SELECT ROWS	217
	PV Is row selected	218
	PV Is all selected	219

	PV SELECT ALL .....	219
<b>第 14 章</b>	<b>PV Style .....</b>	<b>2 2 1</b>
	PV Style、はじめに .....	221
	スタイルシート .....	221
	キャラクタフォント .....	222
	フォーマット .....	222
	PV Add style .....	222
	PV REMOVE STYLE .....	223
	PV SET STYLE NAME .....	224
	PV GET STYLE LIST .....	225
	PV SET STYLE PROPERTY .....	226
	PV Get style property .....	228
	PV Add font .....	229
	PV REMOVE FONT .....	230
	PV GET FONT LIST .....	231
	PV Add format .....	232
	PV REMOVE FORMAT .....	233
	PV SET FORMAT .....	234
	PV GET FORMAT LIST .....	235
<b>第 15 章</b>	<b>PV Drag and drop .....</b>	<b>2 3 7</b>
	Drag and Drop、はじめに .....	237
	PV SET DRAG SIGNATURES .....	238
	PV GET DRAG SIGNATURES .....	241
	PV SET DROP SIGNATURES .....	242
	PV GET DROP SIGNATURES .....	243
	PV Get drag and drop info .....	244
	PV GET DRAG SOURCE .....	245
	PV GET DROP TARGET .....	246
<b>第 16 章</b>	<b>PV Plugin Property .....</b>	<b>2 4 7</b>
	PV Plugin Property、はじめに .....	247
	PV SET PLUGIN PROPERTY .....	248
	PV Get plugin property .....	250
<b>第 17 章</b>	<b>PV Tools .....</b>	<b>2 5 1</b>
	PV Tools、はじめに .....	251
	PV Copy to blob .....	252

	PV PASTE FROM BLOB	253
	PV RGB to color	254
	PV COLOR TO RGB	255
	PV Color to index	256
	PV Index to color	257
	PV SET WINDOW TITLE	258
	PV Get window title	259
<b>第 18 章</b>	<b>PV Allowed Input</b>	<b>261</b>
	PV Allowed Input、はじめに	261
	PV SET ALLOWED VAR LIST	262
	PV GET ALLOWED VAR LIST	262
	PV SET ALLOWED MET LIST	263
	PV GET ALLOWED MET LIST	263
	PV SET ALLOWED COM LIST	264
	PV GET ALLOWED COM LIST	264
<b>第 19 章</b>	<b>付録</b>	<b>265</b>
	付録A、4D View エラーコード一覧	265
<b>定数</b>		<b>269</b>
	PV Area properties	269
	PV Arrow keys	270
	PV Border edge	270
	PV Border style	271
	PV Carriage return	271
	PV Cell properties	272
	PV Cell value type	272
	PV Commands	272
	PV Control	276
	PV Directions	276
	PV Document format	277
	PV Document properties	277
	PV Drag drop allowed	278
	PV Drop action	278
	PV Drop info	279
	PV Drop mode	279
	PV Event	279

PV Header sort	280
PV Headers& footers	280
PV Input enter key mode	280
PV Pane properties	281
PV Picture mapping mode	281
PV Picture properties	282
PV Plugin properties	282
PV Print properties	283
PV Report functions	283
PV Select mode	284
PV Selection action	284
PV Style format date time	284
PV Style properties	285
PV Style special values	286
PV Style values	287
PV Triggers	288

<b>コマンド索引</b>	<b>289</b>
---------------	------------

## 4D View とは

---

4D View プラグインは4th Dimension 言語に各種ルーチンを追加し、手動で行われていたさまざまな作業を自動化します。

4D View コマンドを使用することにより、以下の事柄が実現します。

- あらゆる 4D View メニューコマンドの実行
- ドキュメントのオープンおよび保存
- ドキュメントのヘッダやフッタの設定
- 表示属性や入力属性の設定
- 計算、ピクチャ、フィールド等に対するフォーミュラを用いた作業

4D View のルーチン名は、接頭辞“PV”で始まるため（スペースが1つ入ります）、標準の4th Dimension ルーチンや他のプラグインのルーチンと区別することができます。

## 4D View のドキュメントについて

4D View のドキュメントは、ユーザマニュアルとランゲージマニュアルという2つのマニュアルから構成されています。

本マニュアル『ランゲージリファレンス』では、製品の使用方法および4D View のプログラミング言語のシンタックスについて説明しています。メニューや4D View プラグインの全般的な使用方法については、4D View 『ユーザリファレンス』マニュアルを参照してください。

## 表記方法について

---

本マニュアルにおいて、4D View コマンドは特殊フォントを使用した大文字で表わされています（例：**PV OPEN DOCUMENT**）。

関数（戻り値を返すルーチン）は、頭文字のみが大文字で、後は小文字で表わされています（例：**PV Get on command method**）。

メソッドエディタにおいて、4th Dimension のビルトインコマンドと区別するため、4D View コマンドは太字の斜体（ボールドイタリック）の書体で表示されます。

```
        ` 4th Dimension コマンド
QUERY ([Clients];[Clients]S_Kode=cLi_Scode_V)
If (Records in selection ([Clients])=1)
        ` 4D View コマンド
⇒ PV BLOB TO AREA (theArea;Table(->[Clients]);Field(->[Clients]Pict))
End if
```

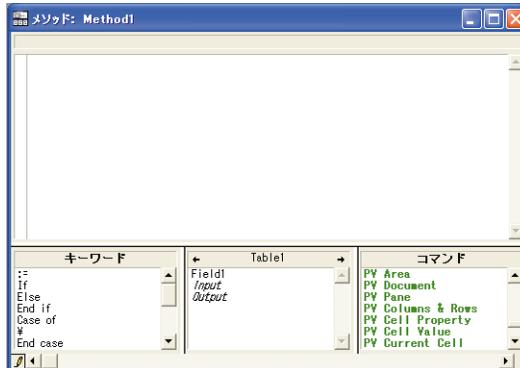
本ドキュメントでは、4D View コマンドの使用方法を説明するため、各種コマンドの例題が数多く提供されています。コマンドのなかには、例題を提示することにより効果的に説明できるものもあります。また、チェックマーク（→）は、説明しているコマンドの名前が例題コード上で現われる行を示しています。

本ドキュメントのいくつかの例題において、余白の都合によりコードの行が2行目または3行目へと続くものがあります。しかし、これらの例題をコードに入力する場合には、改行を行わず1行で記述してください。

## 4D View のコマンドと定数

### メソッドエディタ内のコマンド

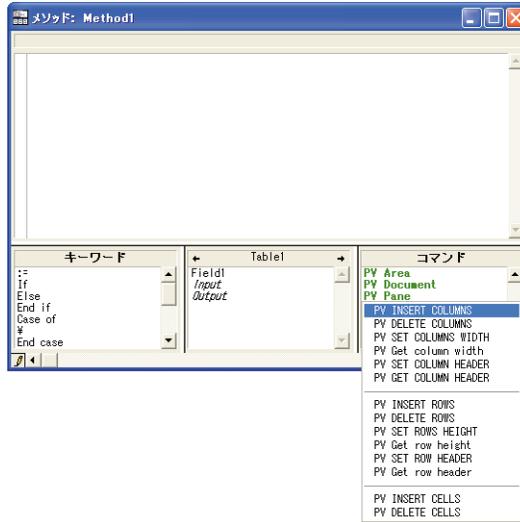
メソッドエディタにおいて、4D View のコマンドはグループ化され、ルーチンリストの最後に配置されています。



複数のプラグインがインストールされている場合、リスト上の各テーマはプラグインがロードされた順に表示されます。

4D View のルーチンは常にテーマ別にグループ化されます (4th Dimension のコマンドはアルファベット順、またはテーマ別に表示されます)。

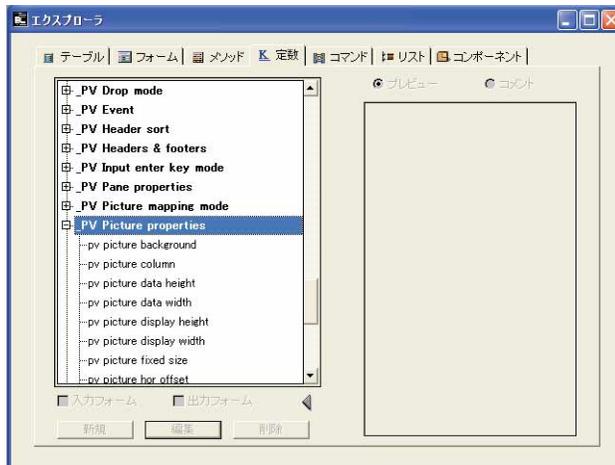
メソッドにおける 4D View コマンドの呼び出し方法は、4th Dimension コマンドと同じです。つまり、ウインドウ内に直接入力するか、またはルーチンリストのポップアップメニューからコマンドを選択します。



4D View ルーチンは、すべてのメソッドタイプ、すなわちデータベース、プロジェクト、フォーム、オブジェクト、トリガの各メソッドで利用することができます。

## エクスプローラ上の定数

4D エクスプローラにおいて、4D View の定数はテーマ別に表示されます。4D View の定数名は “\_” で始まるため、リストの最後に配置されています。



メソッドに4D View の定数を挿入するには、メソッドエディタに定数を入力するか、4D エクスプローラからドラッグします。

本マニュアルにおいて、4D Viewの定数は「4D View 6.8定数テーマリスト」の節にアルファベット順で記載されています。

## 4D View エリアを使用する

---

下記の4D View環境のもとで、プログラムを作成することができます。

- フォームに組み込まれた4D Viewエリア
- 4D Viewのプラグインウインドウ
- オフスクリーン4D Viewエリア

4D Viewドキュメントを利用するには、フォーム上にプラグインエリアを作成するか、またはプラグインウインドウを開きます。

フォーム上にプラグインエリアを作成するには、「デザイン」モードにおいて4Dのフォームエディタでエリアを描画します。

プラグインウインドウを開くには、4Dの「ユーザ」モードで「プラグイン」メニューから4D Viewコマンドを選択するか、または4Dのコマンドである **Open external window** 関数を実行します。

4D Viewでは、表示エリアの作成だけではなく、非表示エリアであるオフスクリーンエリアを作成することもできます。詳細は後述の「4D View オフスクリーンエリア」の節を参照してください。

## 4D View エリアの参照

各ルーチンを使用して4D Viewドキュメントを変更したら、4D ViewエリアのID番号を特定する必要があります。このID番号は4D Viewの内部的なものであり、通常は変数に納められます。

4D Viewは変数を使用して、組み込みエリアのパス名やプラグインウインドウ、およびオフスクリーンエリアを保存します。操作対象のエリアを参照するには、コマンドや関数に対し、エリアのID番号を代入した変数を引数として渡します。

本マニュアルのコマンド説明において、4D Viewドキュメントエリアを識別する倍長整数タイプの変数は、<エリア>と呼ばれます。

以下の2つのタイプの<エリア>変数があります。

■ 組み込みエリア名

■ プラグインウインドウまたはオフスクリーンエリア用に作成した変数

### 組み込みエリア名

フォーム上に4D View エリアを作成して名前を付けると、4th Dimension は4D View エリアの名前がそのエリアを参照するための変数であるものと解釈します。例えば、“Sheet” という名前のエリアを参照するには、引数<エリア>に“Sheet”と指定します。

### プラグインウインドウおよびオフスクリーンエリアのID

**Open external window** 関数または **PV New offscreen area** 関数を使用して、プラグインウインドウやオフスクリーンエリアを作成すると、関数よりエリアのID番号が返されます。このエリアのID番号は、必ず変数に代入して保存しておいてください。この変数を使用し、他のコマンドや関数においてプラグインウインドウやオフスクリーンエリアを参照することができます。値を変数に代入するには、コード行の関数の左側に変数名と代入記号(:=)を記述します。

次の例題は、4D View のプラグインウインドウを作成し、エリアのID番号を変数MyAreaに代入しています。

```
MyArea:=Open external window(30;30;350;450;8;"Sheet";"_4D View")
```

### フォーム上の 4D View プラグインエリア

4D View エリアはあらゆるフォーム上に作成することができます。大抵の場合、エリアは入力フォーム上に作成され、ドキュメントを使った処理が行われますが、出力フォーム上で情報の表示や印刷のために使用される場合もあります。4D View はフォーム全体を使用するか、またはフィールドや他のフォームオブジェクトとフォームスペースを共有します。

アクティブオブジェクトエリアである“プラグインエリア”を4D View で使用しなければなりません。4th Dimension において、プラグインエリアは各種アクティブオブジェクトのうちの一つです（他の例として、ボタンや入力エリア、スクロールエリア等があります）。

また、プラグインエリアを4Dフィールドに関連付けると、保存のたびにエリア内容も一緒に保存することができます。「Validate」のような自動ボタンではなく、**SAVE RECORD** コマンドを使用する場合には、自動保存がアクティブではないため、まず最初に **PV Area to blob** を実行して4D View エリアの内容を4Dフィールドに転送する必要があります。ある点に注意してください。

フォームにおける4D Viewエリアの作成に関する詳細は、4D Viewの『ユーザーファレンス』マニュアルを参照してください。

## 4D View プラグインウインドウ

4th Dimensionの**Open external window**関数を使用して、プラグインウインドウを開き、空の4D Viewドキュメントを表示することができます。

**Open external window**は新しくウインドウを開き、指定されたプラグインを表示してエリアのID番号を返します。

以下に**Open external window**の使用例を示します。この例では、プラグインウインドウを開き、空の4D Viewドキュメントを表示します。

```
PvRefArea:=Open external window (50;50;350;450;8;"Spreadsheet"; "_4D View")
```

このドキュメントを参照する必要がある場合には、常に変数PvRefAreaを使用します。

## 4D View オフスクリーンエリア

オフスクリーンエリアはメモリ上に格納され、プログラマやユーザからは見ることができません。一般的に、2つの状況においてこのエリアが役立ちます。つまり、ユーザが表示する前にドキュメントを修正する場合や、必要があればユーザが元の状態へ復帰できるようにドキュメントを保存しておく場合です。

オフスクリーンエリアではエリアを再描画する必要がないため、4D Viewの処理はより高速に行われます。

オフスクリーンエリアを作成するには、**PV New offscreen area**関数を使用します。

**PV DELETE OFFSCREEN AREA** コマンドを用いてオフスクリーンエリアを消去して、使用されたメモリを解放することを忘れないでください。オフスクリーンエリアをすべて終了せずにデータベースを閉じると、4th Dimensionはエラーメッセージを表示します。

## 4D View メニューコマンドの使用

プログラムから4D Viewメニューコマンドを実行することができます。また、メソッドからメニューやメニューコマンドの状態を調べることもできます。

各メニューコマンドは整数値を用いて参照します。メニューコマンドのコードは、**PV Commands**定数テーマを使用して定義します。例えば、「ファイル」メニューのコマンドは定数“pv cmd file...”で示され、「編集」メニューのコマンドは定数“pv cmd edit...”で示されます。

メニューコマンド定数は固定値であり、4D Viewに変更が加えられた場合でも変わりません。

## クロスプラットフォームドキュメントの管理

4th Dimensionや4D Serverと同様に、4D Viewはクロスプラットフォーム対応です。つまり、4D Viewを使用しているデータベースは、MacOSで作成された場合でも、変更を加えることなくWindows上で開いて使用することができます（その逆も同じです）。もちろん、これらの組み合わせは、ソフトウェアのバージョンが適切である場合にのみ有効となります（詳細は、『4D Product Line インストールマニュアル』を参照してください）。

しかし、クロスプラットフォームの4Dデータベースと4D Viewドキュメントを管理するには、MacOSとWindowsのオペレーティングシステムの相違に関連した一定のガイドラインに従う必要があります。

### MacOS と Windows ドキュメントの対応

次の表は、標準の4D Viewドキュメントに対するMacOSおよびWindowsファイルの対応を示しています。

ドキュメント	MacOS		Windows
	タイプ	クリエータ	拡張子
4D View ドキュメント	4DPV	4DSP	.4PV
4D Calc ドキュメント	4DC	4DSP	.4DC
SYLK 2.0 ドキュメント	TEXT	4DSP	.TXT
タブ区切りテキスト	TEXT	4DSP	.TXT
HTML ドキュメント	TEXT	4DSP	.HTM

### 4D View ドキュメント

以下のガイドラインは重要なので、記憶しておいてください。

- MacOS上では、4D Viewはタイプとクリエータを使用してドキュメントを識別します（例えば、タイプが4DPVでクリエータが4DSPであれば、4D Viewドキュメント）。パス名を記述するには、ディスク名と“:”記号を使用してフォルダを区切ります（例：“MyDisk:Folder1:Folder2:MyBase”）。
- Windows上では、4D Viewは拡張子を使用してドキュメントを識別します（例えば、拡張子が .4PVであれば、4D Viewドキュメント）。パス名を記述するには、ディスクを表わす文字と“¥”記号を使用して（例“D:¥Directory1¥Directory2 ¥Mybase”）ディレクトリを区切ります。

- MacOS上で作成した後、Windowsにコピーした4D Viewドキュメントは、拡張子付きで保存されていれば直接開くことができます。たとえば、ドキュメント“MyDoc”が“MyDoc.4PV”という名前で保存され、PCにコピーされた場合、変更せずにドキュメントを開くことができます。
- Windows上で作成した後、MacOSにコピーした4D Viewドキュメントは、何も変更せずに開くことができます。



## PV Current cell、はじめに

このテーマ内のコマンドは、セルの“位置決め”を行い、更に特定の4D Viewエリアにあるカレントセルを変更します。また、4D Viewエリアのカレントセルの“編集”モードを終了することもできます。

## PV GET CURRENT CELL

---

### PV GET CURRENT CELL (エリア; カラム; 行)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	← アクティブセルのカラム番号
行	倍長整数	← アクティブセルの行番号

### 説明

**PV GET CURRENT CELL** コマンドは、<エリア>のカレントセルの座標を引数<カラム>と<行>に返します。

### 例題

ボタンのオブジェクトコールバックから開始して、カレントセルに文字列“Here”を表示します。

```
C_LONGINT($Column;$Row)    `座標を取得する
⇒ PV GET CURRENT CELL (Area;$Column;$Row)`セルの座標
If ($Column#0) & ($Row#0) `選択されたセルが存在する
PV SET CELL STRING VALUE (Area;$Column;$Row;"Here")
    `このセルには現在“Here”が格納される
End if
```

### 参照

PV GOTO NEXT CELL、PV VALIDATE CURRENT CELL

## PV VALIDATE CURRENT CELL

---

### PV VALIDATE CURRENT CELL (エリア)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア

#### 説明

**PV VALIDATE CURRENT CELL** コマンドは、入力モード中にカレントセルの内容を有効にします。カレントセルはそのまま変わりません。

このコマンドは、コールバックメソッドからのみ呼び出すことができます。コールバックメソッドに関する詳細は、「PV Area、はじめに」の節を参照してください。

#### 例題

まずはじめにコールバックメソッド、EventMethodのインストールを行います。このメソッドはダブルクリック時に呼び出されます。

#### **PV ON EVENT** (Area;pv on double clicked;"EventMethod")

このメソッドは、ユーザのダブルクリックを検知します。

```

`EventMethod メソッド
C_LONGINT($1) `4D View エリア参照
C_LONGINT($2) `イベント
C_LONGINT($3) `モディファイアキーコード
C_LONGINT($4) `カラム番号
C_LONGINT($5) `行番号
C_LONGINT($6) `キーのAscii コード
C_BOOLEAN($0) `戻り値
$0:=False
If($2=pv on double clicked) `他のイベントでも “EventMethod” が呼び出された場合
    BEEP
⇒ PV VALIDATE CURRENT CELL($1) `セル内容が有効になる
End if

```

#### 参照

PV ON EVENT

## PV GET NEXT FREE CELL

---

### PV GET NEXT FREE CELL (エリア; 方向; カラム; 行)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
方向	整数	→ 方向定数
カラム	倍長整数	← カラム番号
行	倍長整数	← 行番号

#### 説明

**PV GET NEXT FREE CELL** コマンドは、指定された<方向>にある次のフリーなセルの座標、<カラム>と<行>を取得します。

<方向>軸は、**PV Directions** 定数テーマの4つの値から1つを指定します（右方向、下方向、左方向、上方向）。

#### 例題

4D View エリア内に、入力されたデータの配列（すべて入力済み）があるものとします。この配列にある行数とカラム数を数えてみましょう。

入力されている最初のセルは、カラムCの4行目であることがわかっています。これからプラグインを使い、入力範囲にあるカラム数と行数を割り出します。

```
C_LONGINT($StartCol;$StartRow) ` 当初のセルの座標
C_LONGINT($RightCol;$RightRow) ` 一番右側の座標
C_LONGINT($LowCol;$LowRow) ` 一番下側の座標
$StartCol:=3 ` 初期化
$StartRow:=4
PV GOTO CELL (Area;$StartCol;$StartRow) ` 位置決めを行う
    ` 一番右側の座標を取得する
⇒ PV GET NEXT FREE CELL (Area;pv to the right;$RightCol;$RightRow)
    ` 一番下側の座標を取得する
⇒ PV GET NEXT FREE CELL (Area;pv to the bottom;$LowCol;$LowRow)
ALERT("入力されたデータは"+String($RightCol$StartCol+1)+"カラム"
    +String($LowCol$StartRow+1)+"行です。")
```

#### 参照

PV GOTO CELL、PV GOTO NEXT CELL

#### 定数

PV Directions テーマ

## PV GOTO CELL

---

### PV GOTO CELL (エリア; カラム; 行)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号

### 説明

**PV GOTO CELL** コマンドが呼び出されると、<カラム>と<行>で指定されたセルがその<エリア>のカレントセルになります。前のカレントセルが入力モードであった場合、その内容は有効になります。

### 例題

このコードでは、8カラム目の5行目にあるセルをカレントセルにします。

⇒ **PV GOTO CELL** (Area;8;5) `新しいカレントセル：H5

### 参照

PV GOTO NEXT CELL

## PV GOTO NEXT CELL

---

### PV GOTO NEXT CELL (エリア; 方向)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
方向	整数	→ 方向定数

### 説明

**PV GOTO NEXT CELL** コマンドが呼び出されると、指定された引数<方向>にある次のセルがその<エリア>のカレントセルになります。前のカレントセルが入力モードであった場合、その内容は有効になります。

<方向>軸は **PV Directions** 定数テーマの4つの値から1つを指定します（右方向、下方向、左方向、上方向）。

### 例題

スプレッドシートにおいて、請求書の“合計金額”に対応するセルを太字にする必要がある場合について考えてみましょう。

```
C_LONGINT(Column;Row) `カレントセルの座標
PV FIND ALL (Area;"合計金額";1;0) `“合計金額”という文字を含むセルを探す
⇒ PV GOTO NEXT CELL (Area; pv to the right) `値を含むセル
PV GET CURRENT CELL (Area;$Column;$Row) `座標を取得する
`選択されたセルを太字に設定する
PV SET CELL PROPERTY (Area;$Column;$Row;pv style text bold;pv value on)
```

### 参照

PV GET CURRENT CELL、PV GOTO CELL

### 定数

PV Directions テーマ

## PV GET PREVIOUS ACTIVE CELL

### PV GET PREVIOUS ACTIVE CELL (エリア; カラム; 行)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	← セルのカラム番号
行	倍長整数	← セルの行番号

#### 説明

**PV GET PREVIOUS ACTIVE CELL** コマンドは、指定された<エリア>の前回アクティブであった（カレント）セルの座標を、引数<カラム>と<行>に返します。

注：カレントセルのための“スタック”は存在しません。最初にカレントセルとなったセルではなく、前回のセルがわかるだけです。複数レベルの取り消し処理を実現したい場合など、必要に応じて一連のカレントセルの記録を管理するかどうかは、ユーザ次第です。

#### 例題

下記の例題では、特定のセルに対して“リバウンド”効果を作成します。イベント管理メソッドを設定して、アクティブセルが変わるたびに呼び出します。

#### **PV ON EVENT**(Area;pv on active cell changed;"EventMethod")

このプロジェクトメソッドEventMethodは、ユーザをC5セルから“追い出し”ます。このセルに到達すると、最後にカレントセルとなったセルが再びアクティブになり、どのような方法でも（4D View コマンドkey stroke、mouse等）、C5セルは選択できなくなります。

メソッド：EventMethod

このメソッドを使用して、C5セルから“追い出し”ます

**C\_LONGINT**(\$1) `4D View エリア参照

**C\_LONGINT**(\$2) `イベント

**C\_LONGINT**(\$3) `モディファイアキーコード

**C\_LONGINT**(\$4) `カラム番号

**C\_LONGINT**(\$5) `行番号

**C\_LONGINT**(\$6) `キーのAsciiコード

**C\_LONGINT**(\$Column;\$Row) `セルの座標（前回カレントであった）

**C\_BOOLEAN**(\$0) `戻り値

\$0:=False

**PV GET CURRENT CELL**(\$1;\$Column;\$Row) `座標を取得する

If(\$Column=3) & (\$Row=5) `セルC5がカレントか？

⇒

**PV GET PREVIOUS ACTIVE CELL**(\$1;\$Column;\$Row) `前回のセル

***PV GOTO CELL*** (\$1;\$Column;\$Row) `再度カレントセルになる  
**End if**

参照

PV GOTO CELL、PV ON ERROR

## PV Area、はじめに

---

このテーマのルーチンを使用し、フォーム上に表示されるオフスクリーンエリアとプラグインエリアを管理することができます。これらルーチンにより、オフスクリーンエリアの作成や消去を行ったり、フィールドやBLOB変数の内容をプラグインエリアやオフスクリーンエリアへペースト、あるいは4D ViewエリアをフィールドやBLOB変数に保存することができます。

更に、このテーマには、4D Viewエリアにより検出された各種タイプのイベントをプログラマが捉えることができるコマンドや、独自のエラーマネージャーを構築するコマンド、およびメニューやパレットを用いて利用できる4D Viewコマンドを管理するコマンドがまとめられています。

## コールバックメソッド

このテーマ内では、数々のコマンドにおいて“コールバック”の概念に触れています。ここでは、この仕組みはイベントやエラー、または4D Viewコマンドにメソッドをリンクするために使用されています。4D Viewは、イベントやエラーまたはメニューコマンドがアクティブになったことを検知すると、その都度エリア設定で定義された4Dのプロジェクトメソッドを実行します。この一連の仕組みの上で、このメソッドは“コールバックメソッド”と呼ばれます。

PV Areaテーマのコマンドのうち、コールバックメソッドについて触れているものは以下の通りです。

- PV ON EVENT
- PV ON COMMAND
- PV ON ERROR
- PV Get on event method
- PV Get on command method
- PV Get on error method

## PV New offscreen area

---

### PV New offscreen area → 倍長整数

引数	タイプ	説明
		このコマンドには引数はありません。
戻り値	倍長整数	← 4D View エリア

### 説明

**PV New offscreen area** コマンドはメモリ内に4D View エリアを作成し、このエリアの参照番号を返します。エリアの参照を必要とする4D View コマンドに、この参照番号を渡します。

エリアが不要になった場合、そのエリアが使用していたメモリスペースを解放するために、**PV DELETE OFFSCREEN AREA** コマンドでエリアを消去するのを忘れないでください。

### 例題

このメソッドは、スクリーンエリアへコピーできるようにテンプレートの内容をコピーします。

```
C_LONGINT($OffscreenArea) ` オフスクリーンエリアの参照番号
QUERY ([Model];[Model]Ref="MyModel") ` 目的のテンプレートを探す
⇒ $OffscreenArea:=PV New offscreen area ` オフスクリーンエリアを作成する
   ` テンプレートを取得する
PV BLOB TO AREA ($OffscreenArea;Table(->[Model]);Field(->[Models]BlobField_))
PV SELECT ALL ($OffscreenArea;pv set as selection) ` すべてを選択
PV EXECUTE COMMAND ($OffscreenArea;pv cmd edit copy) ` 選択内容をコピー
⇒ PV DELETE OFFSCREEN AREA ($OffscreenArea) ` メモリーを解放
   PV EXECUTE COMMAND (Area;pv cmd edit paste)
   ` 選択内容をアクティブなエリアにペースト
```

### 参照

PV BLOB TO AREA、PV DELETE OFFSCREEN AREA

## PV DELETE OFFSCREEN AREA

---

### PV DELETE OFFSCREEN AREA (エリア)

引数	タイプ	説明
エリア	倍長整数	→4D View エリア

#### 説明

**PV DELETE OFFSCREEN AREA** コマンドは、**PV New offscreen area** コマンドで作成された4D View エリアを消去します。

消去できるエリアは、オフスクリーンエリア、つまりフォーム以外の場所に作成されたエリアだけです。**PV New offscreen area** コマンドを使用して作成したエリアは必ず消去して、システムメモリが一杯になることを避けなくてはなりません。オフスクリーンエリアの消去を忘れると、4Dを終了する際に4D Viewから警告メッセージが表示されます。

#### 例題

**PV New offscreen area** コマンドの例題を参照してください。

#### 参照

PV New offscreen area

## PV SET AREA PROPERTY

---

### PV SET AREA PROPERTY (エリア; プロパティ; 値)

引数	タイプ	説明
エリア	倍長整数	→4D View エリア
プロパティ	倍長整数	→プロパティ番号
値	倍長整数	→プロパティの値

#### 説明

**PV SET AREA PROPERTY** コマンドは、指定された4D Viewの<エリア>に対し、<プロパティ>の<値>を設定します。

フィールドに関連付けられているかどうかに関係なく、画面に表示されたエリアに関するプロパティの初期設定（フォーム上に存在する）だけが保存されます。

- エリアを含むフォームが開かれるたびに、ロードされます。
- エリアがクローズされるたびに、保存されます。
- クライアントまたはサーバ上に配置することができます。

<エリア>に0を指定した場合、**PV SET AREA PROPERTY** コマンドは、新しく作成されたすべての4DViewエリアに対して適用されます。この場合、データベースの起動時に実行される「On Startup」データベースメソッドを使用して、このコマンドを呼び出すとよいでしょう。

引数<プロパティ>の定義には、**PV Area properties** 定数を使用します。引数<値>を定義するには、適切な定数テーマを使用するか、特定の値を渡します。各**PV Area properties** 定数および対応する<値>引数を解説したリストを以下に示します。

#### pv select mode

エリアに対して許可する選択動作を設定します。関連する値：**PV Select mode** テーマ内の定数。

- **pv select not allowed**：エリアでの選択は不可（すべてのセルは選択されない）。データ入力も不可（フォーミュラエディタはロックされる）。データ表示のみ可能。
- **pv select single row**：エリアでは一度に1つの行だけを選択できる。
- **pv select adjacent rows**：エリアでは隣接する行だけを選択できる。
- **pv select multiple rows**：エリアでは隣接していなくても複数の行を選択できる。
- **pv select single column**：エリアでは一度に1つのカラムだけを選択できる。
- **pv select adjacent columns**：エリアでは隣接するカラムだけを選択できる。

- **pv select multiple columns** : エリアでは隣接していなくても複数のカラムを選択できる。
- **pv select single cell** : エリアでは一度に1つのセルだけを選択できる。
- **pv select adjacent cells** : エリアでは隣接するセルだけを選択できる。
- **pv select multiple cells** : エリアでは隣接していなくても複数のセルを選択できる。

注：選択範囲内でのデータ入力は可能なままです（pv select not allowed 定数の使用時を除く）。エリアへのデータ入力をすべて禁止したい場合には、さらに、PV SET AREA PROPERTY(area;pv input trigger;pv trigger none)という命令を実行しなければなりません。

#### pv select highlight

エリア内のセル選択範囲を高輝度に設定します。関連する値：**pv value on** または **pv value off**。

- **pv value on** : エリアの選択範囲を高輝度表示する。
- **pv value off** : 選択範囲は高輝度にせず、結果として画面上では非表示となる。

#### pv select null

エリアに対して、現行の選択範囲を持つかどうかを設定します。関連する値：**pv value on** または **pv value off**。

- **pv value on** : エリアの選択は必須ではない。例えば、アクティブなカレントセルを含むカラムや行が削除されると、エリアには選択範囲がなくなる。
- **pv value off** : エリアの選択は必須。

#### pv current cell highlight

エリアのカレントセルを高輝度に設定します。関連する値：**pv value on** または **pv value off**。

- **pv value on** : エリアのカレントセルを高輝度表示する。従って、画面上に表示される。
- **pv value off** : エリアのカレントセルは高輝度にしない。従って、画面上では非表示になる。

デフォルトでは、アクティブセルは高輝度表示される。

## pv resizable columns

カラムサイズの変更の可、不可を設定します。関連する値：**pv value on**または**pv value off**。

- **pv value on**：エリアのカラムサイズを変更できる。
- **pv value off**：エリアのカラムサイズは変更できない。

## pv resizable rows

行サイズの変更の可、不可を設定します。関連する値：**pv value on**または**pv value off**。

- **pv value on**：エリアの行サイズを変更できる。
- **pv value off**：エリアの行サイズは変更できない。

## pv input trigger

エリアの入力トリガを設定します。データ入力にはアクティブなカレントセルでのみ行えます。関連する値：**PV Triggers** テーマ内の定数。

- **pv trigger none**：データ入力モードでキーが許可されている場合でも（**pv input enter key mode** 定数を参照）データ入力にはアクティブではない（イベントによって入力は開始されない）。しかし、フォーミュラエディタのツールバーを使用すればデータ入力は可能であり、選択範囲も変更できる。
- **pv trigger input key**：データ入力は任意のキー入力により開始する。この場合、セル間の移動はキーボードによってのみ可能（水平方向への移動は、Tab キーおよび Shift+Tab キー、垂直方向への移動は、return キーおよび Shift+return キー、または矢印キー）。
- **pv trigger input on enter**：データ入力は Enter キー（数字キーパッド）により開始する。
- **pv trigger input on gain sel**：データ入力はフォーカスのあるセルで開始する。このモードでは、セルは選択直後にフォーカスを取得し、カーソルは入力カーソルになる。
- **pv trigger on click**：データ入力はセルのクリックにより開始する。**pv trigger input on gain sel** 定数とは異なり、入力カーソルは表示されない。
- **pv trigger on double click**：データ入力はセルのダブルクリックにより開始する。シングルクリックでは、入力できない。
- **pv trigger on alt click**：データ入力はセルの Alt キー+クリックにより開始する。
- **pv trigger on alt double click**：データ入力はセルの Alt キー+ダブルクリックにより開始する。

- **pv trigger on ctrl click** : データ入力にはセルのCtrlキー+クリック (MacOSではCommandキー+クリック) により開始する。
- **pv trigger on ctrl double click** : データ入力にはセルのCtrlキー+ダブルクリック (MacOSではCommandキー+ダブルクリック) により開始する。
- **pv trigger on shift click** : データ入力にはセルのShiftキー+クリックにより開始する。
- **pv trigger on shift double clic** : データ入力にはセルのShiftキー+ダブルクリックにより開始する。

#### 注意:

- ・ 1つのトリガに対して、複数の定数を付加することができます。例えば、PV SET AREA PROPERTY (area; pv input trigger;pv trigger on click + pv trigger on alt clic) と指定すると、データ入力のためにクリック、またはAltキー+クリックを使用することができます。
- ・ 入力と選択範囲の両方に対して同じトリガが定義されている場合、入力トリガが優先されます。

#### pv select trigger

エリアの選択範囲のトリガを設定します。関連する値: **PV Triggers** テーマ内の定数。

- **pv trigger none** : エリアでは選択を行えない。コマンド実行前にカレントとなっていた選択範囲でのデータ入力は依然として可能 - Tabキーやreturnキーを押すと選択範囲内でアクティブセルを移動する。
- **pv trigger select on arrow** : 選択範囲の決定 (アクティブセルのみ) は矢印キーを使用して行う。選択範囲の拡張や縮小は不可。
- **pv trigger select on tab** : 選択範囲の決定 (アクティブセルのみ) はTabキー、またはShift+Tabキーを使用して行う。選択範囲の拡張や縮小は不可。
- **pv trigger select on return** : 選択範囲の決定は (アクティブセルのみ) returnキーを使用して行う。選択範囲の拡張や縮小は不可。
- **pv trigger on click** : 選択範囲の決定はマウスクリックによって行う。
- **pv trigger on double click** : 選択範囲の決定は (アクティブセルのみ) マウスのダブルクリックによって行う。選択範囲の拡張や縮小は不可。
- **pv trigger on alt click** : 選択範囲の決定はAltキー+クリックによって行う。
- **pv trigger on alt double click** : 選択範囲の決定はAltキー+ダブルクリックによって行う。
- **pv trigger on ctrl click** : 選択範囲の決定はCtrlキー+クリックによって行う (MacOSではCommandキー+クリック)。

- **pv trigger on ctrl double click** : 選択範囲の決定はCtrlキー+ダブルクリックによって行う (MacOSではCommandキー+ダブルクリック)。
- **pv trigger on shift click** : 選択範囲の決定はShiftキー+クリックによって行う。
- **pv trigger on shift double clic** : 選択範囲の決定はShiftキー+ダブルクリックによって行う。

注意:

- 1つのトリガに対して、複数の定数を追加することができます。例えば、PV SET AREA PROPERTY (area; pv select trigger;pv trigger on click + pv trigger on alt clic) と指定すると選択範囲に対してクリック、または、Altキー+クリックを使用することができます。
- 入力と選択範囲の両方に対して同じトリガが定義されている場合、入力トリガが優先されます。
- ドラッグと選択範囲の両方に対して、同じトリガが定義されている場合、ドラッグトリガが優先されます。

#### pv carriage return

セルに新しい行を作成します (複数行のセル)。関連する値: **PV Carriage return** テーマ内の定数。

- **pv cr not allowed** : エリアでは複数行に渡るデータ入力は許可されない。
- **pv cr allowed** : Returnキーを押すと、セルに新しい行が作成される。
- **pv cr allowed with ctrl** : Ctrlキー+Returnキーを押すと (MacOSではCommandキー+Returnキー)、セルに新しい行が作成される。
- **pv cr allowed with shift** : Shiftキー+Returnキーを押すと、セルに新しい行が作成される。

#### PV arrow keys

データ入力を確定するために、矢印キーの使用を定義します (入力内容の確定と次のセル選択)。セル内容の最初、または最後にカーソルがある場合にのみ、入力が確定されません。関連する値: **PV Arrow keys** テーマ内の定数。

- **pv arrow keys allowed** : すべての矢印キーの使用を許可する。
- **pv top and bottom arrow keys** : 上矢印キーおよび下矢印キーの使用のみ許可する。
- **pv right and left arrow keys** : 右矢印キーおよび左矢印キーの使用のみ許可する。
- **pv arrow keys not allowed** : データ入力の確定のために矢印キーの使用を許可しない。

### pv vert pane count

エリアの縦方向の区画数の読み込みを許可します。この定数は**PV Get area property** コマンドを使用した時にのみ読み込まれます。戻り値：区画数。

覚書：区画とは、2つの分割ボックスの間にあるエリアのことです（縦と横の分割ボックスがあります）。

### pv hor pane count

エリアの横方向の区画数の読み込みを行います。この定数は**PV Get area property** コマンドを使用した時にのみ読み込まれます。戻り値：区画数。

### pv drag trigger

エリアのドラッグトリガの定義を行います。ドロップに関して、特別なトリガはありません。関連する値：PV Triggers テーマ内の定数。

- **pv trigger none**：エリア内でドラッグは行えない。
- **pv trigger on click**：選択範囲のドラッグはマウスクリックを使用して行う。
- **pv trigger on double click**：選択範囲のドラッグはマウスのダブルクリックを使用して行う。
- **pv trigger on alt click**：選択範囲のドラッグはAlt キー+クリックを使用して行う。
- **pv trigger on alt double click**：選択範囲のドラッグはAlt キー+ダブルクリックを使用して行う。
- **pv trigger on ctrl click**：選択範囲のドラッグはCtrl キー+クリックを使用して行う（MacOSではCommand キー+クリック）。
- **pv trigger on ctrl double click**：選択範囲のドラッグはCtrl キー+ダブルクリックを使用して行う（MacOSではCommand キー+ダブルクリック）。
- **pv trigger on shift click**：選択範囲のドラッグはShift キー+クリックを使用して行う。
- **pv trigger on shift double clic**：選択範囲のドラッグはShift キー+ダブルクリックを使用して行う。

注：ドラッグと選択範囲の両方に対して、同じトリガが定義されている場合、ドラッグトリガが優先されます。

## pv drag allowed

ドラッグできる選択範囲のタイプを設定します。関連する値：**PV Drag drop allowed** テーマ内の定数。

- **pv DD not allowed**：ドラッグ&ドロップが可能な場合でも、選択範囲のドラッグは行えない。
- **pv DD single cell**：単一のセルからなる選択範囲をドラッグできる。
- **pv DD adjacent cells**：複数の隣接するセル、または単一のセルからなる選択範囲をドラッグできる。
- **pv DD multiple cells**：複数のセル（隣接しているかどうかに関係なく）、または単一のセルからなる選択範囲をドラッグできる。
- **pv DD single row**：単一の行からなる選択範囲をドラッグできる。
- **pv DD adjacent rows**：複数の隣接する行、または単一の行からなる選択範囲をドラッグできる。
- **pv DD multiple rows**：複数の行（隣接しているかどうかに関係なく）、または単一の行からなる選択範囲をドラッグできる。
- **pv DD single column**：単一のカラムからなる選択範囲をドラッグできる。
- **pv DD adjacent columns**：複数の隣接するカラム、または単一のカラムからなる選択範囲をドラッグできる。
- **pv DD multiple columns**：複数のカラム（隣接しているかどうかに関係なく）、または単一のカラムからなる選択範囲をドラッグできる。

注：1つのエリアに対して、複数の定数を追加することができます。例えば、PV SET AREA PROPERTY (area; pv drag allowed;pv DD multiple cells + pv DD single column + pv DD adjacent rows) と指定すると、複数のセルか単一のカラム、または連続する行のいずれかで構成される選択範囲をドラッグすることができます。

## pv drop mode

エリア内でドラッグした選択範囲のドロップ方法を設定します。このプロパティは、ドラッグした値がドロップ先のエリアにどのようにペーストされるかを定義するだけである点に注意してください。ドラッグした値（ある場合）のコピーは別に管理しなくてはなりません。関連する値：**PV Drop mode** テーマ内の定数。

- **pv drop insert or replace**：ドロップした値はエリアに挿入されるか、または既存の値を置き換える。
- **pv drop insert only**：ドロップした値はエリアに挿入される。
- **pv drop replace only**：ドロップした値はエリアの既存の値を置き換える。

### pv drop allowed

エリアにドロップできる選択範囲のタイプを設定します。関連する値：**PV Drag drop allowed** テーマ内の定数。

- **pv DD not allowed**：ドラッグ&ドロップが可能な場合でも、エリア内で選択範囲のドラッグは行えない。
- **pv DD single cell**：単一のセルからなる選択範囲をドロップできる。
- **pv DD adjacent cells**：複数の隣接するセル、または単一のセルからなる選択範囲をドロップできる。
- **pv DD multiple cells**：複数のセル（隣接しているかどうかに関係なく）、または単一のセルからなる選択範囲をドロップできる。
- **pv DD single row**：単一の行からなる選択範囲をドロップできる。
- **pv DD adjacent rows**：複数の隣接する行、または単一の行からなる選択範囲をドロップできる。
- **pv DD multiple rows**：複数の行（隣接しているかどうかに関係なく）、または単一の行からなる選択範囲をドロップできる。
- **pv DD single column**：単一のカラムからなる選択範囲をドロップできる。
- **pv DD adjacent columns**：複数の隣接するカラム、または単一のカラムからなる選択範囲をドロップできる。
- **pv DD multiple columns**：複数のカラム（隣接しているかどうかに関係なく）、または単一のカラムからなる選択範囲をドロップできる。

注：1つのエリアに対して、複数の定数を追加することができます。例えば、PV SET AREA PROPERTY (area; pv drop allowed;pv DD multiple cells + pv DD single column + pv DD adjacent rows) と指定すると、複数のセルか単一のカラム、または連続する行のいずれかで構成される選択範囲をドラッグすることができます。

### pv input enter key mode

Enterキー（数値キーボード）がデータ入力中に押された場合の動作を設定します。関連する値：**PV Input enter key mode** テーマ内の定数。

- **pv enter key standard**：Enterキーによりカレントセルが有効になり、同一セル内で選択／データ入力のいずれかに切り替わる（カレントセルはそのまま）。
- **pv enter key as tab**：Enterキーによりカレントセルが有効になり、右側にある次のセル内で選択／データ入力のいずれかに切り替わる。Shiftキー+Enterキーの場合には、左側にある次のセル内で選択／データ入力のいずれかに切り替わる。

■ **pv enter key as return** : Enter キーによりカレントセルが有効になり、下側にある次のセル内で選択／データ入力のいずれかに切り替わる。Shift キー +Enter キーの場合には、上側にある次のセル内で選択／データ入力のいずれかに切り替わる。

注 : Enter キーとは異なり、Tab キーおよび Return キーはセルを選択するだけです。

#### pv record tag

レコード区切りを設定します。このプロパティはデータの読み込み／書き出し時にのみ有効です。関連する値 : ASCII 文字コード。

例 : "E1Field1", "E1Field2", "E1Field3"; "E2Field4", "E2Field5";

セミコロンがレコード区切りです (2レコード : E1 および E2)。

#### pv field tag

フィールド区切りを設定します。このプロパティはデータの読み込み／書き出し時にのみ有効です。関連する値 : ASCII 文字コード。

例 : "E1Field1", "E1Field2", "E1Field3"; "E2Field4", "E2Field5";

コンマがフィールド区切りです。

#### pv field wrapper

フィールド囲いを設定します。このプロパティはデータの読み込み／書き出し時にのみ有効です。関連する値 : ASCII 文字コード。

例 : "E1Field1", "E1Field2", "E1Field3"; "E2Field4", "E2Field5";

引用符がフィールド囲いです。

#### pv copy hidden

セルのコピー時に、エリア内に含まれる非表示項目を考慮するかどうかを設定します。関連する値 : **pv value on** または **pv value off**。

■ **pv value on** : セルのコピー時に、非表示項目 (ある場合) を考慮する。

■ **pv value off** : セルのコピー時に、非表示項目 (ある場合) を考慮しない。

### pv column headers height

エリアのカラムヘッダの高さの設定や読み込みを行います。関連する値：ヘッダ高さ (ピクセル単位)。

### pv row headers width

エリアの行ヘッダの幅の設定や読み込みを行います。関連する値：ヘッダ幅 (ピクセル単位)。

### pv show column headers

エリアのカラムヘッダの表示、非表示を設定します。関連する値：**pv value on** または **pv value off**。

■ **pv value on** : カラムヘッダを表示する。

■ **pv value off** : カラムヘッダを表示しない。

### pv show row headers

エリアの行ヘッダの表示、非表示を設定します。関連する値：**pv value on** または **pv value off**。

■ **pv value on** : 行ヘッダを表示する。

■ **pv value off** : 行ヘッダを表示しない。

### pv show menu bar

エリアの4D View メニューバーの表示、非表示を設定します。関連する値：**pv value on** または **pv value off**。

■ **pv value on** : メニューバーを表示する。

■ **pv value off** : メニューバーを表示しない。

### pv show standard toolbar

エリアの4D View 標準ツールバーの表示、非表示を設定します。関連する値：**pv value on** または **pv value off**。

■ **pv value on** : 標準ツールバーを表示する。

■ **pv value off** : 標準ツールバーを表示しない。

#### pv show numbers toolbar

エリアの4D View 数字ツールバーの表示、非表示を設定します。関連する値：**pv value on**または**pv value off**。

■ **pv value on**：数字ツールバーを表示する。

■ **pv value off**：数字ツールバーを表示しない。

#### pv show style toolbar

エリアの4D View スタイルツールバーの表示、非表示を設定します。関連する値：**pv value on**または**pv value off**。

■ **pv value on**：スタイルツールバーを表示する。

■ **pv value off**：スタイルツールバーを表示しない。

#### pv show borders toolbar

エリアの4D View 枠線ツールバーの表示、非表示を設定します。関連する値：**pv value on**または**pv value off**。

■ **pv value on**：枠線ツールバーを表示する。

■ **pv value off**：枠線ツールバーを表示しない。

#### pv show formula toolbar

エリアの4D View フォーミュラツールバーの表示、非表示を設定します。関連する値：**pv value on**または**pv value off**。

■ **pv value on**：フォーミュラツールバーを表示する。

■ **pv value off**：フォーミュラツールバーを表示しない。

#### pv show hor grid

エリアの4D View 水平グリッドの表示、非表示を設定します。関連する値：**pv value on**または**pv value off**。

■ **pv value on**：水平グリッドを表示する。

■ **pv value off**：水平グリッドを表示しない。

### pv show vert grid

エリアの4D View 垂直グリッドの表示、非表示を設定します。関連する値：**pv value on** または **pv value off**。

- **pv value on** : 垂直グリッドを表示する。
- **pv value off** : 垂直グリッドを表示しない。

### pv show hor scrollbar

エリアの4D View 横スクロールバーの表示、非表示を設定します。関連する値：**pv value on** または **pv value off**。

- **pv value on** : 横スクロールバーを表示する。
- **pv value off** : 横スクロールバーを表示しない。

### pv show vert scrollbar

エリアの4D View 縦スクロールバーの表示、非表示を設定します。関連する値：**pv value on** または **pv value off**。

- **pv value on** : 縦スクロールバーを表示する。
- **pv value off** : 縦スクロールバーを表示しない。

### pv zoom factor

エリアのズーム値（パーセント単位）の設定、または読み込みを行います。関連する値：ズーム率25%～1000%。

### pv saving dialog

変更が加えられた4D View ドキュメントのクローズ時に、「ドキュメントを保存」確認メッセージを表示する（または、表示しない）かどうかを設定します。データベースのフィールドに関連付けられていない4D Viewの組み込みエリアを終了する（フォームを有効にするかキャンセルする）際に、この警告メッセージが表示されます。4D Viewのプラグインウィンドウに対して、このプロパティは無効です。関連する値：**pv value on** または **pv value off**。

- **pv value on** : 確認メッセージが表示される（デフォルト値）。
- **pv value off** : 確認メッセージは表示されない。

pv allow undo redo

取り消し機能を使用するかどうかを設定します。関連する値：**pv value on**または**pv value off**。

■ **pv value on**：取り消し機能を使用する（デフォルト値）。

■ **pv value off**：取り消し機能を使用しない（「編集」メニューの「取り消し」コマンドは非アクティブ）。

### 例題

4D View エリアのカラムサイズを固定したり、またはサイズ変更機能が停止している場合にサイズ変更を許可するには、“スイッチ”の役割を果たす以下のようなメソッドを作成します。

**C\_INTEGER(\$Value)** `プロパティの値

`現在の値 (0: 停止, 1 = 許可)

**\$Value:=PV Get area property** (Area;pv resizable columns)

`切り替えコマンド：0 ↔ 1

⇒ **PV SET AREA PROPERTY** (Area;pv resizable columns;**Num(\$Value=0)**)

### 参照

PV Get area propertys

### 定数

PV Area properties、PV Select mode、PV Triggers、PV Carriage return、PV Drag drop allowed、PV Input enter key mode テーマ内の定数

## PV Get area property

---

### PV Get area property (エリア; プロパティ) →倍長整数

引数	タイプ	説明
エリア	倍長整数	→4D View エリア
プロパティ	倍長整数	→プロパティ番号
戻り値	倍長整数	←オプションの値

#### 説明

**PV Get area property**関数は、指定されたオプションに対する4D View<エリア>の<プロパティ>の値を返します。

<プロパティ>の定義には、**PV Area properties**定数を使用します。

プロパティとその値に関する詳細は、**PV SET AREA PROPERTY**コマンドの説明を参照してください。

#### 例題

**PV SET AREA PROPERTY**コマンドの例題を参照してください。

#### 参照

PV SET AREA PROPERTY

定数

PV Area properties テーマ

## PV BLOB TO AREA

---

### PV BLOB TO AREA (エリア; blob)

引数	タイプ	説明
エリア	倍長整数	→4D View エリア
blob	BLOB	→元となる BLOB

### 説明

**PV BLOB TO AREA** コマンドは、< blob >に保存された4D Viewのスプレッドシートを<エリア>内に開きます。

引数<エリア>には、画面上またはオフスクリーン上にある4D Viewエリアを指定します。

### 例題

1番目のメソッド（例えば、「コピー」ボタンのオブジェクトメソッド）は、後で使用する（例えば、エリアを消去した後や別のエリアで）ためにエリア内容をコピーします。

**C\_BLOB**(BlobVariable) `エリアを受け取るプロセス変数  
BlobVariable:=**PV Area to Blob** (Area) `変数に保存する

この2番目のメソッド（例えば、「ペースト」ボタンのオブジェクトメソッド）は、変数に格納されたエリアを貼り付け、画面上にエリア情報をセットします。

⇒ **PV BLOB TO AREA** (Area;BlobVariable) `変数より取得

### 参照

PV Area to blob

## PV Area to blob

---

### PV Area to blob (エリア) → BLOB

引数	タイプ	説明
エリア	倍長整数	→4D View エリア
戻り値	BLOB	←保存先のBLOB

#### 説明

**PV Area to blob**関数は、<エリア>を保存してフィールドまたはBLOB変数に格納します。このBLOB変数は後でフィールドに保存したり、同じタイプの変数を使用して処理することができます。

引数<エリア>には、画面上またはオフスクリーン上にある4D Viewエリアを指定します。

#### 例題

**PV BLOB TO AREA** コマンドの例題を参照してください。

#### 参照

PV BLOB TO AREA

## PV ON EVENT

---

### PV ON EVENT (エリア; イベント; メソッド)

引数	タイプ	説明
エリア	倍長整数	→4D View エリア
イベント	倍長整数	→4D View イベント
メソッド	文字列	→メソッド名

### 説明

**PV ON EVENT** コマンドは、<メソッド>を4D Viewの<イベント>に関連付けるために使用します。<イベント>が発生するたびに、<メソッド>が実行されます。

引数<イベント>の定義には、**PV Event constants** 定数を使用します。

呼び出されたメソッドは6つの倍長整数タイプの引数を受け取り、\$0にブール値を返します。

\$1：4D View エリアの参照番号

\$2：イベント

\$3：モディファイアキーのコード

\$4：カラム番号 (\*)

\$5：行番号 (\*\*)

\$6：キーの Ascii コード（イベントがクリック、右クリック、またはダブルクリックである場合、\$6には0が代入されます）

(\*) イベントが行ヘッダ上でのクリック、右クリック、またはダブルクリックである場合、\$4には0が代入されます。

(\*\*) イベントがカラムヘッダ上でのクリック、右クリック、またはダブルクリックである場合、\$5には0が代入されます。

\$3には以下の値のうちいずれか一つを指定することができます（複数のキーが押された場合には、これらの値を追加します）。

0	なし
512	Shift キー
2048	Alt キー
4096	Ctrl キー（Windows）／Command キー（MacOS）

\$0が“True（真）”の場合、このイベントは考慮されません。

\$0が“False（偽）”の場合、このイベントが考慮されます。

注：データベースをコンパイルする予定であれば、いずれかの引数を使用しない場合でも、\$0をブール、\$1～\$6を倍長整数として宣言しなくてはなりません。

<エリア>に0を渡すと、**PV ON EVENT** コマンドは新たに作成されたすべての4D View エリアに対して適用されます。この場合、データベースのオープン時に実行される「On Startup」データベースメソッドで、このコマンドを使用することをお勧めします。

イベントメソッドを削除するには、最後の引数に空の文字列を渡した上で**PV ON EVENT** コマンドを呼び出します。

### 例題

**PV VALIDATE CURRENT CELL**、**PV GET PREVIOUS ACTIVE CELL**、**PV GET CELL FIELD**、**PV Get on event method**、および**PV SAVE DOCUMENT**の例題を参照してください。

### 参照

PV Get on event method

### 定数

PV Events テーマ

## PV Get on event method

---

### PV Get on event method (エリア; イベント) → 文字列

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
イベント	倍長整数	→ 4D View イベント
戻り値	文字列	← 4D メソッド名

#### 説明

**PV Get on event method** 関数は、指定された<イベント>のコールバックに関連付けられたメソッド名を返します。

<イベント>の定義には、**PV Event** 定数を使用します。

<イベント>に関連付けられたメソッドが存在しなければ、**PV Get on event method** 関数は空の文字列を返します。

#### 例題

イベントコールを一時的に無効にしておき、本来の呼び出しを回復する前に、ある処理を実行するとよい場合があります。

例として、**pv on cell value changed** イベントに対して **PV Get on event method** 関数を使用し、汎用的な方法でこの“解放”を行うかんたんなメソッドを示します。

```
C_STRING(32;$EventMethod)
  `適切と思われるメソッドを保存する
⇒ $EventMethod:=PV Get on event method (Area;pv on cell value changed)
PV ON EVENT (Area;pv on cell value changed ;"`) `キャンセル
  `... ここで処理を実行する
PV ON EVENT (Area;pv on cell value changed;$EventMethod) `回復
```

#### 参照

PV Get on command method、PV Get on error method、PV ON EVENT

#### 定数

PV Event テーマ

## PV ON COMMAND

---

### PV ON COMMAND (エリア; コマンド; メソッド)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
コマンド	倍長整数	→ コマンド番号
メソッド	文字列	→ 4D メソッド名

### 説明

**PV ON COMMAND** コマンドは、4D View メニューの<コマンド>を4Dの<メソッド>に関連付けます。

引数<コマンド>の定義には、**PV Commands** 定数を使用します。

<メソッド>は、3つの引数を受け取ります。

\$1 : 4D View エリアの参照番号

\$2 : メニューコマンド番号

\$3 : モディファイアキー

**on command** メソッド呼び出しを削除するには、3番目の引数に空の文字列を渡した上で**PV ON COMMAND** コマンドを呼び出します。

### 例題

例えば、データベースにおいて、組み込まれた4D View エリアを呼び出すプリントジョブをすべてトレースする場合について考えてみましょう。この対策案として、いくつかのコードを以下に示します。

```

If (PV Get on command method (area;(pv cmd print document)
                                     #"PrintMethod")
    `レコード印刷フォーミュラのトレース
⇒ PV ON COMMAND (area;(pv cmd print document);"PrintMethod")
End if
If (PV Get on command method (area;(pv cmd print formulas)#"PrintMethod")
    `標準のレコード印刷のトレース
⇒ PV ON COMMAND (area;(pv cmd print formulas);"PrintMethod")
End if

```

**PrintMethod** メソッドのコードは以下の通りです。

```

`メソッド: PrintMethod
C_LONGINT($1)`4D View エリアの参照番号

```

**C\_LONGINT(\$2)**`メニューコマンド番号  
**C\_LONGINT(\$3)**`モディファイアキーコード  
**Case of**  
\**(\$2=pv cmd print formulas)**

## PV Get on command method

---

**PV Get on command method** (エリア; コマンド) → 文字列

引数	タイプ	説明
エリア	倍長整数	→4D View エリア
コマンド	倍長整数	→コマンド番号
戻り値	文字列	←4D メソッド名

### 説明

**PV Get on command method**関数は、4D View メニューの<コマンド>に関連付けられたメソッドの名前を返します。

引数<コマンド>の定義には、**PV Commands** 定数を使用します。

関連付けられたメソッドが存在しない場合、**PV Get on command method**関数は空の文字列を返します。

### 例題

**PV ON COMMAND** コマンドの例題を参照してください。

### 参照

PV Get on error method、PV Get on event method、PV ON COMMAND

### 定数

PV Commands テーマ

## PV ON ERROR

---

### PV ON ERROR (メソッド)

引数	タイプ	説明
メソッド	文字列	→4D View メソッド

#### 説明

**PV ON ERROR** コマンドは、4D Viewのエラーを管理する<メソッド>を設定します。

この割り込み用のメソッドは、4D View コマンドのコール中にエラーが発生するたびに実行されます。従って、実行エラーの制御が可能になります。

呼び出されるメソッドは3つの引数を受け取ります。

\$1：4D View エリアの参照番号

\$2：エラー番号

\$3：エラーテキスト

4D View によって生成されるエラーの番号および種類は、付録A「4D View エラーコード一覧」に記載されています。

<メソッド>を削除するには、引数に空の文字列を渡した上で**PV ON ERROR** コマンドを呼び出します。

#### 例題

アクティブな4D View エリアに対して、エラー管理用メソッドを設定します。

```
If (PV Get on error method #"ErrMethMan") `管理メソッドが設定されているか？
⇒      PV ON ERROR ("ErrMethMan") `メソッドのコール
End if
```

“ErrMethMan” のコードは以下の通りです。

```
C_LONGINT ($1) `4D View エリア参照番号
C_LONGINT ($2) `エラー番号
C_TEXT ($3) `エラーテキスト
ALERT ("内部エラー番号 "+String (ErrorNum)+Char (13)+ErrorText)
```

#### 参照

付録A「4D View エラーコード一覧」、PV GET LAST ERROR、PV Get on error method

## PV Get on error method

---

### PV Get on error method → 文字列

引数	タイプ	説明
		このコマンドには引数はありません。
戻り値	文字列	← 4Dメソッド名

### 説明

**PV Get on error method**関数は、**PV ON ERROR** コマンドで設定された現在のエラー管理メソッドの名前を返します。

設定されているエラーメソッドが存在しない場合、**PV Get on error method**関数は空の文字列を返します。

### 例題

**PV ON ERROR** コマンドの例題を参照してください。

### 参照

PV Get on command method、PV Get on event method、PV ON ERROR

## PV GET LAST ERROR

---

### PV GET LAST ERROR (エリア; エラーコード; エラーテキスト)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
エラーコード	倍長整数	← エラー番号
エラーテキスト	テキスト	← エラーの説明テキスト

### 説明

**PV GET LAST ERROR** コマンドは、指定された4D Viewの<エリア>で発生した最新のエラーに関する情報を取得します。エリア参照番号に0を指定すると、すべての4D View エリアから受け取った最新のエラーに対応する情報を取得します。

このコマンドの実行後、<エラーコード>にはエラー番号、<エラーテキスト>には対応するエラーの詳細説明が納められます。4D Viewによって生成されるエラー番号およびエラー名は、付録A「4D View エラーコード一覧」に記載されています。

**PV GET LAST ERROR** コマンドは、<エリア>に対する直近の4D View コマンド呼び出しでエラーが発生した場合にのみ、エラーを返します。つまり、コマンドの呼び出しでエラーが発生しない場合は、最新のエラーがゼロ (0) にリセットされます。発生する可能性のあるエラーを捕捉し、管理するには、このコマンドの代わりに **PV ON ERROR** コマンドを使用してください。

しかし、**PV ON ERROR** コマンドを使用しない場合、エラーが発生すると4D View はユーザに対して警告ダイアログボックスを表示します。従って、**PV GET LAST ERROR** コマンドを使用すると、例えば「4D デバugg」上などで必要な情報を取得することができます。

### 例題

4D View エリアにレコードセレクションの値をロードした後、処理がダウンせずに正常終了できるだけの十分なメモリが利用できるかどうかを調べます。メモリが十分でなければ、この問題の解決案をユーザに提示します。

```
C_LONGINT($ErrorCode) ` エラー番号
C_TEXT($ErrorText) ` エラーの説明テキスト
PV FIELD TO CELLS (Area;1;1;1;0;Table(->[Clients]);Table(->[Clients]);
Field(->[Clients]FirstName))
PV FIELD TO CELLS (Area;1;2;1;0;Table(->[Clients]);Table(->[Clients]);
Field(->[Clients]Name))
⇒ PV GET LAST ERROR (Area;$ErrorCode;$ErrorText) ` エラーが発生したか？
If ($ErrorCode=18) ` メモリ不足
    ALERT("メモリ不足：表示するセレクションを減らすか、"+
    "4D に割り当てるメモリを増やしてください")
End if
```

### 参照

付録A 「4D View エラーコード一覧」、PV ON ERROR

## PV EXECUTE COMMAND

---

### PV EXECUTE COMMAND (エリア; コマンド)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
コマンド	倍長整数	→ コマンド番号

### 説明

**PV EXECUTE COMMAND** コマンドは、<エリア>内において、引数<コマンド>に渡された番号の4D View メニューコマンドを実行します。

引数<コマンド>の定義には、**PV Commands** テーマ内の定数を使用します。

### 例題

縦スクロールバーの表示と非表示を切り替えるメソッドを以下に示します。対応する「表示」メニューも、“縦スクロールバー”の行にあわせてアクティブ/非アクティブになります。

**C\_INTEGER(\$status)** `1=使用可 0=使用不可

**C\_INTEGER(\$checkbox)** `0=チェックなし, 1=チェック

**C\_STRING(30;\$name)** `対応するコマンドの名前

`情報を取得

```
PV GET COMMAND STATUS (Area;pv cmd ViewVscrollbar;$status;  
                                                                    $checkbox;$name)
```

If (\$checkbox=1) `縦スクロールバーは表示されている？

```
⇒ PV EXECUTE COMMAND (Area;pv cmd ViewVscrollbar) `非表示にする  
PV SET COMMAND STATUS (Area;pv cmd ViewVscrollbar;0)  
    `使用不可にする
```

Else

```
⇒ PV SET COMMAND STATUS (Area;pv cmd ViewVscrollbar;1)  
    `コマンドをアクティブにする  
PV EXECUTE COMMAND (Area;pv cmd ViewVscrollbar)  
    `スクロールバーを表示
```

End if

End it

### 参照

PV GET COMMAND STATUS、PV SET COMMAND STATUS

### 定数

PV Commands テーマ

## PV SET COMMAND STATUS

### PV SET COMMAND STATUS (エリア; コマンド; ステータス)

引数	タイプ	説明
エリア	倍長整数	→4D View エリア
コマンド	倍長整数	→コマンド番号
ステータス	整数	→0 = 使用不可：1 = 使用可

#### 説明

**PV SET COMMAND STATUS** コマンドは、<コマンド>で指定されたメニューコマンドを使用可または使用不可に設定します。これらのコマンドはメニュー上（プログラムを介して4D Viewメニューを使用できます）、またはパレット上でローカライズすることができます。

- <ステータス>が0の場合、コマンドは呼び出されても実行されず、メニュー（またはボタン）はグレー表示となる。
- <ステータス>が1の場合、コマンドは呼び出されると実行され、メニュー（またはボタン）はアクティブになる。

使用不可になったメニューコマンドは、**PV EXECUTE COMMAND** コマンドを使用してプログラムから実行することはできません。

また、**PV SET COMMAND STATUS** コマンドで使用不可となったコマンドをユーザが使おうとした場合、**PV ON COMMAND** コマンドも利用できません。

#### 例題

(1)4D Viewの<エリア>に参照番号を表示しないようにするには、次のようなコードを作成します。

⇒ **PV GET COMMAND STATUS** (area;pv cmd Viewreferences;0)  
`3番目の引数を1にすると、再び使用可能になる

(2)**PV EXECUTE COMMAND** コマンドの例題を参照してください。

#### 参照

PV EXECUTE COMMAND、PV GET COMMAND STATUS

#### 定数

PV Commands テーマ

## PV GET COMMAND STATUS

---

**PV GET COMMAND STATUS** (エリア; コマンド; ステータス; チェックボックス; 名前)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
コマンド	倍長整数	→ コマンド番号
ステータス	整数	← 0 = 使用不可; 1 = 使用可
チェックボックス	整数	← 0 = チェックなし; 1 = チェック
名前	文字列	← コマンド名

### 説明

**PV GET COMMAND STATUS** コマンドは、<コマンド>で指定された4D View コマンドに関する<ステータス>、<チェックボックス>、<名前>を取得します。

引数<コマンド>は、情報を取得したいコマンドの番号に対応します。この引数の定義には、**PV Commands** テーマ内の定数を使用します。

引数<ステータス>にはコマンドの状態が返され、その値が0であればそのコマンドは使用不可であり、1の場合は使用可能であることを意味します。

引数<チェックボックス>は、そのコマンドがチェックされているか (値は1)、またはチェックされていないか (値は0) を示します。

引数<名前>には、コマンドラベルが納められます。

### 例題

**PV EXECUTE COMMAND** コマンドの例題を参照してください。

### 参照

PV EXECUTE COMMAND、PV SET COMMAND STATUS

### 定数

PV Commands テーマ

## PV REDRAW

---

### PV REDRAW (エリア)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア

### 説明

**PV REDRAW** コマンドは、4D View の<エリア>を強制的にリフレッシュするために使用します。

### 例題

**PV SET COLUMNS WIDTH** コマンドの例題を参照してください。このコマンドは、行とカラムのサイズ変更を行う例題で説明されています。



## PV Borders、はじめに

---

このテーマ内のルーチンを使用すると、単独のセルやセル選択範囲への枠線の設定や、枠線の属性の定義を行えます。また、メニューコマンドを使用して設定された引数に応じて、枠線タイプに関する情報を取得することもできます。

選択範囲およびセル領域に関する詳細は、「PV Selection、はじめに」の節を参照してください。

## PV SET RANGE BORDER

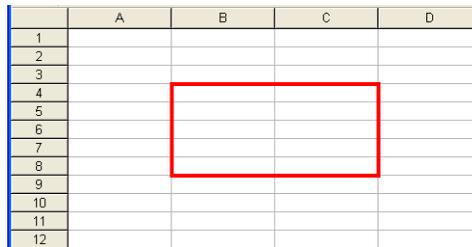
---

### PV SET RANGE BORDER (エリア; 左; 上; 右; 下)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	倍長整数	→ 左側のセルのカラム番号
上	倍長整数	→ 上側のセルの行番号
右	倍長整数	→ 右側のセルのカラム番号
下	倍長整数	→ 下側のセルの行番号

### 説明

**PV SET RANGE BORDER** コマンドは **PV SET BORDER STYLE** コマンドを用いて設定された枠線を指定されたセル領域に対して適用します。セル領域は、引数<左>、<上>、<右>、<下>で定義します。



	A	B	C	D
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

左=2 右=3 上=4 下=8

領域に関する詳細は、「PV Selection、はじめに」の節を参照してください。

### 例題

4D View エリア上で、カラム A と B に入力された 12 行からなる表の最下部に、水色の二重横線を引きたいものとします。

**PV SET BORDER STYLE** (Area;pv border edge top;pv border style 111;

**PV Index to color** (Light blue)) `枠線スタイルと色

⇒ **PV SET RANGE BORDER** (Area;1;12;2;12) `領域の最下部に下線を引く

### 参照

PV SET BORDER STYLE

## PV SET BORDER STYLE

### PV SET BORDER STYLE (エリア; 辺; スタイル; 色)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
辺	倍長整数	→ 枠線の辺
スタイル	倍長整数	→ 枠線のスタイル
色	倍長整数	→ 枠線の色

#### 説明

**PV SET BORDER STYLE** コマンドは、枠線の<辺>に対して<スタイル>と<色>を設定します。

注：このコマンドは、エリアのセルのスタイルには適用されません。**PV SET RANGE BORDER** コマンドを使用した際に適用される枠線スタイルを定義するだけです。

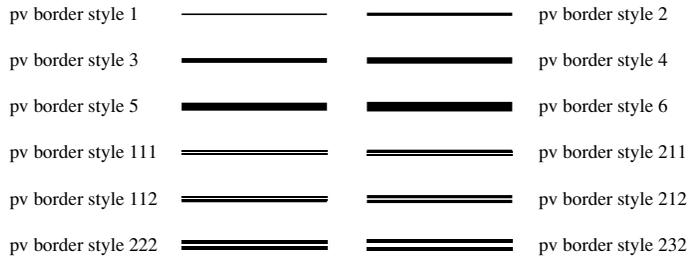
引数<辺>の定義には、**PV Border edge** 定数を使用します。枠線は、枠の縁部（辺）として定義されます（上、下、左、右）。一回の**PV SET BORDER STYLE** コマンドコールで、複数の枠線を設定するには、単に各定数を追加してゆきます。例えば、ある行に対して、“pv border edge top + pv border edge bottom” と指定すると、枠の上側と下側の線の設定を行えます。

セル領域に対し、部分的な枠または完全な枠を形成するように枠線を組み合わせることができます。この場合、内部の枠線はその枠に含まれている各セルの端に相当し、**pv border edge inner vert** および **pv border edge inner hor** 定数を使用して設定することができます。また、各枠線には特定のユニークな特徴を持たせることができます。枠線にさまざまなスタイル設定を行う必要がある場合には、**PV SET BORDER STYLE** コマンドを何度でも呼び出すことが可能です。

引数<スタイル>の定義には、**PV Border style** 定数を使用します。これらの定数の定義は、以下の通りです。

- 単純な枠線の場合、定数はピクセル数を表わします（例えば、**pv border style 4 = 4** ピクセルの線）。
- 枠線の組み合わせの場合、定数は各要素のピクセル数を表わします（例えば、**pv border style 211 = 2** ピクセルの線、1ピクセルのスペース、1ピクセルの線）

これらの定数を以下の図で説明します。



引数<色>は、RGBタイプの倍長整数です。この値として、**PV Index to color**関数を使用し、4Dパレットにある256色から1つを選ぶことができます。厳密に言えば、4Dランゲージの定数である「Colors」テーマを使用することが可能であり、最初の16色（カラーパレットの1番目の行）が利用できます。

4th Dimensionで利用できる色についての詳細は、以下の4Dコマンドの説明を参照してください。

■ **SET RGB COLORS**：4Dで使用するRGBカラーのシステムについて

■ **SET COLOR**：4Dパレットのインデックスカラーについて

### 例題

**PV SET RANGE BORDER** コマンドの例題を参照してください。

### 参照

PV SET RANGE BORDER

### 定数

PV Border edge および PV Border style テーマ

## PV GET BORDER STYLE

### PV GET BORDER STYLE (エリア; 辺; スタイル; 色)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
辺	倍長整数	→ 枠線の辺
スタイル	倍長整数	← 枠線のスタイル
色 倍長整数		← 枠線の色

#### 説明

**PV GET BORDER STYLE** コマンドは、4D View <エリア>の枠線の<辺>に設定された<スタイル>と<色>を取得します。

引数<辺>の定義には、**PV Border edge** 定数を使用します。

引数<スタイル>に返される値は、**PV Border style** 定数の値に相当します。**PV Border style** テーマの定数に関する詳細は、**PV SET BORDER STYLE** コマンドを参照してください。

#### 例題

あるデータ範囲に設定された枠線スタイルが、あなたの希望するものであるかどうかを検証します (**PV SET BORDER STYLE** コマンドを参照)。実際にはどこにも枠線を設定していない点に注意してください。次のコードを実行すると、**PV SET RANGE BORDER** コマンドで作成しようとする枠線の設定を変更するだけです。

```

C_LONGINT($Style;$Color) ` スタイル属性
    ` スタイル情報を取得
⇒ PV GET BORDER STYLE (Area;pv border edge bottom;$Style;$Color)
    ` 希望するスタイルか？
If (Style#pv border style 111) | ($Color#PV Index to color (Light blue))
    ` 希望していたスタイルではない...
        PV SET BORDER STYLE (Area;pv border edge bottom;pv border style 111;
            PV Index to color (Light blue)) ` 枠のスタイルと色
End if

```

#### 参照

PV SET BORDER STYLE、PV SET RANGE BORDER

#### 定数

PV Border edge および PV Border style テーマ



## PV Cell property、はじめに

このテーマ内のコマンドを使用して、単独のセルまたはセル選択範囲に関するプロパティを定義したり、その情報を取得することができます。プロパティには、ロック、非表示、入力フィルタ、表示フォーマット等があります。

このテーマ内にある3つのコマンドを使用し、セル名を管理することができます。つまり、4D Viewエリアにある既存の名を一覧表示したり、新しい名前を作成することができます。

セルに名前を付けることにより、セルの管理がより簡単になります。例えば、ユーザや開発者にとって、セルの位置決めを行えるポップアップメニューにおいて、カラムYの384行にあるセルというよりも“合計”セルと言った方がわかりやすいでしょう。

## PV SET CELL PROPERTY

---

### PV SET CELL PROPERTY (エリア; カラム; 行; プロパティ; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
プロパティ	倍長整数	→ プロパティ番号
値	倍長整数	→ プロパティの値

#### 説明

**PV SET CELL PROPERTY** コマンドは、引数<カラム>と<行>で定義されたセルの<プロパティ>の値を設定します。

<プロパティ>の定義には、以下の定数を使用します。

■ **PV Cell properties** テーマ内の定数。この場合、ピクセル単位で表わしたセルの高さと幅を引数<値>に渡します。

■ **PV Style properties** テーマ内の定数。この場合、引数<値>を定義するには、**PV Style values**、**PV Style special values**、**PV Style format date time**、**PV Picture mapping mode** テーマのいずれかの定数を使用します。

引数<プロパティ>に使用する各種定数、および関連する<値>を以下に説明します。

#### PV Cell properties テーマ

##### pv cell width

セル幅を設定します。関連する値：ピクセル単位で表わされた幅。

##### pv cell height

セルの高さを設定します。関連する値：ピクセル単位で表わされた高さ。

#### PV Style properties テーマ

##### pv style locked

セルのユーザに対するロック設定を行います。ロックされたセルへの変更や選択等はありません。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : セルはロックされる。
- **pv value off** : セルはロックされない。

#### pv style hidden

セルのロックと非表示を設定します。ロックされ隠されたセルへの変更や選択等はありません。関連する値 : **PV Style values** テーマ内の定数。

- **pv value on** : セルはロックされ、隠されている。
- **pv value off** : セルはロックされておらず、表示されている。

#### pv style spellcheck

セルに対してスペルチェックを適用します。関連する値 : **PV Style values** テーマ内の定数。

- **pv value on** : セルへのスペルチェックを適用する。
- **pv value off** : セルへのスペルチェックを適用しない。

#### pv style use picture height

セルに関連付けられたピクチャの高さに応じて、セルのサイズを合わせます。関連する値 : **PV Style values** テーマ内の定数。

- **pv value on** : セルのサイズをそのセル内のピクチャの高さに合わせる。セルに関連付けられたピクチャが存在しない場合には、セルのサイズは変わらない。
- **pv value off** : セルのサイズは、そのセルに関連付けられたピクチャの高さに合わせて変わらない。

#### pv style based on

セルは、引数<値>に渡された番号のスタイルシートをひな形として使用します。関連する値 : スタイルシート番号、または **PV Style special values** テーマ内の定数。

#### pv style format alpha

セルは、引数<値>に渡された番号のテキスト表示フォーマットを使用します。関連する値 : 表示フォーマット番号。

### pv style format num

セルは、引数<値>に渡された番号の数値表示フォーマットを使用します。関連する値：表示フォーマット番号。

注：デフォルトの表示フォーマット番号は、セルの「フォーマット」ダイアログボックスにおいて、形式を選択する際に用いるメニュー上の位置に対応します。

### pv style format bool

セルは、引数<値>に渡された番号のブーリアン表示フォーマットを使用します。関連する値：表示フォーマット番号。

### pv style format date time

セルは、引数<値>に渡された番号の日付&時刻表示フォーマットを使用します。関連する値：**PV Style format date time** テーマ内の定数。

■ **pv Short** : 02/21/02

■ **pv Abbreviated** : thu 21 feb 2002

■ **pv Long** : thursday 21 february 2002

■ **pv Short2** : 02/21/2002

■ **pv Month Day Year** : 21 february, 2002

■ **pv Abbr Month Day Year** : 21 feb, 2002

■ **pv Day Name** : thursday

■ **pv Day Number** : 21

■ **pv Month Name** : february

■ **pv Month Number** : 2

■ **pv Year Number** : 2002

■ **pv Long H MM AM PM** : thursday 21 february 2002 at 12:30 PM

■ **pv Abbreviated H MM AM PM** : thu 21 feb 2002 at 12:30 PM

■ **pv Short HH MM SS** : 02/21/02 at 12:30:00

■ **pv Month Day Year H MM AM PM** : 21 february, 2002 at 12:30 PM

■ **pv Short2 Hour Min Sec** : 21/02/2002 and 12 hours 30 minutes 0 second

■ **pv HH MM SS** : 12:30:00

■ **pv HH MM** : 12:30

■ **pv Hour Min Sec** : 12 hours 30 minutes 0 second

■ **pv Hour Min** : 12 hours 30 minutes

■ **pv HH MM AM PM** : 12:30 PM

注：現在のシステム設定に応じて、表示結果が異なる場合があります。

pv style format picture

セルに関連付けられたピクチャ表示フォーマットを定義します。関連する値：**PV Picture mapping mode** テーマ内の定数。

■ **pv mapping trunc non-centered**

■ **pv mapping truncated centered**

■ **pv mapping replicated**

■ **pv mapping scaled to fit prop**

■ **pv mapping scaled to fit**

■ **pv mapping scaled centered prop**

pv style color back even

偶数番号の行にあるセルの背景色を設定します。関連する値：カラー番号 (**PV RGB to color** 関数、および **PV Index to color** 関数を参照)。

pv style color back odd

奇数番号の行にあるセルの背景色を設定します。関連する値：カラー番号 (**PV RGB to color** 関数、および **PV Index to color** 関数を参照)。

pv style color text even

偶数番号の行にあるセルのテキストの色を設定します。関連する値：カラー番号 (**PV RGB to color** 関数、および **PV Index to color** 関数を参照)。

pv style color text odd

奇数番号の行にあるセルのテキストの色を設定します。関連する値：カラー番号 (**PV RGB to color** 関数、および **PV Index to color** 関数を参照)。

#### `pv style color zero even`

偶数番号の行に位置し、その値が0（ゼロ）であるセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### `pv style color zero odd`

奇数番号の行に位置し、その値が0（ゼロ）であるセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### `pv style color minus even`

偶数番号の行に位置し、その値が負数（マイナス）であるセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### `pv style color minus odd`

奇数番号の行に位置し、その値が負数（マイナス）であるセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### `pv style text font`

セルのフォントを設定します。関連する値：フォント番号（**PV Add font**関数、および**PV GET FONT LIST**コマンドを参照）。

#### `pv style text size`

セルのフォントサイズを設定します。関連する値：ピクセル単位のサイズ。

#### `pv style text face`

セルのスタイルシートを設定します。関連する値：スタイルシート番号、または**PV Style special values**テーマ内の定数。

#### `pv style text bold`

セルのテキストを太字に設定します。関連する値：**PV Style values**テーマ内の定数。

- **pv value on** : セルに太字を適用する。
- **pv value off** : セルに太字を適用しない。

#### pv style text italic

セルのテキストを斜体（イタリック）に設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : セルに斜体を適用する。
- **pv value off** : セルに斜体を適用しない。

#### pv style text underline

セルのテキストに下線（アンダーライン）を設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : セルに下線を適用する。
- **pv value off** : セルに下線を適用しない。

#### pv style text outline

セルのテキストをアウトラインに設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : セルにアウトラインを適用する。
- **pv value off** : セルにアウトラインを適用しない。

#### pv style text shadow

セルのテキストをシャドウに設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : セルにシャドウを適用する。
- **pv value off** : セルにシャドウを適用しない。

#### pv style text condensed

セルのテキストをコンデンス（文字間隔を狭める）に設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** :セルにコンデンスを適用する。
- **pv value off** :セルにコンデンスを適用しない。

#### pv style text extended

セルのテキストをエクステンド（文字間隔を拡げる）に設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** :セルにエクステンドを適用する。
- **pv value off** :セルにエクステンドを適用しない。

#### pv style hor alignment

セル内容の横方向の行揃えを設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value hor alignment default** :セルにデフォルトの横揃えを適用する。
- **pv value hor alignment left** :セルに左横揃えを適用する。
- **pv value hor alignment center** :セルに中央横揃えを適用する。
- **pv value hor alignment right** :セルに右横揃えを適用する。

#### pv style vert alignment

セル内容の縦方向の行揃えを設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value vert alignment top** :セルに上揃えを適用する。
- **pv value vert alignment center** :セルに中央揃えを適用する。
- **pv value vert alignment bottom** :セルに下揃えを適用する。

#### pv style rotation

セル内容の回転を設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value rotation 0** :セルに回転を適用しない。
- **pv value rotation 90** :セルに左90度の回転を適用する。
- **pv value rotation 180** :セルに180度の回転を適用する。
- **pv value rotation 270** :セルに左270度の回転を適用する。

注：1つの選択範囲内で、セル領域のプロパティを定義するには、**PV SET RANGE PROPERTY** コマンドを使用することができます。

## 例題

ロックとロック解除とを切り替えるため、フォーミュラでD20のセルを参照するセルをすべて検索します。

**C\_LONGINT**(\$ProtectedCell) `セルをロックまたはロック解除

**C\_LONGINT**(\$EndColumn;\$EndRow) `右下の境界を保存

**C\_LONGINT**(\$StartColumn;\$StartRow) `左上の境界を保存

**C\_LONGINT**(\$Column;\$Row) `ループ内の座標を取得

**ARRAY LONGINT**(\$LeftTab;0)

**ARRAY LONGINT**(\$UpperTab;0)

**ARRAY LONGINT**(\$RightTab;0)

**ARRAY LONGINT**(\$LowerTab;0)

**PV FIND ALL** (Area;"\$D\$20";0;0) `"\$D\$20"を含むセル

`選択されたセルの座標

**PV GET CURRENT CELL** (Area;\$LeftTab;\$UpperTab;\$RightTab;\$LowerTab)

**If** (Size of array(\$LeftTab)#0) `検索条件に一致するフォーミュラが存在するか？

**SORT ARRAY**(\$LeftTab;\$UpperTab;\$RightTab;\$LowerTab;>)

\$StartColumn:=\$LeftTab{1} `一番左側のセルを取得

**SORT ARRAY**(\$UpperTab;\$LeftTab;\$RightTab;\$LowerTab;>)

\$StartRow:=\$UpperTab{1} `一番上側のセルを取得

**SORT ARRAY**(\$RightTab;\$UpperTab;\$LeftTab;\$LowerTab;>)

\$EndColumn:=\$RightTab{Size of array(\$RightTab)} `一番右側のセルを取得

**SORT ARRAY**(\$LowerTab;\$RightTab;\$UpperTab;\$LeftTab;>)

\$EndRow:=\$LowerTab{Size of array(\$LowerTab)} `一番下側のセルを取得

`セレクションを調べる

**For** (\$Column;\$StartCol;\$EndCol)

**For** (\$Row;\$StartRow;\$EndRow)

**If** (Position("\$D\$20";**PV Get cell formula**

(Area;\$Column;\$Row))>0)

`ロックされている？

\$ProtectedCell:=**PV Get cell property**

(Area;\$Column;\$Row;pv style locked)

`ロックとロック解除を切り替える

**PV SET CELL PROPERTY**(Area;\$Column;\$Row;

pv style locked ;Num(\$ProtectedCell=0)) `0 ↔ 1

⇒

**End if**  
**End for**  
**End for**  
**End if** ` 選択された領域 ?

#### 参照

PV Get cell property、PV Get range property、PV SET RANGE PROPERTY

#### 定数

PV Style properties、PV Cell properties、PV Style values、PV Style special values、PV Style format date time、PV Picture mapping mode テーマ内の定数

## PV SET RANGE PROPERTY

---

### PV SET RANGE PROPERTY (エリア; 左; 上; 右; 下; プロパティ; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	倍長整数	→ 左側セルのカラム番号
上	倍長整数	→ 上側セルの行番号
右	倍長整数	→ 右側セルのカラム番号
下	倍長整数	→ 下側セルの行番号
プロパティ	倍長整数	→ プロパティの番号
値	倍長整数	→ プロパティの値

#### 説明

**PV SET RANGE PROPERTY** コマンドは、引数<左>、<上>、<右>、<下>を使用して定義された、選択セル領域の<プロパティ>の<値>を設定します。セル領域に関する詳細は「PV Selection、はじめに」の節を参照してください。

プロパティの定義には、以下の定数を使用することができます。

- **PV Cell properties** テーマ内の定数。この場合、セル範囲の高さまたは幅をピクセル単位で引数<値>に渡します。
- **PV Style properties** テーマ内の定数。この場合、引数<値>を定義するには、**PV Style values**、**PV Style special values**、**PV Style format date time**、または**PV Picture mapping mode** テーマ内の定数を使用します。

引数<プロパティ>で使用する各種定数、および関連する<値>を以下に説明します。

#### PV Cell properties テーマ

##### pv cell width

定義された領域内にある各セルの幅（つまり各カラム）を設定します。関連する値：ピクセル単位で表わされた幅。

##### pv cell height

定義された領域内にある各セルの高さ（つまり各行）を設定します。関連する値：ピクセル単位で表わされた高さ。

## PV Style properties theme

### pv style locked

領域の各セルにおけるロックを設定します。ロックされたセルへの変更や選択等はありません。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : セルはロックされる。
- **pv value off** : セルはロックされない。

### pv style hidden

領域の各セルにおけるロックと非表示を設定します。ロックされ隠されたセルへの変更や選択等はありません。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : セルはロックされ、隠されている。
- **pv value off** : セルはロックされておらず、表示されている。

### pv style spellcheck

セル領域に対してスペルチェックを適用します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : セル領域へのスペルチェックを適用する。
- **pv value off** : セル領域へのスペルチェックを適用しない。

### pv style use picture height

セルに関連付けられたピクチャの高さに応じて、領域内の各セルのサイズを合わせます。関連する値：**PV Style values** テーマ内の定数。

- **pv value on** : 領域内の各セルのサイズをそのセルに含まれるピクチャの高さに合わせる。セルに関連付けられたピクチャが存在しない場合には、セルのサイズは変わらない。
- **pv value off** : セルのサイズは、そのセルに関連付けられたピクチャの高さに合わせて変わらない。

### pv style based on

領域内の各セルは、引数<値>に渡された番号のスタイルシートをひな形として使用します。関連する値：スタイルシート番号、または**PV Style special values** テーマ内の定数。

### pv style format alpha

領域内の各セルは、引数<値>に渡された番号のテキスト表示フォーマットを使用します。関連する値：表示フォーマット番号。

### pv style format num

領域内の各セルは、引数<値>に渡された番号の数値表示フォーマットを使用します。関連する値：表示フォーマット番号。

注：デフォルトの表示フォーマット番号は、セルの「フォーマット」ダイアログボックスにおいて、形式を選択する際に使用するメニュー上の位置に対応します。

### pv style format bool

領域内の各セルは、引数<値>に渡された番号のブーリアン表示フォーマットを使用します。関連する値：表示フォーマット番号。

### pv style format date time

領域内の各セルは、引数<値>に渡された番号の日付&時刻表示フォーマットを使用します。関連する値：**PV Style format date time** テーマ内の定数。

■ **pv Short** : 02/02/21

■ **pv Abbreviated** : thu 21 feb 2002

■ **pv Long** : thursday 21 february 2002

■ **pv Short2** : 02/21/2002

■ **pv Month Day Year** : 21 february, 2002

■ **pv Abbr Month Day Year** : 21 feb, 2002

■ **pv Day Name** : thursday

■ **pv Day Number** : 21

■ **pv Month Name** : february

■ **pv Month Number** : 2

■ **pv Year Number** : 2002

■ **pv Long H MM AM PM** : thursday 21 february 2002 at 12:30 PM

■ **pv Abbreviated H MM AM PM** : thu 21 feb 2002 at 12:30 PM

■ **pv Short HH MM SS** : 02/21/02 at 12:30:00

- **pv Month Day Year H MM AM PM** : 21 february, 2002 at 12:30 PM
- **pv Short2 Hour Min Sec** : 21/02/2002 and 12 hours 30 minutes 0 second
- **pv HH MM SS** : 12:30:00
- **pv HH MM** : 12:30
- **pv Hour Min Sec** : 12 hours 30 minutes 0 second
- **pv Hour Min** : 12 hours 30 minutes
- **pv HH MM AM PM** : 12:30 PM

注：現在のシステム設定に応じて、表示結果が異なる場合があります。

#### pv style format picture

領域内の各セルに関連付けられたピクチャ表示フォーマットを定義します。関連する値：**PV Picture mapping mode** テーマ内の定数。

- **pv mapping trunc non-centered**
- **pv mapping truncated centered**
- **pv mapping replicated**
- **pv mapping scaled to fit prop**
- **pv mapping scaled to fit**
- **pv mapping scaled centered prop**

#### pv style color back even

領域内の各セルのうち、偶数番号の行に位置するセルの背景色を設定します。関連する値：カラー番号 (**PV RGB to color** 関数、および **PV Index to color** 関数を参照)。

#### pv style color back odd

領域内の各セルのうち、奇数番号の行に位置するセルの背景色を設定します。関連する値：カラー番号 (**PV RGB to color** 関数、および **PV Index to color** 関数を参照)。

#### pv style color text even

領域内の各セルのうち、偶数番号の行に位置するセルのテキストの色を設定します。関連する値：カラー番号 (**PV RGB to color** 関数、および **PV Index to color** 関数を参照)。

#### pv style color text odd

領域内の各セルのうち、奇数番号の行に位置するセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### pv style color zero even

領域内の各セルのうち、偶数番号の行に位置し、その値が0（ゼロ）であるセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### pv style color zero odd

領域内の各セルのうち、奇数番号の行に位置し、その値が0（ゼロ）であるセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### pv style color minus even

領域内の各セルのうち、偶数番号の行に位置し、その値が負数（マイナス）であるセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### pv style color minus odd

領域内の各セルのうち、奇数番号の行に位置し、その値が負数（マイナス）であるセルのテキストの色を設定します。関連する値：カラー番号（**PV RGB to color**関数、および**PV Index to color**関数を参照）。

#### pv style text font

セル領域のフォントを設定します。関連する値：フォント番号（**PV Add font**関数、および**PV GET FONT LIST**コマンドを参照）。

#### pv style text size

セル領域のフォントサイズを設定します。関連する値：ピクセル単位のサイズ。

#### pv style text face

セル領域のスタイルシートを設定します。関連する値：スタイルシート番号、または **PV Style special values** テーマ内の定数。

#### pv style text bold

セル領域のテキストを太字に設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on**：セルに太字を適用する。
- **pv value off**：セルに太字を適用しない。

#### pv style text italic

セル領域のテキストを斜体（イタリック）に設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on**：セルに斜体を適用する。
- **pv value off**：セルに斜体を適用しない。

#### pv style text underline

セル領域のテキストに下線（アンダーライン）を設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on**：セルに下線を適用する。
- **pv value off**：セルに下線を適用しない。

#### pv style text outline

セル領域のテキストをアウトラインに設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value on**：セルにアウトラインを適用する。
- **pv value off**：セルにアウトラインを適用しない。

#### pv style text shadow

セル領域のテキストをシャドウに設定します。関連する値：**PV Style values** テーマ内の定数。

■ **pv value on** :セルにシャドウを適用する。

■ **pv value off** :セルにシャドウを適用しない。

#### pv style text condensed

セル領域のテキストをコンデンス（文字間隔を狭める）に設定します。関連する値：**PV Style values** テーマ内の定数。

■ **pv value on** :セルにコンデンスを適用する。

■ **pv value off** :セルにコンデンスを適用しない。

#### pv style text extended

セル領域のテキストをエクステンド（文字間隔を広げる）に設定します。関連する値：**PV Style values** テーマ内の定数。

■ **pv value on** :セルにエクステンドを適用する。

■ **pv value off** :セルにエクステンドを適用しない。

#### pv style hor alignment

セル領域の内容に横方向の行揃えを設定します。関連する値：**PV Style values** テーマ内の定数。

■ **pv value hor alignment default** :セル領域にデフォルトの横揃えを適用する。

■ **pv value hor alignment left** :セル領域に左横揃えを適用する。

■ **pv value hor alignment center** :セル領域に中央横揃えを適用する。

■ **pv value hor alignment right** :セル領域に右横揃えを適用する。

#### pv style vert alignment

セル領域の内容に縦方向の行揃えを設定します。関連する値：**PV Style values** テーマ内の定数。

■ **pv value vert alignment top** :セル領域に上揃えを適用する。

■ **pv value vert alignment center** :セル領域に中央揃えを適用する。

■ **pv value vert alignment bottom** :セル領域に下揃えを適用する。

## `pv style rotation`

セル領域の内容に回転を設定します。関連する値：**PV Style values** テーマ内の定数。

- **pv value rotation 0**：セル領域に回転を適用しない。
- **pv value rotation 90**：セル領域に左90度の回転を適用する。
- **pv value rotation 180**：セル領域に180度の回転を適用する。
- **pv value rotation 270**：セル領域に左270度の回転を適用する。

注：単一のセルのプロパティを定義するには、**PV SET CELL PROPERTY** コマンドを使用することができます。

## 例題

フォーミュラでD20のセルを参照しているセルをすべて太字に設定します。

```
ARRAY LONGINT($LeftArray;0)
ARRAY LONGINT($TopArray;0)
ARRAY LONGINT($RightArray;0)
ARRAY LONGINT($BottomArray;0)
C_INTEGER($Index)
```

```
PV FIND ALL (Area;"D$20";0;0) `"$D$20"を含むフォーミュラ
```

```
    `選択されたセルの座標
```

```
PV GET SELECTED RANGES LIST
```

```
    (Area;$LeftArray;$TopArray;$RightArray;$BottomArray)
```

```
    `太字に設定する
```

```
For ($Index;1;Size of Array($LeftArray)) `範囲をくまなく処理する
```

```
⇒ PV SET RANGE PROPERTY (Area;$LeftArray{$Index};$TopArray{$Index};
    $RightArray{$Index};$BottomArray{$Index};
    pv style text bold ;pv value on)
```

```
End for
```

## 参照

PV Get cell property、PV Get range property、PV SET CELL PROPERTY

## 定数

PV Style properties、PV Cell properties、PV Style values、PV Style special values、PV Style format date time、PV Picture mapping mode テーマ内の定数

## PV Get cell property

---

### PV Get cell property (エリア; カラム; 行; プロパティ) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
プロパティ	倍長整数	→ プロパティ番号
戻り値	倍長整数	← プロパティの値

### 説明

**PV Get cell property** 関数は、引数<カラム>と<行>で定義されたセルの<プロパティ>の値を返します。

引数<プロパティ>の定義には、**PV Style properties** および **PV Cell properties** テーマ内の定数を使用します。これらの関数に関する詳細は、**PV SET CELL PROPERTY** コマンドを参照してください。

### 例題

**PV SET CELL PROPERTY** コマンドの例題を参照してください。

### 参照

PV Get range property、PV SET CELL PROPERTY、PV SET RANGE PROPERTY

### 定数

PV Style properties および PV Cell properties テーマ

## PV Get range property

---

### PV Get range property (エリア; 左; 上; 右; 下; プロパティ) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	倍長整数	→ 左側セルのカラム番号
上	倍長整数	→ 上側セルの行番号
右	倍長整数	→ 右側セルのカラム番号
下	倍長整数	→ 下側セルの行番号
プロパティ	倍長整数	→ プロパティの番号
戻り値	倍長整数	← プロパティの値

### 説明

**PV Get range property**関数は、引数<左>、<上>、<右>、<下>を使用して定義された選択セル領域の<プロパティ>の値を返します。セル領域に関する詳細は、「PV Selection、はじめに」の節を参照してください。

引数<プロパティ>の定義には、**PV Style properties**および**PV Cell properties**テーマ内の定数を使用します。これらの定数に関する詳細は、**PV SET RANGE PROPERTY**コマンドの説明を参照してください。

### 複合値

プロパティのなかには、**PV Get range property**関数より65535または255という値が返されるものがあります。この値は、**PV Style values**テーマの“複合”タイプの定数に相当します。

たとえば、以下の構文では、

```
$value:=PV Get range property(area; column; row; pv style color back even)
```

領域内にある偶数番号行のセルが、すべて同じ背景色ではない場合に65535という値が返されます。この値は、**PV Style values**テーマの定数**pv value format mixed**に相当します。また、領域内にある偶数番号行のセルが、すべて同じ背景色である場合には、実際のカラー番号が返されます。

この原則は、以下のプロパティに対して適用されます。

プロパティ	複号選択の場合に返される値/定数
pv style format alpha	65535 (pv value format mixed)
pv style format num	65535 (pv value format mixed)
pv style format bool	65535 (pv value format mixed)
pv style format date time	65535 (pv value format mixed)
pv style format picture	65535 (pv value format mixed)
pv style color back even	65535 (pv value format mixed)
pv style color back odd	65535 (pv value format mixed)
pv style color text even	65535 (pv value format mixed)
pv style color text odd	65535 (pv value format mixed)
pv style color zero even	65535 (pv value format mixed)
pv style color zero odd	65535 (pv value format mixed)
pv style hor alignment	255 (pv value hor alignment mixed)
pv style vert alignment	255 (pv value vert alignment mixed)
pv style rotation	255 (pv value rotation mixed)

### 例題

**PV Get cell property** 関数の例題を参照してください。

### 参照

PV Get cell property、PV SET CELL PROPERTY、PV SET RANGE PROPERTY

### 定数

PV Style properties および PV Cell properties テーマ

## PV SET CELL NAME

---

### PV SET CELL NAME (エリア; カラム; 行; 名前)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
名前	文字列	→ セルの名前

### 説明

**PV SET CELL NAME** コマンドは、引数<カラム>と<行>で定義されたセルの<名前>を設定します。

注：セル名の頭3桁には、数字を使用してはいけません。

### 例題

**PV GET CELL NAME LIST** コマンドと **PV GOTO CELL** コマンドを使用すると、セルの絶対参照（カラム番号または行番号）あるいはセル名のいずれかで、セルにアクセスすることができます。4D View エリアのフォーミュラでは、セル名を使ってこのセルを参照することもできます。

```
C_TEXT($CellName) ` セルに割り当てる名前  
C_LONGINT($Column) ` カレントセルのカラム番号  
C_LONGINT($Row) ` カレントセルの行番号
```

```
PV GET CURRENT CELL (Area;$Column;$Row) ` カレントセルの座標を取得  
$CellName:=PV Get cell name(Area;$Column;$Row) ` 既に名前が設定されている  
可能性がある
```

```
If ($CellName="") ` セルには既に名前が付けられている？  
Repeat ` アクティブセルの名前を尋ねる  
$CellName:=Request("このセルに割り当てたい名前は？";"新しい名前")  
If($CellName#"") ` 名前が入力された
```

```
⇒ PV SET CELL NAME (Area;$Column;$Row;$CellName)  
` 入力された名前を割り当てる  
End if  
Else ` カレントセルには既に名前が付けられている  
ALERT("このセルには既に名前が付いています。["+ $CellName+"] ") ` 警告  
End if
```

## 参照

PV Get cell name、PV GET CELL NAME LIST

## PV Get cell name

---

**PV Get cell name** (エリア; カラム; 行) → 文字列

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
カラム	倍長整数	→	セルのカラム番号
行	倍長整数	→	セルの行番号
戻り値	文字列	←	セルの名前

## 説明

**PV Get cell name** 関数は、セルの名前を返します。

## 例題

**PV SET CELL NAME** コマンドの例題を参照してください。

## 参照

PV GET CELL NAME LIST、PV SET CELL NAME

## PV GET CELL NAME LIST

---

### PV GET CELL NAME LIST (エリア; カラム; 行; 名前)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	配列	→ セルのカラム番号の配列
行	配列	→ セルの行番号の配列
名前	配列	← 各セルの名前

### 説明

**PV GET CELL NAME LIST** コマンドは、名前が割り当てられているセルの名前、カラム番号、行番号を取得し、それぞれ<名前>、<カラム>、<行>の配列に代入します。

### 例題

以下のメソッドは、セル名を管理する一般的なメソッドです。

名前を用いてセルを位置付けることができるコマンドは特に存在しません。しかし、配列 \$TabNames において、求めるセル名の位置を割り出し、それに対応する各配列 \$TabColumns および \$TabRows の要素を引数として **PV GOTO CELL** コマンドを使用することにより、これが実現します。

```
C_TEXT($1) ` 移動先のセルの名前
C_LONGINT($Position) ` 名前リスト内でのセル名の位置
ARRAY LONGINT($TabColumns;0) ` カラム配列
ARRAY LONGINT($TabRows;0) ` 行配列
ARRAY TEXT($TabNames;0) ` セル名配列
```

```
⇒ ` 名前と対応する座標を取得する
PV GET CELL NAME LIST (Area;$TabColumns;$TabRows;$TabNames)
```

```
$Position:=Find in array($TabNames;$1) ` セルを検索
If ($Position#-1) ` 存在する場合
    ` そのセルをカレントセルに
    PV GOTO CELL (Area;$TabColumns{$Position};$TabRows{$Position})
Else
    ` エラー結果を管理する
End if
```

### 参照

PV Get cell name、PV GOTO CELL、PV SET CELL NAME

## PV Cell value、はじめに

このテーマ内の定数を使用し、セルに値を割り当てたり、セル内容を復元することができます。

セル内容のタイプは、4th Dimensionに関連したさまざまなタイプがありますが（テキスト、数値、日付等）、変数やフィールド、フォーミュラ、コントロール、クイックレポートなどのように、異なるタイプも存在します（詳細は、**PV SET CELL CONTROL** コマンドを参照）。

コマンドに応じて、セルの値は固定（起動時にデータベースフィールドの値をコピー）、またはダイナミック（対話式にデータベースへリンク）にすることができます。

## PV Get cell value type

---

### PV Get cell value type(エリア; カラム; 行) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
戻り値	倍長整数	← セル値のタイプ

### 説明

**PV Get cell value type**関数は、<カラム>と<行>で指定された<エリア>のセルに格納されている値のタイプを返します。

タイプは倍長整数の値として返され、この値は**PV Cell value type**テーマ内の定数に対応しています。

注：セルに格納されている値のタイプは、セルの内容に応じて4D Viewが自動的に定義します。

### 例題

**PV SET CELL FORMULA** コマンドの例題を参照してください。

### 定数

PV Cell value type テーマ

## PV SET CELL STRING VALUE

### PV SET CELL STRING VALUE(エリア; カラム; 行; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
値	文字列	→ セルの値

#### 説明

**PV SET CELL STRING VALUE** コマンドは、<カラム>と<行>で指定されたセルに文字列タイプの<値>を書き込みます。

<値>が“=”で始まる場合、フォーミュラとして読み込まれます。

#### 例題

次の例題は、文字タイプのセルに対してのみ“下方向へのコピー”を再現します。

```
C_INTEGER($CopyNumber)
```

```
C_LONGINT($Column;$Row) `コピーするセルの座標
```

```
C_INTEGER($Index) `ループの指標
```

```
C_STRING(80;$Value) `コピーする値
```

```
$CopyNumber:=Num(Request("下方向へ何回コピーしますか?";"5")) `5が省略値
```

```
If($CopyNumber>0)
```

```
    PV_GET_CURRENT_CELL(Area;$Column;$Row) `コピーするセル
```

```
    $Value:=PV_Get_cell_string_value(Area;$Column;$Row)
```

```
    For($Index;$Row+1;$Row+$CopyNumber) ` $CopyNumberの
```

```
        回数分コピーする
```

```
⇒        PV_SET_CELL_STRING_VALUE(Area;$Column;$Index;$Value)
```

```
    End for
```

```
End if
```

Tip：このメソッドは、任意のタイプに対して使用可能ですが、その場合“PV Get cell xxx value”を呼び出す前に、PV Get cell value type関数を用いてセルのタイプを調べるジェネリックなメソッドをコールし、この後でタイプとは関係なく“PV SET CELL XXX VALUE”コマンドを使用して値をコピーしたほうがよいでしょう。これは、ジェネリックなプログラム作成の最適な練習になります。

## 参照

PV Get cell string value、PV Get cell value type

## PV Get cell string value

---

**PV Get cell string value**(エリア; カラム; 行) → 文字列

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
カラム	倍長整数	→	セルのカラム番号
行	倍長整数	→	セルの行番号
戻り値	文字列	←	セルの値

## 説明

**PV Get cell string value**関数は、<カラム>と<行>で指定されたセルに格納されている文字列を返します。

**PV Get cell string value**関数は、文字タイプではないセルに対して使用されると、空の文字列を返します。

## 例題

**PV SET CELL STRING VALUE** コマンドの例題を参照してください。

## 参照

PV Get cell value type、PV SET CELL STRING VALUE

## PV SET CELL TEXT VALUE

---

### PV SET CELL TEXT VALUE(エリア; カラム; 行; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
値	テキスト	→ セルの値

### 説明

**PV SET CELL TEXT VALUE** コマンドは、<カラム>と<行>で指定されたセルにテキストタイプの<値>を書き込みます。

### 例題

**PV SET CELL STRING VALUE** コマンドの例題を参照してください。

### 参照

PV Get cell text value、PV Get cell value type

## PV Get cell text value

---

### PV Get cell text value(エリア; カラム; 行) → テキスト

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
戻り値	テキスト	← セルの値

### 説明

**PV Get cell text value**関数は、<カラム>と<行>で指定されたセルに格納されているテキストを返します。

**PV Get cell text value**関数は、テキストタイプではないセルに対して使用されると、空の文字列を返します。

### 例題

**PV SET CELL STRING VALUE**コマンドの例題を参照してください。

### 参照

PV Get cell value type、PV SET CELL TEXT VALUE

## PV SET CELL NUM VALUE

---

### PV SET CELL NUM VALUE(エリア; カラム; 行; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
値	数値	→ セルの値

### 説明

**PV SET CELL NUM VALUE** コマンドは、<カラム>と<行>で指定されたセルに数値タイプの<値>を割り当てます。

### 例題

**PV SET CELL STRING VALUE** コマンドの例題を参照してください。

### 参照

PV Get cell num value、PV Get cell value type

## PV Get cell num value

---

### PV Get cell num value(エリア; カラム; 行) → 数値

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
戻り値	数値	← セルの値

### 説明

**PV Get cell num value**関数は、<カラム>と<行>で指定されたセルに格納されている数値を返します。

**PV Get cell num value**関数は、数値タイプではないセルに対して使用されると、0を返します。

### 例題

**PV SET CELL STRING VALUE**コマンドの例題を参照してください。

### 参照

PV Get cell value type、PV SET CELL NUM VALUE

## PV SET CELL DATE VALUE

---

### PV SET CELL DATE VALUE(エリア; カラム; 行; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
値	日付	→ セルの値

### 説明

**PV SET CELL DATE VALUE** コマンドは、<カラム>と<行>で指定されたセルに日付タイプの<値>を割り当てます。

### 例題

**PV SET CELL STRING VALUE** コマンドの例題を参照してください。

### 参照

PV Get cell date value、PV Get cell value type

## PV Get cell date value

---

### PV Get cell date value(エリア; カラム; 行) → 日付

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
戻り値	日付	← セルの値

### 説明

**PV Get cell date value**関数は、<カラム>と<行>で指定されたセルに格納されている日付を返します。

**PV Get cell date value**関数は、日付タイプではないセルに対して使用されると00/00/00を返します。

### 例題

**PV SET CELL STRING VALUE**コマンドの例題を参照してください。

### 参照

PV Get cell value type、PV SET CELL DATE VALUE

## PV SET CELL TIME VALUE

---

### PV SET CELL TIME VALUE(エリア; カラム; 行; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
値	時間	→ セルの値

### 説明

**PV SET CELL TIME VALUE** コマンドは、<カラム>と<行>で指定されたセルに時間タイプの<値>を割り当てます。

### 例題

**PV SET CELL STRING VALUE** コマンドの例題を参照してください。

### 参照

PV Get cell time value、PV Get cell value type

## PV Get cell time value

---

### PV Get cell time value(エリア; カラム; 行) → 時間

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
カラム	倍長整数	→	セルのカラム番号
行	倍長整数	→	セルの行番号
戻り値	時間	←	セルの値

### 説明

**PV Get cell time value**関数は、<カラム>と<行>で指定されたセルに格納されている時間を返します。

**PV Get cell time value**関数は、時間タイプではないセルに対して使用されると 00:00:00 を返します。

### 例題

**PV SET CELL STRING VALUE** コマンドの例題を参照してください。

### 参照

PV Get cell value type、PV SET CELL TIME VALUE

## PV SET CELL DATE TIME VALUE

### PV SET CELL DATE TIME VALUE(エリア; カラム; 行; 日付; 時間)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
日付	日付	→ 日付タイプのセルの値
時間	時間	→ 時間タイプのセルの値

#### 説明

**PV SET CELL DATE TIME VALUE** コマンドは、<日付>と<時間>を1つの値として<カラム>と<行>で指定されたセルに割り当てます。

#### 例題

このメソッドは、会議の日付と時間を求めると同時に、会議日時を翌日の30分遅い時間に繰り下げます。その後、新しい日程をセルA1に表示します。

```

C_DATE($Date) ` 会議の日付
C_TIME($Time) ` 会議の時間
$Date:=Date(Request("会議の日付";String(Current date)))
If($Date#100/00/00!) ` 有効な日付
    $Time:=Time(Request("会議の時間";Time string(Current time)))
    If($Time# 00:00:00 ) ` 有効な時間
        ` 会議の日程を1日後の30分遅い時間に変更し、
            それをセルA1に割り当てる
⇒ PV SET CELL DATE TIME VALUE(Area;1;1;$Date+1;
    $Time+!00:30:00!!)
PV GET CELL DATE TIME VALUE(Area;1;1;$Date;$Time)
    ` 情報の読み込み
ALERT("会議の日程が繰り下げられました。日付："+string($Date)
    +"時間："+Time string($Time))
End if
End if

```

#### 参照

PV GET CELL DATE TIME VALUE、PV Get cell value type

## PV GET CELL DATE TIME VALUE

---

### PV GET CELL DATE TIME VALUE(エリア; カラム; 行; 日付値; 時間値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
日付値	日付	← 日付タイプのセルの値
時間値	時間	← 時間タイプのセルの値

#### 説明

**PV GET CELL DATE TIME VALUE** コマンドは、<カラム>と<行>で指定されたセル内に連結されている<日付>と<時間>の値を返します。

**PV GET CELL DATE TIME VALUE** コマンドは、日付や時間タイプではないセルに対して使用されると、00/00/00 と 00:00:00 を返します。

#### 例題

**PV SET CELL DATE TIME VALUE** コマンドの例題を参照してください。

#### 参照

PV Get cell value type、PV SET CELL DATE TIME VALUE

## PV SET CELL BOOLEAN VALUE

### PV SET CELL BOOLEAN VALUE(エリア; カラム; 行; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
値	整数	→ セルの値

#### 説明

**PV SET CELL BOOLEAN VALUE** コマンドは、<カラム>と<行>で指定されたセルに整数タイプの<値>を割り当てます。この整数を使用して、ブール値を表現することができます (0=偽、1=真)。これは、Num(ブール式) 関数として4Dのブール値に変換する必要があります。

#### 例題

**PV SET CELL STRING VALUE** コマンドの例題に手を加え、整数で表わされたブールタイプに変更します。

```

C_INTEGER($CopyNumber)
C_LONGINT($Column;$Row) `コピーするセルの座標
C_INTEGER($Index) `ループの指標
C_BOOLEAN($Value) `コピーする値
$CopyNumber:=Num(Request("下方向へ何回コピーしますか?";"5")) `5が省略値
If($CopyNumber>0)
    PV GET CURRENT CELL(Area;$Column;$Row) `コピーするセル
        `整数をブールに変換
    $Value:=(PV Get cell boolean value(Area;$Column;$Row)=1)
    For($Index;$Row+1;$Row+$CopyNumber) `$CopyNumberの
        回数分コピーする
⇒ PV SET CELL BOOLEAN VALUE(Area;$Column;$Index;
    Num($Value)) `値をコピー
End for
End if

```

#### 参照

PV Get cell boolean value、PV Get cell value type

## PV Get cell boolean value

---

### PV Get cell boolean value(エリア; カラム; 行) → 整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
戻り値	整数	← セルの値

### 説明

**PV Get cell boolean value**関数は、<カラム>と<行>で指定されたセルの値をブールとして返します。この値を4Dのブール値に変換するには、“ブール値:=(戻り値=1)”という式を使用します。

**PV Get cell boolean value**関数は、ブールタイプではないセルに対して使用されると0を返します。

### 例題

**PV SET CELL BOOLEAN VALUE** コマンドの例題を参照してください。

### 参照

PV Get cell value type、PV SET CELL BOOLEAN VALUE

## PV SET CELL PICTURE VALUE

---

### PV SET CELL PICTURE VALUE(エリア; カラム; 行; 値)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
カラム	倍長整数	→	セルのカラム番号
行	倍長整数	→	セルの行番号
値	ピクチャ	→	セルの値

### 説明

**PV SET CELL PICTURE VALUE** コマンドは、<カラム>と<行>で指定されたセルにピクチャタイプの<値>を割り当てます。

### 例題

**PV SET CELL STRING VALUE** コマンドの例題を参照してください。

### 参照

PV Get cell picture value、PV Get cell value type

## PV Get cell picture value

---

### PV Get cell picture value(エリア; カラム; 行) → ピクチャ

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
カラム	倍長整数	→	セルのカラム番号
行	倍長整数	→	セルの行番号
戻り値	ピクチャ	←	セルの値

### 説明

**PV Get cell picture value**関数は、<カラム>と<行>で指定されたセルに格納されているピクチャを返します。

**Get cell picture value**関数は、ピクチャタイプではないセルに対して使用されると空のピクチャを返します。

### 例題

**PV SET CELL STRING VALUE**コマンドの例題を参照してください。

### 参照

PV Get cell value type、PV SET CELL PICTURE VALUE

## PV SET CELL VARIABLE

### PV SET CELL VARIABLE(エリア; カラム; 行; 変数)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
変数	文字列	→ 変数名

#### 説明

**PV SET CELL VARIABLE** コマンドは、<カラム>と<行>で指定されたセルを<変数>に関連付けます。セル内容に何らかの変更が加えられると、その変更が変数に反映されます（その逆も同様）。

#### 例題

次のフォームメソッドは、変数vTimeを使用してC3セルに現在時刻を表示します。この変数は毎秒ごとに更新され、セルは時計の役割を果たします。

**C\_TIME(vTime)** `表示する時刻を受け取る変数

**Case of**

\ (Form event=On Load)

**If(PV Get cell variable(Area;3;3)="" )** `C3セルには変数が

まだ関連付けられていない

⇒

**PV SET CELL VARIABLE(Area;3;3;"vTime")** `変数vTimeに  
関連付ける

**End if**

**SET TIMER(60)** `毎秒ごと

\ (Form event=On Timer)

vTime:=**Current time**

**End case**

#### 参照

PV ADD DYNAMIC ARRAYS、PV Get cell value type、PV Get cell variable、PV SET CELL FIELD

## PV Get cell variable

---

### PV Get cell variable(エリア; カラム; 行) → 文字列

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
戻り値	文字列	← 関連付けられた変数の名前

### 説明

**PV Get cell variable** 関数は、<カラム>と<行>で指定されたセルに関連付けられている変数の名前を返します。

### 例題

**PV SET CELL VARIABLE** コマンドの例題を参照してください。

### 参照

PV Get cell value type、PV SET CELL VARIABLE

## PV SET CELL FIELD

### PV SET CELL FIELD(エリア; カラム; 行; テーブル; フィールド)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
テーブル	整数	→ テーブル番号
フィールド	整数	→ フィールド番号

#### 説明

**PV SET CELL FIELD** コマンドは、<カラム>と<行>との交差する場所に相当する座標を持つセルエリアにフィールドを関連付けます。

<テーブル>は、セルに関連付けたいカレントレコードが属するテーブル番号です。カレントレコードの<フィールド>の値が表示されます。

セルは<フィールド>と動的に関連付けられます。つまり、この値に何らかの変更が加えられると、カレントレコードのフィールドにもその変更が自動的に反映されます（その逆も同様）。

#### 例題

4D Viewのセルで構成される入力フォームを作成する場合について考えてみます。**PV SET CELL FIELD** コマンドを使用し、各セルでは関連付けられたフィールドの表示や変更を行うことができます。繰り返し入力を行うために、コールバックメソッドや他のフォームオブジェクトを用いてレコードを処理しても構いません。

**C\_INTEGER(\$Table;\$Field)** ` 関連付けられたフィールドの参照番号

**If**(Form event=On Load)

**PV GET CELL FIELD**(Area;2;1;\$Table;\$Field) ` フィールドが B1 に  
関連付けられているか?

**If**(\$Table=0) & (\$Field=0)

⇒ **PV SET CELL FIELD**(Area;2;1;1;4) ` B1: [顧客]名字

**End if**

**PV GET CELL FIELD**(Area;2;2;\$Table;\$Field) ` フィールドが B2 に  
関連付けられているか?

**If**(\$Table=0) & (\$Field=0)

⇒ **PV SET CELL FIELD**(Area;2;2;1;3) ` B2: [顧客]名前

**End if**

` ラベル

**PV SET CELL STRING VALUE**(Area;1;1;"名字 :")

```
PV SET CELL STRING VALUE(Area;1;2;"名前:")  
CREATE RECORD([顧客])`新しい顧客が入力された  
End if
```

参照

PV ADD DYNAMIC FIELDS、PV GET CELL FIELD、PV Get cell value type、PV SET CELL VARIABLE

## PV GET CELL FIELD

### PV GET CELL FIELD(エリア; カラム; 行; テーブル; フィールド)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
テーブル	整数	← テーブル番号
フィールド	整数	← フィールド番号

### 説明

**PV GET CELL FIELD** コマンドは、<カラム>と<行>で指定されたセルに関連付けられているテーブル番号とフィールド番号を引数<テーブル>と<フィールド>に納めて返します。

### 例題

(1)**PV SET CELL FIELD** コマンドの例題を参照してください。

(2)以下の例題は、メソッド内で**PV GET CELL FIELD** コマンドを使用し、カラムヘッダのクリック時にダイナミックカラムの並べ換えを実現する方法を示しています。このエリアに含まれているのは、ダイナミックカラムだけです。まず、エリアでクリックが行われた場合に呼び出されるコールバックメソッド、EventMethodをインストールします。

```
PV ON EVENT(area;pv on clicked;"EventMethod")
```

この命令は、エリアで行われるクリックをすべて捉えます。メソッドEventMethodは、カラムヘッダ上でのクリックを検出し、その結果としてデータの並べ換えを行います。

```

`EventMethodメソッド
C_BOOLEAN($0)
C_LONGINT($1;$2;$3;$4;$5;$6)
C_INTEGER($tableNum;$fieldNum)
If($5=0) `ヘッダ上でクリックが行われた場合
    $0:=True `イベントをキャンセルする
    PV GET CELL FIELD(area;$4;1;$tableNum;$fieldNum) `並べ替えるデータ
    ORDER BY(Table($tableNum)->;Field($tableNum;$fieldNum)->;>)
    `4Dデータの並べ替え
End if `カラム内の関連付けられた値は自動的に並べ替えられる

```

### 参照

PV Get cell value type、PV SET CELL FIELD

## PV SET CELL FORMULA

---

### PV SET CELL FORMULA(エリア; カラム; 行; フォーミュラ)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
フォーミュラ	文字列	→ フォーミュラ

### 説明

**PV SET CELL FORMULA** コマンドは、<カラム>と<行>で指定されたセルに<フォーミュラ>を設定します。

### 例題

次の例題は、ある数値タイプのセルから開始して、下方向への増分を行います。開始セルの値が後で変更されても、**PV SET CELL FORMULA** を使用してそのセルに設定したフォーミュラによって、この増分値は自動的に更新されます。

**C\_INTEGER**(\$CopyNumber) `コピーする回数

**C\_LONGINT**(\$Column;\$Row) `コピーするセルの座標

**C\_INTEGER**(\$Index) `ループの指標

**C\_TEXT**(\$Name) `開始セル名

**PV GET CURRENT CELL**(Area;\$Column;\$Row)

`タイプの確認

**If**(**PV Get cell value type**(Area;\$Column;\$Row)=**pv number type value**)

\$CopyNumber:=**Num**(**Request**("下方向に増分を行いたいセルの数は?";"5"))

`デフォルト値は5

**If**(\$CopyNumber>0) `確定

\$Name:=**PV Get cell name**(Area;\$Column;\$Row) `セル名の取得

**If**(\$Name="") `名前が付けられていない?

\$Name:="COL"+String(\$Column)+"RW"+String(\$Row)

`名前を付ける

**PV SET CELL NAME**(Area;\$Column;\$Row;\$Name)

`"COL2RW3" タイプの名前

**End if**

**For**(\$Index;\$Row+1;\$Row+\$CopyNumber)

`\$CopyNumber回数をループする

⇒

**PV SET CELL FORMULA**(Area;\$Column;\$Index;"=

+\$Name+" "+String(\$Index-\$Row)) `増分

**End for**

**End if**

Else`正しくないタイプ

ALERT("開始セルは数値タイプでなければなりません。")

End if

参照

PV Get cell formula、PV Get cell value type

## PV Get cell formula

---

**PV Get cell formula**(エリア; カラム; 行) → 文字列

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
戻り値	文字列	← フォーミュラ

説明

**PV Get cell formula**関数は、<カラム>と<行>で指定されたセルに設定されているフォーミュラを返します。

例題

次のメソッドは、あるセルに対して、そのひとつ上のセルに設定されているフォーミュラを再コピーします。

**C\_LONGINT**(\$Column;\$Row) `再コピーするセルの座標

**C\_TEXT**(\$Formula) `再コピーするフォーミュラ

**PV GET CURRENT CELL**(Area;\$Column;\$Row)

⇒ \$Formula:=**PV Get cell formula**(Area;\$Column;\$Row-1) `1つ上のフォーミュラを  
取得

**PV SET CELL FORMULA**(Area;\$Column;\$Row;\$Formula) `再コピー

参照

PV Get cell value type、PV SET CELL FORMULA

## PV SET CELL CONTROL

---

### PV SET CELL CONTROL(エリア; カラム; 行; タイプ; 配列名; メソッド; タイトル)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
タイプ	整数	→ 制御タイプ
配列名	文字列	→ 制御管理配列の名前
メソッド	文字列	→ コールバックメソッド名
タイトル	文字列	→ 制御のタイトル

### 説明

**PV SET CELL CONTROL** コマンドは、<カラム>と<行>で指定されたセルに“制御”を設定します。制御とは、セルに納められるボタンやチェックボックス、ラジオボタン、ドロップダウンリスト、コンボボックスタイプのオブジェクトのことです。セルに制御が既に組み込まれている場合には、タイプに関わらずそれが削除され、新しい制御で置き換えられます。

<タイプ>は、セルに納められる制御タイプ（上記5つのタイプの1つ）を示します。この引数を設定するには、**PV Control** テーマ内の定数を使用します。

<配列名>は、表示する値（ドロップダウンリストやコンボボックスの制御タイプ用）を格納した配列の名前です。

<メソッド>には、設定する制御に関連付けるコールバックメソッドの名前を指定します。コールバックメソッドに関する詳細は、「**PV Area**、はじめに」の節を参照してください。

<タイトル>には、ボタンやチェックボックス等のラベルを指定します。

### 例題

以下の各メソッドにより、さまざまなタイプの制御を具体的に示します。

- ・ C3セルに標準ボタン：

```
C_LONGINT(vStandardButton) `標準ボタン
⇒ PV SET CELL CONTROL(Area;3;3;pv control button;"vStandardButton";
                                                                    "CallbackButton";"Button")
                                                                    `CallbackButton メソッド
C_LONGINT($1) `4D View エリア
C_LONGINT($2) `カラム番号
```

**C\_LONGINT(\$3)** `行番号

**C\_POINTER(\$4)** `オブジェクトをコールするポインタ

**ALERT("メソッドからの制御："+Current method name)**

- ・ E3、E4、E5セルにラジオボタン：

**C\_LONGINT(vRadio1;vRadio2;vRadio3)** `ラジオボタン

vRadio1:=1

vRadio2:=0

vRadio3:=0

- ⇒ **PV SET CELL CONTROL**(Area;5;3;pv control radio button;"vRadio1";"Callback Radio";"日")
- ⇒ **PV SET CELL CONTROL**(Area;5;4;pv control radio button;"vRadio2";"Callback Radio";"月")
- ⇒ **PV SET CELL CONTROL**(Area;5;5;pv control radio button;"vRadio3";"Callback Radio";"年")

    `CallbackRadio メソッド

**C\_LONGINT(\$1)** `4D View エリア

**C\_LONGINT(\$2)** `カラム番号

**C\_LONGINT(\$3)** `行番号

**C\_POINTER(\$4)** `オブジェクトをコールするポインタ

**C\_STRING(6;\$Value)**

**Case of**

  \**(\$4=(->vRadio1))**

    vRadio2:=0

    vRadio3:=0

    \$Value:="Day"

  \**(\$4=(->vRadio2))**

    vRadio1:=0

    vRadio3:=0

    \$Value:="Month"

  \**(\$4=(->vRadio3))**

    vRadio1:=0

    vRadio2:=0

    \$Value:="Year"

**End case**

**PV SET CELL TEXT VALUE(\$1;(\$2+1);3;\$Value)**

- ・ E7、E8、E9セルにチェックボックス：

**C\_LONGINT**(vCheckBox1;vCheckBox2;vCheckBox3) `チェック用のボックス

vCheckBox1:=0

vCheckBox2:=0

vCheckBox3:=0

- ⇒ **PV SET CELL CONTROL**(Area;5;7;pv control check box;"vCheckBox1";  
"CallbackCheckBox";"初心者")
- ⇒ **PV SET CELL CONTROL**(Area;5;8;pv control check box;"vCheckBox2";  
"CallbackCheckBox";"中級")
- ⇒ **PV SET CELL CONTROL**(Area;5;9;pv control check box;"vCheckBox3";  
"CallbackCheckBox";"上級")

`CallbackCheckBox メソッド

**C\_LONGINT**(\$1) `4D View エリア

**C\_LONGINT**(\$2) `カラム番号

**C\_LONGINT**(\$3) `行番号

**C\_POINTER**(\$4) `オブジェクトをコールするポインタ

**C\_TEXT**(\$Value)

**If**(vCheckBox1=1)

\$Value:="初心者"

**End if**

**If**(vCheckBox2=1)

**If**(\$Value#"" )

\$Value:=\$Value+" + "

**End if**

\$Value:=\$Value+"中級"

**End if**

**If**(vCheckBox3=1)

**If**(\$Value#"" )

\$Value:=\$Value+" + "

**End if**

\$Value:=\$Value+"上級"

**End if**

**PV SET CELL TEXT VALUE**(\$1;(\$2+1);7;\$Value)

- ・ C5セルにドロップダウンリスト：

**ARRAY TEXT**(DropDownListArray;5) `ドロップダウンリスト用

DropDownListArray{1}:="月曜日"

DropDownListArray{2}:="火曜日"

DropDownListArray{3}:="水曜日"

DropDownListArray{4}:="木曜日"

DropDownListArray{5}:="金曜日"

DropDownListArray:=3`水曜日がデフォルト

⇒ **PV SET CELL CONTROL**(Area;3;5;pv control drop down;"DropDownListArray";  
"CallbackDropList";")

    `CallbackDropList メソッド

**C\_LONGINT**(\$1)`4D View エリア

**C\_LONGINT**(\$2)`カラム番号

**C\_LONGINT**(\$3)`行番号

**C\_POINTER**(\$4)`オブジェクトをコールするポインタ

**PV SET CELL TEXT VALUE**(\$1;3;6;DropDownListArra{DropDownListArray})

・ C7セルにコンボボックス :

**ARRAY TEXT**(ComboArray;5)`コンボボックスのリスト用

ComboArray{1};="月曜日"

ComboArray{2};="火曜日"

ComboArray{3};="水曜日"

ComboArray{4};="木曜日"

ComboArray{5};="金曜日"

ComboArray:=5`水曜日がデフォルト

⇒ **PV SET CELL CONTROL**(Area;3;7;pv control combo box;"ComboArray";  
"CallbackCombo";")

    `メソッド : CallbackCombo

**C\_LONGINT**(\$1)`4D View エリア

**C\_LONGINT**(\$2)`カラム番号

**C\_LONGINT**(\$3)`行番号

**C\_POINTER**(\$4)`オブジェクトをコールするポインタ

**PV SET CELL TEXT VALUE**(\$1;3;8;ComboArray{0})

参照

PV GET CELL CONTROL、PV Get cell value type

定数

PV Control テーマ

## PV GET CELL CONTROL

---

### PV GET CELL CONTROL(エリア; カラム; 行; タイプ; 名前; メソッド; タイトル)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ セルのカラム番号
行	倍長整数	→ セルの行番号
タイプ	整数	← 制御タイプ
名前	文字列	← 制御名
メソッド	文字列	← コールバックメソッド名
タイトル	文字列	← 制御のタイトル

### 説明

**PV GET CELL CONTROL** コマンドは、4D View エリア内の<カラム>と<行>で指定されたセルに表示される“制御”に関する情報を取得します。

制御とは、セルに納められたボタンやチェックボックス、ラジオボタン、ドロップダウンリスト、コンボボックスタイプのオブジェクトのことです。

<タイプ>は、セルに納められる制御タイプ（上記5つのタイプの1つ）を示します。この引数を設定するには、PV Control テーマ内の定数を使用します。

<名前>は、表示する値（ドロップダウンリストやコンボボックスの制御タイプ用）を格納した配列の名前です。

<メソッド>には、制御に関連付けられたコールバックメソッドの名前が納められます。

<タイトル>には、ボタンやチェックボックス等のラベルが納められます。

### 例題

次のメソッドは、制御セルの下方向にあるセルに、制御に関する説明を代入します。

```
C_LONGINT($Column;$Row) ` “制御される” セルの座標  
C_INTEGER($CtrlType) ` 制御タイプ  
C_STRING(32;$CtrlName) ` 制御の名前 ( ) Name(eventual) of control  
C_STRING(32;$CallbackMethod) ` コールバックメソッド名  
C_STRING(32;$Title) ` 制御のタイトル  
C_STRING(32;$TypeName) ` 制御タイプの名前  
PV GET CURRENT CELL(Area;$Column;$Row) ` 制御されるセル  
⇒ PV GET CELL CONTROL(Area;$Column;$Row;$CtrlType;$CtrlName;  
$CallbackMethod;$Title)
```

### Case of

```
\($CtrlType=0)
    $TypeName:="制御未設定"
\($CtrlType=pv control push button)
    $TypeName:="ボタン"
\($CtrlType=pv control radio button)
    $TypeName:="ラジオボタン"
\($CtrlType=pv control check box)
    $TypeName:="チェックボックス"
\($CtrlType=pv control drop down)
    $TypeName:="ドロップダウンリスト"
\($CtrlType=pv control combo box)
    $TypeName:="コンボボックス"
```

**End case**

```
PV SET CELL TEXT VALUE(Area,$Column,$Row+1;"タイプ : "+$TypeName)
```

```
PV SET CELL TEXT VALUE(Area,$Column,$Row+2;"名前 : "+$CtrlName)
```

```
PV SET CELL TEXT VALUE(Area,$Column,$Row+3;"タイトル : "+$Title)
```

```
PV SET CELL TEXT VALUE(Area,$Column,$Row+4;"メソッド : "
+$CallbackMethod)
```

## 参照

PV Get cell value type、PV SET CELL CONTROL

## 定数

PV Control テーマ

## PV ARRAY TO CELLS

---

### PV ARRAY TO CELLS(エリア; 方向; カラム; 行; 変換; 配列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
方向	整数	→ 0= 行 ; 1= カラム
カラム	倍長整数	→ 開始カラム番号
行	倍長整数	→ 開始行番号
変換	整数	→ 0= 元のタイプ ; 1= テキストに変換
配列	配列	→ 配列名

### 説明

**PV ARRAY TO CELLS** コマンドは、配列<配列>に指定された名前の配列の内容を<エリア>に挿入します。この配列内容は、<カラム>と<行>で指定された座標（セル）から<方向>で定義した方向へ挿入されます。

注：このコマンドの引数<方向>の使い方は、PV GOTO NEXT CELL コマンドおよび PV GET NEXT FREE CELL コマンドとは異なります。これら2つのコマンドでは、<方向>としていずれの方向でも指定できますが、PV ARRAY TO CELLS コマンドでは<方向>として、右方向（0= 行）または下方向（1= カラム）のいずれかのみを指定します。

5番目の引数<変換>を使用すると、セル内容のデータタイプをテキストタイプに変換することができます。ただし、元の値のタイプは、この処理に対応するタイプでなければいけません。BLOBやピクチャデータタイプをテキストに変えるような指定を行うと、変換は行われません。

### 例題

カレントセルより開始して、下方向へ異なるタイプの3つの配列を再コピーします。タイプ変換の確認用ダイアログボックスに対する応答に応じて、配列内容を4D Viewのセルへコピーする際に（テキストへの）変換を行うかどうかを決定します。

```
C_INTEGER($Index) ` ループの指標
C_LONGINT($Column;$Row) ` 開始セルの座標
C_INTEGER($Conversion) ` テキストへの変換を行うか？
ARRAY STRING(20;$ArrayString;10) ` 文字列の値の配列（カレントセルより開始）
ARRAY DATE($ArrayDates;10) ` 日付配列（次のカラム）
ARRAY BOOLEAN($ArrayBooleans;10) ` ブール配列（3番目のカラム）
    ` 初期化
For($Index;1;Size of array($ArrayString))
    $ArrayString{$Index}:=String($Index*10) ` "10", "20", "30"...
    $ArrayDates{$Index}:=Current date(*)+$Index ` 02.08.16、02.08.17...
```

```
$ArrayBooleans{$Index}:=(($Index%2=0) `True= 偶数
```

```
End for
```

```
PV GET CURRENT CELL(Area;$Column;$Row)
```

```
CONFIRM("配列内容をテキストに変換しますか?")
```

```
$Conversion:=OK
```

```
`カラム "$ArrayString"、"$ArrayDates"、"$ArrayBooleans" を取得
```

```
`1= 下方向へ
```

```
⇒ PV ARRAY TO CELLS(Area;1;$Column;$Row;$Conversion;$ArrayString)
```

```
⇒ PV ARRAY TO CELLS(Area;1;$Column+1;$Row;$Conversion;$ArrayDates)
```

```
⇒ PV ARRAY TO CELLS(Area;1;$Column+2;$Row;$Conversion;$ArrayBooleans)
```

参照

PV CELLS TO ARRAY、PV FIELD TO CELLS

## PV CELLS TO ARRAY

---

### PV CELLS TO ARRAY(エリア; 方向; 軸; 開始位置; 配列; セル数)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
方向	整数	→ 0= 行 ; 1= カラム
軸	倍長整数	→ 行番号またはカラム番号
開始位置	倍長整数	→ 開始セル番号
配列	配列	→ 配列名
セル数	倍長整数	→ セル数

### 説明

**PV CELLS TO ARRAY** コマンドは、<軸>、<方向>、<開始位置>、<セル数>で指定されたセル内容を<配列>に代入します。

<方向>は、**PV CELLS TO ARRAY** コマンドにより、連続するセルのコピーを横方向 (0) または縦方向 (1) のいずれの方向に対して行うかを指定します。<方向>に指定した値に応じて、引数<軸>には行番号またはカラム番号を設定します。

注：このコマンドの引数<方向>の使い方は、**PV GOTO NEXT CELL** コマンドおよび **PV GET NEXT FREE CELL** コマンドとは異なります。これら2つのコマンドでは、<方向>としていずれの方向でも指定できますが、**PV CELLS TO ARRAY** コマンドでは<方向>として、右方向 (0= 行) または下方向 (1= カラム) のいずれかのみを指定します。

## 例題

4D View エリアが顧客テーブルの入力画面に組み込まれている例について考えてみましょう。この組み込みエリアの行数は最大で10行で、顧客との交渉の入力や変更のために使われます。入力エリアは10個のセルで構成され、カレントセルからカラム上に配置されています。

```

ARRAY STRING(255;ContactsTab;0) ` 交渉名の配列
C_LONGINT($Column;$Row) ` 開始セルの座標
PV GET CURRENT CELL(Area;$Column;$Row)
    ` カレントセルから開始して、下方向へ最大10行
⇒ PV CELLS TO ARRAY(Area;1;$Column;$Row;ContactsTab;10)
If(Size of array(ContactsTab)#0) ` 対象となるレコードがあるか？
    RELATE MANY([Clients]Code) ` リンク先レコードを取得
    DELETE SELECTION([Contacts]) ` 既存のレコードを消去
        ` 交渉の更新（新規、変更、削除）
    ARRAY TO SELECTION(ContactsTab;[Contacts]ContactName)
        ` 交渉レコードの作成
        ` リレートを維持
    APPLY TO SELECTION([Contacts];[Contacts]CodeClient:=[Clients]Code)
    QUERY SELECTION([Contacts];[Contacts]ContactName='')
        ` 空のレコードを削除

    DELETE SELECTION([Contacts])
End if

```

## 参照

PV ARRAY TO CELLS、PV FIELD TO CELLS

## PV FIELD TO CELLS

---

**PV FIELD TO CELLS**(エリア; 方向; カラム; 行; 変換; マスター; テーブル; フィールド)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
方向	整数	→ 0= 行; 1= カラム
カラム	倍長整数	→ 開始セルのカラム番号
行	倍長整数	→ 開始セルの行番号
変換	整数	→ 0= 元のタイプ; 1= テキストに変換
マスター	整数	→ マスターテーブル番号
テーブル	整数	→ テーブル番号
フィールド	整数	→ フィールド番号

### 説明

**PV FIELD TO CELLS** コマンドは、<マスター>テーブルのカレントセクションに対応しているフィールドの値を<エリア>に挿入します。挿入は、<カラム>と<行>で指定された座標（セル）から<方向>で定義した方向へ行われます。

注：このコマンドの引数<方向>の使い方は、PV GOTO NEXT CELL コマンドおよび PV GET NEXT FREE CELL コマンドとは異なります。これら2つのコマンドでは、<方向>としていずれの方向でも指定できますが、PV FIELD TO CELLS コマンドでは<方向>として、右方向（0=行）または下方向（1=カラム）のいずれかのみを指定します。

5番目の引数<変換>を使用すると、セル内容をテキストタイプに変換することができます。ただし、元の値のタイプは、この処理に対応するタイプでなければなりません。BLOBやピクチャデータタイプをテキストに変えるような指定を行うと、変換は行われません。

このコマンドは、リレートを用いて、マスターテーブルにリンクしているテーブルに対して適用することができます。このリレートは、<マスター>で定義したテーブルから<テーブル>で指定したテーブルへの自動リレート（タイプ）でなければなりません。表示する<フィールド>は<テーブル>に属しています。請求書と請求明細に関する従来の例題では、明細テーブルからフィールド内容を回復することができますが、請求書テーブル明細（マスターテーブル）のセクションにリレートしている請求書テーブルからも、これを回復することができます。

**PV SET CELL FIELD** コマンドは、データベースとの動的なりレートを維持しますが、このコマンドとは異なり、**PV FIELD TO CELLS** コマンドを使用して回復を行った後に4D View エリアで値が変更されても、レコードの内容には反映されません。

## 例題

**PV CELLS TO ARRAY** コマンドの例題において、顧客の入力フォームに組み込まれた4D View エリア内で交渉データが入力されると、交渉テーブルの更新を行いました。ここでは、「On Load」フォームイベント、つまり修正される入力フォームのロード時に、この組み込みエリアの更新を行います。

マスターとなる「交渉」テーブルから「業務上肩書き」テーブル（社長、秘書、開発者）への自動リレートを使用して、各交渉に対する業務上肩書きのラベルの回復を行うことができます。「業務上肩書き」テーブルの特別なフィールドに整数として保存されている肩書きコード（リレートの開始点となるフィールド）を元にしてこの処理を行います。2つのカラムに交渉名とその肩書きを表示します。

```

C_INTEGER($Master) ` マスターテーブル番号
C_INTEGER($Table) ` テーブル番号
C_INTEGER($Field) ` フィールド番号
C_LONGINT($Column;$Row) ` 開始セルの座標
$Master:=Table(->[Contacts]) ` マスターテーブル番号：交渉（2つのカラム用）
RELATE MANY([Clients]code) ` セレクションから対応する交渉を取得
PV GET CURRENT CELL(Area;$Column;$Row)
    `4D View エリア上の名前を更新
$Table:=Table(->[Contacts]) ` 交渉テーブルのテーブル番号
    `内容を回復するフィールド番号
$Field:=Field(->[Contacts]NomDuContact)
    `変換は不要、文字タイプの回復を行う
⇒ PV FIELD TO CELLS(Area;1;$Column;$Row;0;$Master;$Table;$Field)
    `4D View エリア の肩書きを更新
$Table:=Table(->[Titres]) ` 肩書きタイプ（リレート先）テーブルの番号
$Field:=Field(->[Titres]Label) ` 内容を回復するフィールドの番号
⇒ PV FIELD TO CELLS(Area;1;$Column+1;$Row;0;$Master;$Table;$Field)

```

## 参照

PV ARRAY TO CELLS、PV FIELDS LIST TO CELLS、PV SET CELL FIELD

## PV FIELDS LIST TO CELLS

---

**PV FIELDS LIST TO CELLS**(エリア; 方向; カラム; 行; 変換; マスター; テーブル; フィールド)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
方向	整数	→ 0= 行; 1= カラム
カラム	配列	→ 開始セルのカラム番号の配列
行	配列	→ 開始セルの行番号の配列
変換	配列	→ 0= 元のタイプ; 1= テキストに変換
マスター	整数	→ マスターテーブル番号
テーブル	配列	→ テーブル番号
フィールド	配列	→ フィールド番号

### 説明

**PV FIELDS LIST TO CELLS** コマンドは、**PV FIELD TO CELLS** コマンドと同じ働きをしますが、複数のフィールドを使用します。

注：このコマンドの引数<方向>の使い方は、**PV GOTO NEXT CELL** コマンドおよび **PV GET NEXT FREE CELL** コマンドとは異なります。これら2つのコマンドでは、<方向>としていずれの方向でも指定できますが、**PV FIELDS LIST TO CELLS** コマンドでは<方向>として、右方向 (0= 行) または下方向 (1= カラム) のいずれかのみを指定します。

引数<カラム>および<行>には、開始セルのカラム番号と行番号をそれぞれ指定します。

5番目の引数<変換>を使用すると、セル内容をテキストタイプに変換することができます。ただし、元の値のタイプは、この処理に対応するタイプでなければいけません。BLOBやピクチャタイプをテキストに変えるような指定を行うと、変換は行われません。

配列である<テーブル>と<フィールド>には、テーブル番号および元になるフィールドの番号を納めます。

このコマンドは、リレートを用いて、リレートテーブルやマスターテーブルに対し適用することができます。このリレートは、<マスター>テーブルから表示フィールドの属するテーブルへの自動リレートでなければなりません。表示するテーブルとフィールドは配列<テーブル>と<フィールド>で定義します。

**PV ADD DYNAMIC FIELDS** コマンドは、データベースとの動的なりレートを維持しますが、このコマンドとは異なり、**PV FIELDS LIST TO CELLS** コマンドを使用して回復を行った後に4D View エリアで値が変更されても、レコードの内容には反映されません。

## 例題

ここでも **PV CELLS TO ARRAY** コマンドの例題を使用して、顧客テーブルのレコードをロードする入力フォームについて見てみましょう。今回は、3つのフィールドを同時に処理します。

マスターとなる「交渉」テーブルから「業務上肩書き」テーブル（社長、秘書、開発者）への自動リレートを使用して、「業務上肩書き」テーブルの特別なフィールドに整数として保存されている肩書きコード（リレートの開始点となるフィールド）を元に、各交渉に対する肩書きのラベルの回復を行うことができます。さらに、関連する3つ目のカラムを表示します（カラムCより開始するため、カラムEに表示する）。

```

C_INTEGER($Master) ` マスターテーブル番号
C_INTEGER($Index) ` ループの指標
$Master:=Table(->[Contacts]) ` マスターテーブル番号：交渉（全カラム用）
ARRAY LONGINT($ArrayColumns;3) ` 開始セルのカラム番号
ARRAY LONGINT($ArrayRows;3) ` 開始セルの行番号
ARRAY INTEGER($ConversionArray;3) ` 0：元のタイプ、または
                                         1：テキストへ変換

ARRAY INTEGER($TablesArray;3) ` テーブル番号
ARRAY INTEGER($FieldsArray;3) ` フィールド番号
For($Index;1;3)
    $ArrayColumns{$Index}:= $Index+2 ` カラム C から E
    $ArrayRows{$Index}:= 2 ` 2行目より開始
    $ConversionArray{$Index}:= 0 ` 元のフィールドタイプを維持
End for
    ` 取得するテーブルとフィールド
$TablesArray{1}:=Table(->[Contacts]) ` 交渉テーブルの番号
$FieldsArray{1}:=Field(->[Contacts]ContactName) ` 名字フィールドの番号
$TablesArray{2}:=Table(->[Contacts]) ` 交渉テーブルの番号
$FieldsArray{2}:=Field(->[Contacts]ContactFirstName) ` 名前フィールドの番号
$TablesArray{3}:=Table(->[Titles]) ` 肩書きタイプテーブル（リレート先）の番号
    ` 肩書きフィールドの番号（社長、秘書、開発者）
$FieldsArray{3}:=Field(->[Titles]Label)
    ` 対応する「交渉」のセレクションを取得
RELATE MANY([Clients]code)
    ` 4D View エリア上の名字、名前、肩書きフィールドを更新
⇒ PV FIELDS LIST TO CELLS(Area;1;$ArrayColumns;$ArrayRows;
    $ConversionArray;$Master;$TablesArray;$FieldsArray)

```

## 参照

PV ADD DYNAMIC FIELDS、PV ARRAY TO CELLS、PV FIELD TO CELLS

## PV REPORT MANY

---

**PV REPORT MANY**(エリア; カラム; 行; マスター; テーブルブレイク; フィールドブレイク; オペレータ; テーブル; フィールド; 挿入; 明細; タイトル)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ カラム番号
行	倍長整数	→ 行番号
マスター	整数	→ プライマリテーブル番号
テーブルブレイク	整数	→ ブレイクが発生するテーブル番号
フィールドブレイク	配列	→ ブレイクが発生するフィールド番号
オペレータ	整数配列	→ 実行する処理
テーブル	整数配列	→ 表示フィールドの属するテーブルの番号
フィールド	整数配列	→ 表示フィールドの番号
挿入	整数	→ 0=置き換え; 1=行挿入
明細	整数	→ 明細行挿入オプション
タイトル	文字列	→ ブレイクタイトル

### 説明

**PV REPORT MANY** コマンドは、<マスター>テーブルのカレントセレクションから開始して、<テーブル>と<フィールド>で指定した4th Dimension フィールドの値を使用し、4D View<エリア>上にレポートを作成します。各フィールドに適用する処理は、<オペレータ>で定義します（合計、カウント、最大値等）。引数<オペレータ>を定義するには、**PV Report functions** テーマ内の定数を使用します。

生成されたレポートは、<カラム>と<行>で指定したセルより<エリア>上に挿入されます。

<テーブルブレイク>と<フィールドブレイク>は、ブレイクシーケンスを実行するフィールドを指定します。ブレイクにより、同種類のグループごとにレコードを区切り、各グループの中間集計を行うことができます。

ブレイクとは、テーブルの並び替えにおける値の変わり目です。このブレイクレベルは、並び替えの条件に関連付ける必要があります。**PV REPORT MANY** コマンドを使用する前に、<テーブルブレイク>と<フィールドブレイク>で指定したレコードセレクションに対する並び替えを行う必要があります。その際、ブレイクとして考慮されない並び替え条件を更に追加しても構いません。ブレイクをとまなうクイックレポートの作成に関する詳細は、4th Dimensionの『ユーザリファレンス』マニュアルを参照してください。

引数<挿入>は、このコマンドを使用して4D Viewエリアに配置されるデータで既存のセルを置き換えるか、もしくは新しい行に挿入するかを決定します。この引数に0を渡すと、データは消去されて置き換えられます。1を渡した場合、追加の行が挿入されます。

引数<明細>を使用すると、4D Viewが明細行およびブレイク行をレポートに挿入する方法を定義することができます。

- <明細>に0を渡した場合、ブレイク行だけが挿入されます（明細行は挿入されません）。
- <明細>に1を渡した場合、明細行が挿入され、ブレイクの値は行ごとに繰り返し表示されます。
- <明細>に2を渡した場合、明細行が挿入されますが、ブレイクの値は一度しか表示されません。

引数<タイトル>を使用して、結果の行のタイトルを設定することができます（つまり、「合計」や「平均」等です）。4Dの「クイックレポート」エディタと同様に、現在のブレイク値のプレースホルダとして数値記号“#”を使用可能です。例えば、「国」フィールドがブレイクフィールドである場合、タイトルには“合計：#”と指定することができます。すると、レポートには“合計：合衆国”、“合計：日本”というように表示されます。

タイトルを挿入したくない場合には、空の文字列を渡します。

注：<明細>に0を指定し、<タイトル>に空の文字列を渡した場合、最初のコラムは挿入されません（コラムは空の状態になります）。

## 例題

「顧客」の入力フォーム内に組み込まれた4D Viewエリアに、その顧客にリンクしている「交渉」のレポートを表示します。肩書きごとに（秘書、開発者、ドキュメント部門等）、交渉数を示すブレイクを追加します。「交渉」テーブルと「顧客」テーブル間には自動リレートが設定されています。

**C\_LONGINT(\$Column)** ` コラム番号  
**C\_LONGINT(\$Row)** ` 行番号  
**C\_INTEGER(\$Master)** ` マスターテーブル番号  
**C\_INTEGER(\$TableBreak)** ` ブレイクが発生するテーブル番号  
**C\_INTEGER(\$FieldBreak)** ` ブレイクが発生するフィールド番号  
**ARRAY INTEGER(\$Operator;3)** ` 実行する処理  
**ARRAY INTEGER(\$Tables;3)** ` 表示フィールドの属するテーブルの番号  
**ARRAY INTEGER(\$Fields;3)** ` 表示フィールドの番号  
**C\_INTEGER(\$Insert)** ` 0=置き換え；1=行を挿入  
**C\_INTEGER(\$Detail)** ` 明細行挿入オプション

```

C_STRING(30;$Title) ` 結果行のタイトル
    ` 初期化
$Column:=4 ` カラム D から表示を開始する
$Row:=3 ` 3行目から表示を開始する (タイトル + 空の行)
$Master:=Table(->[Contacts]) ` これは「交渉」テーブルの“レポート”である
$TableBreak:=Table(->[Titles])
$FieldBreak:=Field(->[Titles]Label) ` 「交渉」肩書き上でブレイクが発生

$Operator{1}:=pv report function count ` 1カラム目に各タイプの数を表示
$Tables{1}:=Table(->[Titles])
$Fields{1}:=Field(->[Titles]Label) ` 1カラム目に肩書きのラベル

$Operator{2}:=pv report function none ` カラム2における計算は行わない
$Tables{2}:=Table(->[Contacts])
$Fields{2}:=Field(->[Contacts]ContactName) ` 2カラム目に交渉名

$Operator{3}:=pv report function none ` カラム3における計算は行わない
$Tables{3}:=Table(->[Contacts])
$Fields{3}:=Field(->[Contacts]ContactFirstname) ` 3カラム目に交渉の名前

$Insert:=1 ` 挿入
$Detail:=2 ` 明細行を挿入し、値は1度だけ表示する
$Title:="#の交渉数" ` “#”記号は現在のブレイク値で置き変わる
RELATE MANY([Clients]Code) ` 顧客の交渉を取得
    ` ブレイク点でのソートが必要 + アルファベット順に表示
ORDER BY([Contacts];[Titles]Label;[Contacts]ContactName;
            [Contacts]ContactFirstname)
⇒ PV REPORT MANY(Area;$Column;$Row;$Master;$TableBreak;$FieldBreak;
                $Operator;$Tables;$Fields;$Insert;$Detail;$Title)

```

参照

PV ARRAY TO CELLS、PV REPORT ONE

定数

PV Report functions テーマ

## カレントセクションとレコード

セクションは、データベースのストラクチャレベルでのテーブル間のリレート、引数としてコマンドに渡されたフィールド番号と配列によって決定します。

## PV REPORT ONE

**PV REPORT ONE**(エリア; カラム; 行; マスター; テーブルブレイク; フィールドブレイク; オペレータ; テーブル; フィールド; 挿入; 明細; タイトル)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ カラム番号
行	倍長整数	→ 行番号
マスター	整数	→ プライマリテーブル番号
テーブルブレイク	整数	→ ブレイクが発生するテーブル番号
フィールドブレイク	整数	→ ブレイクが発生するフィールド番号
オペレータ	整数	→ 実行する処理
テーブル 番号	整数	→ 表示フィールドの属するテーブルの番号
フィールド	整数	→ 表示フィールドの番号
挿入	整数	→ 0= 置き換え; 1= 行挿入
明細	倍長整数	→ 明細行挿入オプション
タイトル	文字列	→ ブレイクタイトル

### 説明

**PV REPORT ONE** コマンドは、<マスター>テーブルのカレントセレクションから開始して、<テーブル>と<フィールド>で指定した4th Dimension フィールドの値を使用し、4D View<エリア>上にレポートを作成します。このフィールドに適用する処理は、<オペレータ>で定義します（合計、カウント、最大値等）。引数<オペレータ>を定義するには、**PV Report functions** テーマ内の定数を使用します。

生成されたレポートは、<カラム>と<行>で指定したセルより<エリア>上に挿入されます。

<テーブルブレイク>と<フィールドブレイク>は、ブレイクシーケンスを実行するフィールドを指定します。ブレイクにより、同種類のグループごとにレコードを区切り、各グループの中間集計を行うことができます。

ブレイクとは、テーブルの並び替えにおける値の変わり目です。このブレイクレベルは、並び替えの条件に関連付ける必要があります。**PV REPORT ONE** コマンドを使用する前に、<テーブルブレイク>と<フィールドブレイク>で指定したレコードセレクションに対する並び替えを行う必要があります。その際、ブレイクとして考慮されない並び替え条件を更に追加しても構いません。ブレイクをとまなうクイックレポートの作成に関する詳細は、4th Dimension の『ユーザーファレンス』マニュアルを参照してください。

引数<挿入>は、このコマンドを使用して4D Viewエリアに配置されたデータで既存のセルを置き換えるか、もしくは新しい行に挿入するかを決定します。この引数に0を渡すと、データは消去されて置き換えられます。1を渡した場合、追加の行が挿入されます。

引数<明細>を使用すると、4D Viewが明細行およびブレイク行をレポートに挿入する方法を定義することができます。

- <明細>に0を渡した場合、ブレイク行だけが挿入されます（明細行は挿入されません）。
- <明細>に1を渡した場合、明細行が挿入され、ブレイクの値は行ごとに繰り返し表示されます。
- <明細>に2を渡した場合、明細行が挿入されますが、ブレイクの値は一度しか表示されません。

引数<タイトル>を使用して、結果の行のタイトルを設定することができます（つまり、“合計”や“平均”等です）。4Dの「クイックレポート」エディタと同様に、現在のブレイク値のプレースホルダとして数値記号“#”を使用可能です。例えば、「国」フィールドがブレイクフィールドである場合、タイトルには“合計：#”と指定することができます。すると、レポートには“合計：合衆国”、“合計：日本”というように表示されません。タイトルを挿入したくない場合には、空の文字列を渡します。

注：<明細>に0を指定し、<タイトル>に空の文字列を渡した場合、最初のカラムは挿入されません（カラムは空の状態になります）。

## 例題

4D View エリアに顧客リストを表示し、男性と女性を分けて、それぞれのグループのレコード数を示します。

```
C_LONGINT($Column) `カラム番号
C_LONGINT($Row) `行番号
C_INTEGER($Master) `プライマリテーブル番号
C_INTEGER($TableBreak) `ブレイクが発生するテーブル番号
C_INTEGER($FieldBreak) `ブレイクが発生するフィールド番号
C_INTEGER($Operator) `実行する処理
C_INTEGER($Table) `表示フィールドの属するテーブルの番号
C_INTEGER($Field) `表示フィールドの番号
C_INTEGER($Insert) `0=置き換え；1=行を挿入
C_INTEGER($Detail) `明細行挿入オプション
C_STRING(20;$Title) `結果行のタイトル
    `初期化
$Column:=4 `カラムCから表示を開始する
$Row:=3 `3行目から表示を開始する（タイトル+空の行）
```

\$Master:=**Table**(->[Clients]) `「顧客」テーブルをすべて処理する

\$TableBreak:=**Table**(->[Clients])

\$FieldBreak:=**Field**(->[Clients]Type) `顧客タイプでブレイク

\$Operator:=**pv report function count** `男性と女性の数を集計

\$Table:=**Table**(->[Clients]) `「顧客」テーブルのフィールド

\$Field:=**Field**(->[Clients]Name) `名前を印刷

\$Insert:=1 `挿入を行う

\$Detail:=2 `明細行を挿入し、値は1度だけ表示する

\$Title:="合計"

**ALL RECORDS**([Clients])

`ブレイク点でのソートが必要 + アルファベット順に表示

**ORDER BY**([Clients];[Clients]Type;[Clients]Name)

⇒ **PV REPORT ONE**(Area;\$Column;\$Row;\$Master;\$TableBreak;\$FieldBreak;  
\$Operator;\$Table;\$Field;\$Insert;\$Detail;\$Title)

参照

PV ARRAY TO CELLS、PV REPORT MANY

定数

PV Report functions テーマ

## PV ADD DYNAMIC ARRAYS

---

### PV ADD DYNAMIC ARRAYS(エリア; 配列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
配列	文字列配列	→ 配列名を納めた配列

#### 説明

**PV ADD DYNAMIC ARRAYS** コマンドは、配列の値に対応するひとまとまりの行を<エリア>に追加します。この配列の名前は、引数<配列>を使用して渡し、A1セルより行の追加を開始します。

必ず各配列の項目数はすべて同数でなくてはならず、常に縦方向に（カラムとして）表示されます。最初の配列はカラムAに表示され、他の配列はその右側にあるカラムにそれぞれ表示されます。ダイナミック行は、1行目の上に挿入されます。

注：PV ADD DYNAMIC FIELDS コマンドを使用して、4D View エリアにダイナミックフィールドエリアが既に挿入されている場合、そのエリアは削除されダイナミック配列で置き換えられます。

**PV ADD DYNAMIC ARRAYS** コマンドは、エリアのダイナミックエリアに関連している配列との動的リンクを維持します。この結果、4D View エリアで値が変更されると、各配列に対してその変更が反映されます（この逆も同様）。4D 配列の項目を削除したり追加した場合も、エリアに反映されます。

1つの4D View エリア内に、ダイナミックエリアを複数設定することはできません。

注：

- 4D 配列および4D View エリアは同じ手順で定義しなくてはなりません。
- 4D の「ユーザ」モードと4D View のプラグインウィンドウとの間で、動的なデータ更新は行えません。4D の「ユーザ」モードで行われた修正内容を4D View ウィンドウに反映するには、PV REDRAW コマンドを使用してウィンドウをリドロウ（再描画）する必要があります（4D コマンドを用いて行った修正は4D View エリアに自動的に反映されます）。

#### 例題

次の例題では、顧客テーブルのカレントレコードにリレートする交渉の名前を取得して、配列に代入します。その後、修正される顧客の入力フォームに組み込まれた4D View エリアにこの配列内容を表示します。

ユーザーが変更を行うと、直接4Dの配列に反映されます。この次の処理として、例えば入力された交渉のセレクションの更新を行う必要があります。

**ARRAY TEXT**(NamesArray;0) ` 交渉名の配列

**ARRAY TEXT**(FirstnamesArray;0) ` 交渉の名前の配列

**ARRAY STRING**(31;\$ArrayArrays;2) ` 配列名の配列

\$ArrayArrays{1}:="NamesArray" ` 1番目のダイナミックカラム

\$ArrayArrays{2}:="FirstnamesArray" ` 2番目のダイナミックカラム

**RELATE MANY**([Clients]Code) ` リンクした交渉を取得

**SELECTION TO ARRAY**([Contacts]ContactName;NamesArray;

[Contacts]ContactFirstname;FirstnamesArray)

⇒ **PV ADD DYNAMIC ARRAYS**(Area;\$ArrayArrays)

参照

PV ADD DYNAMIC FIELDS、PV CLEAR DYNAMIC COLUMNS、PV SET CELL VARIABLE

## PV ADD DYNAMIC FIELDS

---

### PV ADD DYNAMIC FIELDS(エリア; マスター; テーブル; フィールド; メソッド)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
マスター	整数	→ マスターテーブル番号
テーブル	整数配列	→ テーブル番号の配列
フィールド	整数配列	→ フィールド番号の配列
メソッド	文字列配列	→ コールバックメソッドの配列

#### 説明

**PV ADD DYNAMIC FIELDS** コマンドは、配列<テーブル>と<フィールド>で指定した各フィールドの値に対応するひとまとまりの行を<エリア>に追加します。このフィールドは<マスター>で定義したテーブルのカレントセクションにリレートしており、A1セルより行の追加を開始します。エリアにダイナミックカラムが既に定義されている場合、新しいデータは最初に利用可能なカラムの1行目より挿入されます。これらのフィールドの値は、対応するセルを用いて、直接4D View から修正することができます。

結果は常に縦方向に表示されます（カラム、および配列<テーブル>、<フィールド>、<メソッド>の要素として）。

注：PV ADD DYNAMIC ARRAYS コマンドを使用して、4D View エリアにダイナミック配列エリアが既に挿入されている場合、そのエリアは削除されダイナミックフィールドで置き換えられます。

すべてのカラムに対し、この処理はリレートにより<マスター>テーブルに関連付けられた配列に適用されます。リレートはマスター配列から表示される配列のフィールドへの自動リレートでなければいけません。請求書と明細行に関する従来の例題では、行の配列のフィールド内容を回復したり変更することができますが、明細行配列を元にしてセクションにリレートしている請求書テーブル（マスターテーブル）の内容も回復することができます。

各コールバックメソッドは以下の6つの引数を受け取ります。

- \$1：エリア
- \$2：カラム番号
- \$3：配列タイプ
- \$4：この配列へのポインタ
- \$5：ダイナミックエリアの最初の行
- \$6：このエリアに表示できる行数

注：データベースをコンパイルする予定であれば、使用しない引数がある場合でも、これらの引数を必ず宣言しなくてはなりません。

開発者は、このコールバックメソッドの配列をすべて埋めてください。4D Viewはこの配列を使用して、計算後のカラムに代入します。戻り値（\$0）はありません。

**PV ADD DYNAMIC FIELDS** コマンドは、引数として渡された各フィールドとのダイナミックリンクを維持します。この結果、4D Viewエリアで値が変更されると、各フィールドに対してその変更が反映されます（この逆も同様）。

注：

- ・ 4D メソッドおよびフィールドの値は4D View エリアと同じ手順で定義しなくてはなりません。
- ・ 4Dの「ユーザ」モードと4D Viewのプラグインウィンドウとの間で、ダイナミックなデータ更新は行えません。4Dの「ユーザ」モードで行われた修正内容を4D View ウィンドウに反映するには、PV REDRAW コマンドを使用してウィンドウをリドロウ（再描画）する必要があります（4D コマンドを用いて行った修正は4D View エリアに自動的に反映されます）。

## 例題

この例題では、入力フォームに組み込まれた入力テーブルについて説明します（4D Viewを使用すると、より簡単に行えます）。それぞれの肩書き（リンク先テーブル）と各交渉のイニシャル（計算後のカラム）を用いて、現在の顧客にリレートしている交渉の各フィールドを変更します。

```

ARRAY INTEGER($TablesArray;4) ` テーブル番号
ARRAY INTEGER($FieldsArray;4) ` フィールド番号
ARRAY STRING(30;$MethodsArray;4) ` コールバックメソッド名
    `カラム 1：交渉の名字
$TablesArray{1}:=Table(->[Contacts])
$FieldsArray{1}:=Field(->[Contacts]ContactName)
$MethodsArray{1}:=""
    `カラム 2：交渉の名前
$TablesArray{2}:=Table(->[Contacts])
$FieldsArray{2}:=Field(->[Contacts]ContactFirstname)
$MethodsArray{2}:=""
    `カラム 3：交渉のタイトル（リンク先テーブル）
$TablesArray{3}:=Table(->[Titles])
$FieldsArray{3}:=Field(->[Titles]Label)
$MethodsArray{3}:=""
    `カラム 4：：注文番号/ 合計（計算後のカラム）

```

```

$TablesArray{4}:=0
$fieldsArray{4}:=ls text`結果
$MethodsArray{4}:="CallMethod"
RELATE MANY([Clients]Code)`交渉を取得
⇒ PV ADD DYNAMIC FIELDS(Area;Table(->[Contacts]);$TablesArray;$FieldsArray;
                                         $MethodsArray)

```

プロジェクトメソッド“CallMethod”のコードは次の通りです。

```

C_LONGINT($1)`4D View エリア
C_LONGINT($2)`カラム番号
C_LONGINT($3)`配列のタイプ
C_POINTER($4)`この配列へのポインタ
C_LONGINT($5)`ダイナミックエリアの最初の行
C_LONGINT($6)`このエリアに表示できる行数
GOTO SELECTED RECORD([Contacts];$5)
For($i;1;$6)
    $4->{$i}:=Substring([Contacts]ContactFirstname;1;1)+
                Substring([Contacts]ContactName;1;1)
    NEXT RECORD([Contacts])
End for

```

参照

PV ADD DYNAMIC ARRAYS、PV CLEAR DYNAMIC COLUMNS、PV FIELD TO CELLS、  
PV SET CELL FIELD

## PV CLEAR DYNAMIC COLUMNS

### PV CLEAR DYNAMIC COLUMNS(エリア; 開始; カラム数)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
開始	倍長整数	→ 開始カラム番号
カラム数	倍長整数	→ カラム数

#### 説明

**PV CLEAR DYNAMIC COLUMNS** コマンドは、<開始>で指定した番号のカラムから始めて、ダイナミックカラムを<カラム数>消去します。

消去されるダイナミックカラムは、フィールドまたは配列であり、それぞれ **PV ADD DYNAMIC FIELDS** および **PV ADD DYNAMIC ARRAYS** コマンドで作成されたものです。

残りのダイナミックカラムは再編成されるので、ダイナミックエリアは常にカラム A より始まり、ダイナミックエリアには“穴”が生じないように保たれます。

**PV CLEAR DYNAMIC COLUMNS** コマンドは、ダイナミックなフィールドまたは配列だけを使用したカラムに対してのみ処理を実行します。

#### 例題

**PV ADD DYNAMIC FIELDS** コマンドの例題では、現在の顧客にリンクしている交渉テーブルからそれぞれの肩書きと注文番号を使用して、リスト入力を行いました。以下のメソッドは3番目のカラムを削除しますが、名字と名前は依然として変更可能です。

⇒ **PV CLEAR DYNAMIC COLUMNS**(Area;3;1) `カラム C を削除

この命令が実行されると、4番目のカラム (D) が3番目 (C) に繰り上がるため、ダイナミックエリアには“穴”が発生しません。

#### 参照

PV ADD DYNAMIC ARRAYS、PV ADD DYNAMIC FIELDS



## PV Cell manipulation、はじめに

このテーマ内のコマンドを使用すると、4D Viewエリア上で検索や置き換え、並び替えを行うことができます。

また、このテーマには一連の3つのコマンドがあり、これらのコマンドによりコピー&ペースト機能をアプリケーションのコードへ迅速に組み込むことができます。

## PV FIND ONE

---

### PV FIND ONE(エリア; 条件; 方向; 場所; 一致{; カラム; 行 })

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
条件	文字列	→ 検索する文字列
方向	整数	→ 0= カラム ; 1= 行
場所	整数	→ 0= フォーミュラ ; 1= 値
一致	整数	→ 0= 一部 ; 1= 単語全体
カラム	倍長整数	→ 開始セルのカラム番号 ← 検出されたセルのカラム番号
行	倍長整数	→ 開始セルの行番号 ← 検出されたセルの行番号

### 説明

**PV FIND ONE** コマンドは、<エリア>内のセルを検索し、最初に見つかったセルをカレントセルにします。

新しくカレントとなったセルの座標は、<カラム>と<行>に返されます。検索の結果、セルが見つからなかった場合、カレントセルはそのまま変わりません。

<条件>には、検索する文字列を指定します。

<場所>には、検索を行うスプレッドシートの場所を指定します。

■ 0 : フォーミュラを検索

■ 1 : 値を検索

<一致>には、比較タイプを指定します。

■ 0 : 一部 (条件値を含む)

■ 1 : 全体 (条件値と一致)

任意の引数である<カラム>と<行>は、検索を開始する<エリア>の場所を指定します。次に示すエリアでは、B3セルが開始セルとして設定されています。

	A	B	C	D	E
1	IGNORED AREA				
2					
3					
4					
5			RESEARCH AREA		
6					
7					
8					
9					
10					
11					
12					

コマンドの実行後、発見された最初のセルが新しくカレントセルとなります。<カラム>と<行>を指定した場合、これらの引数にはそのセルの座標が納められます。この後、このセルの座標を発見されたエリアの一番左上に指定するか、値“1”を2つの変数に指定することにより、新しいカレントセルの座標を取得することができます。

検索によりセルが見つからなかった場合、カレントセルはそのまま変わりません。

### 例題

まず初めにユーザへの質問を行い、その応答された条件を含む最初のセルを検索しましょう。前述の図と同様に、検索はB3セルより開始します。

**C\_STRING**(255;\$Criteria) ` 検索する文字列

**C\_INTEGER**(\$QueryWhere) `0= フォーミュラ / 1= 値

**C\_INTEGER**(\$Contains) `0= 一部 / 1= 単語全体

**C\_LONGINT**(\$Column) ` 検出されたセルのカラム番号

**C\_LONGINT**(\$Row) ` 検出されたセルの行番号

\$Criteria:=**Request**("検索文字列を入力してください。";"x")

**If**(\$Criteria#"") ` 条件が指定された

**CONFIRM**("フォーミュラまたは値のいずれを検索しますか?";

"値";"フォーミュラ")

\$QueryWhere:=OK`0= フォーミュラ / 1= 値

**CONFIRM**("一致条件は?";"一部";"単語全体")

\$Contains:=OK`0= 一部 / 1= 単語全体

\$Column:=2` 検索は B3セルより開始

\$Row:=3` カラム A および 行 1 と行 2 は無視される

⇒ **PV FIND ONE**(Area;\$Criteria;\$QueryWhere;\$Contains;\$Column;\$Row)

` 検索

**End if**

参照

PV FIND ALL、PV REPLACE ONE

## PV FIND ALL

---

### PV FIND ALL(エリア; 条件; 場所; 一致)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
条件	文字列	→ 検索する文字列
場所	整数	→ 0= フォーマット ; 1= 値
一致	整数	→ 0= 一部 ; 1= 単語全体

### 説明

**PV FIND ALL** コマンドは **PV FIND ONE** コマンドと同様の働きをしますが、検索条件を満たす<エリア>内のセルをすべて選択します。

このコマンドの実行後、発見されたセルのうち最初のセルが新しくカレントセルとなり、他のセルは選択された状態となります。

検索によりセルが見つからなかった場合、カレントセルはそのまま変わりません。

<条件>には、検索する文字列を指定します。

<場所>には、検索を行うスプレッドシートの場所を指定します。

■ 0：フォーマットを検索

■ 1：値を検索

<一致>には、比較タイプを指定します。

■ 0：一部（条件値を含む）

■ 1：全体（条件値と一致）

### 例題

**PV SET CELL PROPERTY** の例題を参照してください。

### 参照

PV FIND ONE、PV REPLACE ALL

## PV REPLACE ONE

### PV REPLACE ONE(エリア; 文字列; 置換; 方向; 場所; 一致{; カラム; 行})

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
文字列	文字列	→ 置換対象の文字列
置換	文字列	→ 置き換える文字列
方向	整数	→ 0= カラム ; 1= 行
場所	整数	→ 0= フォーミュラ ; 1= 値
一致	整数	→ 0= 一部 ; 1= 単語全体
カラム	倍長整数	→ 開始セルのカラム番号 ← 検出されたセルのカラム番号
行	倍長整数	→ 開始セルの行番号 ← 検出されたセルの行番号

### 説明

**PV REPLACE ONE** コマンドは、<カラム>と<行>で指定したセルより検索を開始し、<場所>と<一致>で定義した検索条件を満たす<エリア>内の最初の<文字列>を<置換>で置き換えます。検索を開始する箇所は、エリアの一番左上または引数で指定したセルです。また、検索条件を満たす最初のセルが見つかった時点で検索を終了します。

<文字列>には検索対象となる文字列を指定し、置き換える文字列は<置換>に指定します。

<場所>には、検索を行うスプレッドシートの場所を指定します。

■ 0：フォーミュラを検索

■ 1：値を検索

<一致>には、比較タイプを指定します。

■ 0：一部（条件値を含む）

■ 1：全体（条件値と一致）

任意の引数である<カラム>と<行>は、検索を開始する<エリア>の場所を指定します。次に示すエリアでは、B3セルが開始セルとして設定されています。

	A	B	C	D	E
1	IGNORED AREA				
2					
3					
4					
5			RESEARCH AREA		
6					
7					
8					
9					
10					
11					
12					

このコマンドの実行後、最初に発見されたセルが新しくカレントセルとなります。＜カラム＞と＜行＞を指定した場合、引数＜カラム＞と＜行＞にはこのセルの座標が納められます。この後、そのセルの座標を発見されたエリアの一番左上に指定するか、または2つの変数に値“1”を指定することにより、新しいカレントセルの座標を取得することができます。

検索によりセルが見つからなかった場合、カレントセルはそのまま変わりません。

### 例題

最初に検出された現在のT.V.A. レートを、新しい基準（新しいレート）で置き換えます。

**C\_INTEGER(\$Where)** `0= フォーミュラ / 1= 値

**C\_INTEGER(\$Contains)** `0= 一部 / 1= 全体

\$Where:=0` フォーミュラを置換

\$Contains:=0` 条件値を含む

`最初に発見された文字列を置換

⇒ **PV REPLACE ONE**(Area;"\$C\$30";"\$C\$31";\$Where;\$Contains)

### 参照

PV FIND ONE、PV REPLACE ALL

## PV REPLACE ALL

---

### PV REPLACE ALL(エリア; 文字列; 置換; 場所; 一致)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
文字列	倍長整数	→ 置換対象の文字列
置換	文字列	→ 置き換える文字列
場所	整数	→ 0= フォーミュラ; 1= 値
一致	整数	→ 0= 一部; 1= 単語全体

### 説明

**PV REPLACE ALL** コマンドは **PV REPLACE ONE** コマンドと同様の働きをしますが、引数<場所>と<一致>で定義した検索条件を満たす<エリア>内のすべてのセルの<文字列>を<置換>で置き換えます。

このコマンドの実行後、発見されたセルのうち最初のセルが新しくカレントセルとなり、他のセルは選択された状態となります。

置換が行われなかった場合、アクティブセルはそのまま変わりません。

### 例題

検出されたすべての文字列 “Amount” を大文字に変えます。

⇒ **PV REPLACE ALL**(Area;"Amount";"AMOUNT";1;1)

### 参照

PV FIND ALL、 PV REPLACE ONE

## PV SORT ONE

---

### PV SORT ONE(エリア; 左; 上; 右; 下; 方向; キー; 並び替え順)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	倍長整数	→ 左側のカラム番号
上	倍長整数	→ 上側の行番号
右	倍長整数	→ 右側のカラム番号
下	倍長整数	→ 下側の行番号
方向	整数	→ 0= カラムを並び替え; 1= 行を並び替え
キー	倍長整数	→ 並び替える値を含むカラム、または行
並び替え順	整数	→ 0= 昇順; 1= 降順

### 説明

**PV SORT ONE** コマンドは、<左>、<上>、<右>、<下>で区切られた選択範囲の内容を、<キー>となる行またはカラム内の値との関連で、<並び替え順>で指定された順序に並び替えます。

<方向>には、行またはカラムのいずれを並び替えるのかを指定します。

■ 0を渡した場合、<キー>行の値に応じてカラムを並び替えます。

■ 1を渡した場合、<キー>カラムの値に応じて行を並び替えます。

このコマンドの処理対象となるのは、スタティックなデータだけです。

ダイナミックエリア（配列およびリンク付きフィールド）の並び替えは、必ず4Dコマンドを使用して行ってください。つまり、原則として元となるデータを並び替えるということです。ヘッダクリック時に、フィールドにリンクするダイナミックカラムを並び替える例題は、**PV GET CELL FIELD** コマンドの説明で提供されています。また、ピクチャタイプの配列やフィールド、および計算結果のカラム（コールバックメソッドを呼び出して、その結果を表示するカラム）に対する並べ替えも行えないということに注意してください。

### 例題

この例題を使用すると、カラムヘッダのクリックによって、スタティックなカラムを昇順に並び替えることができます。エリアにはスタティックなカラムだけが含まれていません。

- まず、コールバックメソッド `HeaderSortMethod` をインストールします。このメソッドは、エリア内でクリックされた場合に呼びだされます。

```
PV ON EVENT(area;pv on clicked;"HeaderSortMethod")
```

- HeaderSortMethod メソッドは、カラムヘッダ上で行われたクリックをすべて検出し、対応するデータを昇順に並び替えます（1行目から25行目まで）。

```
HeaderSortMethod メソッド
C_BOOLEAN($0)
C_LONGINT($1;$2;$3;$4;$5;$6)
If($5=0) `ヘッダ上でクリックされた場合
    $0:=True `イベントをキャンセルする
⇒ PV SORT ONE(area;$4;1;$4;25;1;$4;0) `選択されたカラムを昇順に
    並び替える

End if
```

参照

PV SORT MANY

## PV SORT MANY

---

### PV SORT MANY(エリア; 左; 上; 右; 下; 方向; キー; 並び替え順)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	倍長整数	→ 左側のカラム番号
上	倍長整数	→ 上側の行番号
右	倍長整数	→ 右側のカラム番号
下	倍長整数	→ 下側の行番号
方向	整数	→ 0= カラムを並び替え; 1= 行を並び替え
キー	配列	→ 並び替える値を含むカラム、または行
並び替え順	配列	→ 並び替え方向: 0= 昇順; 1= 降順

### 説明

**PV SORT MANY** コマンドは **PV SORT ONE** コマンドと同様の働きをしますが、このコマンドはマルチソートを行います。並び替え条件となるカラムまたは行を<キー>に渡し、並び替えの実行順（昇順または降順）を<並び替え順>に渡します。

<左>、<上>、<右>、<下>で区切られたセルの選択範囲に対して、並び替えが行われます。

<方向>には、行またはカラムのいずれを並び替えるのかを指定します。

■ 0を渡した場合、<キー>行の値に応じてカラムを並び替えます。

■ 1を渡した場合、<キー>カラムの値に応じて行を並び替えます。

### 例題

セルの選択範囲（5カラム X 9行）に対し、2番目のカラムを最初の条件として昇順に、3番目のカラムを第2の条件として降順に並び替えます。

```
ARRAY LONGINT($Keys;2) `並び替える値を含むカラム、または行  
ARRAY INTEGER($Orders;2) `0= 昇順 / 1= 降順  
`初期化
```

```
$Keys{1}:=3` 選択範囲の2番目のカラムを並び替えの第1条件として使用  
$Keys{2}:=4` 選択範囲の3番目のカラムを並び替えの第2条件として使用  
$Orders{1}:=0` 選択範囲の2番目のカラムとの関係で行を昇順に並び替える  
$Orders{2}:=1` 選択範囲の3番目のカラムとの関係で行を降順に並び替える
```

⇒ **PV SORT MANY**(Area;2;2;6;10;0;\$Keys;\$Orders)

### 参照

PV SORT ONE

## PV SPECIAL CLEAR

---

### PV SPECIAL CLEAR(エリア; 値; フォーマチュラ; フォーマット; 枠線)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
値	整数	→ 0= クリアしない ; 1= クリア
フォーマチュラ	整数	→ 0= クリアしない ; 1= クリア
フォーマット	整数	→ 0= クリアしない ; 1= クリア
枠線	整数	→ 0= クリアしない ; 1= クリア

### 説明

**PV SPECIAL CLEAR** コマンドは、<エリア>におけるカレントセルの選択範囲から引数<値>、<フォーマチュラ>、<フォーマット>、<枠線>で指定された情報をクリアします。

### 例題

選択されたセルのフォーマチュラ、フォーマット、枠線を消去し、フォーマチュラではなく値を含んだセルの値は保持します。

⇒ **PV SPECIAL CLEAR**(Area;0;1;1;1) `フォーマチュラ、フォーマット、枠線を消去

### 参照

PV SPECIAL CUT、PV SPECIAL PASTE

## PV SPECIAL PASTE

---

### PV SPECIAL PASTE(エリア; 値; フォーミュラ; フォーマット; 枠線)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
値	整数	→ 0= ペーストしない; 1= ペースト
フォーミュラ	整数	→ 0= ペーストしない; 1= ペースト
フォーマット	整数	→ 0= ペーストしない; 1= ペースト
枠線	整数	→ 0= ペーストしない; 1= ペースト

### 説明

**PV SPECIAL PASTE** コマンドは、クリップボードに保存されたセルのデータを元に、引数<値>、<フォーミュラ>、<フォーマット>、<枠線>で指定された情報を<エリア>のカレントセルにペーストします。

このコマンドは、先に **PV SPECIAL CUT** コマンドで“カット”されたか、または **PV EXECUTE COMMAND(area; pv cmd edit copy)** コマンドでコピーされたデータを再利用します。

### 例題

この例題は、値とフォーマットのコピー／ペーストをシミュレートします。B2セルの値とフォーマットをカットした後、その情報をすべてE2セルにペーストします。

**PV GOTO CELL**(Area;2;2) `開始セル： B2

**PV SPECIAL CUT**(Area;1;0;1;0) `値とフォーマットをカット

**PV GOTO CELL**(Area;5;2) `コピー先のセル： E2

⇒ **PV SPECIAL PASTE**(Area;1;0;1;0) `値とフォーマットのコピー

### 参照

PV SPECIAL CLEAR、PV SPECIAL CUT

## PV SPECIAL CUT

---

### PV SPECIAL CUT(エリア; 値; フォーミュラ; フォーマット; 枠線)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
値	整数	→ 0= カットしない; 1= カット
フォーミュラ	整数	→ 0= カットしない; 1= カット
フォーマット	整数	→ 0= カットしない; 1= カット
枠線	整数	→ 0= カットしない; 1= カット

#### 説明

**PV SPECIAL CUT** コマンドは、<エリア>におけるカレントセルの選択範囲に対して、引数<値>、<フォーミュラ>、<フォーマット>、<枠線>で指定された情報をカットします。

**PV SPECIAL CLEAR** コマンドとは異なり、**PV SPECIAL CUT** コマンドは後で利用するためにカットした情報をクリップボードに保存します。例えば、**PV SPECIAL PASTE** コマンドでこの情報を使用します。

#### 例題

**PV SPECIAL PASTE** コマンドの例題を参照してください。

#### 参照

PV SPECIAL CLEAR、PV SPECIAL PASTE



## PV Columns & rows、はじめに

---

このテーマ内のコマンドにより、プログラムを介して4D View スプレッドシートのカラムや行を管理することができます。

- 挿入
- 削除
- サイズの読み込みと設定（高さと幅）
- カラムと行のヘッダの読み込みと設定（行タイトルとカラムタイトル、およびカラムの並び替えオプション）

### 行とカラムの名前

4D View エリアの行やカラムに名前を関連付けることができ、この名前によりドキュメント内での行とカラムの取り扱いがより容易になります。デフォルトとして、それぞれの行とカラムには名前が付けられています。以下に説明する方法で行とカラムは自動的に命名されます。

- 行：実際の行番号に厳密に対応する名前。

行番号を参照するコマンドに対しては、名前に関する取り決めがあります。エリアの行数は、**PV SET DOCUMENT PROPERTY** コマンドを使用して設定することができます。

- カラム：文字を使用してカラム名が付けられます。エリアのプロパティに応じ、カラム数はアルファベット 26 文字より多くても構いません。複数の文字を使って記号化が行われ、文字“A”より繰り返されます（AA、AB、AC等。AZ、BA、BB等）。カラム番号を呼び出すコマンドに対して、カラム番号とカラム名との取り決めはデフォルトとして以下の通りになります。

カラム名	カラム番号
A	1
B	2
C	3
[...]	
Y	25
Z	26
AA	27
AB	28
AC	29
他	

## PV INSERT COLUMNS

---

### PV INSERT COLUMNS (エリア; 開始; カラム数)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
開始	倍長整数	→ 新しいカラムの挿入箇所となるカラム
カラム数	倍長整数	→ カラム数

### 説明

**PV INSERT COLUMNS** コマンドは、<開始>カラム番号から開始して<エリア>に<カラム数>の数のカラムを挿入します。カラムは、<開始>で指定したカラムの直前に挿入されます。

### 例題

表の1番目のカラム (A) の前にカラムを1つ挿入し、カラムエリアの内容を右側にシフトします。

⇒ **PV INSERT COLUMNS**(Area;1;1)

### 参照

PV DELETE COLUMNS、PV INSERT ROWS

## PV INSERT ROWS

---

### PV INSERT ROWS (エリア; 開始; 行数)

引数	タイプ	説明
エリア	倍長整数 →	4D View エリア
開始	倍長整数 →	新しい行の挿入箇所となる行
行数	倍長整数 →	行数

### 説明

**PV INSERT ROWS** コマンドは、<開始>行番号から開始して<エリア>に<行数>の数の行を挿入します。行は、<開始>で指定した行の直前に挿入されます。

### 例題

表の1番目の行 (1) の前に行を1行挿入し、行エリアの内容を下側にシフトします。

⇒ **PV INSERT ROWS**(Area;1;1)

### 参照

PV DELETE ROWS、PV INSERT COLUMNS

## PV INSERT CELLS

---

### PV INSERT CELLS (エリア; カラム; 行; セル数; 方向)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ カラム番号
行	倍長整数	→ 行番号
セル数	倍長整数	→ 挿入するセル数
方向	倍長整数	→ 挿入する方向

#### 説明

**PV INSERT CELLS** コマンドは、<カラム>と<行>で指定したセルより開始して、<エリア>に<セル数>の数のセルを挿入します。

引数<方向>を使用して、既存のセルをシフトする方向（下側または右側）を指定します。この引数の値を設定するには、**PV Directions** テーマ内の定数である **pv to the right** または **pv to the bottom** を使用します。

#### 例題

次の例題では、1カラム目の1行目より開始して2つのセルを挿入します。既存のセルは下側にシフトします。

⇒ **PV INSERT CELLS**(area;1;1;2; pv to the bottom)

#### 参照

PV DELETE CELLS

#### 定数

PV Directions テーマ

## PV DELETE COLUMNS

---

### PV DELETE COLUMNS (エリア; 開始; カラム数)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
開始	倍長整数	→ 開始カラム番号
カラム数	倍長整数	→ カラム数

#### 説明

**PV DELETE COLUMNS** コマンドは、<開始>カラム番号から始めて、<エリア>の<カラム数>の数のカラムを削除します。

#### 例題

表の1番目のカラム (A) を削除し、残りのカラムエリアの内容を左側にシフトします。

⇒ **PV DELETE COLUMNS**(Area;1;1)

#### 参照

PV DELETE ROWS、PV INSERT COLUMNS

## PV DELETE ROWS

---

### PV DELETE ROWS (エリア; 開始; 行数)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
開始	倍長整数	→ 開始行番号
行数	倍長整数	→ 行数

#### 説明

**PV DELETE ROWS** コマンドは、<開始>行番号から始めて、<エリア>の<行数>の数の行を削除します。

#### 例題

表の1番目の行 (1) を削除し、残りの行エリアを上側にシフトします。

⇒ **PV DELETE ROWS**(Area;1;1)

#### 参照

PV DELETE COLUMNS、PV INSERT ROWS

## PV DELETE CELLS

---

### PV DELETE CELLS (エリア; カラム; 行; セル数; 方向)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ カラム番号
行	倍長整数	→ 行番号
セル数	倍長整数	→ 削除するセル数
方向	倍長整数	→ セルをシフトする方向

#### 説明

**PV DELETE CELLS** コマンドは、<カラム>と<行>で指定したセルより開始して、<エリア>内のセルを<セル数>の数だけ削除します。

引数<方向>を使用し、既存のセルをシフトする方向（上側または左側）を指定します。この引数の値を設定するには、**PV Directions** テーマ内の定数である **pv to the left** または **pv to the top** を使用します。

#### 例題

次の例題では、2カラム目の2行目から1つのセルを削除します。その他のセルは上側にシフトします。

⇒ **PV DELETE CELLS**(エリア;2;2;1; pv to the top)

#### 参照

PV INSERT CELLS

#### 定数

PV Directions テーマ

## PV SET COLUMNS WIDTH

### PV SET COLUMNS WIDTH (エリア; 始め; 終わり; 幅)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
始め	倍長整数	→ 最初のコラム番号
終わり	倍長整数	→ 最後のコラム番号
幅	整数	→ コラム幅 (ピクセル単位)

### 説明

**PV SET COLUMNS WIDTH** コマンドを使用して、<エリア>内の<始め>と<終わり>の間にあるコラムの幅 (ピクセル単位) を変更することができます。

注: 引数<始め>と<終わり>に0 (ゼロ) を渡すと、指定した幅がエリアの全コラムに対して適用され、この値がそのエリアのコラムの新しいデフォルト幅になります。ユーザがコラムの右側のセパレータをダブルクリックした際には、このデフォルト幅が適用されます。

### 例題

この例題では、行とコラムのサイズを変更する方法を説明します。

```

C_LONGINT($Column;$Row) `主要ループの指標
C_LONGINT($Width;$Height) `コラムと行を拡張するループの指標
C_LONGINT($StartWidth) `X番目のコラムの元の幅
C_LONGINT($RequestWidth) `X番目のコラムに求められる幅
    `初期化を行う
$StartWidth:=5 `元の幅を設定する
$RequestWidth:=$StartWidth+5 `最初のコラムを10ポイントの幅に設定
For($Column;1;5) `最初の5コラムを処理する
    $Row:=$Column `最初の5行だけを処理する
    For($Width;$StartWidth;$RequestWidth;2) `"$Column" ...番目のコラム
        $Height:=$Width-5 `高さを更新
        `1行
        PV SET ROWS HEIGHT(area;$Row;$Row;PV Get row height
            (area;$Row)+$Height)
⇒ PV SET COLUMNS WIDTH(area;$Column;$Column;PV Get column
    width(area;$Row)+$Width) `1コラム
    PV REDRAW(area) `リフレッシュ
End for
$StartWidth:=$RequestWidth `最初から開始しない
$RequestWidth:=$RequestWidth+5 `次のコラムを拡張する

```

## End for

### 参照

PV Get column width、PV SET ROWS HEIGHT

## PV SET ROWS HEIGHT

---

### PV SET ROWS HEIGHT (エリア; 始め; 終わり; 高さ)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
始め	倍長整数	→ 最初の行番号
終わり	倍長整数	→ 最後の行番号
高さ	整数	→ 行の高さ (ピクセル単位)

### 説明

**PV SET ROWS HEIGHT** コマンドは、<エリア>内の<始め>と<終わり>の間にある行の高さ (ピクセル単位) を設定します。

注: 引数<始め>と<終わり>に0 (ゼロ) を渡すと、指定した高さがエリアのすべての行に対して適用され、この値がそのエリアの行の新しいデフォルト高さになります。ユーザが行の下側のセパレータをダブルクリックした際には、このデフォルト幅が適用されます。

### 例題

**PV SET COLUMNS WIDTH** コマンドの例題を参照してください。

### 参照

PV Get row height、PV SET COLUMNS WIDTH

## PV Get column width

---

### PV Get column width (エリア; カラム) → 整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ カラム番号
戻り値	整数	← カラム幅 (ピクセル単位)

#### 説明

**PV Get column width**関数は、指定された<カラム>の幅 (ピクセル単位) を返します。

#### 例題

**PV SET COLUMNS WIDTH** コマンドおよび **PV ADD VERT SPLITTER** コマンドの例題を参照してください。

#### 参照

PV Get row height、PV SET COLUMNS WIDTH

## PV Get row height

---

### PV Get row height (エリア; 行) → 整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
行	倍長整数	→ 行番号
戻り値	整数	← 行の高さ (ピクセル単位)

#### 説明

**PV Get row height**関数は、指定された<行>の高さ (ピクセル単位) を返します。

#### 例題

**PV SET COLUMNS WIDTH** コマンドの例題を参照してください。

#### 参照

PV Get column width、PV SET ROWS HEIGHT

## PV SET COLUMN HEADER

---

PV SET COLUMN HEADER(エリア; カラム; タイトル; 並び替え)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
カラム	倍長整数	→ カラム番号
タイトル	文字列	→ カラム名

### 説明

**PV SET COLUMN HEADER** コマンドは、指定された<カラム>の<タイトル>オプションを設定します。

行とカラムのデフォルト名に関する詳細は、「PV Columns & Rows、はじめに」の節を参照してください。

### 例題

この例題では、エリアの最初から10カラムと10行に新しい名前を割り当てます。並び替えオプションはそのまま変わりません。

**C\_INTEGER(\$Index)** `ループの指標

**C\_STRING(80;\$Title)** `カラム／行の名前

**For(\$Index;1;10)**

`\$Indexカラム目の名前を取得する

**PV GET COLUMN HEADER**(Area;\$Index;\$Title)

\$Title:="Column"+\$Title `名前を変更

⇒ **PV SET COLUMN HEADER**(Area;\$Index;"C"+\$Title)

`新しい名前を割り当てる

`\$Index行目の名前を読み込んで変更し、新しい名前を割り当てる

**PV SET ROW HEADER**(Area;\$Index;"L"+**PV Get row header**(Area;\$Index))

**End for**

### 参照

PV GET COLUMN HEADER、PV SET ROW HEADER

## PV GET COLUMN HEADER

---

### PV GET COLUMN HEADER (エリア; カラム; タイトル; 並び替え)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
カラム	倍長整数	→	カラム番号
タイトル	文字列	←	カラム名

#### 説明

**PV GET COLUMN HEADER** コマンドは、指定された<カラム>の現在の<タイトル>の状況を取得します。

行とカラムのデフォルト名に関する詳細は、「PV Columns & Rows、はじめに」の節を参照してください。

#### 例題

**PV SET COLUMN HEADER** コマンドの例題を参照してください。

#### 参照

PV Get row header、PV SET COLUMN HEADER

## PV SET ROW HEADER

---

### PV SET ROW HEADER (エリア; 行; タイトル)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
行	倍長整数	→	行番号
タイトル	文字列	→	行の名前

#### 説明

**PV SET ROW HEADER** コマンドは、指定された<行>の<タイトル>を設定します。

行とカラムのデフォルト名に関する詳細は、「PV Columns & Rows、はじめに」の節を参照してください。

#### 例題

**PV SET COLUMN HEADER** コマンドの例題を参照してください。

## 参照

PV Get row header、PV SET COLUMN HEADER

## PV Get row header

---

### PV Get row header (エリア; 行) → 文字列

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
行	倍長整数	→	行番号
戻り値	文字列	←	行の名前

### 説明

**PV Get row header** 関数は、指定された<行>の現在の名前を取得します。

行とカラムのデフォルト名に関する詳細は、「PV Columns & Rows、はじめに」の節を参照してください。

### 例題

**PV SET COLUMN HEADER** コマンドの例題を参照してください。

## 参照

PV GET COLUMN HEADER、PV SET ROW HEADER

## PV Document、はじめに

このテーマ内のコマンドを使用して、4D Viewで読み込むことができるドキュメントの操作を行うことができます。

これらのコマンドにより、ディスク上のドキュメントの保存やオープンが可能になるだけでなく、題名や作成者、およびデフォルトの行数やカラム数のような関連情報の設定や取得もプログラムを使用して行うことができます。

## PV OPEN DOCUMENT

---

### PV OPEN DOCUMENT (エリア; ドキュメント; テンプレート)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
ドキュメント	文字列	→ ドキュメントの名前
テンプレート	整数	→ 0= ドキュメント; 1= テンプレート

#### 説明

**PV OPEN DOCUMENT** コマンドは、要求された<ドキュメント>を<エリア>内で開きます。

<ドキュメント>には、開こうとするドキュメントのフルパス名を渡します。この引数に空の文字列 ("") を渡した場合、標準の「ファイルを開く」ダイアログボックスが表示され、ユーザはドキュメントを選択することができます。このダイアログボックスでユーザが「キャンセル」をクリックすると、ドキュメントは開かれません。

引数<テンプレート>により、ドキュメントを標準ドキュメントとして開くか (テンプレート=0)、またはテンプレートとして開くか (テンプレート=1) を指定することができます。

ドキュメントがテンプレートとして開かれると、“名称未設定”という新しいドキュメントが実際に作成され、要求されたドキュメントのコピーが納められます。オリジナルのドキュメントはそのままです。<テンプレート>の値に1を指定すると、**PV SAVE DOCUMENT** コマンドで設定された実際の状態とは関係なく (ドキュメントまたはテンプレート)、ドキュメントはテンプレートであるものとして開かれます。

注：“ドキュメント”のテンプレートのメカニズムは、OSにより管理されています。これは組み込みエリアに付属する4D Viewの“テンプレート”で使用するものとは異なります (「テンプレートとして保存」メニューコマンドを使ってアクセスする)。エリアのテンプレートに関する詳細は、4D Viewの『ユーザリファレンス』マニュアルを参照してください。

#### 例題

標準の「ファイルを開く」ダイアログボックスを4D Viewで開きます。

⇒ **PV OPEN DOCUMENT**(Area;"";0) `ドキュメントの選択

#### 参照

PV SAVE DOCUMENT

### システム変数およびセット

システム変数Documentには、最後に開かれたディスクファイルの名前またはアクセスパスのいずれかが代入されます（4th Dimensionの『ランゲージリファレンス』マニュアルの「システム変数」の章を参照してください）。

ドキュメントが正常に開かれた場合、システム変数OKには値として1が代入されます。

## PV SAVE DOCUMENT

---

### PV SAVE DOCUMENT (エリア; ドキュメント; テンプレート; 消去; フォーマット)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
ドキュメント	文字列	→ ドキュメントの名前
テンプレート	整数	→ 0= ドキュメント; 1= テンプレート
消去	整数	→ 0= 消去しない; 1= 消去
フォーマット	倍長整数	→ ドキュメントのフォーマット

#### 説明

**PV SAVE DOCUMENT** コマンドは、4D View<エリア>をディスクドキュメントとして保存します。

<ドキュメント>には、保存するドキュメントの名前とフルパス名を渡します。この引数に空の文字列 ("") を渡した場合、標準の「ファイル保存」ダイアログボックスが表示され、ユーザはドキュメントの名前とパスを選択することができます。このダイアログボックスでユーザが「キャンセル」ボタンをクリックすると、ドキュメントは保存されません。

引数<テンプレート>により、ドキュメントを標準ドキュメントとして保存するか（テンプレート=0）、またはテンプレートとして保存するか（テンプレート=1）を指定することができます。テンプレートに関する詳細は、**PV OPEN DOCUMENT** コマンドの説明を参照してください。

指定された<エリア>内に同じ名前のドキュメントが既に存在する場合、**PV SAVE DOCUMENT** コマンドは、引数<消去>の値に応じて既存のファイルを上書きします。ただしその場合、<消去>に0が指定されていると番号26のエラー「このドキュメントは既に存在しています」が返されます。また、この名前がユーザにより指定された場合（<ドキュメント>に空の文字列を渡す）には、<消去>の値とは関係なく「このドキュメントは既に存在しています」という通常のメッセージがOSより表示されます。

対象となるドキュメントが他の4D Viewエリアで使用されている場合、ドキュメントがテンプレートであるかどうかには関わらず、**PV SAVE DOCUMENT** コマンドはエラーを返します。

引数<フォーマット>を定義するには、**PV Document format** テーマの定数を使用します。この定数を用いて、ドキュメントを保存するフォーマットを設定することができます。



## PV EXPORT

---

### PV EXPORT(エリア; ドキュメント; 置換; フォーマット)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
ドキュメント	文字列	→	ドキュメント名または空の文字列
置換	整数	→	0= 置換しない; 1= 置換
フォーマット	倍長整数	→	ドキュメントのフォーマット

### 説明

**PV EXPORT** コマンドは、4D View エリアや書き出しエリアをディスク上のドキュメントとして書き出します。

<ドキュメント>には、書き出しを行うドキュメントの名前またはフルパス名を渡します。この引数に空の文字列を渡すと、「別名で書き出し」ダイアログボックスが表示され、ユーザがドキュメントの名前および保存場所を指定することができます。この場合、ユーザが「キャンセル」ボタンをクリックすると、ドキュメントの書き出しは行われません。

指定した場所に同じ名前を持つドキュメントが既に存在している場合、**PV EXPORT** コマンドは引数<置換>の値に応じて、ファイルの上書きを実行するかどうかを決定します。この場合、<置換>の値が0であれば、エラー番号26「ドキュメントは既に存在します。」が発生します。ユーザにより名前が指定された場合には (<ドキュメント>に空の文字列を指定)、引数<置換>の値に関係なくOSより従来の確認メッセージ「このドキュメントは既に存在します...」が表示されます。

引数<フォーマット>の定義には、**PV Document format** テーマの定数を使用してください。これにより、ドキュメントを保存するフォーマットを指定することができます。

注：このコマンドでは、ドキュメントフォーマットの **pv view** 定数を適用することはできません。

**PV SAVE DOCUMENT** コマンド（エリア全体を保存する）とは異なり、**PV EXPORT** コマンドを使用すると、ドキュメントの書き出しエリアだけを保存することができます。書き出しエリアの指定は、ユーザが手動で行うか（「ファイル：エリア書き出し：設定」コマンド）、または**PV EXECUTE COMMAND**(area;pv cmd export area set)を使用してプログラムから行うことができます。書き出しエリアを定義する時点で選択されているセルが、書き出しエリアになります。デフォルトでは、書き出しエリアはドキュメント全体になります。

## 例題

以下の例題を使用して、エリア内で選択されているすべてのセルをHTMLフォーマットで書き出すことができます。書き出し時にセルが選択されていない場合、任意の領域が設定されます。

```

ARRAY LONGINT($ALleft;0) `左側のセルのカラム番号
ARRAY LONGINT($ALtop;0) `上側のセルの行番号
ARRAY LONGINT($ALright;0) `右側のセルのカラム番号
ARRAY LONGINT($ALbottom;0) `下側のセルの行番号
PV GET SELECTED RANGES LIST(area;$ALleft;$ALtop;$ALright;$ALbottom)
If(Size of array($ALleft)=0) `セルが選択されていない
    PV SELECT RANGE(area;2;4;5;7;pv selection set) `任意の領域
Else
    PV SELECT RANGES LIST(area;$ALleft;$ALtop;$ALright;$ALbottom;
                                pv selection set)
End if
    `書き出しエリアを選択領域にまで縮める
PV EXECUTE COMMAND(area;pv cmd export area set)
PV EXPORT(area;"";1;pv html)
    `書き出しエリアをドキュメント全体に初期設定する
PV EXECUTE COMMAND(area;pv cmd export area clear)

```

## 参照

PV SAVE DOCUMENT

## 定数

PV Document format テーマ

## PV SET DOCUMENT PROPERTY

---

### PV SET DOCUMENT PROPERTY (エリア; オプション; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
オプション	倍長整数	→ プロパティの番号
値	倍長整数	→ プロパティの値

#### 説明

**PV SET DOCUMENT PROPERTY** コマンドは、<エリア>内の4D View ドキュメントに対し、<オプション>で指定されたプロパティの<値>を設定します。

これらの定数および関連する値について、以下に説明します。

#### pv document modified

エリアに関する“変更状況”属性の設定や読み込みを行います。関連する値：pv value on または pv value off。

- この定数を書き込みモード (**PV SET DOCUMENT PROPERTY** コマンド) で使用し、引数<値>に pv value on を渡した場合、エリアのクローズ時にエリアが修正されたことを知らせる警告ダイアログボックスが表示されます。引数<値>に pv value off を渡した場合、ドキュメントがその後ユーザやプログラムにより変更されなければ、このダイアログボックスは表示されません。
- この定数を読み込みモードで使用し、**PV Get document property** 関数を実行した場合、戻り値としてドキュメントが変更されていれば1が、そうでなければ0が返されません。

#### pv column count

この定数はリードオンリーで使用され (**PV Get document property** 関数)、エリアに定義されたカラム数を返します。

#### pv picture count

この定数はリードオンリーで使用され (**PV Get document property** 関数)、エリアにペーストされたピクチャ数を返します。

### pv row count

この定数はリードオンリーで使用され（**PV Get document property**関数）、エリアに定義された行数を返します。

### pv no formula external call

エリアのフォーミュラ上で、4D変数やメソッド、およびコマンドの呼び出しを禁止します。関連する値：pv value onまたはpv value off。

- **pv value on**：フォーミュラにおいて、4D変数やメソッド、およびコマンドの呼び出しを禁止します（この場合、“PV Allows Input”テーマのコマンドを使用して、呼び出し可能な4Dオブジェクトを指定することができます）。
- **pv value off**：フォーミュラでは、すべての4D変数、メソッド、およびコマンドの呼び出しが可能です（デフォルト値）。

### 例題

1) この汎用メソッドを使用して、新しい4D View エリアのカラム数と行数を設定することができます（例えば、フォームのロード時）。

⇒ **PV SET DOCUMENT PROPERTY**(Area;pv column count;10) `10 カラム

⇒ **PV SET DOCUMENT PROPERTY**(Area;pv row count;20) `20 行

(2)このメソッドは、例えば4D View ドキュメントのクローズボタンに関連付けて、エリアが変更されたことを示す警告ダイアログボックスを表示しないように設定することができます。

```
If(PV Get document property(Area;pv document modified)#0)  
    `0= 変更なし, 1= 変更
```

⇒ **PV SET DOCUMENT PROPERTY**(Area;pv document modified;0)

```
End if
```

### 参照

PV Get document property

### 定数

PV Document properties テーマ

## PV Get document property

---

### PV Get document property (エリア; オプション) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
オプション	倍長整数	→ プロパティの番号
戻り値	倍長整数	← プロパティの値

#### 説明

**PV Get document property** 関数は、<エリア>内のドキュメントに関する引数<オプション>の現在の値を返します。

引数<オプション>を定義するには、**PV Document properties** 定数を使用します。

#### 例題

1番目の引数として渡された4D View エリアのカラム数および行数を格納した変数を更新するメソッドを作成します。

```
C_LONGINT($ColNum) `カラム数
C_LONGINT($RowNum) `行数
C_LONGINT($PicNum) `ピクチャ数
⇒ $ColNum:=PV Get document property(Area;pv column coun)
⇒ $RowNum:=PV Get document property(Area;pv row coun)
⇒ $PicNum:=PV Get document property(Area;pv picture count)
ALERT("この4D View エリアには、"+String($ColNum)+" カラム"+
      ("s"*Num($ColNum>1))+ "と "+String($RowNum)+"行"+("s"*Num($RowNum>1))
      +"があり、ピクチャは"+String($PicNum)+"個あります。"
      +("s"*Num($PicNum>1))+".")
ALERT(" The 4D View area contains"+String($ColNum)+" column"+
      ("s"*Num($ColNum>1))+ "and"+String($RowNum)+"row"+("s"*Num($RowNum>1))
      +". It contains"+String($PicNum)+"picture"+("s"*Num($PicNum>1))+".")
```

#### 参照

PV SET DOCUMENT PROPERTY

#### 定数

PV Document properties テーマ

## PV SET DOCUMENT INFO

### PV SET DOCUMENT INFO (エリア; タイトル; 題名; 作成者; 会社名; コメント)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
タイトル	文字列	→ ドキュメントのタイトル
題名	文字列	→ ドキュメントの題名
作成者	文字列	→ ドキュメントの作成者
会社名	文字列	→ 会社名
コメント	テキスト	→ コメント

#### 説明

**PV SET DOCUMENT INFO** コマンドは、引数<タイトル>、<題名>、<作成者>、<会社名>、<コメント>に渡された情報を<エリア>のドキュメントに関連付けます。この情報は、ドキュメントの「ドキュメント情報」ダイアログボックスに表示される項目に相当します（「ツール」メニューの「ドキュメント情報」メニューコマンド）。

#### 例題

フォームに組み込まれた4D Viewエリアの場合、ユーザがこのフォームを使用してレコードの作成や修正を行うたびに、このエリアに関連する情報を更新します。

```

C_STRING(255;$Title) `ドキュメントのタイトル
C_STRING(255;$Subject) `ドキュメントの題名
C_STRING(255;$Author) `ドキュメントの作成者
C_STRING(255;$Company) `会社名
C_TEXT($Comment) `コメント
    `ドキュメントの作成日と時間、および最終更新日と時間
C_DATE($CreationDate;$ModificationDate)
C_TIME($CreationTime;$ModificationTime)
    `ドキュメントに関する情報を取得する
PV GET DOCUMENT INFO($1;$Title;$Subject;$Author;$Company;$Comment;
$CreationDate;$CreationTime;$ModificationDate;$ModificationTime)
$Title:=Request("ドキュメントのタイトルは?";$Title)
$Subject:=Request("ドキュメントの題名は?";$Subject)
$Author:=Request("作成者の名前は?";$Author)
$Company:=Request("会社名は?";$Company)
$Comment:=Request("コメントは?";$Comment)
⇒ PV SET DOCUMENT INFO(Area;$Title;$Subject;$Author;$Company;$Comment)
    `情報を更新する

```

## 参照

PV GET DOCUMENT INFO

## PV GET DOCUMENT INFO

---

**PV GET DOCUMENT INFO** (エリア; タイトル; 題名; 作成者; 会社名; コメント; 作成日; 作成時刻; 変更日; 変更時刻)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
タイトル	文字列	←	ドキュメントのタイトル
題名	文字列	←	ドキュメントの題名
作成者	文字列	←	ドキュメントの作成者
会社名	文字列	←	会社名
コメント	テキスト	←	コメント
作成日	日付	←	作成日
作成時刻	時間	←	作成時刻
変更日	日付	←	最終変更日
変更時刻	時間	←	最終変更時刻

## 説明

**PV GET DOCUMENT INFO** コマンドは、4D View<エリア>に表示されたドキュメントの情報を、引数<タイトル>、<題名>、<作成者>、<会社名>、<コメント>に納めて取得します。この情報は、「ユーザ」モードで入力するか、または **PV SET DOCUMENT INFO** コマンドを使用してプログラムから設定することができます。

また、このコマンドは、ドキュメントの作成/変更の日付および時刻に関する情報を、引数<作成日>、<作成時刻>、<変更日>、<変更時刻>に納めて返します。これらの情報は、ドキュメントの保存時にOSによって自動的に更新されます。

## 例題

**PV SET DOCUMENT INFO** コマンドの例題を参照してください。

## 参照

PV SET DOCUMENT INFO

## PV Panes、はじめに

---

4D View エリアの異なる箇所を同時に表示するため、他の部分の表示に影響を及ぼさないようにこのエリアの一部をスクロールしたい場合があります。これら各部分のことを“区画”と呼びます。

水平区画は、表示される2つの境界線の間に位置するスペースです。例えば、スプレッドシートの上部分、縦スクロールバー上のセパレータ、スプレッドシートの下部分に分かれます。

垂直区画は、表示される2つのターゲットの間に位置するスペースです。例えば、スプレッドシートの左部分、横スクロールバー上のセパレータ、スプレッドシートの右部分に分かれます。

このように、デフォルトとして水平区画と垂直区画があり、それぞれエリア全体を対象としています。

また、スプレッドシートの2つの範囲のそれぞれで、同時に複数の区画を設定することができます。

このテーマ内のコマンドを使用すると、4D View エリアの各区画を操作することができます。例えば、区画（水平または垂直）の追加や削除、区画のプロパティの取得や設定などを行うことができます。

## PV ADD VERT SPLITTER

---

### PV ADD VERT SPLITTER(エリア; 分割ボックス; 位置; ロック)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
分割ボックス	整数	→ 垂直方向のセパレータの番号
位置	整数	→ 最後のセパレータに関し、セパレータの位置 (ピクセル単位)
ロック	整数	→ 0=ロックしない; 1=ロック

#### 説明

**PV ADD VERT SPLITTER** コマンドは、<分割ボックス>に指定した番号の垂直分割ボックスを新しく<エリア>に作成します。分割ボックスは<エリア>の左側境界からの<位置> (ピクセル単位で指定) に作成されます。

引数<ロック>に1を指定すると、区画のサイズは変更できません。0を指定した場合は、ユーザが自由に区画のサイズを変更することができます。

#### 注:

- 区画の本来の位置が行ヘッダの右側に配置されていたのと同様に、区画の位置はこれからヘッダの幅や、<エリア>におけるヘッダの表示状況に左右されません。
- 垂直区画の最小サイズは8ピクセルです。
- 2番目の引数として **pv vert pane count** 定数を渡した上で、**PV Get area property** 関数を実行すると、エリアにある垂直区画の数を調べることができます。垂直分割ボックスがまだ設定されていない場合、**PV Get area property** 関数は1を返します。この場合、唯一存在する区画はエリア全体ということになります。

#### 例題

20数カラムからなるスプレッドシートを例に考えてみましょう。1番目のカラムには参照番号 (例えば製品コード) が格納されており、ユーザがこのセルを変更可能かどうかに関わらず、これは常に表示されていなければなりません。このカラム A を表示する垂直区画を作成してみましょう。

```
C_LONGINT($ColumnWidth)`カラム A の幅 (ピクセル単位)  
$ColumnWidth:=PV Get column width(Area; 1)`カラム A  
⇒ PV ADD VERT SPLITTER(Area;1;$ColumnWidth;0)`サイズ変更可
```

#### 参照

PV ADD HOR SPLITTER、PV REMOVE VERT SPLITTER

## PV ADD HOR SPLITTER

### PV ADD HOR SPLITTER (エリア; 区画; 位置; ロック)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
分割ボックス	整数	→ 水平方向のセパレータの番号
位置	整数	→ 最後のセパレータに関し、セパレータの位置 (ピクセル単位)
ロック	整数	→ 0=ロックしない; 1=ロック

#### 説明

**PV ADD HOR SPLITTER** コマンドは、<分割ボックス>に指定した番号の水平分割ボックスを新しく<エリア>に作成します。分割ボックスは<エリア>の最後の分割ボックスからの<位置> (ピクセル単位で指定) に作成されます。エリアに分割ボックスが設定されていない場合は、エリアの上側境界からの<位置>に作成されます (ツールバーの外側)。

引数<ロック>に1を指定すると、区画のサイズは変更できません。0を指定した場合は、ユーザが自由に区画のサイズを変更することができます。

#### 注:

- 区画の本来の位置がカラムヘッダの下側に配置されていたのと同様に、区画の位置はこれらヘッダの高さや、<エリア>におけるヘッダの表示状況に左右されません。
- 水平区画の最小サイズは8ピクセルです。
- 2番目の引数として **pv hor pane count** 定数を渡した上で、**PV Get area property** 関数を実行すると、エリアにある水平区画の数を調べることができます。水平分割ボックスがまだ設定されていない場合、**PV Get area property** 関数は1を返します。この場合、唯一存在する区画はエリア全体ということになります。

#### 例題

エリアに既存する区画に続けて、30ピクセルの高さの水平区画を追加します。

```

C_LONGINT($HeaderHeight) `カラムヘッダの高さ (ピクセル単位)
C_LONGINT($HorPaneNum) `既存する水平区画の数
    `水平区画数
$HorPaneNum:=PV Get area property(Area;pv hor pane count)
    `最後のセパレータの位置
$Position:=PV Get hor pane property(Area;$HorPaneNum;pv pane size)
    +30`+30ピクセル
⇒ PV ADD HOR SPLITTER(Area;$HorPaneNum;$Position;0) `サイズ変更可
  
```

## 参照

PV ADD VERT SPLITTER、PV REMOVE HOR SPLITTER

## PV REMOVE VERT SPLITTER

---

### PV REMOVE VERT SPLITTER (エリア; 区画)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
分割ボックス	整数	→ 垂直区画の番号

### 説明

**PV REMOVE VERT SPLITTER** コマンドは、<分割ボックス>に渡した番号の垂直分割ボックスを<エリア>から削除します。

このコマンドを使用して、ユーザあるいは**PV ADD VERT SPLITTER** コマンドにより作成されたあらゆるタイプの垂直区画を削除することができます。

### 例題

最後の垂直区画を削除します。

```
C_LONGINT($VertPaneNum) ` 垂直区画数  
` 垂直区画の数  
$VertPaneNum:=PV Get area property(Area;pv vert pane count)  
⇒ PV REMOVE VERT SPLITTER(Area;$VertPaneNum)
```

### 参照

PV ADD VERT SPLITTER、PV REMOVE HOR SPLITTER

## PV REMOVE HOR SPLITTER

---

### PV REMOVE HOR SPLITTER (エリア; 区画)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
分割ボックス	整数	→ 水平区画の番号

### 説明

**PV REMOVE HOR SPLITTER** コマンドは、<分割ボックス>に渡した番号の水平分割ボックスを<エリア>から削除します。

このコマンドを使用して、ユーザあるいは**PV ADD HOR SPLITTER** コマンドにより作成されたあらゆるタイプの水平区画を削除することができます。

### 例題

最後の水平区画を削除します。

```
C_LONGINT($HorPaneNum) `水平区画数
`水平区画の数
$HorPaneNum:=PV Get area property(Area;pv hor pane count)
⇒ PV REMOVE HOR SPLITTER(Area;$HorPaneNum)
```

### 参照

PV ADD HOR SPLITTER、PV REMOVE VERT SPLITTER

## PV SET VERT PANE PROPERTY

---

### PV SET VERT PANE PROPERTY (エリア; 区画; プロパティ; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
区画	整数	→ 垂直区画の番号
プロパティ	倍長整数	→ プロパティ番号
値	倍長整数	→ プロパティの値

### 説明

**PV SET VERT PANE PROPERTY** コマンドは、<エリア>において<区画>に渡された番号の垂直区画に関する<プロパティ>の<値>を設定します。

<プロパティ>を定義するには、**PV Pane properties** 定数を使用します。

注： **pv pane relative scroll** 定数は、“PV SET...” コマンドでのみ使用することができます。

### 例題

1 番目の垂直区画を 30 ピクセル拡張します。

```
C_LONGINT($Size) `1 番目の垂直区画のサイズ
$Size:=PV Get vert pane property(Area;1;pv pane size)
⇒ PV SET VERT PANE PROPERTY(Area;1;pv pane size;$Size+30)
`30 ピクセル増やす
```

### 参照

PV Get vert pane property、PV SET HOR PANE PROPERTY

### 定数

PV Pane properties テーマ

## PV SET HOR PANE PROPERTY

---

### PV SET HOR PANE PROPERTY (エリア; 区画; プロパティ; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
区画	整数	→ 水平区画の番号
プロパティ	倍長整数	→ プロパティ番号
値	倍長整数	→ プロパティの値

#### 説明

**PV SET HOR PANE PROPERTY** コマンドは、<エリア>において<区画>に渡された番号の水平区画に関する<プロパティ>の<値>を設定します。

<プロパティ>を定義するには、**PV Pane properties** 定数を使用します。

注：pv pane relative scroll 定数は、“PV SET...” コマンドでのみ使用することができません。

#### 例題

1 番目の水平区画を 30 ピクセル拡張します。

```

C_LONGINT($Size) `1 番目の水平区画のサイズ
$Size:=PV Get hor pane property(Area;1;pv pane size)
⇒ PV SET HOR PANE PROPERTY(Area;1;pv pane size;$Size+30)
`30 ピクセル増やす

```

#### 参照

PV Get hor pane property、PV SET VERT PANE PROPERTY

#### 定数

PV Pane properties テーマ

## PV Get vert pane property

---

### PV Get vert pane property (エリア; 区画; プロパティ) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
区画	整数	→ 垂直区画の番号
プロパティ	倍長整数	→ プロパティ番号
戻り値	倍長整数	← プロパティの値

### 説明

**PV Get vert pane property** 関数は、<エリア>において<区画>に渡された番号の垂直区画に関する<プロパティ>の現在値を返します。

<プロパティ>を定義するには、**PV Pane properties** 定数を使用します。

注：pv pane relative scroll 定数は、“PV SET...” コマンドでのみ使用することができません。

### 例題

**PV SET VERT PANE PROPERTY** コマンドの例題を参照してください。

### 参照

PV Get hor pane property、PV SET VERT PANE PROPERTY

### 定数

PV Pane properties テーマ

## PV Get hor pane property

---

### PV Get hor pane property (エリア; 区画; プロパティ) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
区画	整数	→ 水平区画の番号
プロパティ	倍長整数	→ プロパティ番号
戻り値	倍長整数	← プロパティの値

### 説明

**PV Get hor pane property** 関数は、<エリア>において<区画>に渡された番号の水平区画に関する<プロパティ>の現在値を返します。

<プロパティ>を定義するには、**PV Pane properties** 定数を使用します。

注：pv pane relative scroll 定数は、“PV SET...” コマンドでのみ使用することができません。

### 例題

**PV ADD HOR PANE** コマンドと **PV SET HOR PANE PROPERTY** コマンドの例題を参照してください。

### 参照

PV Get vert pane property、PV SET HOR PANE PROPERTY

### 定数

PV Pane properties テーマ



## PV Pictures、はじめに

---

このテーマ内のコマンドや関数を使用して、4D Viewエリア内のピクチャを操作することができます。

プログラムを使用して、ピクチャの挿入や削除を行うことができます。また、これらのコマンドにより、任意のピクチャのプロパティを設定したり、変更することも可能です。ピクチャのプロパティを変更すると、その外観や透過度、サイズ、位置などを変えることができます。

### ピクチャの位置

プログラムにより貼り付けられたピクチャは、アクティブセルの左上隅との関係から自動的に位置決めされます。しかし、ピクチャはセル内に挿入されるのではなく、セルの上に配置されます。つまりピクチャは、セルではなくドキュメントに付属しています。従って、ピクチャサイズに合わせて列と行のサイズが調整されることはありません。

### ピクチャ番号について

このテーマ内のすべてのコマンドでは、4D Viewエリアに挿入されたピクチャを参照する際に、引数<ピクチャ番号>を使用しています。この引数は、エリアにおけるピクチャのインデックス番号であり、ピクチャの挿入時に4D Viewにより設定されます。挿入された各ピクチャはインデックス番号を受け取り、この番号はプログラムあるいはユーザーによって追加されます。

この番号はエリアに対してユニークですが、絶対的なものではありません。つまり、エリアでピクチャが削除されると、削除されたピクチャよりも大きなインデックス番号が付けられたピクチャはすべて、その番号が小さくなります。

4D Viewエリアに貼り付けられたピクチャの番号を随時調べるには、**PV Get document property** コマンドと **pv picture number** 定数を使用してください。

## PV Add picture

---

**PV Add picture**(エリア; ピクチャ{; 式{; テーブル番号{; フィールド番号}}}) →  
倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
ピクチャ	ピクチャ	→ 4D ピクチャ
式	文字列	→ 4D ピクチャを返す任意の式
テーブル番号	整数	→ テーブル番号
フィールド番号	整数	→ フィールド番号
戻り値	倍長整数	← ピクチャ番号

### 説明

**PV Add picture**関数は、<エリア>のカレントセルの位置にピクチャをペーストし、ピクチャのユニークなID番号を返します。このID番号は、“PV Pictures” テーマ内の他のコマンドで使用することができます。

ピクチャは、有効な4Dピクチャでなければなりません。追加されるピクチャは以下のタイプです。

- ピクチャ変数：この場合、引数<ピクチャ>に変数名を渡します。その他の引数は省略することができます。
- 4D式：この場合、引数<式>に式の名前を渡します（引数<ピクチャ>は使用されず、後の引数は省略できます）。例えば、ピクチャ変数やピクチャフィールドの参照（“[テーブル]ピクチャフィールド”）を返す4Dメソッド名を引数<式>に納めることができます。
- ピクチャフィールド番号：この場合、引数<テーブル番号>および<フィールド番号>にテーブルおよびフィールドの番号を渡します（引数<ピクチャ>および<式>は使用されません）。

4D Viewは、エリアにペーストされたピクチャと元のピクチャとの間に動的参照を維持します。4D側の元のピクチャに対して何らかの修正が加えられると、エリアにペーストされたピクチャへその変更が反映されます。

### 例題

カレントレコードとなっている顧客の写真を4D Viewエリアのカレントセルに貼り付けます。

```
C_LONGINT($PicRef) `追加されるピクチャの参照番号  
C_PICTURE($Picture) `空のピクチャ（無視される）  
⇒ $PicRef:=PV Add picture(Area;$Picture;"";Table(->[Clients]);Field(->[Clients]Photo))
```

## 参照

PV Get picture、PV REMOVE PICTURE

## PV REMOVE PICTURE

---

### PV REMOVE PICTURE (エリア; ピクチャ番号)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
ピクチャ番号	倍長整数	→	ピクチャ番号

## 説明

**PV REMOVE PICTURE** コマンドは、<ピクチャ番号>で指定した番号のピクチャを<エリア>から削除します。

注：あるピクチャが4D View エリアから削除されると、エリア内にある他のピクチャのインデックス番号が削除されたものよりも大きい場合には、その番号が振り替えられません。詳細については、「PV Pictures、はじめに」の説を参照してください。

## 例題

4D View エリアに最初に追加されたピクチャを削除します。

⇒ ***PV REMOVE PICTURE***(Area;1)

## 参照

PV Add picture

## PV Get picture

---

### PV Get picture (エリア; ピクチャ番号) → ピクチャ

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
ピクチャ番号	倍長整数	→ ピクチャ番号
戻り値	ピクチャ	← ピクチャ

### 説明

**PV Get picture** 関数は、<エリア>内の<ピクチャ番号>で指定した番号のピクチャを返します。

### 例題

ピクチャ番号 1 のピクチャをカレントセルにコピーします。

```
C_LONGINT($PicRef) `追加されるピクチャの参照番号
C_PICTURE($picNum) `コピーするピクチャ
⇒ $Picture:=PV Get picture(Area;1) `ピクチャ番号 1
   $PicRef:=PV Add picture(Area;$Picture;0;0) `カレントセルにコピーする
```

### 参照

PV Add picture

## PV Create picture

**PV Create picture** (エリア; 左; 上; 右; 下; 空セルを無視) → ピクチャ

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	倍長整数	→ 左側セルの列番号
上	倍長整数	→ 一番上のセルの行番号
右	倍長整数	→ 右側セルの列番号
下	倍長整数	→ 一番下のセルの行番号
空セルを無視	整数	→ 0= 空のセルを無視しない; 1= 空のセルを無視
戻り値	ピクチャ	← 領域内のセルのピクチャ

### 説明

**PV Create picture** 関数は、引数<左>、<上>、<右>、<下>で指定されたセル範囲のピクチャを返します。

引数<空セルを無視>に1を指定すると、空ではないセルだけがピクチャの一部となります。空ではない最後のセルの座標（右下）が、指定された枠の座標<右>と<下>よりも小さい場合、引数<左>、<上>、<右>、<下>で指定された枠の大きさが縮小されます。

### 例題

次のコードは、B2、E2、B5、E5で区切られた範囲におけるセル内容の図をピクチャフィールドに代入します。

```

`ピクチャを取得し、フィールドに割り当てる
⇒ [Templates]ReducedView:=PV Create picture(Area;2;2;5;5;0)

```

### 参照

PV COPY TO BLOB

## PV SET PICTURE PROPERTY

---

### PV SET PICTURE PROPERTY (エリア; ピクチャ番号; プロパティ; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
ピクチャ番号	倍長整数	→ ピクチャ番号
プロパティ	倍長整数	→ プロパティ番号
値	倍長整数	→ プロパティの値

#### 説明

**PV SET PICTURE PROPERTY** コマンドは、指定された<プロパティ>の<値>を<ピクチャ番号>のピクチャに設定します。

引数<プロパティ>を定義するには、**PV Picture properties** 定数を使用します。

また、引数<値>の定義には、**PV Picture mapping mode** テーマ内の定数を使用することができます。

#### 例題

ピクチャ番号1のピクチャの表示フォーマットを、“scaled centered” に設定します。

⇒ **PV SET PICTURE PROPERTY**(Area;1;pv picture mapping mode;  
pv mapping scaled centered prop)

#### 参照

PV Get picture property

#### 定数

PV Picture properties および PV Picture mapping mode テーマ

## PV Get picture property

### PV Get picture property (エリア; ピクチャ番号; プロパティ) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
ピクチャ番号	倍長整数	→ ピクチャ番号
プロパティ	倍長整数	→ プロパティ番号
戻り値	倍長整数	← プロパティの値

### 説明

**PV Get picture property** 関数は、<ピクチャ番号>で指定された<エリア>のピクチャに関するプロパティの値を返します。

引数<プロパティ>を定義するには、**PV Picture properties** 定数を使用します。

また、引数<プロパティ>に **pv picture mapping mode** の値を渡した場合には、戻り値を比較するために **PV Picture mapping mode** テーマ内の定数を使用することができます。

### 例題

このメソッドは、ピクチャ番号1のピクチャに関する情報を表示します。

```

C_INTEGER($Index) ` プロパティ配列のループ指標
C_LONGINT($Value) ` オプションに対応する値
ARRAY STRING($PropertiesCodes;12) ` プロパティのコード
ARRAY STRING(80;$PropertiesLabels;12) ` プロパティのラベル
    ` プロパティ配列の初期化
$PropertiesCodes{1}:=pv picture column
$PropertiesLabels{1}:="参照列"
$PropertiesCodes{2}:=pv picture row
$PropertiesLabels{2}:="参照行"
$PropertiesCodes{3}:=pv picture horz offset
$PropertiesLabels{3}:="H オフセット"
$PropertiesCodes{4}:=pv picture vert offset
$PropertiesLabels{4}:="V オフセット"
$PropertiesCodes{5}:=pv picture data width
$PropertiesLabels{5}:="実際の幅"
$PropertiesCodes{6}:=pv picture data height
$PropertiesLabels{6}:="実際の高さ"
$PropertiesCodes{7}:=pv picture display width
$PropertiesLabels{7}:="表示幅"

```

```

$PropertiesCodes{8}:=pv picture display height
$PropertiesLabels{8}:="表示高さ"
$PropertiesCodes{9}:=pv picture background
$PropertiesLabels{9}:="背景"
$PropertiesCodes{10}:=pv picture mapping mode
$PropertiesLabels{10}:="マッピング"
$PropertiesCodes{11}:=pv picture fixed size
$PropertiesLabels{11}:="固定サイズ"
$PropertiesCodes{12}:=pv picture locked
$PropertiesLabels{12}:="ロック"
$PictureInfo:="ピクチャ番号1の情報："+Character(Carriage return)
For($Index;1;12)`異なるプロパティを調査
    `プロパティの読み込み
⇒    $Value:=PV Get picture property(Area;1;$PropertiesCodes{$Index})
    `情報を更新
    $PictureInfo:=$PictureInfo+$PropertiesLabels{$Index}+
        "："+String($Value)+"。"

End for
ALERT($PictureInfo)`情報を表示

```

参照

PV SET PICTURE PROPERTY

定数

PV Picture properties および PV Picture mapping mode テーマ

## PV Printing、はじめに

このテーマ内のコマンドは、4th Dimension内でプログラムを使用してスプレッドシートを印刷する際の管理を行います。印刷する要素（ヘッダ、脚注等）の設定や取得を行ったり、値またはフォーミュラのいずれを印刷するかを選択することができます。

ユーザに「ファイル」メニューの「プリント」コマンドを選択させずに、ドキュメントの印刷を行いたい場合、これらのコマンドが非常に役立ちます。

## PV PRINT

---

### PV PRINT (エリア)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア

#### 説明

**PV PRINT** コマンドは、引数として渡された4D View<エリア>を印刷します。

**pv cmd print page setup** および **pv cmd print preview** 定数を指定した上で、**PV EXECUTE COMMAND** コマンドを実行すると、用紙設定と印刷プレビューを利用することができます。

4D View エリアのリフレッシュオプションに注意してください。リフレッシュが自動ではない場合は、プリント領域の印刷やプレビューを行う前に必ずリフレッシュを実行してください。

#### 例題

フッタ設定を行った上で印刷を実行します。

```
PV SET PRINT HEADER(Area;pv footer center;" #D" Printed) `ページフッタを設定  
CONFIRM("値とフォーミュラのいずれを印刷しますか?";"フォーミュラ";"値")  
If(OK=1)  
    PV PRINT FORMULAS(Area) `フォーミュラを印刷  
Else  
⇒ PV PRINT(Area) `値を印刷  
End if
```

#### 参照

PV PRINT FORMULAS

## PV PRINT FORMULAS

---

### PV PRINT FORMULAS (エリア)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア

### 説明

**PV PRINT FORMULAS** コマンドは、引数として渡された 4D View <エリア> で使用されているフォーミュラをすべて印刷します。

### 例題

**PV PRINT** コマンドの例題を参照してください。

### 参照

PV PRINT

## PV SET PRINT HEADER

---

### PV SET PRINT HEADER (エリア; ヘッダ; 文字列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
ヘッダ	倍長整数	→ ヘッダの位置
文字列	文字列	→ ヘッダに設定する文字列

#### 説明

**PV SET PRINT HEADER** コマンドは、<ヘッダ>で指定された位置に、<エリア>のヘッダまたはフッタとなる<文字列>を設定します。

引数<ヘッダ>の定義には、**PV Headers& footers** 定数を使用します。

引数<文字列>には、次の特殊文字を挿入することができます。

#d	Current date abbreviated	Wed, Apr 3, 1996
#(Macintosh)	Current date in short form	04/03/1996
#c(Windows)	Forced special	04/03/1996
#D	Current date in long form	Wednesday, April 3, 1996
#p	Page number	2
#h	Time without seconds	09:42
#H	Time with seconds	09:42:47
#F	Table or area name	Forecast(SP) or _Forecast
#P	Total page number	10

#### 例題

**PV PRINT** コマンドおよび **PV Get print header** 関数の例題を参照してください。

#### 参照

PV Get print header

#### 定数

PV Headers & footers テーマ

## PV Get print header

---

### PV Get print header (エリア; ヘッダ) → 文字列

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
ヘッダ	倍長整数	→ ヘッダの位置
戻り値	文字列	← ヘッダの文字列

### 説明

**PV Get print header**関数は、<ヘッダ>で指定された位置に設定されているヘッダまたはフッタ文字列を返します。

引数<ヘッダ>の定義には、**PV Headers & footers**定数を使用します。

### 例題

中央のヘッダのテキストをページフッタに移動します。

```

C_TEXT($Header) `中央のヘッダテキスト
⇒ $Header:=PV Get print header(Area;pv header center)
                                `中央のヘッダテキストを取得
PV SET PRINT HEADER(Area;pv header center;") `ヘッダをクリア
PV SET PRINT HEADER(Area;pv footer center;$Header) `フッタに渡す
PV PRINT(Area)

```

### 参照

PV SET PRINT HEADER

### 定数

PV Headers& footers テーマ

## PV SET PRINT PROPERTY

---

### PV SET PRINT PROPERTY (エリア; プロパティ; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
プロパティ	倍長整数	→ プロパティ番号
値	倍長整数	→ プロパティの値

#### 説明

**PV SET PRINT PROPERTY** コマンドは、指定された4D View<エリア>に対して<プロパティ>の<値>を設定します。

引数<プロパティ>を定義するには、**PV Print properties** 定数を使用します。引数<プロパティ>および<値>の両方で使用される定数について、以下のリストで説明します。

#### pv print left margin

左マージンとは、用紙の左端（左デッドマージンを含む）とテキストとの間の領域です。  
関連する値：ピクセル単位のマージン

#### pv print top margin

上マージンとは、用紙の上端（上デッドマージンを含む）とヘッダとの間の領域です。  
関連する値：ピクセル単位のマージン

#### pv print right margin

右マージンとは、用紙の右端（右デッドマージンを含む）とテキストとの間の領域です。  
関連する値：ピクセル単位のマージン

#### pv print bottom margin

下マージンとは、用紙の下端（下デッドマージンを含む）とフッタとの間の領域です。  
関連する値：ピクセル単位のマージン

注：デッドマージンに関する詳細は、後述の説明を参照してください。

### pv print repeat first column

各ページに印刷する領域の最初の列番号を示します。この定数は、必ず **pv print repeat last column** 定数と組み合わせて使用します。

関連する値：列番号

### pv print repeat last column

各ページに印刷する領域の最後の列番号を示します。この定数は、必ず **pv print repeat first column** 定数と組み合わせて使用します。

関連する値：列番号

### pv print repeat first row

各ページに印刷する領域の最初の行番号を示します。この定数は、必ず **pv print repeat last row** 定数と組み合わせて使用します。

関連する値：行番号

### pv print repeat last row

各ページに印刷する領域の最後の行番号を示します。この定数は、必ず **pv print repeat first row** 定数と組み合わせて使用します。

関連する値：行番号

### pv print headers

行ヘッダと列ヘッダの印刷を行います。関連する値：pv value on または pv value off

- pv value on：行ヘッダと列ヘッダを印刷する。
- pv value off：行ヘッダと列ヘッダを印刷しない。

### pv print centered

ページの中央に印刷を行うかどうかを設定します。関連する値：pv value on または pv value off

- pv value on：ページの中央に印刷を行う。
- pv value off：ページの中央に印刷を行わない。

### pv print adjust area

印刷領域の調整を行うかどうかを設定します。関連する値：pv value onまたはpv value off

- pv value on：印刷領域の調整を行う。
- pv value off：印刷領域の調整を行わない。

### pv print frame each page

印刷される各ページの周囲に枠を印刷するかどうかを設定します。関連する値：pv value onまたはpv value off

- pv value on：各ページに枠を印刷する。
- pv value off：枠を印刷しない。

### pv print grid

領域上にグリッドを印刷するかどうかを設定します。関連する値：pv value onまたはpv value off

- pv value on：グリッドを印刷する。
- pv value off：グリッドを印刷しない。

### pv print orientation

印刷時に、用紙の方向の設定や読み込みを行います。関連する値：PV Print values テーマ内の定数。

- pv portrait orientation：用紙は縦方向に合わせる。
- pv landscape orientation：用紙は横方向に合わせる。

### pv print paper width

この定数はリードオンリー（**PV Get print property** 関数）です。用紙の幅を設定します。関連する値：ピクセル単位の幅

### pv print paper height

この定数はリードオンリー（**PV Get print property** 関数）です。用紙の高さを設定します。関連する値：ピクセル単位の高さ

## Dead Margin

デッドマージンとは、用紙の端にある印刷不可領域のことです。この領域はプリンタドライバにより設定されています。

### pv print dead left margin

この定数は読み取り専用で (**PV Get print property** コマンド)、左デッドマージンのサイズをピクセル単位で返します。

### pv print dead top margin

この定数は読み取り専用で (**PV Get print property** コマンド)、上デッドマージンのサイズをピクセル単位で返します。

### pv print dead right margin

この定数は読み取り専用で (**PV Get print property** コマンド)、右デッドマージンのサイズをピクセル単位で返します。

### pv print dead bottom margin

この定数は読み取り専用で (**PV Get print property** コマンド)、下デッドマージンのサイズをピクセル単位で返します。

```
$margin:=PV Get print property(area; pv print dead bottom margin)  
`下デッドマージンのサイズをピクセル単位で返す
```

## 参照

PV Get print property

## 定数

PV Print properties および PV Print values テーマ

## PV Get print property

---

### PV Get print property (エリア; プロパティ) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
プロパティ	倍長整数	→ プロパティ番号
戻り値	倍長整数	← プロパティの値

#### 説明

**PV Get print property** 関数は、指定された4D View<エリア>の<プロパティ>の現在値を返します。

引数<プロパティ>を定義するには、**PV Print properties** 定数を使用します。これらの定数に関する詳細は、**PV SET PRINT PROPERTY** コマンドの説明を参照してください。

注：“pv print dead...” で始まる4つの定数、ならびに pv print paper width と pv print paper height は、読み取り専用です。

#### 例題

実際の印刷可能領域を知りたいものとします。

```
C_LONGINT($paperWidth;$paperHeight)
C_LONGINT($bottomMargin;$topMargin;$rightMargin;$leftMargin)
C_LONGINT($usableWidth;$usableHeight)
```

```
$paperWidth:=PV Get print property(area; pv print paper width)
$paperHeight:=PV Get print property(area; pv print paper height)
$bottomMargin:=PV Get print property(area; pv print dead bottom margin)
$topMargin:=PV Get print property(area; pv print dead top margin)
$rightMargin:=PV Get print property(area; pv print dead right margin)
$leftMargin:=PV Get print property(area; pv print dead left margin)
```

```
$usableWidth:=$paperWidth-($rightMargin+$leftMargin)
$usableHeight:=$paperHeight-($topMargin+$bottomMargin)
```

#### 参照

PV SET PRINT PROPERTY

#### 定数

PV Print properties および PV Print values テーマ

## PV Selection、はじめに

---

このテーマ内のコマンドを使用すると、以下の事柄を実行することができます。

- 一連のセル（連続、または非連続の選択範囲）、行、列を選択する。
- 4D View エリアの現在の選択範囲を調べる。

### 選択範囲とセル範囲

コマンドに応じて、セルの選択範囲には連続したもの（範囲）と単独のものがあります。

セル範囲とは、連続する一連のセルのことで、例えばA1、A2、B1、B2のようなセルです。範囲は必ずしも選択されている必要はありません。

選択セルとは、高輝度表示された選択範囲内の一連のセルのことであり、例えばA1、A2、B1、B2、C15のようなセルです。

選択範囲には、1つ以上の範囲および1つ以上の単独のセル、あるいは1つ以上の単独セルのみが含まれます。

4D View エリアが新しく開かれると、他のスプレッドシートと同様にカーソルはセル上に置かれますが、マウスクリックや他の選択動作、あるいはコマンドを使用してユーザや開発者が明示的に指定しないかぎり、そのセルは選択されません。

## PV SELECT CELL

---

### PV SELECT CELL (エリア; 列; 行; アクション)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
列	倍長整数	→ セルの列番号
行	倍長整数	→ セルの行番号
アクション	整数	→ 選択動作

### 説明

**PV SELECT CELL** コマンドは、<列>と<行>が交差する位置にあるセルを選択します。

引数<アクション>により、選択セルが既に存在する場合に行う選択動作を定義することができます。新しい選択範囲を作成するか、そのセルを既存の選択範囲に加えることができます。

<アクション>の定義は、**PV Selection action** 定数を使用して行います。

### 例題

E2セルを選択したい場合について考えてみましょう。選択動作は、状況に応じて変わります（既存の選択セルがあるかどうか）。

```
        `既存の選択範囲を定義する配列
ARRAY LONGINT($Left;0) ` 左側のセルの列番号
ARRAY LONGINT($Top;0) ` 上側のセルの行番号
ARRAY LONGINT($Right;0) ` 右側のセルの列番号
ARRAY LONGINT($Bottom;0) ` 下側のセルの行番号
        `選択された範囲が存在する場合には、その座標を取得する
PV GET SELECTED RANGES LIST(Area;$Left;$Top;$Right;$Bottom)
If(Size of array($Left)=0) `現在の選択範囲はない
⇒      PV SELECT CELL(Area;5;2;pv selection set)
                                                `E2セルを現在の選択範囲に設定

Else
⇒      PV SELECT CELL(Area;5;2;pv selection add)
                                                `E2セルを現在の選択範囲に追加

End if
```

参照

PV Is cell selected

定数

PV Selection action テーマ

## PV Is cell selected

---

**PV Is cell selected** (エリア; 列; 行) → 整数

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
列	倍長整数	→	セルの列番号
行	倍長整数	→	セルの行番号
戻り値	整数	←	0= 未選択、1= 選択

説明

**PV Is cell selected**関数は、<列>と<行>で指定された<エリア>のセルが、現在の選択範囲に含まれる場合には1を、そうでない場合には0を返します。

例題

**PV SELECT CELL** コマンドの例題を参考してください。

参照

PV SELECT CELL

## PV SELECT RANGE

---

### PV SELECT RANGE (エリア; 左; 上; 右; 下; アクション)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	倍長整数	→ 左側のセルの列番号
上	倍長整数	→ 上側のセルの行番号
右	倍長整数	→ 右側のセルの列番号
下	倍長整数	→ 下側のセルの行番号
アクション	整数	→ 選択動作

### 説明

**PV SELECT RANGE** コマンドは、<左>、<上>、<右>、<下>で定義された範囲のセルを選択します。

引数<アクション>により、選択セルが既に存在する場合に行う選択動作を定義することができます。このセルを既存の選択範囲に加えるか、または選択範囲をこのセルだけに縮小することができます。<アクション>の定義は、**PV Selection action** 定数を使用して行います。

### 例題

セル範囲、E2、E3、F2、F3 を選択したい場合について考えてみます。選択動作は状況に応じて変わります（既存の選択セルがあるかどうか）。

```
        `既存の選択範囲を定義する配列
ARRAY LONGINT($Left;0) ` 左側のセルの列番号
ARRAY LONGINT($Top;0) ` 上側のセルの行番号
ARRAY LONGINT($Right;0) ` 右側のセルの列番号
ARRAY LONGINT($Bottom;0) ` 下側のセルの行番号
        `選択された範囲が存在する場合には、その座標を取得する
PV GET SELECTED RANGES LIST(Area;$Left;$Top;$Right;$Bottom)
If(Size of array($Left)=0) ` 現在の選択範囲はない
⇒      PV SELECT RANGE(Area;5;2;6;3;pv selection set)
        ` この範囲を現在の選択範囲に設定
Else
⇒      PV SELECT RANGE(Area;5;2;6;3;pv selection add)
        ` この範囲を現在の選択範囲に追加
End if
```

## 参照

PV GET CURRENT CELL、PV Is range selected

## 定数

PV Selection action テーマ

**PV Is range selected**

---

**PV Is range selected** (エリア; 左; 上; 右; 下) → 整数

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
左	倍長整数	→	左側のセルの列番号
上	倍長整数	→	上側のセルの行番号
右	倍長整数	→	右側のセルの列番号
下	倍長整数	→	下側のセルの行番号
戻り値	整数	←	0= 未選択、1= 選択

## 説明

**PV Is range selected** 関数は、引数<左>、<上>、<右>、<下>で定義された範囲のセルが、現在の選択範囲に含まれる場合には1を、そうでない場合には0を返します。

## 例題

**PV SELECT RANGE** コマンドの例題を参考してください。

## 参照

PV SELECT RANGE

## PV SELECT RANGES LIST

---

### PV SELECT RANGES LIST (エリア; 左; 上; 右; 下; アクション)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	配列	→ 左側のセルの列番号の配列
上	配列	→ 上側のセルの行番号の配列
右	配列	→ 右側のセルの列番号の配列
下	配列	→ 下側のセルの行番号の配列
アクション	整数	→ 選択動作

#### 説明

このコマンドは **PV SELECT RANGE** コマンドと同様の働きをしますが、引数<左>、<上>、<右>、<下>の配列に納められている座標を持つ複数のセル範囲に対して適用されます。

<アクション>の定義は、**PV Selection action** 定数を使用して行います（詳細については、**PV SELECT RANGE** コマンドの説明を参照してください）。

#### 例題

プログラムを使用して、サイズが少しずつ大きな5つの範囲を選択します。

```
C_INTEGER($Index) ` ループの指標
C_INTEGER($Number) ` 範囲の数
$Number:=5 ` 範囲内に5つのグループ
ARRAY LONGINT($Left;5)
ARRAY LONGINT($Top;5)
ARRAY LONGINT($Right;5)
ARRAY LONGINT($Bottom;5)
` 初期化
For($Index;1;5)
  $Left{$Index}:=$Index*3 ` 左境界
  $Top{$Index}:=$Index*6 ` 上境界
  $Right{$Index}:=$Index*4 ` 右境界
  $Bottom{$Index}:=$Index*7 ` 下境界
End for
```

⇒ **PV SELECT RANGES LIST**(Area;\$Left;\$Top;\$Right;\$Bottom;pv selection add)

## 参照

PV GET SELECTED RANGES LIST.

## 定数

PV Selection action テーマ

## PV GET SELECTED RANGES LIST

---

### PV GET SELECTED RANGES LIST (エリア; 左; 上; 右; 下)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
左	倍長整数 配列	← 左側のセルの列番号の配列
上	倍長整数 配列	← 上側のセルの行番号の配列
右	倍長整数 配列	← 右側のセルの列番号の配列
下	倍長整数 配列	← 下側のセルの行番号の配列

## 説明

**PV GET SELECTED RANGES LIST** コマンドは、<エリア>内の選択範囲の座標を取得し、引数として渡された配列に代入します。

## 例題

**PV SET CELL PROPERTY**、**PV SET RANGE PROPERTY**、**PV SELECT CELL** および **PV SELECT RANGE** コマンドの例題を参照してください。

## 参照

PV FIND ALL、PV SELECT RANGES LIST

## PV SELECT COLUMNS

---

### PV SELECT COLUMNS (エリア; 始め; 終わり; アクション)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
始め	倍長整数	→ 最初の選択列
終わり	倍長整数	→ 最後の選択列
アクション	整数	→ 選択動作

#### 説明

**PV SELECT COLUMNS** コマンドは、<始め>と<終わり>で指定された列番号の間に存在する<エリア>の列を選択します。

引数<アクション>により、選択列が既に存在する場合に行なう選択動作を定義することができます。これらの列を既存の選択範囲に加えるか、または選択範囲をこれらの列だけに縮小することができます。<アクション>の定義は、**PV Selection action** 定数を使用して行います。

#### 例題

カレントセルの列と行を共に選択したいものとします。

```
C_LONGINT($Column;$Row) `座標を取得
C_INTEGER($ColSelect;$OwSelect) `列/行が既に選択済みかどうかを調べる
PV GET CURRENT CELL(Area;$Column;$Row) `カレントセルの座標を取得
$ColSelect:=PV Is column selected(Area;$Column)
$RowSelect:=PV Is row selected(Area;$Row)
If($ColSelect=0) `列は未選択
⇒ PV SELECT COLUMNS(Area;$Column;$Column;pv selection add) `選択する
End if
If($RowSelect=0) `行は未選択
PV SELECT ROWS(Area;$Row;$Row;pv selection add) `選択する
End if
```

#### 参照

PV Is column selected

#### 定数

PV Selection action テーマ

## PV Is column selected

---

### PV Is column selected (エリア; 列) → 整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
列	倍長整数	→ 列番号
戻り値	整数	← 0= 未選択、1= 選択

#### 説明

**PV Is column selected**関数は、<エリア>内の列番号である<列>が現在の選択範囲に含まれる場合には1を、そうでない場合には0を返します。

#### 例題

**PV SELECT COLUMNS**の例題を参考してください。

#### 参照

PV SELECT COLUMNS

## PV SELECT ROWS

---

### PV SELECT ROWS (エリア; 始め; 終わり; アクション)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
始め	倍長整数	→ 最初の選択行
終わり	倍長整数	→ 最後の選択行
アクション	整数	→ 選択動作

#### 説明

**PV SELECT ROWS** コマンドは、<始め>と<終わり>で指定された行番号の間に存在する<エリア>の行を選択します。

引数<アクション>により、選択行が既に存在する場合に行なう選択動作を定義することができます。これらの行を既存の選択範囲に加えるか、または選択範囲をこれらの行だけに縮小することができます。<アクション>の定義は、**PV Selection action** 定数を使用して行います。

#### 例題

**PV SELECT COLUMNS** コマンドの例題を参照してください。

参照

PV Is row selected

定数

PV Selection action テーマ

## PV Is row selected

---

**PV Is row selected** (エリア; 行) → 整数

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
行	倍長整数	→	行番号
戻り値	整数	←	0= 未選択、1= 選択

説明

**PV Is row selected**関数は、<エリア>の行番号である<行>が現在の選択範囲に含まれる場合には1を、そうでない場合には0を返します。

例題

**PV SELECT COLUMNS** コマンドの例題を参照してください。

参照

PV SELECT ROWS

## PV Is all selected

---

### PV Is all selected (エリア) → 整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
戻り値	整数	← 0= 未選択、1= 選択

#### 説明

**PV Is all selected**関数は、〈エリア〉の一連のセルが選択されている場合には1を、そうでない場合には0を返します。

#### 例題

**PV SELECT ALL** コマンドの例題を参照してください。

#### 参照

PV SELECT ALL

## PV SELECT ALL

---

### PV SELECT ALL (エリア)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア

#### 説明

**PV SELECT ALL** コマンドは、〈エリア〉のセルをすべて選択します。

#### 参照

PV Is all selected



## PV Style、はじめに

---

このテーマ内のコマンドと関数を使用して、4D Viewエリアに関連付けたスタイルを管理することができます。

これらのコマンドにより、プログラムを通して既存のスタイルシートへのアクセスが可能になり、表示フォーマット、利用可能なフォント、色、属性等の各フォーマットプロパティを変更することができます。

さらに、これらのコマンドを使用すると、作成したドキュメント内からアプリケーションを管理したり、スタイルシートを更新することができます。

### スタイルシート

4D Viewでは、倍長整数タイプの参照番号を用いてスタイルシートにアクセスします。

デフォルトとして、3種類の固定タイプのスタイルシートがあります。

- 行/列ヘッダ
- セル
- ヘッダとフッタ

特定のエリアにリンクしている独自のスタイルシートの作成や変更、削除を行うことができます。これらのスタイルシートは、外部ドキュメント内、または4th Dimensionのデータ内に4D Viewエリアと一緒に保存されます。

## キャラクタフォント

デフォルトとして、4D View エリアではあらゆるシステムフォントを使用することができます。しかし、4D View エリアで使用可能な、または使用不可能なフォントを管理するコマンドもあります。4D View においてフォントを削除すると、そのフォントは関連するエリア内では使用できなくなります。従って、そのエリアからは削除されたフォントを呼び出すことができなくなりますが、その他の4D View エリア、および4Dや他のアプリケーションからは依然として使用可能です。

## フォーマット

4th Dimension と同様に、情報を表示する際にフォーマットが適用されます。

表示フォーマットの定義方法に関する詳細は、4Dの『デザインリファレンス』マニュアルを参照してください。

## PV Add style

---

### PV Add style (エリア; 名前) → 倍長整数

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
名前	文字列	→	スタイルシートの名前
戻り値	倍長整数	←	スタイルシートのID

### 説明

**PV Add style** 関数は、引数<名前>を使用して<エリア>にスタイルシートを追加し、そのIDを返します。

このスタイルシートの名前が既に存在している場合、**PV Add style** 関数はそのIDを返します（この番号は、**PV GET STYLE LIST** コマンドを使用して取得することもできます）。

### 例題

**PV SET STYLE PROPERTY** コマンドの例題を参照してください。

### 参照

PV GET STYLE LIST、PV Get style property、PV REMOVE STYLE、PV SET STYLE NAME、PV SET STYLE PROPERTY

## PV REMOVE STYLE

---

### PV REMOVE STYLE (エリア; スタイルシート)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
スタイルシート	倍長整数	→ スタイルシートのID

#### 説明

**PV REMOVE STYLE** コマンドは、<エリア>から<スタイルシート>を削除します。

注：そのエリアに追加されたスタイルのみ削除することができます。

#### 例題

このメソッドを使用して、不要なスタイルを削除することができます。

```

C_STRING(255;$StyleName) ` エリアの不要なスタイルの名前
ARRAY LONGINT($StyleNumArray;0) ` スタイル番号の配列
ARRAY STRING(255;$StyleNameArray;0) ` スタイル名の配列
C_INTEGER($Position) ` 番号配列と名前配列内における不要なスタイルの位置
$StyleName:="subparagraph" ` スタイル “Subparagraph” は不要
PV GET STYLE LIST(Area;$StyleNumArray;$StyleNameArray)
` 利用可能なスタイルのリスト
$Position:=Find in array($StyleNameArray;$StyleName) ` 不要なスタイルを検索
If($Position#-1) ` エリア内に不要なスタイルが存在するか？
⇒ PV REMOVE STYLE(Area;$StyleNumArray{$Position}) ` 削除
End if

```

#### 参照

PV Add style

## PV SET STYLE NAME

---

### PV SET STYLE NAME (エリア; スタイルシート; 名前)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
スタイルシート	倍長整数	→ スタイルシートのID
名前	文字列	→ スタイルシートの名前

### 説明

**PV SET STYLE NAME** コマンドは、引数<名前>に渡された文字列で<スタイルシート>の名前を変更します。

### 例題

このメソッドを使用して、スタイル名を変更することができます。

```
C_STRING(255;$StyleName) `名前を変更するエリアのスタイル名
C_STRING(255;$NewName) `エリアのスタイルに割り当てる新しい名前
ARRAY LONGINT($StyleNumArray;0) `スタイル番号の配列
ARRAY STRING(255;$StyleNameArray;0) `スタイル名の配列
C_INTEGER($Position) `番号配列と名前配列内での名前を変更するスタイルの位置
$StyleName:="subparagraph" `スタイル "Subparagraph" の名前を
$NewName:="Paragraph" ` "Paragraph" に変更する
PV GET STYLE LIST(Area;$StyleNumArray;$StyleNameArray)
`利用可能なスタイルのリスト
$Position:=Find in array($StyleNameArray;$StyleName) `変更するスタイルを検索
If($Position#-1) `エリア内に名前変更対象のスタイルが存在するか?
⇒ PV SET STYLE NAME(Area;$StyleNumArray{$Position};$NewName)
`名前を変更する

Else
ALERT("スタイル "+$StyleName+" はエリア内に存在しません。")
End if
```

### 参照

PV Add style

## PV GET STYLE LIST

---

### PV GET STYLE LIST (エリア; スタイルシート; 名前)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
スタイルシート	倍長整数配列	← スタイルシートIDの配列
名前	文字列配列	← スタイルシート名の配列

### 説明

**PV GET STYLE LIST** コマンドは、<エリア>内に存在する各スタイルシートのIDと名前を取得し、引数<スタイルシート>と<名前>に代入します。

### 例題

**PV REMOVE STYLE**、**PV SET STYLE NAME**、**PV SET STYLE PROPERTY** コマンドの例題を参照してください。

### 参照

PV Add style、PV REMOVE STYLE

## PV SET STYLE PROPERTY

---

### PV SET STYLE PROPERTY (エリア; スタイル; プロパティ; 値)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
スタイル	倍長整数	→ スタイルシートのID
プロパティ	倍長整数	→ プロパティ番号
値	倍長整数	→ プロパティの値

#### 説明

**PV SET STYLE PROPERTY** コマンドは、<スタイル>で指定した番号のスタイルシートに対して、<プロパティ>の<値>を設定します。

引数<プロパティ>を定義するには、**PV Style properties** テーマ内の定数を使用します。

引数<値>を定義するには、**PV Style values** テーマ内の定数を使用します。引数<値>に割り当てられる定数は、選択した<プロパティ>によって変わります。

これらの定数については、**PV SET CELL PROPERTY** コマンドの解説で詳しく説明しています。

デフォルトスタイルシートのプロパティを編集したい場合、**PV Style special values** 定数の1つを引数<スタイル>に渡します。

- **pv style cells** : セルのデフォルトスタイルシート (名前は “セル”)。
- **pv style col row headers** : 列ヘッダと行ヘッダのデフォルトスタイルシート (名前は “列/行ヘッダ”)。
- **pv style page footer header** : 印刷されるページのヘッダとフッタ (名前は “ページヘッダ&フッタ”)。

注: 4D Viewの「スタイル」メニューから「スタイルシート」コマンドを使用すると、これらのスタイルシートを編集することができます。

#### 例題

このメソッドを使用して、新しいスタイルのプロパティを定義することができます。この例題では、3つの各スタイルプロパティに値を設定し、カスタマイズを行います。

- “横揃え” プロパティは “左横揃え” に設定。
- “縦方向の行揃え” プロパティは “中央揃え” に設定。
- “回転” プロパティは “90度” に設定。

```

ARRAY LONGINT($ArrayProps;3) `スタイルシートプロパティの配列
ARRAY LONGINT($ArrayValues;3) `各プロパティの値
C_STRING(255;$StyleName) `エリアに追加するスタイルの名前
ARRAY LONGINT($StyleNumArray;0) `スタイル番号の配列
ARRAY STRING(255;$StyleNameArray;0) `スタイル名の配列
C_INTEGER($Index) `ループの指標
C_INTEGER($Position) `番号配列と名前配列内での新しいスタイルの位置
    `初期化
$ArrayProps{1}:=pv style hor alignment `対応するプロパティ...
$ArrayProps{2}:=pv style vert alignment
$ArrayProps{3}:=pv style rotation
$ArrayValues{1}:=pv value hor alignment left `...および値
$ArrayValues{2}:=pv value hor alignment center
$ArrayValues{3}:=pv value rotation 90
$StyleName:="subparagraph"
PV GET STYLE LIST(Area;$StyleNumArray;$StyleNameArray)
    `利用可能なスタイルのリスト
If(Find in array($StyleNameArray;$StyleName)=-1)
    `スタイル $StyleName が存在しない?
    $Position:=Size of array($StyleNameArray)+1 `このスタイルを追加する
    INSERT ELEMENT($StyleNameArray;$Position) `配列のサイズを変更
    INSERT ELEMENT($StyleNumArray;$Position)
    $StyleNameArray{$Position}:=$StyleName
    `新しいスタイルに名前を割り当てる
    `新しいスタイルに番号を割り当てる
    $StyleNumArray{$Position}:=PV Add style(Area;$StyleName)
    For($Index;1;Size of array($ArrayProps))
        `設定するすべてのプロパティをループする
        If(PV Get style property(Area;$StyleNumArray{$Position};
            $ArrayProps{$Index})
            # $ArrayValues{$Index})
            ⇒ PV SET STYLE PROPERTY(Area;$StyleNumArray
                {$Position};
                $ArrayProps{$Index};$ArrayValues{$Index})
        End if `プロパティが希望する値ではない?
    End for `スタイルシートのプロパティの配列 $ArrayProps を再調査
Else
    ALERT("スタイル "+$StyleName+" は既にエリア内に存在しています。")
End if

```

## 参照

PV Get style property

## 定数

PV Style properties および PV Style values テーマ

## PV Get style property

---

**PV Get style property** (エリア; スタイル; プロパティ) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
スタイル	倍長整数	→ スタイルシートのID
プロパティ	倍長整数	→ プロパティ番号
戻り値	倍長整数	← プロパティの値

## 説明

**PV Get style property** 関数は、<スタイル>で指定したスタイルシートの<プロパティ>の現在値を返します。

引数<プロパティ>を定義するには、**PV Style properties** テーマ内の定数を使用します。

引数<値>を定義するには、**PV Style values** テーマ内の定数を使用します。

## 例題

**PV SET STYLE PROPERTY** コマンドの例題を参照してください。

## 参照

PV SET STYLE PROPERTY

## 定数

PV Style properties および PV Style values テーマ

## PV Add font

---

### PV Add font (エリア; フォント名) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
フォント名	文字列	→ フォント名
戻り値	倍長整数	← フォントID

### 説明

**PV Add font** 関数は、<フォント名>で指定したフォントを<エリア>に追加し、そのIDを返します。

<フォント名>が既に存在している場合、**PV Add font** 関数はそのIDを返します（この番号は、**PV GET FONT LIST** コマンドを使用して取得することもできます）。

4D View エリアに関連付けたフォントの機能に関する詳細は、「PV Style、はじめに」の節を参照してください。

### 例題

このメソッドは、切り替えスイッチの役割を果たします。4D View からフォントを削除したり、またはそのフォントが既に削除されていれば4D View に関連付けます。

```

C_STRING(255;$FontName) ` エリアに追加または削除するフォント名
ARRAY LONGINT($FontNumArray;0) ` フォント番号の配列
ARRAY STRING(255;$FontNameArray;0) ` フォント名の配列
C_INTEGER($Position) ` 番号配列と名前配列内でのフォントの位置
$FontName:="Arial"
PV GET FONT LIST(Area;$FontNumArray;$FontNameArray)
` 利用可能なフォントのリスト
$Position:=Find in array($FontNameArray;$FontName)
If($Position=-1) ` エリアでは、フォント $FontName が使用できない？
    $Position:=Size of array($FontNameArray)+1 ` このフォントを追加
    INSERT ELEMENT($FontNameArray;$Position) ` 配列を...
    INSERT ELEMENT($FontNumArray;$Position) ` ...リサイズする
    $FontNameArray{$Position}:=$FontName
    ` 新しいフォントに名前を割り当てる
⇒ $FontNumArray{$Position}:=PV Add font(Area;$FontName)
    ` フォント番号を割り当てる
Else ` フォント $FontName はエリア内に既に存在している
    PV REMOVE FONT(Area;$FontNumArray{$Position}) ` このフォントを削除
End if

```

## 参照

PV GET FONT LIST、PV REMOVE FONT

## PV REMOVE FONT

---

### PV REMOVE FONT (エリア; フォント)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
フォント	倍長整数	→	フォント ID

### 説明

**PV REMOVE FONT** コマンドは、<エリア>から<フォント>を削除します。

このコマンドを使用してフォントを削除すると、関連する 4D View エリアでは使用できなくなります。もちろん、このフォントは物理的にシステムから削除されるわけではありません。

### 例題

**PV Add font** コマンドの例題を参照してください。

## 参照

PV Add font、PV GET FONT LIST

## PV GET FONT LIST

---

### PV GET FONT LIST (エリア; フォント; フォント名)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
フォント	倍長整数配列	←	フォントIDの配列
フォント名	文字列配列	←	フォント名の配列

### 説明

**PV GET FONT LIST** コマンドは、<エリア>内の各フォントのIDと名前を取得して、配列<フォント>および<フォント名>に代入します。

### 例題

**PV Add font** コマンドの例題を参照してください。

### 参照

PV Add font、PV REMOVE FONT

## PV Add format

---

### PV Add format (エリア; 文字列) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
文字列	文字列	→ フォーマットの文字列
戻り値	倍長整数	← フォーマット ID

### 説明

**PV Add format** 関数は、フォーマットの<文字列>を<エリア>に追加し、そのIDを返します。

<文字列>が既に存在する場合、**PV Add format** 関数はそのIDを返します（この番号は、**PV GET FORMAT LIST** コマンドを使用して取得することもできます）。

### 例題

アクティブなエリアで使用可能なアメリカ通貨フォーマットを削除しますが、ヨーロッパ通貨フォーマット（ユーロ）は使用可能なままにしておきます。

```
C_STRING(255;$OldFormat) ` エリアから削除するフォーマット文字列
C_STRING(255;$NewFormat) ` エリアに追加するフォーマット文字列
ARRAY LONGINT($FormatNumArray;0) ` フォーマット番号の配列
ARRAY STRING(255;$FormatStringArray;0) ` フォーマット文字列の配列
C_INTEGER($Position) ` 番号配列と文字列配列内における
                                削除対象フォーマットの位置
$OldFormat:="$###,##0.00"
$NewFormat:="### ##0,00 EUR"
    ` 利用可能なフォーマットのリスト
PV GET FORMAT LIST(Area;$FormatNumArray;$FormatStringArray)
$Position:=Find in array($FormatStringArray;$OldFormat)
If($Position#-1) ` 削除対象のフォーマットがエリアに存在するか?
    PV REMOVE FORMAT(Area;$FormatNumArray{$Position})
                                ` これを削除する
End if
    ` 追加するフォーマットはエリアに存在しない?
If(Find in array($FormatStringArray;$NewFormat)=-1)
    $Position:=Size of array($FormatStringArray)+1 ` これを追加する
    INSERT ELEMENT($FormatStringArray;$Position) ` 配列を...
    INSERT ELEMENT($FormatNumArray;$Position) ` ...リサイズする
    $FormatStringArray{$Position}:=$NewFormat
                                ` 新しいフォーマットを割り当てる
```

```
⇒ $FormatNumArray{$Position}:=PV Add format(Zone;$NewFormat)
                                     `番号を割り当てる
```

End if

参照

PV GET FORMAT LIST、PV REMOVE FORMAT

## PV REMOVE FORMAT

---

**PV REMOVE FORMAT** (エリア; フォーマット)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
フォーマット	倍長整数	→ フォーマットID

説明

**PV REMOVE FORMAT** コマンドは、<エリア>から<フォーマット>を削除します。

**PV Add format** 関数を使用して作成されたフォーマットのみ削除することができます。本来の4D Viewのフォーマットは削除できません。

例題

**PV Add format** コマンドの例題を参照してください。

参照

PV Add format

## PV SET FORMAT

---

### PV SET FORMAT (エリア; フォーマット; 文字列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
フォーマット	倍長整数	→ フォーマット ID
文字列	文字列	→ フォーマットの文字列

### 説明

**PV SET FORMAT** コマンドは、<フォーマット>のフォーマット<文字列>を変更します。

### 例題

**PV Add format** 関数の例題を簡単にしたものを以下に示します。ただし、ここではフォーマットが突然置き換えられます。以前のフォーマットが存在しない場合には、新しいフォーマットは作成されません。

```
ARRAY LONGINT($ArrayFormatNum;0) ` フォーマット番号の配列
ARRAY STRING(255;$ArrayFormatStrings;0) ` フォーマット文字列の配列
C_INTEGER($Position) ` 番号配列と文字列配列内における
                                変更対象フォーマットの位置
                                ` 利用可能なフォーマットのリスト
PV GET FORMAT LIST(Area;$ArrayFormatNum;$ArrayFormatStrings)
$Position:=Find in array($ArrayFormatStrings;"###,##0.00")
If($Position#-1) ` フォーマットはエリアに存在するか?
    ` フォーマットの変更
⇒ PV SET FORMAT(Area;$ArrayFormatNum{$Position}
                                ;"### ##0,00 EUR")
```

End if

### 参照

PV Add format

## PV GET FORMAT LIST

---

### PV GET FORMAT LIST (エリア; フォーマット; 文字列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
フォーマット	倍長整数配列	← フォーマットIDの配列
文字列	文字列配列	← フォーマット文字列の配列

### 説明

**PV GET FORMAT LIST** コマンドは、<エリア>に存在する各フォーマットのIDとその文字列を取得して、配列<フォーマット>および<文字列>に代入します。

### 例題

**PV Add format** コマンドの例題を参照してください。

### 参照

PV Add format



## Drag and Drop、はじめに

このテーマ内のコマンドと関数を使用して、同一の4D Viewエリア、または2つの4D Viewエリア間におけるドラッグ&ドロップの管理を行うことができます。

4D Viewにおいて、ドラッグ&ドロップは3つの原則に基づいて作用します。

- ソースオブジェクト（ドラッグが実行されるエリア）
- ターゲットオブジェクト（ドロップが実行されるエリア）
- 署名を使用すると、特定のエリア間でのドラッグ&ドロップを許可するかどうかを指定できます。

このテーマ内のコマンドは、ソースとターゲットおよびその署名を識別したり、ドロップが行われるターゲットエリアの位置に関する情報を取得する目的に使用されます。

必要に応じて、この情報を他の4D Viewコマンドで使用することも可能です。例えば、処理の確定管理が行われた後、ソースエリアからデータのコピーやカットを行い、ターゲットエリアへペーストしたり、または選択に基づいて別の処理を実行することができます。

## PV SET DRAG SIGNATURES

---

### PV SET DRAG SIGNATURES (エリア; 署名)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
署名	配列	→ 署名の配列

#### 説明

**PV SET DRAG SIGNATURES** コマンドは、配列<署名>の内容を<エリア>の“ドラッグ”署名として設定します。

署名は英数字の文字列で、その内容は自由に指定することができます。最長で32桁までの署名を設定できます。

**PV SET AREA PROPERTY** コマンドにより定義されるエリアのプロパティを使用して、4D Viewのドラッグ&ドロップに関するオプションを指定することができます。

■ エリアに適用するドラッグ&ドロップのプロパティを定義するには、**PV Area properties** テーマ内の定数を使用します。

■ これら各種プロパティの値を定義するには、**PV Drag drop allowed** および **PV Drop mode** テーマ内の定数を使用します。

ドラッグ&ドロップが許可されると、ソースエリアのドラッグとターゲットエリアのドロップに対して共通する署名が少なくとも1つはエリアに存在する場合に、2つのエリア間（または同一のエリア）でのやり取りを行うことができます。

ドラッグ&ドロップイベントに関しては、**PV ON EVENT** コマンドを使用して、このイベントを検出することができます。

#### 例題

次の例題では、2つの4D View エリア間のドラッグ&ドロップについて説明しています。まずはじめに署名およびこれらのエリアの動作（例えばフォームの変更時）を設定します。

```
ARRAY TEXT($DDSignatureArray;1)
```

```
$DDSignatureArray{1}:="Signature_1"
```

　　`ドラッグを行うには、alt キーを押しながらクリックする

```
PV SET AREA PROPERTY(SourceArea;pv drag trigger;pv trigger on alt click)
```

```
PV SET AREA PROPERTY(SourceArea;pv drag allowed;pv DD multiple cells)
```

```
PV ON EVENT(SourceArea;pv on drag;"DragDropMethod")
```

⇒ **PV SET DRAG SIGNATURES**(SourceArea;\$DDSignatureArray)

```
PV SET AREA PROPERTY(TargetArea;pv drop allowed;pv DD single cell+
```

pv DD adjacent cells+pv DD multiple cells)

**PV SET AREA PROPERTY**(TargetArea;pv drop mode;pv drop replace only)

**PV ON EVENT**(TargetArea;pv on drop;"DragDropMethod")

**PV SET DROP SIGNATURES**(TargetArea;\$DDSignatureArray)

　　`ドラッグと同じ署名

altキー+クリックを使用してドラッグが実行されるか、またはターゲットエリア上でドロップが行われた場合に、プロジェクトメソッド“DragDropMethod”が呼び出されます。

**C\_LONGINT**(\$1) `4D View 参照エリア

**C\_LONGINT**(\$2) `イベント

**C\_LONGINT**(\$3) `キーボードモディファイアキーのコード

**C\_LONGINT**(\$4) `カラム番号

**C\_LONGINT**(\$5) `行番号

**C\_LONGINT**(\$6) `キーのASCIIコード

　　`2度のコールバックメソッドの間で保持しておく場所がない

**C\_POINTER**(SourceAreaPointer;TargetAreaPointer)

**C\_BLOB**(\$blob) `一時的なドラッグ&ドロップノートパッド

**C\_LONGINT**(\$currentColumn;\$currentRow) `カレントセルの座標

　　`ドロップが行われるセルの座標

**C\_LONGINT**(\$destinationColumn;\$destinationRow)

**Case of**

　　\**(\$2=pv on drag)**

**PV GET DRAG SOURCE**(\$1;SourceAreaPointer) `移動元は？

　　\**(\$2=pv on drop)**

**PV GET DROP TARGET**(\$1;TargetAreaPointer) `移動先は？

**\$blob:=PV Copy to blob**(SourceAreaPointer->)

　　　`ノートパッドにコピー

**\$destinationColumn:=PV Get drag and drop info**

　　　(TargetAreaPointer->;pv drop column)

　　　`目標となる....

**\$destinationRow:=PV Get drag and drop info**

　　　(TargetAreaPointer->;pv drop row)

　　　`...座標

**PV GET CURRENT CELL**(TargetAreaPointer->;\$currentColumn;

　　　\$currentRow)

　　　`指定されたエリアにペースト

**PV GOTO CELL**(TargetAreaPointer->;\$destinationColumn;

　　　\$destinationRow)

**PV PASTE FROM BLOB**(TargetAreaPointer->;\$blob;1;1;1)

　　　`処理が終了したら、カレントセルを再設定する

***PV GOTO CELL***(TargetAreaPointer->,\$currentColumn,\$currentRow)

**End case**

参照

PV GET DRAG SIGNATURES、PV SET DROP SIGNATURES

定数

PV Area properties、PV Drag drop allowed および PV Drop mode テーマ

## PV GET DRAG SIGNATURES

---

### PV GET DRAG SIGNATURES (エリア; 署名)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
署名	配列	← 署名の配列

#### 説明

**PV GET DRAG SIGNATURES** コマンドは、<エリア>のドラッグ署名を取得し、<署名>配列に代入します。

署名は英数字の文字列で、その内容は自由に指定することができます。最長で32桁までの署名を設定できます。

#### 例題

エリアが内部的なドラッグ&ドロップのオブジェクトとなり得る場合に、ヘルプメッセージを表示します。

```

ARRAY TEXT($DragSignatureArray;0)
ARRAY TEXT($DropSignatureArray;0)
C_TEXT(HelpMessage)
C_INTEGER($Index)
⇒ PV GET DRAG SIGNATURES(Area;$DragSignatureArray)
PV GET DROP SIGNATURES(Area;$DropSignatureArray)
HelpMessage:=""
For($Index;1;Size of array($DragSignatureArray)) `共通する署名を探す
    If(Find in array($DropSignatureArray;$DragSignatureArray{$Index})#-1)
        HelpMessage:="このエリア内でドラッグ&ドロップを行うことが
                                                    できます。"
        $Index:=Size of array($DragSignatureArray)
    End if
End for

```

#### 参照

PV GET DROP SIGNATURES、PV SET DRAG SIGNATURES

## PV SET DROP SIGNATURES

---

### PV SET DROP SIGNATURES (エリア; 署名)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
署名	配列	→ 署名の配列

### 説明

**PV SET DROP SIGNATURES** コマンドは、配列<署名>の内容を<エリア>の“ドロップ”署名として設定します。

署名は英数字の文字列で、その内容は自由に指定することができます。最長で32桁までの署名を設定できます。

### 例題

**PV SET DRAG SIGNATURES** コマンドの例題を参照してください。

### 参照

PV GET DROP SIGNATURES、PV SET DRAG SIGNATURES

## PV GET DROP SIGNATURES

---

### PV GET DROP SIGNATURES (エリア; 署名)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
署名	配列	← 署名の配列

#### 説明

**PV GET DROP SIGNATURES** コマンドは、<エリア>のドロップ署名を取得して<署名>配列を作成します。

署名は英数字の文字列で、その内容は自由に指定することができます。最長で32桁までの署名を設定できます。

#### 例題

**PV GET DRAG SIGNATURES** コマンドの例題を参照してください。

#### 参照

PV GET DRAG SIGNATURES、PV SET DROP SIGNATURES

## PV Get drag and drop info

---

### PV Get drag and drop info (エリア; オプション) → 倍長整数

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
オプション	倍長整数	→ オプション番号
戻り値	倍長整数	← オプションの値

#### 説明

**PV Get drag and drop**関数は、指定された<オプション>に対するドラッグ&ドロッププロパティの値を返します。

<オプション>を定義するには、**PV Drop info**定数を使用します。

#### ■ pv drag X offset

ドラッグ対象の<エリア>で使用します。ドラッグ動作が行われたセルのX座標（セルの左上隅から開始）を返します。

#### ■ pv drag Y offset

ドラッグ対象の<エリア>で使用します。ドラッグ動作が行われたセルのY座標（セルの左上隅から開始）を返します。

#### ■ pv drop X offset

ドロップ対象の<エリア>で使用します。ドロップ動作が行われたセルのX座標（セルの左上隅から開始）を返します。

#### ■ pv drop Y offset

ドロップ対象の<エリア>で使用します。ドロップ動作が行われたセルのY座標（セルの左上隅から開始）を返します。

#### ■ pv drop action

ドロップ対象の<エリア>で使用します。この定数を使用して、ユーザが行ったドロップ動作を取得することができます。**PV Drop action** テーマ内の定数を返します。

#### 例題

**PV SET DRAG SIGNATURES** コマンドの例題を参照してください。

#### 参照

PV SET DRAG SIGNATURES

## 定数

PV Drop info および PV Drop action テーマ

## PV GET DRAG SOURCE

---

### PV GET DRAG SOURCE (エリア; ソース)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
ソース のポインタ	ポインタ	←	ドラッグ対象のソースオブジェクトへの ポインタ

### 説明

**PV GET DRAG SOURCE** コマンドは、ドラッグ対象のソースオブジェクトへのポインタを<ソース>にセットします。

### 例題

**PV SET DRAG SIGNATURES** コマンドの例題を参照してください。

### 参照

PV GET DROP TARGET

## PV GET DROP TARGET

---

### PV GET DROP TARGET (エリア; ターゲット)

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
ターゲット	ポインタ	←	ドロップ対象のターゲット オブジェクトへのポインタ

### 説明

**PV GET DROP TARGET** コマンドは、ドロップ対象のターゲットオブジェクトへのポインタを<ターゲット>にセットします。

### 例題

**PV SET DRAG SIGNATURES** コマンドの例題を参照してください。

### 参照

PV GET DRAG SOURCE

## PV Plugin Property、はじめに

このテーマ内のコマンドを使用して、汎用的な4D Viewプラグインのプロパティの値を設定したり、取得することができます。

これらの汎用的なプロパティとは、新しい4D Viewエリアにデフォルトで含まれる行数とカラム数、組み込みエリアの最小サイズ、および4D Viewテンプレートの読み込みと書き込みを行う場所に関する情報です。

## PV SET PLUGIN PROPERTY

---

### PV SET PLUGIN PROPERTY (プロパティ; 値)

引数	タイプ	説明
プロパティ	倍長整数	→ プロパティ番号
値	倍長整数	→ プロパティの値

#### 説明

**PV SET PLUGIN PROPERTY** コマンドを使用して、汎用的な 4D View プラグインの<プロパティ>の<値>を設定することができます。

このコマンドは、例えば「**On Startup**」データベースメソッド内に配置することができます。定義されたプロパティは、すべての新しい 4D View エリアへ即座に適用されます。

引数<プロパティ>の設定には、**PV Plugin properties** テーマ内の定数を使用します。

プロパティに設定する値は、引数<値>に渡します。この値は、定義するプロパティによって異なります。

引数<プロパティ>および<値>に使用できる定数について、以下のリストで説明します。

#### pv write template on server

クライアント/サーバアプリケーションにおいて、4D View ドキュメントのテンプレートを各クライアントマシン上に書き込みます。デフォルトでは、サーバ上にテンプレートが書き込まれます。関連する値：0または1。

■ 0：テンプレートは各クライアントマシン上に書き込まれる

■ 1：テンプレートはサーバ上に書き込まれる

#### pv load template on server

クライアント/サーバアプリケーションにおいて、各クライアントマシンから 4D View ドキュメントのテンプレートをロードします。デフォルトでは、サーバからテンプレートのロードが行われます。関連する値：0または1。

■ 0：テンプレートは各クライアントマシンからロードされる。

■ 1：テンプレートはサーバからロードされる。

### pv default columns count

新しい4D Viewドキュメント内のデフォルトのカラム数を定義します。この値は、ユーザまたはプログラムにより随時変更することができます。デフォルトでは、新しい4D Viewドキュメントには256カラムが含まれます。関連する値：カラム数。

### pv default rows count

新しい4D Viewドキュメント内のデフォルトの行数を定義します。この値は、ユーザまたはプログラムにより随時変更することができます。デフォルトでは、新しい4D Viewドキュメントには、8192行が含まれます。関連する値：行数。

### pv button width

4D View内に含まれるエリアの最小の幅を定義します。この値より小さい場合、そのエリアはボタンとして表示されます（このエリアをウインドウ全体の大きさで表示するには、このボタンをクリックします）。デフォルトでは、4D Viewエリアの幅が150ピクセルより小さい場合に、エリアはボタンとして表示されます。関連する値：幅（ピクセル数）。

### pv button height

4D View内に含まれるエリアの最小の高さを定義します。この値より小さい場合、そのエリアはボタンとして表示されます（このエリアをウインドウ全体の大きさで表示するには、このボタンをクリックします）。デフォルトでは、4D Viewエリアの高さが100ピクセルより小さい場合に、エリアはボタンとして表示されます。関連する値：高さ（ピクセル数）。

### pv confirm convert dialog

4D Calc 6.7ドキュメントを4D Viewで開いた際に、変換メッセージを表示するかどうかを指定します。表示されるメッセージは、4D Viewのリソース内に保存されています。関連する値：0または1。

■ 0：変換メッセージを表示しない。

■ 1：変換メッセージを表示する。

### 例題

作成されるすべての4D Viewエリアが、最初に100カラム、50行で構成されるように設定します。

- ⇒ **PV SET PLUGIN PROPERTY**(pv default columns count;100)
- ⇒ **PV SET PLUGIN PROPERTY**(pv default rows count;50)

参照

PV Get plugin property

定数

PV Plugin properties テーマ

## PV Get plugin property

---

### PV Get plugin property (プロパティ) → 倍長整数

引数	タイプ	説明
プロパティ	倍長整数	→ プロパティ番号
戻り値	倍長整数	← プロパティの値

説明

**PV Get plugin property** 関数は、汎用的な 4D View プラグインの<プロパティ>の現在値を返します。

引数<プロパティ>の設定には、**PV Plugin properties** テーマ内の定数を使用します。これらの定数に関する詳細は、**PV SET PLUGIN PROPERTY** コマンドの説明を参照してください。

例題

4D View の組み込みエリアがボタンに変わる最小幅（ピクセル数）を知りたいものとします。

```

C_LONGINT($vWidth)
⇒ $vWidth:=PV Get plugin property(pv button width)
ALERT("4D View エリアの最小幅は、"+String($vWidth)+" ピクセルです。")

```

参照

PV SET PLUGIN PROPERTY

定数

PV Plugin properties テーマ

## PV Tools、はじめに

---

このテーマ内のコマンドと関数により、セル選択範囲の“BLOBへの格納”や“BLOBからの取得”を行ったり、さまざまな方法でのカラー参照を管理する各種ツールを提供します。

更に、プラグインウィンドウのタイトルの取得や設定を行える2つのコマンドについて説明しています。

## PV Copy to blob

---

### PV Copy to blob (エリア) → Blob

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
戻り値	Blob ←	選択範囲を納めた Blob

### 説明

**PV Copy to blob** 関数は、<エリア>の現在のセル選択範囲を BLOB に納めて返します。

これらの選択範囲は、連続（範囲）している必要はありません。

### 例題

次のメソッドは、範囲 A1、A2、B1、B2 のコンテンツと A3、A4、B3、B4 のコンテンツとを入れ替えます。

**C\_BLOB**(\$Blob) `保存する BLOB

**PV SELECT RANGE**(Area;1;1;2;2;pv selection set) `範囲 A1、A2、B1、B2 を選択

**PV SPECIAL CUT**(Area;1;1;1) `コンテンツをカットしてクリップボードに配置する

**PV SELECT RANGE**(Area;1;3;2;4;pv selection set) `範囲 A3、A4、B3、B4 を選択

⇒ \$Blob:=**PV Copy to blob**(Area) `選択範囲を BLOB に納める

**PV SPECIAL PASTE**(Area;1;1;1) `クリップボードの内容をペースト

**PV SELECT RANGE**(Area;1;1;2;2;pv selection set) `A1 セルを選択

**PV PASTE FROM BLOB**(Area;\$Blob;1;1;1) `クリアした内容を再度設定する

**PV SELECT RANGE**(Area;1;1;1;1;pv selection remove) `A1 セルを未選択にする

### 参照

PV Create picture、PV PASTE FROM BLOB

## PV PASTE FROM BLOB

---

### PV PASTE FROM BLOB(エリア; BLOB; 値; フォーミュラ; フォーマット)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
BLOB	BLOB	→ 選択範囲を納めた BLOB
値	整数	→ 0= ペーストしない; 1= ペースト
フォーミュラ	整数	→ 0= ペーストしない; 1= ペースト
フォーマット	整数	→ 0= ペーストしない; 1= ペースト

### 説明

**PV PASTE FROM BLOB** コマンドは、BLOB に納められた選択範囲を元に、引数<値>、<フォーミュラ>、<フォーマット>、<枠線>で指定された情報を<エリア>のカルレントセルにペーストします。この BLOB は、**PV Copy to blob** 関数を使用してあらかじめ作成します。

### 例題

**PV Copy to blob** コマンドの例題を参照してください。

### 参照

PV COPY TO BLOB

## PV RGB to color

---

### PV RGB to color (赤; 緑; 青) → 倍長整数

引数	タイプ		説明
赤	整数	→	赤の成分 (0 ~ 255)
緑	整数	→	緑の成分 (0 ~ 255)
青	整数	→	青の成分 (0 ~ 255)
戻り値	倍長整数	←	カラー

### 説明

**PV RGB to color** 関数は、RGB カラーを定義する倍長整数タイプの値を返します。この値により、<赤>、<緑>、<青>の成分に基づいた色が決定します。

4D で使用されるカラーシステムに関する詳細は、4th Dimension コマンドである **SET RGB COLORS** を参照してください。

### 例題

**PV Color to index** コマンドの例題を参照してください。

### 参照

PV COLOR TO RGB

## PV COLOR TO RGB

---

### PV COLOR TO RGB (カラー; 赤; 緑; 青)

引数	タイプ		説明
カラー	倍長整数	→	カラー
赤	整数	←	赤の成分 (0 ~ 255)
緑	整数	←	緑の成分 (0 ~ 255)
青	整数	←	青の成分 (0 ~ 255)

### 説明

**PV COLOR TO RGB** コマンドは、RGB <カラー> の3つの成分を取得し、引数 <赤>、<緑>、<青> に代入します。

4D で使用されるカラーシステムに関する詳細は、4th Dimension コマンドである **SET RGB COLORS** を参照してください。

### 例題

A1 セルの背景色に関する3つのRGBカラー成分を表示します。

```

C_LONGINT($Color)
C_INTEGER($Red;$Green;$Blue)
PV SET CELL STRING VALUE(Area;2;1;"カラー") `ラベル
PV SET CELL STRING VALUE(Area;2;2;"赤")
PV SET CELL STRING VALUE(Area;2;3;"緑")
PV SET CELL STRING VALUE(Area;2;4;"青")
$Color:=PV Get cell property(Area;1;1;pv style color back odd)
⇒ PV COLOR TO RGB($Color;$Red;$Green;$Blue)
PV SET CELL NUM VALUE(Area;1;2;$Red) `値
PV SET CELL NUM VALUE(Area;1;3;$Green)
PV SET CELL NUM VALUE(Area;1;4;$Blue)

```

### 参照

PV RGB to color

## PV Color to index

---

### PV Color to index (カラー) → 整数

引数	タイプ		説明
カラー	倍長整数	→	カラー番号
戻り値	整数	←	インデックスカラー番号

### 説明

**PV Color to index**関数は、デフォルトの4Dカラーパレットにおける指定されたRGB<カラー>の番号を返します。

4Dのインデックスカラーの範囲は0から255までですが、プラグインカラーは1から256までのインデックスが付けられている点に注意してください。開発者が4Dのカラー定数を使用したい場合には、このことが特に重要となります。

4Dで使用されるRGBカラーに関する詳細は、以下の4th Dimension コマンドの説明を参照してください。

■ SET RGB COLORS：4Dで使用されるRGBカラーシステムについて。

■ SET COLOR：4Dのカラーパレットについて。

### 例題

このメソッドは、セルA2、A3、A4に入力されたRGBの値（0～255）に基づいて設定された背景色をセルA1に設定します。セルC1には、暫定的な背景色を使って、4Dパレット内で最も近いインデックスカラー番号を表示します。一方、セルC2、C3、C4にはこのインデックスカラーに対応するRGBの値を表示し、はじめの値との違いを示します。

```
C_LONGINT($Color) `Background color for A1 セルの背景色 (RGB)、  
後インデックス後のカラー  
C_INTEGER($Red;$Green;$Blue) `はじめのRGBカラーと終了後のRGBカラー  
C_INTEGER($Index) `4Dパレット内での番号  
PV SET CELL STRING VALUE(Area;2;1;"カラー") `ラベル  
PV SET CELL STRING VALUE(Area;2;2;"赤")  
PV SET CELL STRING VALUE(Area;2;3;"緑")  
PV SET CELL STRING VALUE(Area;2;4;"青")  
$Red:=PV Get cell num value(Area;1;2) `はじめのRGBカラー  
$Green:=PV Get cell num value(Area;1;3)  
$Blue:=PV Get cell num value(Area;1;4)  
$Color:=PV RGB to color($Red;$Green;$Blue)  
`A1の背景色を割り当てる  
PV SET CELL PROPERTY(Area;1;1;pv style color back odd;$Color)
```

```
⇒ $Index:=PV Color to index($Color) `インデックス化"
PV SET CELL NUM VALUE(Area;3;1;$Index) `4Dパレット内での番号
$Color:=PV Index to color($Index) `新しいカラー
`C1セルの背景色を割り当てる
PV SET CELL PROPERTY(Area;3;1;pv style color back odd;$Color)
PV COLOR TO RGB($Color;$Red;$Green;$Blue) `分解
PV SET CELL NUM VALUE(Area;3;2;$Red) `“インデックス化”の後の
RGBカラー
PV SET CELL NUM VALUE(Area;3;3;$Green)
PV SET CELL NUM VALUE(Area;3;4;$Blue)
```

## 参照

PV Index to color

## PV Index to color

---

### PV Index to color (インデックス) → 倍長整数

引数	タイプ	説明
インデックス	整数	→ インデックスカラー番号
戻り値	倍長整数	← カラー番号

## 説明

**PV Index to color**関数は、<インデックス>番号に対応する4Dパレット上のRGBカラー番号を返します。

この結果を元に、**PV COLOR TO RGB**コマンドを使用してRGBカラーの3つの成分を取り出すことができます。

4Dのインデックスカラーの範囲は0から255までですが、プラグインカラーは1から256までのインデックスが付けられている点に注意してください。開発者が4Dカラーの定数を使用したい場合には、このことが特に重要となります。

4Dで使用されるRGBカラーに関する詳細は、以下の4th Dimensionコマンドの説明を参照してください。

■ SET RGB COLORS : 4Dで使用されるRGBカラーシステムについて。

■ SET COLOR : 4Dのカラーパレットについて。

## 例題

**PV SET RANGE BORDER** および **PV Color to index** コマンドの例題を参照してください。

## 参照

PV Color to index

## PV SET WINDOW TITLE

---

### PV SET WINDOW TITLE (エリア; タイトル)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
タイトル	文字列	→ プラグインウィンドウの新しいタイトル

## 説明

**PV SET WINDOW TITLE** コマンドは、4D View のプラグイン<エリア>のウィンドウ<タイトル>を設定します。

## 例題

ウィンドウに現在日付を追加します。

```
C_TEXT($Title) `既存のタイトル
$Title:=PV Get window title(Area)
⇒ PV SET WINDOW TITLE(Area,$Title+" (" +String(Current date)+")")
```

## 参照

PV Get window title

## PV Get window title

---

### PV Get window title (エリア) → 文字列

引数	タイプ		説明
エリア	倍長整数	→	4D View エリア
戻り値	文字列	←	プラグインウィンドウのタイトル

### 説明

**PV Get window title** 関数は、4D View のプラグイン<エリア>のウィンドウタイトルを返します。

### 例題

**PV SET WINDOW TITLE** コマンドの例題を参照してください。

### 参照

PV SET WINDOW TITLE



## PV Allowed Input、はじめに

---

このテーマ内のコマンドを使用して、4D View ユーザがカレントエリアのフォーミュラで利用可能な4Dオブジェクト（変数、メソッド、コマンド）の設定、および読み込みを行うことができます。この操作により、4D View エリア内でのユーザ動作をコントロールできるようになります。

デフォルトでは、認可入力システムはアクティブではありません（ユーザはすべての4D変数、メソッド、コマンドにアクセス可能）。このテーマのコマンドを使用する前に、まず `pv no formula external call` 定数に1を指定した上で（pv value on）**PV SET DOCUMENT PROPERTY** コマンドを実行し、これら4Dオブジェクトに対する呼び出しをすべて禁止しておかなければなりません。

- 呼び出しを禁止し、認可入力システムをアクティブにする。

**PV SET DOCUMENT PROPERTY**(area;pv no formula external call;pv value on)

- このシステムを非アクティブにする（デフォルト処理）。

**PV SET DOCUMENT PROPERTY**(area;pv no formula external call;pv value off)

## PV SET ALLOWED VAR LIST

---

### PV SET ALLOWED VAR LIST (エリア; 変数名配列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
変数名配列	文字列配列	→ 変数名の配列

#### 説明

**PV SET ALLOWED VAR LIST** コマンドは、フォーミュラで許可される変数のリスト（プロセス変数とインタープロセス変数のみ）を用いて〈変数名配列〉を設定します。

注：このコマンドを有効にするためには、許可される入力システムがアクティブでなければなりません。「PV Allowed Input、はじめに」の節を参照してください。

#### 参照

PV GET ALLOWED VAR LIST

## PV GET ALLOWED VAR LIST

---

### PV GET ALLOWED VAR LIST (エリア; 変数名配列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
変数名配列	文字列配列	← 変数名の配列

#### 説明

**PV GET ALLOWED VAR LIST** コマンドは、フォーミュラで許可される変数のリスト（プロセス変数とインタープロセス変数のみ）を〈変数名配列〉に代入します。

#### 参照

PV SET ALLOWED VAR LIST

## PV SET ALLOWED MET LIST

---

### PV SET ALLOWED MET LIST(エリア; メソッド配列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
メソッド配列	文字列配列	→ メソッド名の配列

#### 説明

**PV SET ALLOWED MET LIST** コマンドは、フォーミュラで許可されるメソッドのリストを用いて<メソッド配列>を設定します。

注：このコマンドを有効にするためには、許可される入力システムがアクティブでなければなりません。「PV Allowed Input、はじめに」の節を参照してください。

#### 参照

PV GET ALLOWED MET LIST

## PV GET ALLOWED MET LIST

---

### PV GET ALLOWED MET LIST (エリア; メソッド配列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
メソッド配列	文字列配列	← メソッド名の配列

#### 説明

**PV GET ALLOWED MET LIST** コマンドは、フォーミュラで許可されるメソッドのリストを<メソッド配列>に代入します。

#### 参照

PV SET ALLOWED MET LIST

## PV SET ALLOWED COM LIST

---

### PV SET ALLOWED COM LIST (エリア; コマンド配列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
コマンド配列	文字列配列	→ 4D コマンド名の配列

#### 説明

**PV SET ALLOWED COM LIST** コマンドは、フォーミュラで許可される4Dコマンドのリストを用いて<メソッド配列>を設定します。

注：このコマンドを有効にするためには、許可される入力システムがアクティブでなければなりません。「PV Allowed Input、はじめに」の節を参照してください。

#### 参照

PV GET ALLOWED COM LIST

## PV GET ALLOWED COM LIST

---

### PV GET ALLOWED COM LIST (エリア; コマンド配列)

引数	タイプ	説明
エリア	倍長整数	→ 4D View エリア
コマンド配列	文字列配列	← 4D コマンド名の配列

#### 説明

**PV GET ALLOWED COM LIST** コマンドは、フォーミュラで許可される4Dコマンドのリストを<コマンド配列>に代入します。

#### 参照

PV SET ALLOWED COM LIST

## 付録 A、4D View エラーコード一覧

---

エラー管理用メソッド内で4D View から返されるエラーコードの一覧を以下に示します。  
これらのコードは、**PV GET LAST ERROR** および **PV ON ERROR** コマンドで使用します。

番号 エラーメッセージ

- 1 未知のエラーです。
- 2 コマンドが正しくありません。
- 3 これは古いコマンドです。このバージョンで使用することはできません。
- 4 これは古い引数です。このバージョンで使用することはできません。
- 5 引数が範囲外です。
- 6 配列名が正しくありません。
- 7 配列のタイプが正しくありません。
- 8 数値タイプの配列が必要です。
- 9 テキストタイプの配列が必要です。
- 10 配列の数が正しくありません。
- 11 配列の数が同じでなければなりません。
- 12 変数のタイプが正しくありません。
- 13 空のピクチャです。
- 14 プラグインエリアが必要です。
- 15 4D View プラグインエリアが必要です。
- 16 4D Calc プラグインエリアが必要です。
- 17 ALP プラグインエリアが必要です。
- 18 メモリが足りません。
- 19 ドキュメントを読み込みまたは書き込みしている最中にエラーが発生しました。
- 20 4D View ドキュメントではありません。
- 21 4D Calc ドキュメントではありません。
- 22 SYLK ドキュメントではありません。
- 23 ドキュメントフォーマットが正しくありません。
- 24 ドキュメントのバージョンが新しすぎます。
- 25 ドキュメントは壊れているようです。
- 26 ドキュメントは既に存在します。

- 27 ドキュメントが存在しません。
- 28 プロパティが正しくありません。
- 29 このプロパティは読み込みのみ許可されています。
- 30 選択モードの値が正しくありません。
- 31 選択アクションの値が正しくありません。
- 32 改行値が正しくありません。
- 33 矢印キーの値が正しくありません。
- 34 Enterキーの値が正しくありません。
- 35 並び替えの値が正しくありません。
- 36 枠線の角の値が正しくありません。
- 37 枠線スタイルの値が正しくありません。
- 38 スタイルのターゲットが正しくありません。
- 39 方向が正しくありません。
- 40 整列が正しくありません。
- 45 ピクチャフォーマットが正しくありません。
- 46 カラーが正しくありません。
- 47 スタイル値が正しくありません。
- 48 ドラッグ&ドロップ動作が正しくありません。
- 49 スタイルシート参照が正しくありません。
- 50 フォーマット参照が正しくありません。
- 51 フィルタ参照が正しくありません。
- 52 フォント参照が正しくありません。
- 53 ピクチャ番号が正しくありません。
- 54 選択範囲番号が正しくありません。
- 55 縦スプリッタの番号が正しくありません。
- 56 横スプリッタの番号が正しくありません
- 57 最後のペインを削除することはできません。
- 58 ペインの幅が正しくありません。
- 59 ペインの高さが正しくありません。
- 60 列番号が正しくありません。
- 61 行番号が正しくありません。
- 62 セルがリンクされています。
- 63 セル範囲が正しくありません。
- 64 挿入する列/行の数が正しくありません。
- 65 削除する列/行の数が正しくありません。
- 66 フォーミュラが正しくありません。
- 67 列幅が正しくありません。
- 68 行の高さが正しくありません。
- 69 セル名が正しくありません。
- 70 その名前はすでに使用されています。
- 71 リンクされた列がありません。

- 72 リンクされた列の行数が異なります。
- 73 リンクされたフィールドのマスタテーブルが異なります。
- 74 計算された値のタイプが正しくありません。
- 75 テーブルまたはフィールドが正しくありません。
- 76 テーブルが正しくありません。
- 77 フィールドが正しくありません。
- 78 フィールドタイプが正しくありません。
- 79 カレントセクションがありません。
- 80 カレントレコードがありません。
- 81 印刷する値付きセルがありません。
- 82 統計が正しくありません。
- 83 条件が正しくありません。
- 84 テーブルが正しくありません。
- 85 このスタイルシートは既に存在します。
- 86 リンクされた領域内では無効な操作です。

#### 参照

PV GET LAST ERROR、PV ON ERROR



## PV Area properties

---

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv allow undo redo	倍長整数	39
pv arrow keys	倍長整数	9
pv carriage return	倍長整数	8
pv column headers height	倍長整数	21
pv copy hidden	倍長整数	19
pv current cell highlight	倍長整数	3
pv drag allowed	倍長整数	13
pv drag trigger	倍長整数	12
pv drop allowed	倍長整数	33
pv drop mode	倍長整数	14
pv field tag	倍長整数	17
pv field wrapper	倍長整数	18
pv header sort	倍長整数	20
pv hor pane count	倍長整数	11
pv input enter key mode	倍長整数	15
pv input trigger	倍長整数	6
pv record tag	倍長整数	16
pv resizable columns	倍長整数	4
pv resizable rows	倍長整数	5
pv row headers width	倍長整数	22
pv saving dialog	倍長整数	37
pv select highlight	倍長整数	1
pv select mode	倍長整数	0
pv select null	倍長整数	2
pv select trigger	倍長整数	7
pv show borders toolbar	倍長整数	29
pv show column headers	倍長整数	23

定数	タイプ	値
pv show formula toolbar	倍長整数	30
pv show hor grid	倍長整数	31
pv show hor scrollbar	倍長整数	34
pv show menu bar	倍長整数	25
pv show numbers toolbar	倍長整数	27
pv show row headers	倍長整数	24
pv show standard toolbar	倍長整数	26
pv show style toolbar	倍長整数	28
pv show vert grid	倍長整数	32
pv show vert scrollbar	倍長整数	35
pv vert pane count	倍長整数	10
pv zoom factor	倍長整数	36

## PV Arrow keys

これらの定数は、“pv arrows” エリアプロパティの値として使用することができます。

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv arrow keys allowed	倍長整数	0
pv arrow keys not allowed	倍長整数	3
pv right and left arrow keys	倍長整数	2
pv top and bottom arrow keys	倍長整数	1

## PV Border edge

このテーマ内の定数を使用して、セル範囲に枠線を設定することができます。1つ以上の枠線を定義するには、複数の定数を追加してゆきます。

複数のセルを選択した場合、最初から4つの定数はその範囲の外側の角を示します。この場合、範囲の内側の角は“pv border edge inner hor” および “pv border edge inner vert” 定数を使用して設定することができます。

関連するコマンド：PV GET BORDER STYLE、PV SET BORDER STYLE

定数	タイプ	値
pv border edge bottom	倍長整数	8
pv border edge inner hor	倍長整数	16
pv border edge inner vert	倍長整数	32
pv border edge left	倍長整数	1
pv border edge right	倍長整数	4
pv border edge top	倍長整数	2

## PV Border style

---

関連するコマンド：PV GET BORDER STYLE、PV SET BORDER STYLE

定数	タイプ	値
pv border style 1	倍長整数	1
pv border style 111	倍長整数	7
pv border style 112	倍長整数	9
pv border style 2	倍長整数	2
pv border style 211	倍長整数	8
pv border style 212	倍長整数	10
pv border style 222	倍長整数	11
pv border style 232	倍長整数	12
pv border style 3	倍長整数	3
pv border style 4	倍長整数	4
pv border style 5	倍長整数	5
pv border style 6	倍長整数	6
pv border style none	倍長整数	0

## PV Carriage return

---

このテーマ内の定数を使用して、入力時の改行キーによる確定処理を定義することができます。

一度“許可”されると、改行キーがモディファイアキーに関連付けられているかどうかには関わらず（指定された引数に準ずる）、改行キーを用いて入力の確定を行うことができます。

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv cr allowed	倍長整数	1
pv cr allowed with ctrl	倍長整数	2
pv cr allowed with shift	倍長整数	4
pv cr not allowed	倍長整数	0

## PV Cell properties

---

関連するコマンド：PV Get cell property、PV Get range property、PV SET CELL PROPERTY、PV SET RANGE PROPERTY

定数	タイプ	値
pv cell height	倍長整数	101
pv cell width	倍長整数	100

## PV Cell value type

---

関連するコマンド：PV Get cell value type

定数	タイプ	値
pv value type boolean	倍長整数	6
pv value type date	倍長整数	4
pv value type date time	倍長整数	5
pv value type none	倍長整数	0
pv value type numeric	倍長整数	1
pv value type picture	倍長整数	7
pv value type string	倍長整数	2
pv value type time	倍長整数	3

## PV Commands

---

このテーマ内の定数を使用すると、「ユーザ」モードでの4D Viewの機能のいずれかを使用できるようになります。これらの定数には、次のように接頭辞が付けられています。

- “cmd” は、メニューコマンド（およびツールパレット上の対応するアイコン）であることを示します。

- “pal” は、ツールパレットアイコンを使用した場合にのみ利用できる機能を示します。  
これらの定数は、**PV SET COMMAND STATUS** および **PV GET COMMAND STATUS** コマンドでしか使用することはできません。

関連するコマンド：PV EXECUTE COMMAND、PV GET COMMAND STATUS、PV Get on command method、PV ON COMMAND、PV SET COMMAND STATUS

定数	タイプ	値
pv cmd calculate now	倍長整数	120
pv cmd calculation mode	倍長整数	119
pv cmd db import fields	倍長整数	213
pv cmd db import report	倍長整数	214
pv cmd db linked cells	倍長整数	215
pv cmd db linked pictures	倍長整数	217
pv cmd document information	倍長整数	109
pv cmd edit clear all	倍長整数	234
pv cmd edit clear borders	倍長整数	233
pv cmd edit clear formats	倍長整数	232
pv cmd edit clear formulas	倍長整数	230
pv cmd edit clear other	倍長整数	235
pv cmd edit clear values	倍長整数	231
pv cmd edit copy	倍長整数	4
pv cmd edit cut	倍長整数	3
pv cmd edit delete	倍長整数	136
pv cmd edit fill down	倍長整数	134
pv cmd edit fill right	倍長整数	135
pv cmd edit find	倍長整数	125
pv cmd edit find next	倍長整数	126
pv cmd edit go to	倍長整数	129
pv cmd edit go to last cell	倍長整数	130
pv cmd edit move	倍長整数	124
pv cmd edit paste	倍長整数	5
pv cmd edit redo	倍長整数	2
pv cmd edit repeat	倍長整数	122
pv cmd edit replace	倍長整数	127
pv cmd edit replace next	倍長整数	128
pv cmd edit select all	倍長整数	7
pv cmd edit set name	倍長整数	170
pv cmd edit sort	倍長整数	131
pv cmd edit special paste	倍長整数	123
pv cmd edit undo	倍長整数	1

定数	タイプ	値
pv cmd export	倍長整数	105
pv cmd export area clear	倍長整数	107
pv cmd export area set	倍長整数	106
pv cmd export area show	倍長整数	108
pv cmd file new	倍長整数	100
pv cmd file open	倍長整数	101
pv cmd file page setup	倍長整数	110
pv cmd file preferences	倍長整数	118
pv cmd file print document	倍長整数	114
pv cmd file print formulas	倍長整数	113
pv cmd file print preview	倍長整数	112
pv cmd file printing options	倍長整数	111
pv cmd file save	倍長整数	102
pv cmd file save as	倍長整数	103
pv cmd file save template	倍長整数	104
pv cmd format borders	倍長整数	202
pv cmd format cells	倍長整数	187
pv cmd format col default W	倍長整数	175
pv cmd format column auto width	倍長整数	174
pv cmd format column hide	倍長整数	179
pv cmd format column show	倍長整数	180
pv cmd format column width	倍長整数	173
pv cmd format row auto height	倍長整数	177
pv cmd format row default H	倍長整数	178
pv cmd format row height	倍長整数	176
pv cmd format row hide	倍長整数	181
pv cmd format row show	倍長整数	182
pv cmd format style sheets	倍長整数	188
pv cmd freeze references	倍長整数	121
pv cmd insert cell	倍長整数	154
pv cmd insert column	倍長整数	155
pv cmd insert column break	倍長整数	211
pv cmd insert row	倍長整数	156
pv cmd insert row break	倍長整数	212
pv cmd linked columns arrays	倍長整数	219
pv cmd linked columns fields	倍長整数	218
pv cmd print area clear	倍長整数	116
pv cmd print area set	倍長整数	115
pv cmd print area show	倍長整数	117

定数	タイプ	値
pv cmd security hide	倍長整数	183
pv cmd security lock	倍長整数	185
pv cmd security show	倍長整数	184
pv cmd security unlock	倍長整数	186
pv cmd swap screenmode	倍長整数	20
pv cmd view col headers	倍長整数	142
pv cmd view formula	倍長整数	141
pv cmd view grid	倍長整数	144
pv cmd view Hscrollbar	倍長整数	146
pv cmd view menu bar	倍長整数	140
pv cmd view page breaks	倍長整数	147
pv cmd view pictures	倍長整数	149
pv cmd view references	倍長整数	148
pv cmd view row headers	倍長整数	143
pv cmd view toolbar border	倍長整数	153
pv cmd view toolbar number	倍長整数	151
pv cmd view toolbar standard	倍長整数	150
pv cmd view toolbar style	倍長整数	152
pv cmd view Vscrollbar	倍長整数	145
pv pal border all	倍長整数	207
pv pal border bottom	倍長整数	203
pv pal border color	倍長整数	210
pv pal border columns	倍長整数	205
pv pal border frame	倍長整数	204
pv pal border kind	倍長整数	209
pv pal border none	倍長整数	208
pv pal border rows	倍長整数	206
pv pal format string	倍長整数	160
pv pal formula cancel	倍長整数	221
pv pal formula validate	倍長整数	222
pv pal number align auto	倍長整数	189
pv pal number align center	倍長整数	191
pv pal number align left	倍長整数	190
pv pal number align right	倍長整数	192
pv pal number money	倍長整数	201
pv pal number percentage	倍長整数	200
pv pal number scientific	倍長整数	199
pv pal standard sort asc	倍長整数	132
pv pal standard sort desc	倍長整数	133

定数	タイプ	値
pv pal standard zoom	倍長整数	139
pv pal style bold	倍長整数	196
pv pal style font name	倍長整数	193
pv pal style font size	倍長整数	194
pv pal style italic	倍長整数	197
pv pal style style sheet	倍長整数	195
pv pal style underline	倍長整数	198

## PV Control

---

関連するコマンド：PV GET CELL CONTROL、PV SET CELL CONTROL

定数	タイプ	値
pv control check box	倍長整数	3
pv control combo box	倍長整数	5
pv control drop down	倍長整数	4
pv control none	倍長整数	0
pv control push button	倍長整数	1
pv control radio button	倍長整数	2

## PV Directions

---

関連するコマンド：PV GET NEXT FREE CELL、PV GOTO NEXT CELL

定数	タイプ	値
pv to the bottom	倍長整数	1
pv to the left	倍長整数	2
pv to the right	倍長整数	0
pv to the top	倍長整数	3

## PV Document format

---

関連するコマンド：PV SAVE DOCUMENT

定数	タイプ	値
pv html	倍長整数	3
pv report	倍長整数	4
pv sylk	倍長整数	2
pv tab tab return	倍長整数	1
pv view	倍長整数	0

## PV Document properties

---

このテーマ内の定数を使用して、4D View 書類のプロパティの設定や取得を行うことができます。

■ “pv picture count” 定数は、書き込みにのみ使用することができます。

■ “pv document modified” 定数は、読み込みまたは書き込みに使用することができます。

—読み込みには、**PV Get document property** 関数を使用します。この場合、書類が変更されていれば1が、そうでなければ0が値として返されます。

—書き込みには、**PV SET DOCUMENT PROPERTY** コマンドを使用します。この場合、プロパティに1を設定すると、エリアをクローズする際にそのエリアが変更されたことを示すダイアログボックスが表示されます。プロパティに0を設定すると、ユーザやプログラムにより書類が後に変更されない場合は、ダイアログボックスが表示されません。

関連するコマンド：PV Get document property、PV SET DOCUMENT PROPERTY

定数	タイプ	値
pv column count	倍長整数	0
pv document modified	倍長整数	4
pv no formula external call	倍長整数	3
pv picture count	倍長整数	2
pv row count	倍長整数	1

## PV Drag drop allowed

---

このテーマ内の定数を使用して、ドラッグ&ドロップ動作に対し、4D Viewエリアで受け入れるソース項目およびターゲット項目のタイプを指定、または取得することができます。複数タイプの項目を許可するには、これらの定数を追加してゆきます。

この定数の使用例は、**PV SET DRAG SIGNATURES** コマンドの説明で提供されています。

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv DD adjacent cells	倍長整数	4
pv DD adjacent columns	倍長整数	256
pv DD adjacent rows	倍長整数	32
pv DD multiple cells	倍長整数	8
pv DD multiple columns	倍長整数	512
pv DD multiple rows	倍長整数	64
pv DD not allowed	倍長整数	0
pv DD single cell	倍長整数	2
pv DD single column	倍長整数	128
pv DD single row	倍長整数	16

## PV Drop action

---

関連するコマンド：PV Get drag and drop info

定数	タイプ	値
pv entire area	倍長整数	1
pv insert cell down	倍長整数	2
pv insert cell right	倍長整数	3
pv insert column	倍長整数	7
pv insert row	倍長整数	5
pv replace cell	倍長整数	4
pv replace column	倍長整数	8
pv replace row	倍長整数	6

## PV Drop info

---

関連するコマンド：PV Get drag and drop info

定数	タイプ	値
pv drag column	倍長整数	2
pv drag content	倍長整数	6
pv drag plugin	倍長整数	1
pv drag process	倍長整数	0
pv drag row	倍長整数	3
pv drag X offset	倍長整数	4
pv drag Y offset	倍長整数	5
pv drop action	倍長整数	14
pv drop column	倍長整数	9
pv drop content	倍長整数	13
pv drop plugin	倍長整数	8
pv drop process	倍長整数	7
pv drop row	倍長整数	10
pv drop X offset	倍長整数	11
pv drop Y offset	倍長整数	12

## PV Drop mode

---

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv drop insert only	倍長整数	1
pv drop insert or replace	倍長整数	0
pv drop replace only	倍長整数	2

## PV Event

---

関連するコマンド：PV Get on event method、PV ON EVENT

定数	タイプ	値
pv on active cell changed	倍長整数	8
pv on cell value changed	倍長整数	9
pv on clicked	倍長整数	2
pv on double clicked	倍長整数	4
pv on drag	倍長整数	10

定数	タイプ	値
pv on drop	倍長整数	11
pv on getting focus	倍長整数	0
pv on keyboard	倍長整数	6
pv on losing focus	倍長整数	1
pv on right clicked	倍長整数	3
pv on scrolled	倍長整数	5
pv on selection changed	倍長整数	7

## PV Header sort

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv header sort allowed	倍長整数	1
pv header sort callback	倍長整数	2
pv header sort indexed	倍長整数	3
pv header sort not allowed	倍長整数	0

## PV Headers & footers

関連するコマンド：PV Get print header、PV SET PRINT HEADER

定数	タイプ	値
pv footer center	倍長整数	5
pv footer left	倍長整数	4
pv footer right	倍長整数	6
pv header center	倍長整数	2
pv header left	倍長整数	1
pv header right	倍長整数	3

## PV Input enter key mode

このテーマ内の定数を使用して、入力時のEnterキーの動作を定義することができます。“標準”モードで使用した場合、Enterキーは入力の確定だけを行います。変更されたセルの右側 (“pv enter key as tab”) あるいは上側 (“pv enter key as return”) に位置するセルをアクティブにすることもできます。

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv enter key as return	倍長整数	2
pv enter key as tab	倍長整数	1
pv enter key standard	倍長整数	0

## PV Pane properties

---

- “pv pane true scroll” 定数は、書き込み用コマンド (**PV SET PANE PROPERTY...**) のみ使用することができます。
- “pv pane col row count” 定数は、読み込み用コマンド (**PV Get pane property...**) のみ使用することができます。

関連するコマンド：PV Get hor pane property、PV Get vert pane property、PV SET HOR PANE PROPERTY、PV SET VERT PANE PROPERTY

定数	タイプ	値
pv pane columns count	倍長整数	5
pv pane first column	倍長整数	4
pv pane first row	倍長整数	4
pv pane lock scrollbar	倍長整数	0
pv pane lock splitter	倍長整数	1
pv pane relative scroll	倍長整数	7
pv pane rows count	倍長整数	5
pv pane size in pixels	倍長整数	3
pv pane true scroll	倍長整数	6
pv pane view splitter cursor	倍長整数	2

## PV Picture mapping mode

---

関連するコマンド：PV Get picture property、PV SET PICTURE PROPERTY

定数	タイプ	値
pv mapping replicated	倍長整数	3
pv mapping scaled centered prop	倍長整数	6
pv mapping scaled to fit	倍長整数	5
pv mapping scaled to fit prop	倍長整数	4
pv mapping trunc non-centered	倍長整数	1
pv mapping truncated centered	倍長整数	2

## PV Picture properties

---

pv picture data widthおよびpv picture data height定数は、読み込み専用としてのみ使用可能です。

関連するコマンド：PV Get picture property、PV SET PICTURE PROPERTY

定数	タイプ	値
pv picture background	倍長整数	8
pv picture column	倍長整数	0
pv picture data height	倍長整数	5
pv picture data width	倍長整数	4
pv picture display height	倍長整数	7
pv picture display width	倍長整数	6
pv picture fixed size	倍長整数	10
pv picture horz offset	倍長整数	2
pv picture locked	倍長整数	11
pv picture mapping mode	倍長整数	9
pv picture row	倍長整数	1
pv picture vert offset	倍長整数	3

## PV Plugin properties

---

関連するコマンド：PV Get plugin property、PV SET PLUGIN PROPERTY

定数	タイプ	値
pv button height	倍長整数	5
pv button width	倍長整数	4
pv confirm convert dialog	倍長整数	6
pv default columns count	倍長整数	2
pv default rows count	倍長整数	3
pv load template on server	倍長整数	1
pv write template on server	倍長整数	0

## PV Print properties

---

関連するコマンド：PV Get print property、PV SET PRINT PROPERTY

定数	タイプ	値
pv print adjust area	倍長整数	10
pv print bottom margin	倍長整数	3
pv print centered	倍長整数	9
pv print dead bottom margin	倍長整数	18
pv print dead left margin	倍長整数	15
pv print dead right margin	倍長整数	17
pv print dead top margin	倍長整数	16
pv print frame each page	倍長整数	11
pv print grid	倍長整数	12
pv print headers	倍長整数	8
pv print left margin	倍長整数	0
pv print paper height	倍長整数	14
pv print paper width	倍長整数	13
pv print repeat first column	倍長整数	4
pv print repeat first row	倍長整数	6
pv print repeat last column	倍長整数	5
pv print repeat last row	倍長整数	7
pv print right margin	倍長整数	2
pv print top margin	倍長整数	1

## PV Report functions

---

関連するコマンド：PV REPORT MANY、PV REPORT ONE

定数	タイプ	値
pv report function average	倍長整数	1
pv report function count	倍長整数	4
pv report function max	倍長整数	3
pv report function min	倍長整数	2
pv report function none	倍長整数	-1
pv report function sum	倍長整数	0

## PV Select mode

---

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv select adjacent cells	倍長整数	8
pv select adjacent columns	倍長整数	5
pv select adjacent rows	倍長整数	2
pv select multiple cells	倍長整数	9
pv select multiple columns	倍長整数	6
pv select multiple rows	倍長整数	3
pv select not allowed	倍長整数	0
pv select single cell	倍長整数	7
pv select single column	倍長整数	4
pv select single row	倍長整数	1

## PV Selection action

---

このテーマ内の定数により、選択コマンドを使用する際、選択セルが既に存在している場合に行なう動作を指定することができます。

■ “pv selection set”：新しい選択範囲で既存の選択範囲を置き換えます。

■ “pv selection add”：新しい選択範囲は既存の選択範囲に追加されます。

関連するコマンド：PV SELECT ALL、PV SELECT CELL、PV SELECT COLUMNS、PV SELECT RANGE、PV SELECT RANGES LIST、PV SELECT ROWS

定数	タイプ	値
pv selection add	倍長整数	1
pv selection set	倍長整数	0

## PV Style format date time

---

関連するコマンド：PV Get cell property、PV Get range property、PV Get style property、PV SET CELL PROPERTY、PV SET RANGE PROPERTY、PV SET STYLE PROPERTY

定数	タイプ	値
pv Abbr Month Day Year	倍長整数	6
pv Abbreviated	倍長整数	2
pv Abbreviated H MM AM PM	倍長整数	13
pv Day Name	倍長整数	7
pv Day Number	倍長整数	8
pv HH MM	倍長整数	18
pv HH MM AM PM	倍長整数	21
pv HH MM SS	倍長整数	17
pv Hour Min	倍長整数	20
pv Hour Min Sec	倍長整数	19
pv Long	倍長整数	3
pv Long H MM AM PM	倍長整数	12
pv Month Day Year	倍長整数	5
pv Month Day Year H MM AM PM	倍長整数	15
pv Month Name	倍長整数	9
pv Month Number	倍長整数	10
pv Short	倍長整数	1
pv Short HH MM SS	倍長整数	14
pv Short2	倍長整数	4
pv Short2 Hour Min Sec	倍長整数	16
pv Year Number	倍長整数	11

## PV Style properties

---

関連するコマンド：PV Get cell property、PV Get range property、PV Get style property、PV SET CELL PROPERTY、PV SET RANGE PROPERTY、PV SET STYLE PROPERTY

定数	タイプ	値
pv style based on	倍長整数	4
pv style color back even	倍長整数	11
pv style color back odd	倍長整数	12
pv style color minus even	倍長整数	17
pv style color minus odd	倍長整数	18
pv style color text even	倍長整数	13
pv style color text odd	倍長整数	14
pv style color zero even	倍長整数	15
pv style color zero odd	倍長整数	16
pv style format alpha	倍長整数	6

定数	タイプ	値
pv style format bool	倍長整数	8
pv style format date time	倍長整数	9
pv style format num	倍長整数	7
pv style format picture	倍長整数	10
pv style hidden	倍長整数	1
pv style hor alignment	倍長整数	29
pv style locked	倍長整数	0
pv style rotation	倍長整数	31
pv style spellcheck	倍長整数	2
pv style text bold	倍長整数	22
pv style text condensed	倍長整数	27
pv style text extended	倍長整数	28
pv style text face	倍長整数	21
pv style text font	倍長整数	19
pv style text italic	倍長整数	23
pv style text outline	倍長整数	25
pv style text shadow	倍長整数	26
pv style text size	倍長整数	20
pv style text underline	倍長整数	24
pv style use picture height	倍長整数	3
pv style vert alignment	倍長整数	30

## PV Style special values

これらの定数を使用して、**PV SET STYLE PROPERTY** コマンドおよび **PV Get style property** 関数を標準の4D Viewスタイルシート（引数<スタイル>）に適用することができます。

関連するコマンド：PV Get style property、PV SET STYLE PROPERTY

定数	タイプ	値
pv style cells	倍長整数	-1
pv style col row headers	倍長整数	-2
pv style page footer header	倍長整数	-3

## PV Style values

---

後ろに“mixed”が付く定数は、プロパティを読み込むコマンドでのみ使用することができます。

関連するコマンド：PV Get cell property、PV Get range property、PV Get style property、PV SET CELL PROPERTY、PV SET RANGE PROPERTY、PV SET STYLE PROPERTY

定数	タイプ	値
pv value base style mixed	倍長整数	65535
pv value color mixed	倍長整数	65535
pv value filter mixed	倍長整数	65535
pv value font name mixed	倍長整数	65535
pv value font size mixed	倍長整数	65535
pv value format mixed	倍長整数	65535
pv value hor alignment center	倍長整数	2
pv value hor alignment default	倍長整数	0
pv value hor alignment left	倍長整数	1
pv value hor alignment mixed	倍長整数	255
pv value hor alignment right	倍長整数	3
pv value ignore	倍長整数	2
pv value ignore base style	倍長整数	65535
pv value ignore color	倍長整数	-1
pv value ignore filter	倍長整数	65535
pv value ignore font name	倍長整数	65535
pv value ignore font size	倍長整数	65535
pv value ignore format	倍長整数	65535
pv value ignore hor alignment	倍長整数	255
pv value ignore rotation	倍長整数	255
pv value ignore vert alignment	倍長整数	255
pv value mixed	倍長整数	2
pv value off	倍長整数	0
pv value on	倍長整数	1
pv value rotation 0	倍長整数	0
pv value rotation 180	倍長整数	2
pv value rotation 270	倍長整数	3
pv value rotation 90	倍長整数	1
pv value rotation mixed	倍長整数	255
pv value vert alignment bottom	倍長整数	3
pv value vert alignment center	倍長整数	2

定数	タイプ	値
pv value vert alignment mixed	倍長整数	255
pv value vert alignment top	倍長整数	1

## PV Triggers

---

- “pv trigger input” で始まるトリガは、データ入力動作に対してのみ使用することができます (“pv input trigger” プロパティ定数)。
- “pv trigger selection” で始まるトリガは、データ選択動作に対してのみ使用することができます (“pv select trigger” プロパティ定数)。
- その他のトリガは、データ入力と選択動作の両方で使用することができます。また、“pv drag trigger” プロパティ定数と一緒に使用することもできます。

複数のキーを許可したい場合には、Triggers定数を追加することができます。

データ入力と選択動作の両方に対して同じトリガが設定されている場合、入力トリガが優先されます。

関連するコマンド：PV Get area property、PV SET AREA PROPERTY

定数	タイプ	値
pv trigger input key	倍長整数	1
pv trigger input on enter	倍長整数	2
pv trigger input on gain sel	倍長整数	4
pv trigger none	倍長整数	0
pv trigger on alt click	倍長整数	32
pv trigger on alt double click	倍長整数	64
pv trigger on click	倍長整数	8
pv trigger on ctrl click	倍長整数	128
pv trigger on ctrl double click	倍長整数	256
pv trigger on double click	倍長整数	16
pv trigger on shift click	倍長整数	512
pv trigger on shift double clic	倍長整数	1024
pv trigger select on arrow	倍長整数	1
pv trigger select on return	倍長整数	4
pv trigger select on tab	倍長整数	2

## A

PV ADD DYNAMIC ARRAYS	136
PV ADD DYNAMIC FIELDS	138
PV Add font	229
PV Add format	232
PV ADD HOR SPLITTER	183
PV Add picture	192
PV Add style	222
PV ADD VERT SPLITTER	182
PV Area to blob	47
PV ARRAY TO CELLS	122

## B

PV BLOB TO AREA	46
-----------------	----

## C

PV CELLS TO ARRAY	124
PV CLEAR DYNAMIC COLUMNS	141
PV Color to index	256
PV COLOR TO RGB	255
PV Copy to blob	252
PV Create picture	195

## D

PV DELETE CELLS	162
PV DELETE COLUMNS	161
PV DELETE OFFSCREEN AREA	31
PV DELETE ROWS	161

## E

PV EXECUTE COMMAND	56
PV EXPORT	174

## F

PV FIELD TO CELLS	126
PV FIELDS LIST TO CELLS	128
PV FIND ALL	146
PV FIND ONE	144

## G

PV GET ALLOWED COM LIST	264
PV GET ALLOWED MET LIST	263
PV GET ALLOWED VAR LIST	262
PV Get area property	45
PV GET BORDER STYLE	65
PV Get cell boolean value	106
PV GET CELL CONTROL	120
PV GET CELL DATE TIME VALUE	104
PV Get cell date value	100
PV GET CELL FIELD	113
PV Get cell formula	115
PV Get cell name	89
PV GET CELL NAME LIST	90
PV Get cell num value	98
PV Get cell picture value	108
PV Get cell property	85
PV Get cell string value	94
PV Get cell text value	96
PV Get cell time value	102
PV Get cell value type	92
PV Get cell variable	110
PV GET COLUMN HEADER	167
PV Get column width	165
PV GET COMMAND STATUS	58
PV GET CURRENT CELL	22
PV GET DOCUMENT INFO	180
PV Get document property	178
PV Get drag and drop info	244
PV GET DRAG SIGNATURES	241
PV GET DRAG SOURCE	245

PV GET DROP SIGNATURES	243
PV GET DROP TARGET	246
PV GET FONT LIST	231
PV GET FORMAT LIST	235
PV Get hor pane property	189
PV GET LAST ERROR	54
PV GET NEXT FREE CELL	24
PV Get on command method	52
PV Get on error method	54
PV Get on event method	50
PV Get picture	194
PV Get picture property	197
PV Get plugin property	250
PV GET PREVIOUS ACTIVE CELL	27
PV Get print header	203
PV Get print property	208
PV Get range property	86
PV Get row header	168
PV Get row height	165
PV GET SELECTED RANGES LIST	215
PV GET STYLE LIST	225
PV Get style property	228
PV Get vert pane property	188
PV Get window title	259
PV GOTO CELL	25
PV GOTO NEXT CELL	26
I	
PV Index to color	257
PV INSERT CELLS	160
PV INSERT COLUMNS	158
PV INSERT ROWS	159
PV Is all selected	219
PV Is cell selected	211
PV Is column selected	217
PV Is range selected	213
PV Is row selected	218

## N

PV New offscreen area . . . . . 30

## O

PV ON COMMAND . . . . . 51

PV ON ERROR . . . . . 53

PV ON EVENT . . . . . 48

PV OPEN DOCUMENT . . . . . 170

## P

PV PASTE FROM BLOB . . . . . 253

PV PRINT . . . . . 200

PV PRINT FORMULAS . . . . . 201

## R

PV REDRAW . . . . . 59

PV REMOVE FONT . . . . . 230

PV REMOVE FORMAT . . . . . 233

PV REMOVE HOR SPLITTER . . . . . 185

PV REMOVE PICTURE . . . . . 193

PV REMOVE STYLE . . . . . 223

PV REMOVE VERT SPLITTER . . . . . 184

PV REPLACE ALL . . . . . 149

PV REPLACE ONE . . . . . 147

PV REPORT MANY . . . . . 130

PV REPORT ONE . . . . . 133

PV RGB to color . . . . . 254

## S

PV SAVE DOCUMENT . . . . . 172

PV SELECT ALL . . . . . 219

PV SELECT CELL . . . . . 210

PV SELECT COLUMNS . . . . . 216

PV SELECT RANGE . . . . . 212

PV SELECT RANGES LIST . . . . . 214

PV SELECT ROWS . . . . . 217

PV SET ALLOWED COM LIST . . . . . 264

PV SET ALLOWED MET LIST . . . . . 263

PV SET ALLOWED VAR LIST . . . . . 262

PV SET AREA PROPERTY . . . . . 32

PV SET BORDER STYLE	63
PV SET CELL BOOLEAN VALUE	105
PV SET CELL CONTROL	116
PV SET CELL DATE TIME VALUE	103
PV SET CELL DATE VALUE	99
PV SET CELL FIELD	111
PV SET CELL FORMULA	114
PV SET CELL NAME	88
PV SET CELL NUM VALUE	97
PV SET CELL PICTURE VALUE	107
PV SET CELL PROPERTY	68
PV SET CELL STRING VALUE	93
PV SET CELL TEXT VALUE	95
PV SET CELL TIME VALUE	101
PV SET CELL VARIABLE	109
PV SET COLUMN HEADER	166
PV SET COLUMNS WIDTH	163
PV SET COMMAND STATUS	57
PV SET DOCUMENT INFO	179
PV SET DOCUMENT PROPERTY	176
PV SET DRAG SIGNATURES	238
PV SET DROP SIGNATURES	242
PV SET FORMAT	234
PV SET HOR PANE PROPERTY	187
PV SET PICTURE PROPERTY	196
PV SET PLUGIN PROPERTY	248
PV SET PRINT HEADER	202
PV SET PRINT PROPERTY	204
PV SET RANGE BORDER	62
PV SET RANGE PROPERTY	77
PV SET ROW HEADER	167
PV SET ROWS HEIGHT	164
PV SET STYLE NAME	224
PV SET STYLE PROPERTY	226
PV SET VERT PANE PROPERTY	186
PV SET WINDOW TITLE	258
PV SORT MANY	152

PV SORT ONE	150
PV SPECIAL CLEAR	153
PV SPECIAL CUT	155
PV SPECIAL PASTE	154
V	
PV VALIDATE CURRENT CELL	23