

4D View

-  はじめに
-  PVエリア
-  PVカレントセル
-  PVスタイル
-  PVセルの値
-  PVセルプロパティ
-  PVセル操作
-  PVセレクション
-  PVツール
-  PVドキュメント
-  PVドラッグアンドドロップ
-  PVピクチャ
-  PVプラグインプロパティ
-  PVプリント
-  PVペイン
-  PV入力認可
-  PV列と行
-  PV罫線
-  定数テーマリスト
-  付録
-  コマンドリスト (文字順)

🌱 はじめに

- 🌱 4D Viewとは
- 🌱 表記方法について
- 🌱 4D Viewのコマンドと定数
- 🌱 4D Viewエリアを使用する
- 🌱 4D Viewメニューコマンドの使用
- 🌱 クロスプラットフォームドキュメントの管理

4D Viewとは

4D Viewプラグインは4D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4D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。

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: *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: *PV Get on command method* メソッドエディタにおいて、4D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。

```
  \ 4D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。

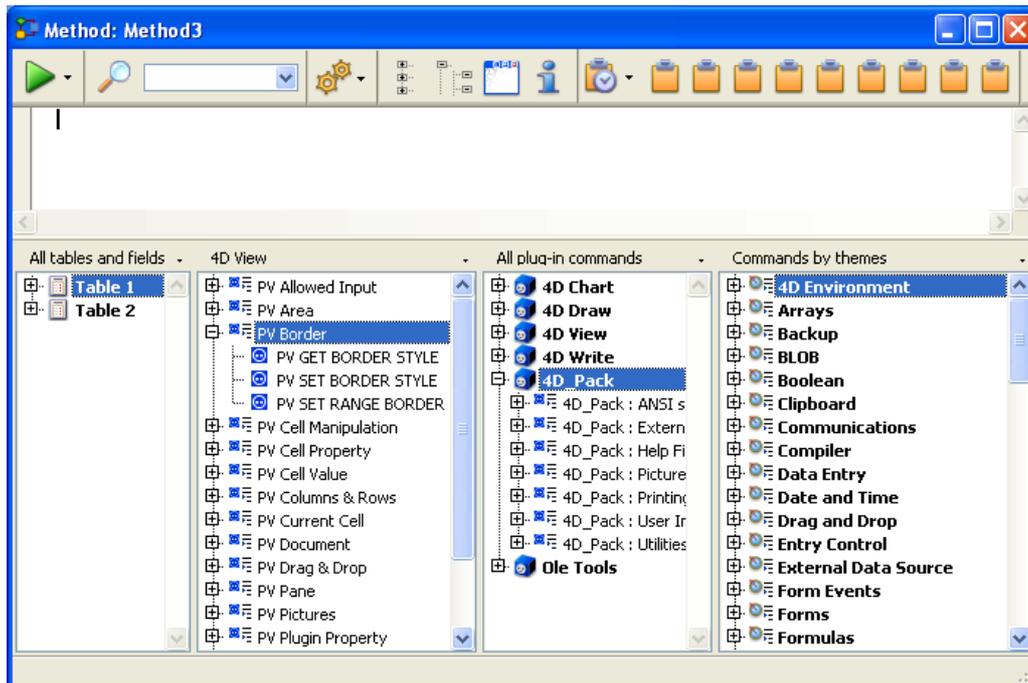
4D Viewのコマンドと定数

メソッドエディタ内のコマンド

4D Viewコマンドを4Dメソッドエディタ内のコマンドリストに表示させることができます。リストには4D Viewコマンドのみ、あるいは利用可能なすべてのプラグインコマンドを表示させることができます：



プラグインコマンドは階層リスト内で"テーマ"ごとにグループ化されます：



プラグインコマンドはエクスプローラの**プラグイン**ページにも表示されます。

Note: プラグイン定数は4D定数リストに追加されます。4D View定数のテーマ名には先頭に"_"がついているので、リストの最後にグループ化されます。

4Dコマンドと同様に、メソッドエディタの必要な場所に4D Viewコマンドを挿入できます。メソッドエディタに直接タイプしたり、リスト内のコマンド名をダブルクリックしたりして入力できます。

4D Viewルーチンはデータベース、プロジェクト、フォーム、オブジェクト、トリガなどどのタイプのメソッド内でも使用できます。

🌱 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 のプラグインウィンドウ
- 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ィンドウを開きます。

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。

プラグインウィンドウを開くには、4Dの**ツール**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**を実行します。

4D Viewでは、表示エリアの作成だけでなく、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"を参照してください。

4D Viewエリア参照

ルーチンを使用して4D View ドキュメントを操作するには、4D View エリアのIDを特定する必要があります。このID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ィンドウ、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`area`と呼ばれます。

以下の2つのタイプの`area`変数があります:

- 組み込みエリアの変数
- プラグインウィンドウまたはオフスクリーンエリア用に作成した変数

組み込みエリアの名前

フォーム上に4D View エリアを作成して名前を付けると、4D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`area`にSheetと指定します。

プラグインウィンドウおよびオフスクリーンエリアのID

Open external windowまたは**PV New offscreen area**を使用してプラグインウィンドウやオフスクリーンエリアを作成すると、関数からエリアの識別番号が変数に返されます。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ィ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 のプラグインウィンドウを作成し、エリア識別番号を変数`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 でフォーム全体を使用したり、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たりできます。

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で使用します。4D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、レコード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。OK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 のユーザーリファレンスマニュアルを参照してください。

4D Viewプラグインウィンドウ

4Dの**Open external window**を使用してプラグインウィンドウを開き、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は新しくウィンドウを開き、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ィンドウを開き、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。**Open external window**コマンドに関する説明は4Dランゲージリファレンスマニュアルを参照してください。

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はできません。一般的に2つの状況においてこのエリアが役立ちます。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*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、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、4D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**テーマの定数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

4D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MacOSで作成された4D Viewを使用しているデータベースを、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。

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/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	Mac OSタイプ	クリエーター	Windows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、心に留めておいてください:

- 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、ディスク名とフォルダを":" (コロン) を使用して区切ります（例: "MyDisk:Folder1:Folder2:MyBase"）。
- Windows上で、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"¥" (バックスラッシュ、日本語環境では円マークで表示される場合もあります) 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4D Viewドキュメントは、拡張子付きで保存されていればWindows にコピーして直接開くことができます。例えばドキュメント"MyDoc"が"MyDoc.4PV"という名前で作成され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4D Viewドキュメントは、何も変更せずにMacOSにコピーして開くことができます。

テンプレート

4D Viewはサーバープラットフォームに関わらず、Mac OSとWindowsクライアントマシン両方で、ユーザーに対し完全に透過的にテンプレートを管理します。

- Mac OSでサーバーが動作している場合、テンプレートは"AreaName_"と命名されます。
- Windowsでサーバーが動作している場合、テンプレートは"AreaName_.4PV"と命名されます。

PVエリア

エリアコマンドについて

-  PV Area to blob
-  PV BLOB TO AREA
-  PV DELETE OFFSCREEN AREA
-  PV EXECUTE COMMAND
-  PV Get area property
-  PV GET COMMAND STATUS
-  PV GET LAST ERROR
-  PV Get on command method
-  PV Get on error method
-  PV Get on event method
-  PV New offscreen area
-  PV ON COMMAND
-  PV ON ERROR
-  PV ON EVENT
-  PV REDRAW
-  PV SCROLL AREA
-  PV SET AREA PROPERTY
-  PV SET COMMAND STATUS

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。これらのルーチンを使用して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したり、あるいは4D ViewエリアをフィールドやBLOB変数に保存したり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開発者が捉えることのできるコマンドや、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エリアテーマのコマンドのうち、コールバックメソッドに関連する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 Area to blob

PV Area to blob (area) -> 戻り値

引数	型		説明
area	倍長整数	→	4D Viewエリア
戻り値	BLOB	↩	保存先のBLOB

説明

PV Area to blob コマンドは、*area* をBLOBフィールドや変数に格納します。その後、フィールドを保存したり、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 *area* 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](#) コマンドの説明を参照

PV BLOB TO AREA

PV BLOB TO AREA (area ; BLOB)

引数	型		説明
area	倍長整数	→	4D View エリア
BLOB	BLOB	→	エリアが格納されたBLOB

説明

PV BLOB TO AREA コマンドは、*blob* に格納された 4D View のスプレッドシートを *area* 内に開きます。

引数 *area* 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) `エリアの内容を格納するプロセス変数  
BlobVariable:=PV Area to blob(Area) `変数に格納
```

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) `変数から内容を取得
```

PV DELETE OFFSCREEN AREA

PV DELETE OFFSCREEN AREA (area)

引数	型	説明
area	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

PV EXECUTE COMMAND

PV EXECUTE COMMAND (area ; command)

引数	型		説明
area	倍長整数	⇒	4D View エリア
command	倍長整数	⇒	コマンド番号

説明

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

PV Commands テーマ内の定数を使用して引数 *command* を指定します。

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($status) `1=有効 0=無効
C_INTEGER($checkbox) `0=チェックされていない, 1=チェックされている
C_STRING(30;$name) `対応するコマンド名

`情報を取得
PV GET COMMAND STATUS(Area;pv_cmd_view_Vscrollbar;$status;$checkbox;$name)
If($checkbox=1) `縦スクロールバーは表示されているか?
    PV EXECUTE COMMAND(Area;pv_cmd_view_Vscrollbar) `縦スクロールバーを隠す
    PV SET COMMAND STATUS(Area;pv_cmd_view_Vscrollbar;0) `無効にする
Else
    PV SET COMMAND STATUS(Area;pv_cmd_view_Vscrollbar;1) `コマンドを有効にする
    PV EXECUTE COMMAND(Area;pv_cmd_view_Vscrollbar) `スクロールバーを表示
End if
```

PV Get area property

PV Get area property (area ; property) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
property	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 コマンドは、*area* で指定された 4D View の、*option* に対応する値を *property* に返します。
option の指定には **PV Area properties** テーマの定数を使用します。
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 コマンドの例題参照

PV GET COMMAND STATUS

PV GET COMMAND STATUS (area ; command ; status ; checkbox ; name)

引数	型		説明
area	倍長整数	→	4D View エリア
command	倍長整数	→	コマンド番号
status	整数	←	0 = 使用不可; 1 = 使用可
checkbox	整数	←	0 = チェックなし; 1 = チェックあり
name	文字	←	コマンド名

説明

PV GET COMMAND STATUS コマンドは、*command* で指定された 4D View コマンドに関する *status*、*checkbox*、*name* の情報を取得するために使用します。

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

引数 *status* 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 *checkbox* 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 *name* には、コマンドラベルが返されます。

例題

PV EXECUTE COMMAND コマンドの例題を参照

PV Get on command method

PV Get on command method (area ; command) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
command	倍長整数	→	コマンド番号
戻り値	文字	↩	4D メソッド名

説明

PV Get on command method コマンドは、4D Viewメニューの*command*に関連付けられたメソッドの名前を返します。
引数*command*の指定には、**PV Commands**テーマの定数を使用します。
関連付けられたメソッドが存在しない場合、*PV Get on command method*は空の文字列を返します。

例題

PV ON COMMAND コマンドの例題参照

PV Get on error method

PV Get on error method -> 戻り値

引数	型		説明
戻り値	文字		4Dメソッド名

説明

PV Get on error method コマンドは、*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* は空の文字列を返します。

例題

PV ON ERROR コマンドの例題参照

🔧 PV Get on event method

PV Get on event method (area ; event) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
event	倍長整数	→	4D View イベント
戻り値	文字	↩	4D メソッド名

説明

PV Get on event method コマンドは、指定された *event* に関連付けられたコールバックメソッド名を返します。
event の指定には、**PV Event** テーマの定数を使用します。

定数	型	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 column resize	倍長整数	12
pv on column sort	倍長整数	14
pv on contextual click	倍長整数	15
pv on double clicked	倍長整数	4
pv on drag	倍長整数	10
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 row resize	倍長整数	13
pv on scrolled	倍長整数	5
pv on selection changed	倍長整数	7

event 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* は空の文字列を返します。

例題

イベント呼び出しを一時的に無効にしておき、本来の呼び出しを実行する前に、特定の処理を実行しておくとい場合があります。

例として、pv on cell value changed イベントに対して *PV Get on event method* 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)

  `現在インストールされているメソッド名を保存する
  $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

  `... 実行する処理をここに書く

  PV ON EVENT(Area;pv on cell value changed;$EventMethod) `復元
```

PV New offscreen area

PV New offscreen area -> 戻り値

引数	型		説明
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参照

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

PV BLOB TO AREA (*\$OffscreenArea*; [Model]BlobField_)

PV SELECT RANGE (*\$OffscreenArea*; 1; 1; 3; 3; *pv selection set*)

PV EXECUTE COMMAND (*\$OffscreenArea*; *pv cmd edit copy*) `セレクションをコピー

PV DELETE OFFSCREEN AREA (*\$OffscreenArea*) `メモリーを解放

PV GOTO CELL (Area; 1; 5)

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

PV ON COMMAND (area ; command ; method)

引数	型		説明
area	倍長整数	→	4D View エリア
command	倍長整数	→	コマンド番号
method	文字	→	4D メソッド名

説明

PV ON COMMANDコマンドは、4D View メニューのcommand を4D のmethodに関連付けます。

引数command の指定には、**PV Commands**テーマの定数を使用します。

method 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

コマンドに関連付けたメソッド呼び出しを削除するには、method引数に空の文字列を渡して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 file 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 file print document;"PrintMethod")
End if

If (PV Get on command method(area;pv cmd file print formulas) #"PrintMethod")
  \標準のレコード印刷のトレース
  PV ON COMMAND(area;pv cmd file 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エリア参照
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 file print formulas)
  CREATE RECORD ([PrintSpy]) \新規レコード
  [PrintSpy]CurUser:=Current user \プリントジョブを要求したのはだれか?
  [PrintSpy]Dte:=Current date (*) \印刷日
  [PrintSpy]Tme:=Current time \印刷時刻
  [PrintSpy]Subject:="Print area formulas"
  SAVE RECORD ([PrintSpy]) \作成したレコードを保存

: ($2=pv cmd file print document) \これはプリントリクエストか?
  CREATE RECORD ([PrintSpy]) \新規レコード
  [PrintSpy]CurUser:=Current user \プリントジョブを要求したのはだれか?
  [PrintSpy]Dte:=Current date (*) \印刷日
  [PrintSpy]Tme:=Current time \印刷時刻
  [PrintSpy]Subject:="Standard area print"
  SAVE RECORD ([PrintSpy]) \作成したレコードを保存

Else
  TRACE \その他のケース?
End case

```


PV ON ERROR (method)

引数	型	説明
method	文字 →	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 *method* をインストールします。

この割り込み用のメソッドは、4D View コマンドの呼び出し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 *method* は3 つの引数を受け取ります：

\$1 : 4D View エリアの参照

\$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エラーコードリスト](#)に記載されています。

method 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") `管理メソッドがインストールされているか?  
    PV ON ERROR("ErrMethMan") `メソッドをインストール  
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  
C_LONGINT ($2) `エラー番号  
C_TEXT ($3) `エラーテキスト  
ALERT ("エラー: "+String($2)+Char(13)+$3)
```

PV ON EVENT (area ; event ; method)

引数	型		説明
area	倍長整数	→	4D View エリア
event	倍長整数	→	4D View イベント
method	文字	→	メソッド名

説明

PV ON EVENTコマンドは、*method* を4D Viewの*event*に関連付けるために使用します。*event*が発生するたびに、*method* が実行されます。

引数*event*の指定には、**PV Event**テーマの定数を使用します。

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
- \$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)

\$3には以下の値のうちいずれかが設定されます (複数のキーが押された場合には、これらの値を加算されます)：

0	なし
512	Shiftキー
2048	Altキー
4096	Ctrlキー (Windows) / Commandキー (Mac OS).

- **クリック管理** ([pv on clicked](#), [pv on right clicked](#), [pv on double clicked](#), [pv on contextual click](#) イベント):
 - *event* (クリック, 右クリック, ダブルクリック, コンテキストクリック) がセル上で生成されると、\$4には列番号が、\$5には行番号が渡されます。クリックが行ヘッダーで生成されると、\$4には0が渡されます。クリックが列ヘッダー上で生成されると、\$5に0が設定されます。クリックが左上の角で生成されると、\$4と\$5に0が渡されます。
 - [pv on contextual click](#)イベントは、ユーザーがマウスボタンを離れたときに生成されます。対して[pv on right clicked](#)イベントはボタンが押されたときに生成されます。これら2つのイベントを使用して、ポップアップコンテキストメニューのインターフェースを置くことができます。[pv on contextual click](#)イベントはWindowsの処理に、[pv on right clicked](#)はMac OSの処理に対応します。これら2つのイベントを同時に使用できます。
 - *event*がクリック, 右クリック, ダブルクリック, コンテキストクリックの場合、\$6は0に設定されます。
- **セクションの変更** ([pv on selection changed](#)イベント):
 - 新しいセクションが複数のセルや列、行を含む場合、\$4と\$5は0に設定されます。
 - 新しいセクションが1つのセルを含む場合、\$4と\$5にはそれぞれ列番号と行番号が返されます。
 - 新しいセクションが列の場合、\$4に列番号が渡され、\$5は0に設定されます。
 - 新しいセクションが行の場合、\$4は0に設定され、\$5には行番号が渡されます。
- **ファンクションキー**: [pv on keyboard](#) イベントのコンテキストで、ファンクションキーが有効になっていると、\$6引数に0が設定されます。この場合、4Dの*Keycode*システム変数を使用して、有効にされたファンクションキーのコードを知ることができます。
- **並び替え**: [pv on column sort](#) イベントは列がソートされた直後に生成されます。ここでユーザーアクションをコントロールすることができます。この場合\$6にはソート順を表す値が設定されます。値は**PV Header sort**テーマの以下の定数と比較できます：


```
[#table_kst id="96691,96695"/]
```
- **サイズ変更**: [pv on column resize](#)と[pv on row resize](#)イベントは、ユーザーが列や行のサイズを変更すると生成されます。(PV SET COLUMNS WIDTHやPV SET ROWS HEIGHTコマンドを使用して) プログラムで変更した場合は生成されません。

\$0 がTrueの場合、*event* は考慮されません。

\$0 がFalseの場合、*event* が考慮されます。

Note: 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。

area が0の場合、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、データベースが開かれる際に実行される**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、method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 1

PV VALIDATE CURRENT CELL, PV GET PREVIOUS ACTIVE CELL, PV GET CELL FIELD, PV Get on event method, PV SAVE DOCUMENT コマンドの例題参照

例題 2

ユーザーはヘッダーをクリックして並び替えを行います。**PM_Event**メソッドを使用してどの列がソートされたか、およびソート順を取得します。

```
`pv on column sort イベントで呼び出されるメソッドをインストール
PV ON EVENT(area;pv on column sort;"PM_Event")
```

```
`PM_Event メソッド
C_BOOLEAN($0)
C_LONGINT($1;$2;$3;$4;$5;$6)
C_STRING(12;$SortOrder)
If($2=pv on column sort)
  Case of
    :($6=pv ascending sort)
      $SortOrder:="昇順"
    :($6=pv descending sort)
      $SortOrder:="降順"
  End case
  ALERT("並び替えは列"+String($4)+"で"+$SortOrder+"順に行われました。")
End if
```

例題 3

列ヘッダをダブルクリックすると列がリサイズされます。しかしダブルクリックは2つのイベント、pv on clickedそのあとpv on double clickedを連続して生成します。

結果として、並び替えがPV SET AREA PROPERTYの呼び出しで許可されているとき、ヘッダをダブルクリックするとまず列がソートされ、そしてリサイズが行われます。ダブルクリックの場合にリサイズのみを行わせたい場合は、pv on clickedイベントを捕捉して処理をさせないようにしなければなりません。これを行うには、pv on clicked イベント生成時に実行されるメソッドをインストールします:

```
`pv on clickedイベントで呼び出されるメソッドをインストール
PV ON EVENT(area;pv on clicked;"PM_Event")
```

```
`PM_Event メソッド
C_BOOLEAN($0)
C_LONGINT($1;$2;$3;$4;$5;$6)
If($2=pv on clicked)
  $0:=True `イベントは無視され、ソートは行われぬ
End if
```

PV REDRAW (area)

引数	型		説明
area	倍長整数	→	4D View エリア

説明

*PV REDRAW*コマンドは、4D View の*area*を強制的に再描画するために使用します。

例題

行と列のサイズ変更を行う*PV SET COLUMNS WIDTH*コマンドの例題を参照してください。

PV SCROLL AREA (area ; horizontal ; vertical ; mode)

引数	型		説明
area	倍長整数	⇒	4D Viewエリア
horizontal	倍長整数	⇒	ピクセル数または列番号
vertical	倍長整数	⇒	ピクセル数または行番号
mode	整数	⇒	スクロールモード: 0 = 絶対、1 = 相対、2 = セル

説明

*PV SCROLL AREA*コマンドを使用して、*area* で指定した4D Viewエリアの内容を、*horizontal*と*vertical*に渡された値だけ、プログラムでスクロールできます。*mode*引数でスクロールタイプを選択することで、ピクセルあるいはセル座標を渡すことができます。

Note: ドキュメントに複数のペインがあるとき、スクロールはカレントのペインに適用されます。

- *mode*引数に0を渡すと、*horizontal*と*vertical*に渡された値はピクセルを表し、スクロールはエリアの先頭のセルを基点として実行されます (絶対指定)。
- *mode*引数に1を渡すと、*horizontal*と*vertical*に渡された値はピクセルを表し、スクロールはエリアに表示されている先頭のセルを基点として実行されます (相対指定)。

*horizontal*と*vertical*に正数を渡すと、スクロールは右および下方向に向かって行われます。負数を渡すと、左および上方向に向かって行われます。

Note: 4D Viewはスクロール後に先頭のセルが完全に表示されるよう、表示位置を調整します。

- *mode*引数に2を渡すと、*horizontal*と*vertical*に渡された値はそれぞれ列と行番号を表します。この座標はスクロール後に左上に表示されるセルを指定します。

例題

この例題はクエリ後に自動でドキュメントをスクロールします:

```
PV FIND ONE(area;"Smith";1;1;$col;$row)
PV SCROLL AREA(area;$col;$row;2)
```

PV SET AREA PROPERTY (area ; property ; value)

引数	型		説明
area	倍長整数	⇒	4D Viewエリア
property	倍長整数	⇒	プロパティ番号
value	倍長整数	⇒	プロパティ値

説明

PV SET AREA PROPERTY コマンドは、指定された4D Viewの*areal*に対し、*property*の*value*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。

area 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*property*の指定には、**PV Area properties** テーマの定数を使用します。引数*value*を指定するには、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*value*引数を解説したリストを以下に示します：

定数	型	値	コメント
pv allow undo redo	倍長整数	39	<p>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：</p> <ul style="list-style-type: none"> 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)。 <p>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、入力が確定されます。関連する値：PV Arrow keys テーマ内の定数。</p> <ul style="list-style-type: none"> 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。 <p>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。</p> <ul style="list-style-type: none"> 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 carriage return	倍長整数	8	
pv column headers height	倍長整数	21	<p>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)</p>
pv copy hidden	倍長整数	19	<p>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：</p> <ul style="list-style-type: none"> 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 current cell highlight	倍長整数	3	<p>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：</p> <ul style="list-style-type: none"> 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ならない。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 drag allowed	倍長整数	13	<p>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。</p> <ul style="list-style-type: none"> 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

の列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列、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。

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 drag trigger 12 倍長整数

- 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。

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 drop allowed 33 倍長整数

- 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 : 単一の列からなる選択範囲をドロップできる。
- pv DD adjacent columns : 複数の隣接する列、または単一の列からなる選択範囲をドロップできる。
- pv DD multiple columns : 複数の列 (隣接しているかどうかに関係なく)、または単一の列からなる選択範囲をドロップできる。
- pv DD 4D objects: 4Dオブジェクトをドロップできる。4Dフィールド (BLOBとサブテーブルを除く) フィールドと、(BLOBを除く) 変数をドロップできます。

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列、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。

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 mode 14 倍長整数

- 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 field tag 倍長整数数 17 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 倍長整数数 18 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。

列ヘッダーがクリックされた時に、(動的あるいは静的なデータの) 標準のソートを行うか禁止するかを設定します。 . 関連する値: **PV Header sort** テーマの以下の値:

pv headers sort 倍長整数数 20

- [pv sort not allowed](#) (デフォルト値): ユーザーが列ヘッダをクリックした際に、4D Viewは標準のソートを行わない (その代わりに開発者によるカスタマイズされたソートを実行できます)
- [pv sort allowed](#): ユーザーが列ヘッダをクリックした際に、4D Viewは標準のソートを実行する。この場合、ソート順を示すマークがヘッダーに表示されます。次に久利いくすると昇順と降順が入れ替わります。

動的な列をソートすると、他の列も同期してソートされます。結果レコードの並びを保つことができます。静的な列のソートはその列のみをソートします。

pv hor pane count 倍長整数数 11 エリアの横方向のペイン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: ペイン数。

Enter キー (数値キーパッド) 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 input enter key mode 倍長整数数 15

- [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。

エリアの入カトリガを設定します。データ入力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 input trigger 倍長整数数 6

- [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 click](#)) 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 record tag	倍長整数 16	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 resizable columns	倍長整数 4	カラムサイズの変更の可、不可を設定します。関連する値 : <ul style="list-style-type: none"> ● 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	倍長整数 5	行サイズの変更の可、不可を設定します。関連する値 : <ul style="list-style-type: none"> ● 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 row headers width	倍長整数 22	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 saving dialog	倍長整数 37	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 : <ul style="list-style-type: none"> ● 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 select highlight	倍長整数 1	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 : <ul style="list-style-type: none"> ● 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 mode	倍長整数 0	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。 <ul style="list-style-type: none"> ● 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。

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 select null | 倍長整数
2 | <ul style="list-style-type: none">● <code>pv value on</code> : 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。● <code>pv value off</code> : エリアの選択は必須。 |
|----------------|-----------|---|

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 select trigger | 倍長整数
7 | <ul style="list-style-type: none">● <code>pv trigger none</code> : 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。● <code>pv trigger select on arrow</code> : 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。● <code>pv trigger select on tab</code> : 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。● <code>pv trigger select on return</code> : 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。● <code>pv trigger on click</code> : 選択範囲の決定はマウスクリックによって行う。● <code>pv trigger on double click</code> : 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。● <code>pv trigger on alt click</code> : 選択範囲の決定はAlt キー+ クリックによって行う。● <code>pv trigger on alt double click</code> : 選択範囲の決定はAlt キー+ ダブルクリックによって行う。● <code>pv trigger on ctrl click</code> : 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）。● <code>pv trigger on ctrl double click</code> : 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）。● <code>pv trigger on shift click</code> : 選択範囲の決定はShift キー+ クリックによって行う。● <code>pv trigger on shift double clic</code> : 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 click)` 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 show borders toolbar | 倍長整数
29 | 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： <ul style="list-style-type: none">● <code>pv value on</code> : 枠線ツールバーを表示する。● <code>pv value off</code> : 枠線ツールバーを表示しない。 |
|-------------------------|------------|---|

- | | | |
|------------------------|------------|--|
| pv show column headers | 倍長整数
23 | 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： <ul style="list-style-type: none">● <code>pv value on</code> : カラムヘッダを表示する。● <code>pv value off</code> : カラムヘッダを表示しない。 |
|------------------------|------------|--|

- | | | |
|-------------------------|------------|---|
| pv show formula toolbar | 倍長整数
30 | エリアの4D View フォーマットツールバーの表示、非表示を設定します。関連する値： <ul style="list-style-type: none">● <code>pv value on</code> : フォーマットツールバーを表示する。● <code>pv value off</code> : フォーマットツールバーを表示しない。 |
|-------------------------|------------|---|

- | | | |
|------------------|------------|--|
| pv show hor grid | 倍長整数
31 | 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： <ul style="list-style-type: none">● <code>pv value on</code> : 水平グリッドを表示する。● <code>pv value off</code> : 水平グリッドを表示しない。 |
|------------------|------------|--|

🔧 PV SET COMMAND STATUS

PV SET COMMAND STATUS (area ; command ; status)

引数	型		説明
area	倍長整数	→	4D View エリア
command	倍長整数	→	コマンド番号
status	整数	→	0 = 使用不可 : 1 = 使用可

説明

PV SET COMMAND STATUS コマンドは、*command* 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できます。**PV Commands** テーマの定数を使用して *command* 引数を指定します。*command* 引数に0を渡すと、コマンドはすべての4D Viewメニューコマンドのステータスを変更します。

- *status* 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。
- *status* 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も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でも、*PV ON COMMAND* コマンドも利用できません。

例題 1

参照の表示をさせたくない場合は、4D Viewエリアに以下のコードを記述します：

```
3番目の引数に0を渡して選択不可にする  
PV SET COMMAND STATUS(area;pv_cmd_view_references;0)
```

例題 2

PV EXECUTE COMMAND コマンドの例題参照

PVカレントセル

 カレントセルコマンドについて

 PV GET CURRENT CELL

 PV GET NEXT FREE CELL

 PV GET PREVIOUS ACTIVE CELL

 PV GOTO CELL

 PV GOTO NEXT CELL

 PV VALIDATE CURRENT CELL

カレントセルコマンドについて

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 (area ; column ; row)

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	←	アクティブセルの列番号
row	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、*area*のカレントセルの座標を引数*column*と*row*に返します。

例題

S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 ($Column#0) &NBSP; &&NBSP; ($Row#0) &NBSP; &NBSP; `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“Here”が格納される
End if
```

PV GET NEXT FREE CELL (area ; direction ; column ; row)

引数	型		説明
area	倍長整数	→	4D View エリア
direction	整数	→	方向定数
column	倍長整数	←	列番号
row	倍長整数	←	行番号

説明

PV GET NEXT FREE CELLコマンドは、指定されたdirectionにある次のフリーなセルの座標をcolumn とrow に返します。direction 軸はPV Directions 定数テーマの4 つの値から1 つを指定します:

定数	型	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

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）があるものとして。この配列にある行数と列数を数えてみましょう。

入力されている最初のセルは、列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</p>
$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)+"列"+String ($LowRow-$StartRow)+"行です。")
    
```

PV GET PREVIOUS ACTIVE CELL

PV GET PREVIOUS ACTIVE CELL (area ; column ; row)

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	←	セルのカラム番号
row	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 *area* の前回アクティブであった（カレント）セルの座標を、引数 *column* と *row* に返します。

注: カレントセルの“スタック”は存在しません。最初にか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履歴を管理するかどうかは、開発者に任されています。

例題

下記の例題では、特定のセルに対して“追い出し”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) `列番号
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

If ($Column=3) & ($Row=5) `セルC5 がカレントか
  PV GET PREVIOUS ACTIVE CELL($1;$Column;$Row) &NBSP; &NBSP; `前回のセル
  PV GOTO CELL($1;$Column;$Row) &NBSP; &NBSP; `再度カレントセルになる
End if
```

PV GOTO CELL (area ; column ; row)

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	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、*column*と*row*で指定されたセルがその*area*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
```

PV GOTO NEXT CELL (area ; direction)

引数	型	説明
area	倍長整数	4D View エリア
direction	整数	方向定数

説明

PV GOTO NEXT CELLコマンドが呼び出されると、*direction* で指定された方向にある次のセルがその*area*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。

direction 軸は**PV Directions**定数テーマの4 つの値から1 つを指定します:

定数	型	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

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;"Total Amount";1;0) ` "Total Amount"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
```

` 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)
```

PV VALIDATE CURRENT CELL

PV VALIDATE CURRENT CELL (area)

引数	型	説明
area	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[エリアコマンドについて](#)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) `列番号
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)
  BEEP
  PV VALIDATE CURRENT CELL($1) `セル内容が有効になる
End if
```

PVスタイル

スタイルコマンドについて

-  PV Add font
-  PV Add format
-  PV Add style
-  PV GET FONT LIST
-  PV GET FORMAT LIST
-  PV GET STYLE LIST
-  PV Get style property
-  PV REMOVE FONT
-  PV REMOVE FORMAT
-  PV REMOVE STYLE
-  PV SET FORMAT
-  PV SET STYLE NAME
-  PV SET STYLE PROPERTY

スタイルコマンドについて

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。

デフォルトで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4Dのデータ内に4D Viewエリアと一緒に保存されます。

文字フォント

デフォルトで、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

4D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 format

PV Add format (area ; string) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
string	文字	→	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 コマンドは、フォーマットの *string* を *area* に追加し、そのID を返します。

string が既に存在する場合、*PV Add format* 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"
PV GET FORMAT LIST (Area; $FormatNumArray; $FormatStringArray) `利用可能なフォーマットのリスト

$Position := Find in array ($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で利用できない?
If (Find in array ($FormatStringArray; $NewFormat) = -1)
    $Position := Size of array ($FormatStringArray) + 1 `これを追加する
    INSERT IN ARRAY ($FormatStringArray; $Position) `配列をリサイズする
    INSERT IN ARRAY ($FormatNumArray; $Position)
    $FormatStringArray { $Position } := $NewFormat `フォーマットを代入
    $FormatNumArray { $Position } := PV Add format (Area; $NewFormat) `番号を代入
End if
```

PV Add style

PV Add style (area ; name) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
name	文字	→	スタイルシート名
戻り値	倍長整数	↩	スタイルシートID

説明

PV Add style コマンドは、引数 *name* を使用して *area* 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* 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 コマンドの例題参照

エラー管理

同じ名前のスタイルシートが既に存在する場合、*area* によりエラーが返されます。

PV GET FONT LIST

PV GET FONT LIST (area ; fonts ; names)

引数	型		説明
area	倍長整数	→	4D View エリア
fonts	倍長整数配列	←	フォントID の配列
names	文字配列	←	フォント名の配列

説明

PV GET FONT LIST コマンドは、*area*内の全フォントのID と名前を取得して、配列*fonts* および*names*に返します。

例題

PV Add font コマンドの例題参照

PV GET FORMAT LIST

PV GET FORMAT LIST (area ; formats ; strings)

引数	型		説明
area	倍長整数	⇒	4D View エリア
formats	倍長整数配列	⇐	フォーマットID の配列
strings	文字配列	⇐	フォーマット文字列の配列

説明

PV GET FORMAT LIST コマンドは、*area* に存在する全フォーマットのID とその文字列を取得して、配列 *formats* および *strings* に返します。

例題

PV Add format コマンドの例題参照

PV GET STYLE LIST

PV GET STYLE LIST (area ; stylesheets ; names)

引数	型		説明
area	倍長整数	→	4D View エリア
stylesheets	倍長整数配列	←	スタイルシートID の配列
names	文字配列	←	スタイルシート名の配列

説明

PV GET STYLE LIST コマンドは、*area*内に存在する全スタイルシートのID と名前を取得し、引数 *stylesheets* と *names* に返します。

例題

PV REMOVE STYLE, *PV SET STYLE NAME*, そして *PV SET STYLE PROPERTY* コマンドの例題参照

PV Get style property

PV Get style property (area ; style ; property) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
style	倍長整数	→	スタイルシートのID
property	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 コマンドは、*style* 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 *property* を指定するには、**PV Style properties** テーマ内の定数を使用します。

value に返される値を、**PV Style values** テーマの定数と比較できます。

例題

PV SET STYLE PROPERTY コマンドの例題を参照

PV REMOVE FONT

PV REMOVE FONT (area ; font)

引数	型		説明
area	倍長整数	→	4D View エリア
font	倍長整数	→	フォントID

説明

PV REMOVE FONT コマンドは、*area* から *font* 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 コマンドの例題参照

PV REMOVE FORMAT

PV REMOVE FORMAT (area ; format)

引数	型		説明
area	倍長整数	→	4D View エリア
format	倍長整数	→	フォーマットID

説明

PV REMOVE FORMAT コマンドは、*area* から *format* を削除します。

PV Add format コマンド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。4D Viewネイティブのフォーマットは削除できません。

例題

Refer to the example for the *PV Add format* command.

PV REMOVE STYLE (area ; stylesheet)

引数	型		説明
area	倍長整数	⇒	4D View エリア
stylesheet	倍長整数	⇒	スタイルシートID

説明

PV REMOVE STYLE コマンドは、*area* から *stylesheet* 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
```

PV SET FORMAT (area ; format ; string)

引数	型		説明
area	倍長整数	⇒	4D View エリア
format	倍長整数	⇒	フォーマットID
string	文字	⇒	フォーマットの文字列

説明

PV SET FORMAT コマンドは、*format* のフォーマットを *string* に変更します。

例題

PV Add format コマンド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
    
```

PV SET STYLE NAME

PV SET STYLE NAME (area ; stylesheet ; name)

引数	型		説明
area	倍長整数	⇒	4D View エリア
stylesheet	倍長整数	⇒	スタイルシートID
name	文字	⇒	スタイルシート名

説明

PV SET STYLE NAMEコマンドは、引数name に渡された文字列でstylesheet 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
```

エラー管理

エリア内に同じ名前のスタイルシートが既に存在する場合、4D Viewによりエラーが返されます。

PV SET STYLE PROPERTY

PV SET STYLE PROPERTY (area ; style ; property ; value)

引数	型		説明
area	倍長整数	→	4D View エリア
style	倍長整数	→	スタイルシートID
property	倍長整数	→	プロパティ番号
value	倍長整数	→	プロパティの値

説明

PV SET STYLE PROPERTY コマンドは、*style* で指定した番号のスタイルシートに対して、*property* の *value* を設定します。引数 *property* を指定するには、**PV Style properties** テーマ内の定数を、*value* を指定するには **PV Style values** テーマ内の定数を使用します。指定可能な *value* は選択した *property* に依存します:

定数	型	値	コメント
pv style automatic word wrap	倍長整数	33	セルの内容が幅を超えたときの自動ワードラップ機能を設定できます。関連する値: PV Style values テーマの定数 <ul style="list-style-type: none"> ● <u>pv value on</u>: 必要に応じてセルの内容が自動で次の行に送られる ● <u>pv value off</u>: 必要に応じてセルの内容が隣接したセルにはみ出して表示される
pv style based on	倍長整数	4	領域内の各セルは、引数value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 color back even	倍長整数	11	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color back odd	倍長整数	12	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color minus even	倍長整数	17	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color minus odd	倍長整数	18	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color text even	倍長整数	13	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color text odd	倍長整数	14	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color zero even	倍長整数	15	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color zero odd	倍長整数	16	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style format alpha	倍長整数	6	領域内の各セルは、引数value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 bool	倍長整数	8	領域内の各セルは、引数value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。
			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。 <ul style="list-style-type: none"> ● pv Short: 02/21/02 ● pv Abbreviated: Thu 21 Feb 2002 ● pv Long: Thursday 21 February 2002 ● pv Short2: 02/21/2002

pv style format date time	倍 長 整 数	9	<ul style="list-style-type: none"> • 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 21Feb 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
pv style format forced text	倍 長 整 数	32	<p>Note: 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。</p> <p>セルに対し生テキストの表示を強制します。セルの内容 (数値、日付、テキスト等) に基づき 4D Viewが適用する自動表示フォーマットは使用されません。関連する値: PV Style valuesテーマの定数</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルの内容は自動フォーマットなしで表示される • <u>pv value off</u> (デフォルト): セルの内容は自動フォーマットされて表示される
pv style format num	倍 長 整 数	7	<p>領域内の各セルは、引数value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。</p> <p>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。</p> <p>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。</p>
pv style format picture	倍 長 整 数	10	<ul style="list-style-type: none"> • 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 hidden	倍 長 整 数	1	<p>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できません。関連する値: PV Style valuesテーマ内の定数。</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルはロックされ、隠されている。 • <u>pv value off</u>: セルはロックされておらず、表示されている。
pv style hor alignment	倍 長 整 数	29	<p>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。</p> <ul style="list-style-type: none"> • <u>pv value hor alignment default</u>: セル領域にデフォルトの横揃えを適用する。 • <u>pv value hor alignment left</u>: セル領域に左横揃えを適用する。 • <u>pv value hor alignment center</u>: セル領域に中央横揃えを適用する。 • <u>pv value hor alignment right</u>: セル領域に右横揃えを適用する。
pv style locked	倍 長 整 数	0	<p>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できません。関連する値: PV Style valuesテーマ内の定数。</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルはロックされる。 • <u>pv value off</u>: セルはロックされない。
倍			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 style rotation	倍長整数	31	<ul style="list-style-type: none"> ● 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。
pv style spellcheck	倍長整数	2	<p>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。</p> <ul style="list-style-type: none"> ● 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 text bold	倍長整数	22	<p>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。</p> <ul style="list-style-type: none"> ● 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 condensed	倍長整数	27	<p>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。</p> <ul style="list-style-type: none"> ● 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	倍長整数	28	<p>セル領域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。</p> <ul style="list-style-type: none"> ● 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 text face	倍長整数	21	<p>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。</p>
pv style text font	倍長整数	19	<p>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 (<i>PV Add font</i> 関数、および <i>PV GET FONT LIST</i> コマンドを参照)。</p>
pv style text italic	倍長整数	23	<p>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。</p> <ul style="list-style-type: none"> ● 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 outline	倍長整数	25	<p>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。</p> <ul style="list-style-type: none"> ● 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	倍長整数	26	<p>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。</p> <ul style="list-style-type: none"> ● 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 size	倍長整数	20	<p>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。</p>
pv style text underline	倍長整数	24	<p>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。</p> <ul style="list-style-type: none"> ● 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 use picture height	倍長整数	3	<p>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。</p> <ul style="list-style-type: none"> ● <u>pv value on</u> : 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。 ● <u>pv value off</u> : 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 vert alignment	倍長整数	30	<p>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。</p> <ul style="list-style-type: none"> ● <u>pv value vert alignment top</u> : セル領域に上揃えを適用する。 ● <u>pv value vert alignment center</u> : セル領域に中央揃えを適用する。 ● <u>pv value vert alignment bottom</u> : セル領域に下揃えを適用する。

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**テーマの定数を引数style に渡します:

定数	型	値	コメント
pv style cells	倍長整数	- 1	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	倍長整数	- 2	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	倍長整数	- 3	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”）。

Note: これらのスタイルは4D Viewの**スタイルメニューのスタイルシート**コマンドを使用して編集できます。

例題

このメソッドでは新しいスタイルのプロパティを設定します。この例では3つのスタイルプロパティをカスタマイズします。

- "横揃え"プロパティを"左揃え"に
- "縦揃え"プロパティを"中央揃え"に
- "回転"プロパティを"90度"にします。

```

ARRAY LONGINT ($ArrayProps;3) `スタイルシートプロパティ配列
ARRAY LONGINT ($ArrayValues;3) `それぞれのプロパティの値

C_STRING (255;$StyleName) `エリアに追加するスタイルの名前

ARRAY LONGINT ($StyleNumArray;0) `スタイル番号配列
ARRAY STRING (255;$StyleNameArray;0) `スタイル名配列
C_INTEGER ($Index) `ループ用のインデックス
C_INTEGER ($Position) `番号および名前配列中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 `対応する値
$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) ` $StyleNameスタイルが存在しない場合
    $Position:=Size of array ($StyleNameArray)+1 `追加する
    INSERT IN ARRAY ($StyleNameArray;$Position) `配列をリサイズ
    INSERT IN ARRAY ($StyleNumArray;$Position)
    $StyleNameArray{$Position}:=$StyleName `新しいスタイルの名称を代入

```

```
$StyleNumArray{$Position}:=PV Add style(Area;$StyleName) `新しいスタイルの番号を代入
```

```
For($Index;1;Size of array($ArrayProps)) `設定するプロパティ毎に
```

```
  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
```

```
End for
```

```
Else
```

```
  ALERT("スタイル'" + $StyleName + "'は既にエリアに存在します。")
```

```
End if
```



PVセルの値

セルの値コマンドについて

-  PV ADD DYNAMIC ARRAYS
-  PV ADD DYNAMIC FIELDS
-  PV ARRAY TO CELLS
-  PV CELLS TO ARRAY
-  PV CLEAR DYNAMIC COLUMNS
-  PV FIELD TO CELLS
-  PV FIELDS LIST TO CELLS
-  PV Get cell boolean value
-  PV GET CELL CONTROL
-  PV GET CELL DATE TIME VALUE
-  PV Get cell date value
-  PV GET CELL FIELD
-  PV Get cell formula
-  PV Get cell num value
-  PV Get cell picture value
-  PV Get cell string value
-  PV Get cell text value
-  PV Get cell time value
-  PV Get cell value type
-  PV Get cell variable
-  PV REPORT MANY
-  PV REPORT ONE
-  PV SET CELL BOOLEAN VALUE
-  PV SET CELL CONTROL
-  PV SET CELL DATE TIME VALUE
-  PV SET CELL DATE VALUE
-  PV SET CELL FIELD
-  PV SET CELL FORMULA
-  PV SET CELL NUM VALUE
-  PV SET CELL PICTURE VALUE
-  PV SET CELL STRING VALUE
-  PV SET CELL TEXT VALUE
-  PV SET CELL TIME VALUE
-  PV SET CELL VARIABLE
-  PV UPDATE DYNAMIC AREA

セルの値コマンドについて

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、4D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 ADD DYNAMIC FIELDS (area ; master ; tables ; fields ; methods)

引数	型		説明
area	倍長整数	➡	4D View エリア
master	整数	➡	マスターテーブル番号
tables	整数配列	➡	テーブル番号の配列
fields	整数配列	➡	フィールド番号の配列
methods	文字配列	➡	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 *tables* と *fields* 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 *area* に追加します。このフィールドは *master* で定義したテーブルのカレントセレ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。*area* 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(列、および配列 *tables*、*fields*、*methods* の要素として)、常に縦方向に表示されます。

Note: *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 *master* 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レ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: 列番号

\$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: このエリアに表示される行数

Note: 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。

\$5と\$6: ユーザーが *area* 中で行(レコード)をスクロールすると、新しく表示された行のみが再描画されます。\$5と\$6引数により、どの行が対象となるか知ることができます。

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 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)。

Note:

- 4D メソッドおよびフィールドの値は4D View エリアと同じプロセス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連絡先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
```

```

`列1: 連絡先名
$TablesArray{1}:=Table(->[Contacts])
$fieldsArray{1}:=Field(->[Contacts]ContactName)
$MethodsArray{1}:=""

`列2: 連絡先名前
$TablesArray{2}:=Table(->[Contacts])
$fieldsArray{2}:=Field(->[Contacts]ContactFirstname)
$MethodsArray{2}:=""

`列3: 敬称 (リンクしたテーブル)
$TablesArray{3}:=Table(->[Titles])
$fieldsArray{3}:=Field(->[Titles]Label)
$MethodsArray{3}:=""

`列4: 注文数/トータル (計算列)
$TablesArray{4}:=0
$fieldsArray{4}:=Is_text&NBSP;&NBSP; `結果
$MethodsArray{4}:="CallMethod"

RELATE MANY ([Clients]Code)&NBSP;&NBSP; `連絡先を取得
PV ADD DYNAMIC FIELDS(Area;Table (->[Contacts]);$TablesArray;$FieldsArray;$MethodsArray)

```

CallMethod プロジェクトメソッドのコードは以下の通り:

```

C_LONGINT ($1) `4D View エリア
C_LONGINT ($2) `列番号
C_LONGINT ($3) `配列の型
C_POINTER ($4) `この配列へのポインター
C_LONGINT ($5) `ダイナミックエリアの先頭行
C_LONGINT ($6) `エリアに表示可能な行数

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

```

PV ARRAY TO CELLS (area ; direction ; column ; row ; conversion ; array)

引数	型		説明
area	倍長整数	➡	4D View エリア
direction	整数	➡	0= 行 ; 1= 列
column	倍長整数	➡	開始列番号
row	倍長整数	➡	開始行番号
conversion	整数	➡	0= テキストに変換 ; 1= 元のタイプ
array	配列	➡	配列名

説明

PV ARRAY TO CELLSコマンドは、配列arrayに指定された名前の配列の内容をareaに挿入します。この配列内容は、columnとrowで指定された座標（セル）からdirectionで定義した方向へ挿入されます。

注：このコマンドの引数direction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、direction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directionとして、右方向（0= 行）または下方向（1= 列）のいずれかのみを指定します。

5番目の引数conversion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($Index) `ループインデックス
C_LONGINT($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(次の列)
ARRAY BOOLEAN($ArrayBooleans;10) `ブール配列(3番目の列)

`初期化
For($Index;1;Size of array($ArrayString))
    $ArrayString{$Index}:=String($Index*10) ` "10", "20", "30"...
    $ArrayDates{$Index}:=Current date (*)+$Index ` 25/06/2001, 26/06/2001...
    $ArrayBooleans{$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:=1-OK

`"$ArrayString", "$ArrayDates", "$ArrayBooleans"を取得 :
PV ARRAY TO CELLS(Area;1;$Column;$Row;$Conversion;$ArrayString) `1 = 下方向へ
PV ARRAY TO CELLS(Area;1;$Column+1;$Row;$Conversion;$ArrayDates)
PV ARRAY TO CELLS(Area;1;$Column+2;$Row;$Conversion;$ArrayBooleans)
    
```

PV CELLS TO ARRAY (area ; direction ; column ; row ; array ; number)

引数	型		説明
area	倍長整数	→	4D View エリア
direction	整数	→	0= 行 ; 1= 列
column	倍長整数	→	開始列番号
row	倍長整数	→	開始行番号
array	配列	→	配列名
number	倍長整数	→	使用する列の数

説明

PV CELLS TO ARRAY コマンドは、*direction*、*column*、*row*、*number*で指定されたセル内容を*array* に代入します。

*direction*は*column*と*row*で設定したセルから開始して、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。*direction*に指定した値に応じて、引数<軸>には行番号または列番号を設定します。

注：このコマンドの引数*direction*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、*direction*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*direction*として、右方向 (0= 行) または下方向 (1= 列) 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用します。入力エリアは10個のセルで構成され、カレントセルから列上に配置されています。

```

ARRAY STRING (255;ContactsTab;0) `連絡先名配列
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
`連絡先の更新(新規、変更、削除)
    ARRAY TO SELECTION (ContactsTab; [Contacts]ContactName) `連絡先の作成
    APPLY TO SELECTION ([Contacts]; [Contacts]CodeClient:= [Clients]Code) `リレーを維持
    QUERY SELECTION ([Contacts]; [Contacts]ContactName="") `空のレコードを削除
    DELETE SELECTION ([Contacts])
End if

```

PV CLEAR DYNAMIC COLUMNS (area ; start ; number)

引数	型		説明
area	倍長整数	⇒	4D View エリア
start	倍長整数	⇒	開始列番号
number	倍長整数	⇒	列数

説明

PV CLEAR DYNAMIC COLUMNSコマンドは、*start*で指定した番号の列から始めて、ダイナミック列を*number* 列数だけ消去します。

消去されるダイナミック列は、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列は再編成されるので、ダイナミックエリアは常に列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列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 番目の列を削除しますが、名字と名前は依然として変更可能です。

```
PV CLEAR DYNAMIC COLUMNS(Area;3;1) `C列を削除
```

この命令が実行されると、4 番目の列 (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。

PV FIELD TO CELLS (area ; direction ; column ; row ; conversion ; master ; numTable ; numField)

引数	型		説明
area	倍長整数	→	4D View エリア
direction	整数	→	0= 行 ; 1= 列
column	倍長整数	→	開始セルの列番号
row	倍長整数	→	開始セルの行番号
conversion	整数	→	0= 元のタイプ ; 1= テキストに変換
master	整数	→	マスターテーブル番号
numTable	整数	→	テーブル番号
numField	整数	→	フィールド番号

説明

PV FIELD TO CELLS コマンドは、*master* テーブルのカレントセレクションに対応しているフィールドの値を *area* に挿入します。挿入は、*column* と *row* で指定された座標 (セル) から *direction* で定義した方向へ行われます。

注 : このコマンドの引数 *direction* 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、*direction* 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 *direction* として、右方向 (0= 行) または下方向 (1= 列) のいずれかのみを指定します。

5番目の引数 *conversion* 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、*master* で定義したテーブルから *table* 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 *field* は *table* 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) のセレ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リ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 エリア内で連絡先データが入力されると、連絡先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「連絡先」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、各連絡先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つの列に連絡先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

$Master:=Table(->[Contacts]) `マスターテーブル番号:連絡先(2つの列用)
RELATE MANY([Clients]code) `セレクションから対応する連絡先を取得
PV GET CURRENT CELL(Area;$Column;$Row)

`4D View エリア上の名前を更新
$Table:=Table(->[Contacts]) `連絡先テーブルのテーブル番号
$Field:=Field(->[Contacts]ContactName) `内容を取得するフィールド番号
`変換は不要、文字タイプの回復を行う
PV FIELD TO CELLS(Area;1;$Column;$Row;0;$Master;$Table;$Field)

```

`4D View エリアの肩書きを更新

\$Table:=Table (->[Titles]) `肩書きタイプ(リレート先)テーブルの番号

\$Field:=Field (->[Titles]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**)

PV FIELDS LIST TO CELLS

PV FIELDS LIST TO CELLS (area ; direction ; columns ; rows ; conversions ; master ; tables ; fields)

引数	型		説明
area	倍長整数	→	4D View エリア
direction	整数	→	0= 行 ; 1= 列
columns	倍長整数配列	→	開始セルの列番号の配列
rows	倍長整数配列	→	開始セルの行番号の配列
conversions	倍長整数配列	→	0= 元のタイプ ; 1= テキストに変換
master	倍長整数	→	マスターテーブル番号
tables	倍長整数配列	→	テーブル番号
fields	倍長整数配列	→	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 *direction* 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、*direction* 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 *direction* として、右方向 (0= 行) または下方向 (1= 列) のいずれかのみを指定します。

引数 *columns* および *rows* には、開始セルの列番号と行番号をそれぞれ指定します。

5番目の引数 *conversion* 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 *tables* と *fields* 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、*master* 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 *tables* と *fields* 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リ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「連絡先」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、各連絡先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つ目の列を表示します (列Cより開始するため、列Eに表示する)。

```
C_INTEGER($Master) `マスターテーブル番号
C_INTEGER($Index) `ループインデックス

$Master:=Table(->[Contacts]) `マスターテーブル番号:連絡先(全列用)

ARRAY LONGINT($ArrayColumns;3) `開始セルの列番号
ARRAY LONGINT($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 `列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]) `連絡先テーブルの番号
$FieldsArray{1}:=Field(->[Contacts]ContactName) `名字フィールドの番号

$TablesArray{2}:=Table(->[Contacts]) `連絡先テーブルの番号
$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
$FieldsArray{3}:=Field(->[Titles]Label) `肩書きフィールドの番号(社長、秘書、開発者)

`対応する「連絡先」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)
```

PV Get cell boolean value

PV Get cell boolean value (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	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*は、*column*と*row*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*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

PV GET CELL CONTROL (area ; column ; row ; type ; varName ; method ; title)

引数	型		説明
area	倍長整数	➡	4D View エリア
column	倍長整数	➡	セルの列番号
row	倍長整数	➡	セルの行番号
type	整数	←	制御タイプ
varName	文字	←	制御管理変数名
method	文字	←	コールバックメソッド名
title	文字	←	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 *column* と *row* 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。

type 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。

varName 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。

method 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。

title 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
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:="No control"

: ($CtrlType=pv_control_push_button)
  $TypeName:="button"

: ($CtrlType=pv_control_radio_button)
  $TypeName:="radio button"

: ($CtrlType=pv_control_check_box)
  $TypeName:="check box"

: ($CtrlType=pv_control_drop_down)
  $TypeName:="drop-down list"

: ($CtrlType=pv_control_combo_box)
  $TypeName:="combo box"
End case

PV SET CELL TEXT VALUE (Area; $Column; $Row+1; "Type: "+$TypeName)
PV SET CELL TEXT VALUE (Area; $Column; $Row+2; "Name: "+$CtrlName)

```

```
PV SET CELL TEXT VALUE(Area;$Column;$Row+3;"Title: "+$Title)
PV SET CELL TEXT VALUE(Area;$Column;$Row+4;"Method: "+$CallbackMethod)
```

PV GET CELL DATE TIME VALUE

PV GET CELL DATE TIME VALUE (area ; column ; row ; dateValue ; timeValue)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	セルの列番号
row	倍長整数	⇒	セルの行番号
dateValue	日付	⇐	日付タイプのセルの値
timeValue	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、*column* と *row* で指定されたセル中の値 *date* と *time* を返します。

PV GET CELL DATE TIME VALUE が日付/時間タイプでないセルに使用された場合、00/00/00と00:00:00が返されます。

例題

PV SET CELL DATE TIME VALUE の例題参照

PV Get cell date value

PV Get cell date value (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	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*は、*column*と*row*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*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*の例題参照

PV GET CELL FIELD (area ; column ; row ; numTable ; numField)

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
numTable	整数	←	テーブル番号
numField	整数	←	フィールド番号

説明

PV GET CELL FIELD コマンドは、*column* と *row* 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 *table* と *field* に返します。

例題 1

PV SET CELL FIELD 例題参照

例題 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、列ヘッダのクリック時にダイナミック列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、ダイナミック列だけです。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、列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 `列内の関連付けられた値は自動的に並べ替えられる
```

⚙️ PV Get cell formula

PV Get cell formula (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	➡	4D View エリア
column	倍長整数	➡	セルの列番号
row	倍長整数	➡	セルの行番号
戻り値	文字	↻	フォーミュラ

説明

*PV Get cell formula*は、*column*と*row*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
```

PV Get cell num value

PV Get cell num value (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
戻り値	実数	↩	セルの値

説明

*PV Get cell num value*は、*column*と*row*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*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*コマンドの例題参照

PV Get cell picture value

PV Get cell picture value (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
戻り値	ピクチャー	↩	セルの値

説明

*PV Get cell picture value*は、*column*と*row*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。

*PV Get cell picture value*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*コマンドの例題参照

PV Get cell string value

PV Get cell string value (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
戻り値	文字	↩	セルの値

説明

*PV Get cell string value*は、*column*と*row*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*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*コマンドの例題参照

PV Get cell text value

PV Get cell text value (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	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*は、*column*と*row*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*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*コマンドの例題参照

PV Get cell time value

PV Get cell time value (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	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*は、*column*と*row*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*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*コマンドの例題参照

PV Get cell value type

PV Get cell value type (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	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*は、*column*と*row*で指定された*area*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

PV Get cell variable

PV Get cell variable (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
戻り値	文字	↩	関連付けられた変数の名前

説明

*PV Get cell variable*は、*column*と*row*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](#)コマンドの例題参照

PV REPORT MANY (area ; column ; row ; master ; tableBreak ; fieldBreak ; operator ; tables ; fields ; insert ; detail ; title)

引数	型	説明
area	倍長整数	→ 4D View エリア
column	倍長整数	→ 列番号
row	倍長整数	→ 行番号
master	整数	→ プライマリテーブル番号
tableBreak	整数	→ ブレークが発生するテーブル番号
fieldBreak	配列	→ ブレークが発生するフィールド番号
operator	整数配列	→ 実行する処理
tables	整数配列	→ 表示フィールドの属するテーブルの番号
fields	整数配列	→ 表示フィールドの番号
insert	整数	→ 0= 置き換え ; 1= 行挿入
detail	整数	→ 明細行挿入オプション
title	文字	→ ブレークタイトル

説明

PV REPORT MANYコマンドは、*master* テーブルのカレントセレクションから開始して、*tables* と *fields* で指定した4D フィールドの値を使用し、4D View *area* 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、*operator*で定義します（合計、カウント、最大値等）。引数*operator*を定義するには、**PV Report functions**テーマ内の定数を使用します：

定数	型	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

生成されたレポートは、*column* と *row*で指定したセルより*area* 上に挿入されます。

tableBreak と *fieldBreak* は、ブレークシーケンスを実行するフィールドを指定します。ブレークにより、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。

ブレークとは、テーブルの並び替えにおける値の変わり目です。このブレークレベルは、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、*tableBreak* と *fieldBreak* 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、ブレー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。ブレークをとまなうクイックレポートの作成に関する詳細は、4D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*insert* 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*detail* を使用すると、4D View が明細行およびブレー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：

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

引数*title*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、現在のブレー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」フィールドがブレー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。

Note: *detail*に0 を指定し、<タイトル> に空の文字列を渡した場合、最初の列は挿入されません（列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「連絡先」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）、連絡先数を示すブレイクを追加します。「連絡先」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($Column) `列番号
C_LONGINT($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) `表示フィールド数
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 `列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「連絡先」テーブルの「レポート」である
$TableBreak:=Table(->[Titles])
$FieldBreak:=Field(->[Titles]Label) `「連絡先」肩書き上でブレイクが発生

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

$Operator{2}:=pv_report_function_none `列2における計算は行わない
$Tables{2}:=Table(->[Contacts])
$Fields{2}:=Field(->[Contacts]ContactName) `2 列目に連絡先名

$Operator{3}:=pv_report_function_none `列3における計算は行わない
$Tables{3}:=Table(->[Contacts])
$Fields{3}:=Field(->[Contacts]ContactFirstname) `3 列目に連絡先の名前

$Insert:=1 `挿入
$Detail:=2 `明細行を挿入し、値は1度だけ表示する
$Title:="Number of contacts for #" ` `#`記号は現在のブレイク値で置き変わる

RELATE MANY([Clients]Code) `顧客の連絡先を取得
`ブレイクポイントでのソートが必要+ 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)
```

カレントセクションおよびレコード

セクションはデータベースストラクチャーレベルでのリレーションやコマンドに渡されるフィールド番号や配列に基づきます。

PV REPORT ONE (area ; column ; row ; master ; tableBreak ; fieldBreak ; operator ; tables ; fields ; insert ; detail ; title)

引数	型	説明
area	倍長整数	→ 4D View エリア
column	倍長整数	→ 列番号
row	倍長整数	→ 行番号
master	整数	→ プライマリテーブル番号
tableBreak	整数	→ ブレークが発生するテーブル番号
fieldBreak	整数	→ ブレークが発生するフィールド番号
operator	整数	→ 実行する処理
tables	整数	→ 表示フィールドの属するテーブルの番号
fields	整数	→ 表示フィールドの番号
insert	整数	→ 0= 置き換え ; 1= 行挿入
detail	倍長整数	→ 明細行挿入オプション
title	文字	→ ブレークタイトル

説明

PV REPORT ONEコマンドは、master テーブルのカレントセレクションから開始して、tables とfields で指定した4Dフィールドの値を使用し、4D View area 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、operatorで定義します（合計、カウント、最大値等）。引数operatorを定義するには、PV Report functionsテーマ内の定数を使用します：

定数	型	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

生成されたレポートは、column とrowで指定したセルよりarea 上に挿入されます。

tableBreakとfieldBreakは、ブレークシーケンスを実行するフィールドを指定します。ブレークにより、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。

ブレークとは、テーブルの並び替えにおける値の変わり目です。このブレークレベルは、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、tableBreak とfieldBreak 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、ブレー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。ブレークをとまなうクイックレポートの作成に関する詳細は、4D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insert 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detail を使用すると、4D View が明細行およびブレー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。

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

引数title 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、現在のブレー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」フィールドがブレー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。

Note : detail に0 を指定し、<タイトル> に空の文字列を渡した場合、最初の列は挿入されません（列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($Column) `列番号
C_LONGINT($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 `列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:="Total"

ALL RECORDS([Clients])
ORDER BY([Clients];[Clients]Type;[Clients]Name) `ブレイクポイントでのソートが必要+ アルファベット順に表示

PV REPORT
ONE(Area;$Column;$Row;$Master;$TableBreak;$FieldBreak;$Operator;$Table;$Field;$Insert;$Detail;$Title)
```

🔧 PV SET CELL BOOLEAN VALUE

PV SET CELL BOOLEAN VALUE (area ; column ; row ; value)

引数	型		説明
area	倍長整数	➡	4D View エリア
column	倍長整数	➡	セルの列番号
row	倍長整数	➡	セルの行番号
value	整数	➡	セルの値

説明

PV SET CELL BOOLEAN VALUE コマンドは、*column* と *row* で指定されたセルに整数タイプの *value* 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) &nbsp; &nbsp; &nbsp; `ループのインデックス
C_BOOLEAN($Value) `コピーする値

$CopyNumber:=Num(Request("How many copies toward the bottom would you like?";"5")) `デフォルトで5

If($CopyNumber>0)
  PV GET CURRENT CELL(Area;$Column;$Row) `コピーするセル
  $Value:=(PV Get cell boolean value(Area;$Column;$Row)=1) `整数をブールに変換
  For($Index;$Row+1;$Row+$CopyNumber) &NBSP;&NBSP; ` $CopyNumberループ
    PV SET CELL BOOLEAN VALUE(Area;$Column;$Index;Num($Value)) `値をコピー
  End for
End if
```


PV SET CELL DATE TIME VALUE

PV SET CELL DATE TIME VALUE (area ; column ; row ; date ; time)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	セルの列番号
row	倍長整数	⇒	セルの行番号
date	日付	⇒	日付タイプのセルの値
time	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、*date* と *time* を1 つの値として *column* と *row* 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

$Date:=Date(Request("Date of meeting";String(Current date)))
If($Date#!00/00/00!) `有効な日付
    $Time:=Time(Request("Time of meeting";Time string(Current time)))
    If($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) `情報を読み込む
        ALERT("The meeting has been pushed ahead to "+String($Date)+" at "+Time string($Time))
    End if
End if
```

PV SET CELL DATE VALUE

PV SET CELL DATE VALUE (area ; column ; row ; value)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	セルの列番号
row	倍長整数	⇒	セルの行番号
value	日付	⇒	セルの値

説明

PV SET CELL DATE VALUE コマンドは、*column* と *row* で指定されたセルに日付タイプの *value* を割り当てます。

例題

PV SET CELL STRING VALUE コマンドの例題参照

PV SET CELL FIELD (area ; column ; row ; numTable ; numField)

引数	型		説明
area	倍長整数	➡	4D View エリア
column	倍長整数	➡	セルの列番号
row	倍長整数	➡	セルの行番号
numTable	整数	➡	テーブル番号
numField	整数	➡	フィールド番号

説明

PV SET CELL FIELD コマンドは、*column* と *row* 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。

table 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 *field* の値が表示されます。

セルは *field* 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 : [Clients]LastName
  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 : [Clients]FirstName
  End if

`ラベル
  PV SET CELL STRING VALUE(Area;1;1;"Last Name :")
  PV SET CELL STRING VALUE(Area;1;2;"First Name :")

  CREATE RECORD([Clients]) `新しい顧客が入力された
End if

```

🔧 PV SET CELL FORMULA

PV SET CELL FORMULA (area ; column ; row ; formula)

引数	型		説明
area	倍長整数	➡	4D View エリア
column	倍長整数	➡	セルの列番号
row	倍長整数	➡	セルの行番号
formula	文字	➡	フォーミュラ

説明

PV SET CELL FORMULAコマンドは、columnとrowで指定されたセルにformula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) `ループインデックス
C_TEXT($Name) `開始セル名

PV GET CURRENT CELL(Area;$Column;$Row)
If (PV Get cell value type(Area;$Column;$Row)=pv number type value) `タイプの確認
`デフォルトで5
  $CopyNumber:=Num(Request("How many cells to the bottom do you want to increment?";"5"))
  If ($CopyNumber>0) `確定
    $Name:=PV Get cell name(Area;$Column;$Row) `セル名の取得
    If ($Name="" &NBSP;&NBSP; `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) ` $CopyNumber ループ
`増分
      PV SET CELL FORMULA(Area;$Column;$Index;"="+$Name+"+"+String($Index-$Row))
    End for
  End if
Else `正しくないタイプ
  ALERT("The start cell must be a numeric type")
End if
```

PV SET CELL NUM VALUE

PV SET CELL NUM VALUE (area ; column ; row ; value)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	セルの列番号
row	倍長整数	⇒	セルの行番号
value	実数	⇒	セルの値

説明

PV SET CELL NUM VALUE コマンドは、*column* と *row* で指定されたセルに数値タイプの *value* を割り当てます。

例題

PV SET CELL STRING VALUE コマンドの例題参照

PV SET CELL PICTURE VALUE

PV SET CELL PICTURE VALUE (area ; column ; row ; value)

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
value	ピクチャー	→	セルの値

説明

PV SET CELL PICTURE VALUE コマンドは、*column*と*row*で指定されたセルにピクチャタイプの*value*を割り当てます。

例題

PV SET CELL STRING VALUE コマンドの例題参照

🔧 PV SET CELL STRING VALUE

PV SET CELL STRING VALUE (area ; column ; row ; value)

引数	型		説明
area	倍長整数	➡	4D View エリア
column	倍長整数	➡	セルの列番号
row	倍長整数	➡	セルの行番号
value	文字	➡	セルの値

説明

PV SET CELL STRING VALUE コマンドは、*column*と*row*で指定されたセルに文字列タイプの*value*を書き込みます。

*value*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) `ループインデックス
C_STRING(80;$Value) `コピーする値

$CopyNumber:=Num(Request("How many times should it copy towards the bottom?";"5")) `デフォルトで5

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) ` $CopyNumberループ
    PV SET CELL STRING VALUE(Area;$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型を調べる汎用メソッドを呼び出し、この後で型とは関係なく“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。これは汎用なプログラム作成の最適な練習になります。

PV SET CELL TEXT VALUE

PV SET CELL TEXT VALUE (area ; column ; line ; value)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	セルの列番号
line	倍長整数	⇒	セルの行番号
value	テキスト	⇒	セルの値

説明

PV SET CELL TEXT VALUE コマンドは、*column* と *row* で指定されたセルにテキストタイプの *value* を書き込みます。

例題

PV SET CELL STRING VALUE コマンドの例題参照

PV SET CELL TIME VALUE

PV SET CELL TIME VALUE (area ; column ; row ; value)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	セルの列番号
row	倍長整数	⇒	セルの行番号
value	時間	⇒	セルの値

説明

PV SET CELL TIME VALUE コマンドは、*column*と*row*で指定されたセルに時間タイプの*value*を割り当てます。

例題

PV SET CELL STRING VALUE コマンドの例題参照

PV SET CELL VARIABLE

PV SET CELL VARIABLE (area ; column ; row ; variable)

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
variable	文字	→	変数名

説明

PV SET CELL VARIABLE コマンドは、*column*と*row*で指定されたセルを*variable*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
```

PV UPDATE DYNAMIC AREA

PV UPDATE DYNAMIC AREA (area)

引数	型	説明
area	倍長整数	4D View エリア

説明

PV UPDATE DYNAMIC AREA コマンドは 4D View *area* に表示されているデータと 4D データベースのデータとの同期を実行します。

このコマンドはダイナミックエリア内でのみ使用されるコールバックメソッドのコンテキストで有効です。実際、コールバックメソッドが 4D のカレントセレクションを変更しても、4D View エリアはメソッド全体が実行された後に更新されます。しかし *PV GOTO CELL* のようなコマンドがコールバックメソッド内で使用されるようなケースでは、4D データと 4D View データとの更新ラグのため、結果が異なってしまいます。このリスクを避けるため、**PV UPDATE DYNAMIC AREA** コマンドをコールバックメソッド内で呼び出して、4D に対して行われた更新を 4D View に反映させる必要があります。

PV UPDATE DYNAMIC AREA は以下の変更の少なくとも 1 つが 4D のセレクションに対して行われた場合に行為 *s* を行います:

- リンクされたフィールドのコンテキスト内でカレントセレクションが変更された
- リンクされた配列サイズが変更された
- リンクされたフィールドまたは配列のデータが変更された

例題

この例題では **PV ADD DYNAMIC FIELDS** コマンドを使用して 4D セレクションと 4D View エリアの同期を有効にする例を示します。"右クリック" イベントメソッドをインストールし、4D セレクションにレコードを作成します。そして 4D View エリアでこのレコードを指示します。

```
PV ON EVENT(area;pv_on_right_clicked;"CallbackMethod")
```

CallbackMethod プロジェクトメソッドは以下の通りです:

```
C_BOOLEAN($0)
C_LONGINT($1;$2;$3;$4;$5;$6)

$0:=True `Blocks event
CREATE RECORD([Contacts]) `新規レコード
SAVE RECORD([Contacts]) `作成したレコードを保存
ALL RECORDS([Contacts])

`4D Viewのセレクションを4Dのセレクションと同期
PV UPDATE DYNAMIC AREA(area)

`新しいレコードを選択
PV GOTO CELL(area;1;Records in selection([Contacts]))
```

PVセルプロパティ

セルプロパティコマンドについて

-  PV Get cell name
-  PV GET CELL NAME LIST
-  PV Get cell property
-  PV Get range property
-  PV SET CELL NAME
-  PV SET CELL PROPERTY
-  PV SET RANGE PROPERTY

セルプロパティコマンドについて

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前を一覧表示したり、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、列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 Get cell name

PV Get cell name (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
戻り値	文字	↩	セルの名前

説明

*PV Get cell name*は、セルの名前を返します。

例題

*PV SET CELL NAME*コマンドの例題参照

PV GET CELL NAME LIST (area ; columns ; rows ; names)

引数	型		説明
area	倍長整数	→	4D View エリア
columns	倍長整数配列	←	セルの列番号の配列
rows	倍長整数配列	←	セルの行番号の配列
names	テキスト配列	←	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、列番号、行番号を取得し、それぞれ *names*、*columns*、*rows* 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 ($TabColumns;0) `列配列
ARRAY LONGINT ($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
    PV GOTO CELL(Area;$TabColumns{$Position};$TabRows{$Position}) `そのセルをカレントセルに
Else
    `エラーを管理する
End if

```

PV Get cell property

PV Get cell property (area ; column ; row ; property) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
property	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*は、引数`column`と`row`で定義されたセルの`property`の値を返します。

引数`property`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 CELL PROPERTY*コマンドを参照してください。

例題

*PV SET CELL PROPERTY*コマンドの例題参照

🔧 PV Get range property

PV Get range property (area ; left ; top ; right ; bottom ; property) -> 戻り値

引数	型		説明
area	倍長整数	➡	4D View エリア
left	倍長整数	➡	左側セルの列番号
top	倍長整数	➡	上側セルの行番号
right	倍長整数	➡	右側セルの列番号
bottom	倍長整数	➡	下側セルの行番号
property	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*は、引数`left`, `top`, `right`そして`bottom`を使用して定義された選択セル領域の`property`の値を返します。セル領域に関する詳細は[セクションコマンドについて](#)を参照してください。

引数`property`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*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	複合値の場合に返される値/定数
<code>pv_style_format_alpha</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_format_num</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_format_bool</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_format_date_time</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_format_picture</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_color_back_even</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_color_back_odd</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_color_text_even</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_color_text_odd</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_color_zero_even</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_color_zero_odd</code>	65535 (<code>pv_value_format_mixed</code>)
<code>pv_style_hor_alignment</code>	255 (<code>pv_value_hor_alignment_mixed</code>)
<code>pv_style_vert_alignment</code>	255 (<code>pv_value_vert_alignment_mixed</code>)
<code>pv_style_rotation</code>	255 (<code>pv_value_rotation_mixed</code>)

例題

*PV Get cell property*コマンドの例題参照

PV SET CELL NAME

PV SET CELL NAME (area ; column ; row ; name ; mode)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	セルの列番号
row	倍長整数	⇒	セルの行番号
name	文字	⇒	セルの名前
mode	倍長整数	⇒	0または省略 = 名前を追加, 1 = 名前を置き換え

説明

PV SET CELL NAMEコマンドは、引数column とrow で定義されたセルの名前をname に設定します。

セルには絶対参照 (列番号と行番号)、または名前でアクセスできます。4D Viewエリアのフォーミュラはnameを使用してセルを参照することができます。

Notes:

- セルの先頭3文字は数字であってはなりません。
- セル名にスペースを含めることはできません。name引数にスペースが含まれる場合、コマンドはそれを取り除きます。

オプションのmode引数を使用して、セルに既に名前が付けられていたときの新しい名称の振る舞いを設定できます。PV Cell propertiesテーマの以下の定数を使用できます:

定数	型	値	コメント
pv add name	倍長整数	0	セルに既につけられた名称に新しい名前が追加される。
pv replace name	倍長整数	1	セルに付けられている名称が新しい名前でも置き換えられる。

セルに割り当てられたすべての名称を削除するには、nameに空の文字列 ("") を渡し、modeにpv replace nameを指定します。

例題

この例題ではカレントセルの名称を変更します。既に名前がつけられていた場合、置き換えることができます:

```
C_TEXT ($CellName) `セルに割り当てる名前
C_TEXT ($CellNameOld) `セルにすでに割り当てられている名前
C_LONGINT ($Column) `カレントセルの列番号
C_LONGINT ($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
$CellNameOld:=PV Get cell name(Area;$Column;$Row) `既に名前が設定されている可能性がある
$CellName:=Request("このセルに与える名称";"New name")

If ($CellName# "") `名前が入力された
  If ($CellNameOld="") `セルにはまだ名前が無い
    PV SET CELL NAME(Area;$Column;$Row;$CellName) `入力された名前を割り当てる
  End if
Else `カレントセルには既に名前が付けられている
  CONFIRM("This cell is already named "+$CellNameOld+". Do you want to rename it?";"Yes";"No")
  If (OK=1) `The user wants to rename the cell
    PV SET CELL NAME(Area;$Column;$Row;$CellName;pv replace name)
  End if
End if
```

PV SET CELL PROPERTY

PV SET CELL PROPERTY (area ; column ; row ; property ; value)

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	倍長整数	→	セルの行番号
property	倍長整数	→	プロパティ番号
value	倍長整数	→	プロパティの値

説明

PV SET CELL PROPERTY コマンドは、引数 *column* と *row* で定義されたセルの *property* の値を *value* に設定します。
property 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 *value* に渡します。
- **PV Style properties** テーマ内の定数。この場合、引数 *value* 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 *property* に使用する各種定数、および関連する *values* を以下に説明します：

PV Cell properties テーマ

定数	型	値	コメント
pv cell height	倍長整数	101	セルの高さの設定を許可。関連する値: ピクセル単位の高さ
pv cell width	倍長整数	100	セルの幅の設定を許可。関連する値: ピクセル単位の幅

PV Style properties テーマ

定数	型	値	コメント
pv style automatic word wrap	倍長整数	33	セルの内容が幅を超えたときの自動ワードラップ機能を設定できます。関連する値: PV Style values テーマの定数 <ul style="list-style-type: none"> • <u>pv value on</u>: 必要に応じてセルの内容が自動で次の行に送られる • <u>pv value off</u>: 必要に応じてセルの内容が隣接したセルにはみ出して表示される
pv style based on	倍長整数	4	領域内の各セルは、引数value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 color back even	倍長整数	11	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color back odd	倍長整数	12	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color minus even	倍長整数	17	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color minus odd	倍長整数	18	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color text even	倍長整数	13	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color text odd	倍長整数	14	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color zero even	倍長整数	15	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color zero odd	倍長整数	16	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style format alpha	倍長整数	6	領域内の各セルは、引数value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 bool	倍長整数	8	領域内の各セルは、引数value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。
			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。 <ul style="list-style-type: none"> • pv Short: 02/21/02 • pv Abbreviated: Thu 21 Feb 2002 • pv Long: Thursday 21 February 2002 • pv Short2: 02/21/2002

pv style format date time	倍 長 整 数	9	<ul style="list-style-type: none"> • 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 21Feb 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 <p>Note: 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。</p> <p>セルに対し生テキストの表示を強制します。セルの内容 (数値、日付、テキスト等) に基づき 4D Viewが適用する自動表示フォーマットは使用されません。関連する値: PV Style valuesテーマの定数</p>
pv style format forced text	倍 長 整 数	32	<ul style="list-style-type: none"> • <u>pv value on</u>: セルの内容は自動フォーマットなしで表示される • <u>pv value off</u> (デフォルト): セルの内容は自動フォーマットされて表示される
pv style format num	倍 長 整 数	7	<p>領域内の各セルは、引数value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。</p> <p>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。</p> <p>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。</p>
pv style format picture	倍 長 整 数	10	<ul style="list-style-type: none"> • 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 hidden	倍 長 整 数	1	<p>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できません。関連する値: PV Style valuesテーマ内の定数。</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルはロックされ、隠されている。 • <u>pv value off</u>: セルはロックされておらず、表示されている。
pv style hor alignment	倍 長 整 数	29	<p>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。</p> <ul style="list-style-type: none"> • <u>pv value hor alignment default</u>: セル領域にデフォルトの横揃えを適用する。 • <u>pv value hor alignment left</u>: セル領域に左横揃えを適用する。 • <u>pv value hor alignment center</u>: セル領域に中央横揃えを適用する。 • <u>pv value hor alignment right</u>: セル領域に右横揃えを適用する。
pv style locked	倍 長 整 数	0	<p>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できません。関連する値: PV Style valuesテーマ内の定数。</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルはロックされる。 • <u>pv value off</u>: セルはロックされない。 <p>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。</p>
	倍		

pv style rotation	倍長整数	31	<ul style="list-style-type: none"> ● 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。
pv style spellcheck	倍長整数	2	<p>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。</p> <ul style="list-style-type: none"> ● 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 text bold	倍長整数	22	<p>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。</p> <ul style="list-style-type: none"> ● 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 condensed	倍長整数	27	<p>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。</p> <ul style="list-style-type: none"> ● 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	倍長整数	28	<p>セル領域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。</p> <ul style="list-style-type: none"> ● 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 text face	倍長整数	21	<p>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。</p>
pv style text font	倍長整数	19	<p>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（<i>PV Add font</i> 関数、および<i>PV GET FONT LIST</i>コマンドを参照）。</p>
pv style text italic	倍長整数	23	<p>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。</p> <ul style="list-style-type: none"> ● 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 outline	倍長整数	25	<p>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。</p> <ul style="list-style-type: none"> ● 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	倍長整数	26	<p>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。</p> <ul style="list-style-type: none"> ● 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 size	倍長整数	20	<p>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。</p>
pv style text underline	倍長整数	24	<p>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。</p> <ul style="list-style-type: none"> ● 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。

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 style use picture height	倍 長 整 数	3	<ul style="list-style-type: none"> ● <u>pv value on</u> : 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。 ● <u>pv value off</u> : 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。
-----------------------------	------------------	---	---

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 style vert alignment	倍 長 整 数	30	<ul style="list-style-type: none"> ● <u>pv value vert alignment top</u> : セル領域に上揃えを適用する。 ● <u>pv value vert alignment center</u> : セル領域に中央揃えを適用する。 ● <u>pv value vert alignment bottom</u> : セル領域に下揃えを適用する。
-------------------------	------------------	----	--

Note: セル範囲のプロパティを指定する場合は **PV SET RANGE PROPERTY** コマンドを使用できます。

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$20"を含むセル
`選択されたセルの座標
PV GET SELECTED RANGES LIST (Area;$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?

```

PV SET RANGE PROPERTY

PV SET RANGE PROPERTY (area ; left ; top ; right ; bottom ; property ; value)

引数	型		説明
area	倍長整数	⇒	4D View エリア
left	倍長整数	⇒	左側セルの列番号
top	倍長整数	⇒	上側セルの行番号
right	倍長整数	⇒	右側セルの列番号
bottom	倍長整数	⇒	下側セルの行番号
property	倍長整数	⇒	プロパティの番号
value	倍長整数	⇒	プロパティの値

説明

`PV SET RANGE PROPERTY` コマンドは、引数 `left`, `top`, `right` そして `bottom` で定義されたセルの `property` の値を `value` に設定します。セル範囲については [セレクトコマンドについて](#) を参照してください。

`property` 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 `value` に渡します。
- **PV Style properties** テーマ内の定数。この場合、引数 `value` 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 `property` に使用する各種定数、および関連する `values` を以下に説明します：

PV Cell properties テーマ

定数	型	値	コメント
pv cell height	倍長整数	101	セルの高さの設定を許可。関連する値: ピクセル単位の高さ
pv cell width	倍長整数	100	セルの幅の設定を許可。関連する値: ピクセル単位の幅

PV Style properties テーマ

定数	型	値	コメント
pv style automatic word wrap	倍長整数	33	セルの内容が幅を超えたときの自動ワードラップ機能を設定できます。関連する値: PV Style values テーマの定数 <ul style="list-style-type: none"> ● <u>pv value on</u>: 必要に応じてセルの内容が自動で次の行に送られる ● <u>pv value off</u>: 必要に応じてセルの内容が隣接したセルにはみ出して表示される
pv style based on	倍長整数	4	領域内の各セルは、引数value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 color back even	倍長整数	11	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color back odd	倍長整数	12	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color minus even	倍長整数	17	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color minus odd	倍長整数	18	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color text even	倍長整数	13	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color text odd	倍長整数	14	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color zero even	倍長整数	15	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color zero odd	倍長整数	16	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style format alpha	倍長整数	6	領域内の各セルは、引数value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 bool	倍長整数	8	領域内の各セルは、引数value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。
			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。 <ul style="list-style-type: none"> ● pv Short: 02/21/02 ● pv Abbreviated: Thu 21 Feb 2002 ● pv Long: Thursday 21 February 2002 ● pv Short2: 02/21/2002

pv style format date time	倍 長 整 数	9	<ul style="list-style-type: none"> • 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 21Feb 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 <p>Note: 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。</p> <p>セルに対し生テキストの表示を強制します。セルの内容 (数値、日付、テキスト等) に基づき 4D Viewが適用する自動表示フォーマットは使用されません。関連する値: PV Style valuesテーマの定数</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルの内容は自動フォーマットなしで表示される • <u>pv value off</u> (デフォルト): セルの内容は自動フォーマットされて表示される
pv style format forced text	倍 長 整 数	32	<p>領域内の各セルは、引数value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。</p> <p>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。</p> <p>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。</p>
pv style format num	倍 長 整 数	7	<ul style="list-style-type: none"> • 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 format picture	倍 長 整 数	10	<p>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できません。関連する値: PV Style valuesテーマ内の定数。</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルはロックされ、隠されている。 • <u>pv value off</u>: セルはロックされておらず、表示されている。
pv style hidden	倍 長 整 数	1	<p>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。</p> <ul style="list-style-type: none"> • <u>pv value hor alignment default</u>: セル領域にデフォルトの横揃えを適用する。 • <u>pv value hor alignment left</u>: セル領域に左横揃えを適用する。 • <u>pv value hor alignment center</u>: セル領域に中央横揃えを適用する。 • <u>pv value hor alignment right</u>: セル領域に右横揃えを適用する。
pv style hor alignment	倍 長 整 数	29	<p>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できません。関連する値: PV Style valuesテーマ内の定数。</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルはロックされる。 • <u>pv value off</u>: セルはロックされない。
pv style locked	倍 長 整 数	0	<p>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。</p>
	倍		

pv style rotation	倍長整数	31	<ul style="list-style-type: none"> ● 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。
pv style spellcheck	倍長整数	2	<p>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。</p> <ul style="list-style-type: none"> ● 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 text bold	倍長整数	22	<p>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。</p> <ul style="list-style-type: none"> ● 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 condensed	倍長整数	27	<p>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。</p> <ul style="list-style-type: none"> ● 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	倍長整数	28	<p>セル領域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。</p> <ul style="list-style-type: none"> ● 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 text face	倍長整数	21	<p>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。</p>
pv style text font	倍長整数	19	<p>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（<i>PV Add font</i> 関数、および<i>PV GET FONT LIST</i>コマンドを参照）。</p>
pv style text italic	倍長整数	23	<p>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。</p> <ul style="list-style-type: none"> ● 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 outline	倍長整数	25	<p>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。</p> <ul style="list-style-type: none"> ● 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	倍長整数	26	<p>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。</p> <ul style="list-style-type: none"> ● 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 size	倍長整数	20	<p>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。</p>
pv style text underline	倍長整数	24	<p>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。</p> <ul style="list-style-type: none"> ● 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 use picture height	倍 長 整 数	3	<p>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。</p> <ul style="list-style-type: none"> ● <u>pv value on</u> : 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。 ● <u>pv value off</u> : 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 vert alignment	倍 長 整 数	30	<p>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。</p> <ul style="list-style-type: none"> ● <u>pv value vert alignment top</u> : セル領域に上揃えを適用する。 ● <u>pv value vert alignment center</u> : セル領域に中央揃えを適用する。 ● <u>pv value vert alignment bottom</u> : セル領域に下揃えを適用する。

Note: 一つのセルのプロパティを指定するには**PV SET CELL PROPERTY** コマンドを使用します。

例題

セルD20を参照するフォーミュラを含むすべてのセルを太字にします。

```

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"を参照するフォーミュラ

`選択されたセルの座標
PV GET SELECTED RANGES LIST(Area;$LeftArray;$TopArray;$RightArray;$BottomArray)

`太字にする
For ($Index;1;Size of array ($LeftArray)) `すべての範囲
  PV SET RANGE
  PROPERTY (Area;$LeftArray{$Index};$TopArray{$Index};$RightArray{$Index};$BottomArray{$Index};pv style text bold;pv value on)
End for

```

PVセル操作

セル操作コマンドについて

-  PV Copy to blob
-  PV FIND ALL
-  PV FIND ONE
-  PV PASTE FROM BLOB
-  PV REPLACE ALL
-  PV REPLACE ONE
-  PV SORT COLUMN
-  PV SORT MANY
-  PV SORT ONE
-  PV SPECIAL CLEAR
-  PV SPECIAL CUT
-  PV SPECIAL PASTE

セル操作コマンドについて

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 Copy to blob

PV Copy to blob (area) -> 戻り値

引数	型		説明
area	倍長整数	→	4D Viewエリア
戻り値	BLOB	↩	セレクションを含むBLOB

説明

PV Copy to blob コマンドは *area* の現在のセルセレクションを BLOB として返します。
セレクションは連続している必要はありません。

例題

このメソッドは範囲 A1, A2, B1, B2 と範囲 A3, A4, B3, B4 の内容を入れ替えます。

C_BLOB (\$Bblob) `結果を受け取るBLOB

PV SELECT RANGE(Area;1;1;2;2;pv_selection_set) `A1, A2, B1, B2 を選択

PV SPECIAL CUT(Area;1;1;1) `内容をカットし、クリップボードに格納

PV SELECT RANGE(Area;1;3;2;4;pv_selection_set) `A3, A4, B3, B4 を選択

\$Bblob := *PV Copy to blob*(Area) `セレクションを BLOB に置く

PV SPECIAL PASTE(Area;1;1;1) `クリップボードから内容をペースト

PV GOTO CELL(Area;1;1) `A1 を選択

PV PASTE FROM BLOB(Area;**\$Bblob**;1;1;1;1) `クリアされたものを再代入

PV FIND ALL (area ; criteria ; where ; contains)

引数	型		説明
area	倍長整数	→	4D View エリア
criteria	文字	→	検索する文字列
where	整数	→	0= フォーミュラ ; 1= 値
contains	整数	→	0= 含む ; 1= 等しい

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 *area* 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。

criteria には、検索する文字列を指定します。

where には、検索を行うスプレッドシートを指定します：

- 0: フォーミュラを検索
- 1: 値を検索

contains 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 の例題参照

PV FIND ONE (area ; criteria ; where ; contains { ; column ; row })

引数	型		説明
area	倍長整数	→	4D View エリア
criteria	文字	→	検索する文字列
where	整数	→	0= フォーミュラ ; 1= 値
contains	整数	→	0= 含む ; 1= 等しい
column	倍長整数	→	開始セルの列番号
		←	検出されたセルの列番号
row	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、*area*内のセルで*criteria*を検索し、(あれば) 最初に見つかったセルをカレントセルにします。

criteria には、検索する文字列を指定します。

where 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

contains 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)

オプションの*column*と*row*引数を省略すると、検索は*area*の左上コーナーから開始されます。

*column*と*row*が渡されると、検索を開始する*area*内のセルを指定します。

検索は上から下、左から右に向かって行われます。

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。*column* と *row* 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。

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、入力された条件を含む最初のセルを検索します。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 ($Contains) `0=含む/1=等しい
C_LONGINT ($Column) `検出されたセルの列番号
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=Formulas / 1=Values

    CONFIRM ("一致条件は?"; "含む"; "=")
    $Contains:=OK `0=Contains / 1=Equals

    $Column:=2 `検索はB3セルより開始
    $Row:=3

    PV FIND ONE (Area; $Criteria; $QueryWhere; $Contains; $Column; $Row) `検索
End if
    
```

PV PASTE FROM BLOB

PV PASTE FROM BLOB (area ; BLOB ; value ; formula ; format ; borders)

引数	型		説明
area	倍長整数	⇒	4D View エリア
BLOB	BLOB	⇒	セレクションを含むBLOB
value	整数	⇒	0 = ペーストしない; 1 = ペーストする
formula	整数	⇒	0 = ペーストしない; 1 = ペーストする
format	整数	⇒	0 = ペーストしない; 1 = ペーストする
borders	整数	⇒	0 = ペーストしない; 1 = ペーストする

説明

PV PASTE FROM BLOB コマンドは、*PV Copy to blob* コマンドで作成された *blob* を、*value*, *formula*, *format*, そして *borders* 引数で指定された条件に基づき、*area* のカレントセルにペーストします。

例題

PV Copy to blob コマンドの例題参照

PV REPLACE ALL (area ; string ; replace ; where ; contains)

引数	型		説明
area	倍長整数	⇒	4D View エリア
string	文字	⇒	置換対象の文字列
replace	文字	⇒	置き換え後の文字列
where	整数	⇒	0=フォーミュラ ; 1= 値
contains	整数	⇒	0=含む ; 1=等しい

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 *where* と *contain* で定義した検索条件を満たす *area* 内のすべてのセルの *string* を *replace* 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)
```

PV REPLACE ONE (area ; string ; replace ; where ; contains { ; column ; row })

引数	型		説明
area	倍長整数	→	4D View エリア
string	文字	→	置換対象の文字列
replace	文字	→	置き換える文字列
where	整数	→	0= フォーミュラ ; 1= 値
contains	整数	→	0=含む ; 1=等しい
column	倍長整数	→	開始セルの列番号
		←	検出されたセルの列番号
row	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、column とrowで指定したセルより検索を開始し、where とcontainsで定義した検索条件を満たすarea 内の最初のstring をreplace で置き換えます。

string 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replaceに指定します。

where 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

contains 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)

オプションのcolumnとrow引数を省略すると、検索はareaの左上コーナーから開始されます。columnとrowが渡されると、検索を開始するarea内のセルを指定します。

検索は上から下、左から右に 向かって行われます。

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。column とrow 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。

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($Where) `0=Formula / 1=Value
C_INTEGER($Contains) `0=Contains / 1=Equals

$Where:=0 `Replacement formula
$Contains:=0 `Contains

`Replace first instance
PV REPLACE ONE(Area;"C$30";"C$31";$Where;$Contains)
```

PV SORT COLUMN (area ; column ; order)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	列番号
order	倍長整数	⇒	ソート方向

説明

`PV SORT COLUMN`コマンドを使用して、4D View *area*エリアの*column* 引数で指定した列に標準の並び替え (昇順または降順) を実行できます。

ダイナミック列をソートすると他の列も同期されるため、行は初期の状態を保ちます。スタティックな列をソートすると、その列のみがソートされます。

*order*引数を設定するには**PV Header sort**テーマの以下の定数を使用できます:

定数	型	値	コメント
pv ascending sort	倍長整数	2	4D Viewは昇順ソートを実行します。
pv descending sort	倍長整数	3	4D Viewは降順ソートを実行します。

Note: このコマンドはあらかじめ**PV SET AREA PROPERTY**コマンドでソートが許可されている場合にのみ動作します。

例題

以下のステートメントでは、列を昇順にソートします:

```
PV SORT COLUMN(area;2;pv ascending sort)
```

PV SORT MANY (area ; left ; top ; right ; bottom ; direction ; keys ; order)

引数	型		説明
area	倍長整数	⇒	4D View エリア
left	倍長整数	⇒	左側の列番号
top	倍長整数	⇒	上側の行番号
right	倍長整数	⇒	右側の列番号
bottom	倍長整数	⇒	下側の行番号
direction	整数	⇒	0=行； 1=列
keys	配列	⇒	並び替える値を含む列、または行
order	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列または行をkeyに渡し、並び替えの実行順（昇順または降順）をorderに渡します。

left, top, right, そしてbottom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。

direction には、行または列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、keys行の値に応じて列を並び替えます。
- 1 を渡した場合、keys列の値に応じて行を並び替えます。

例題

セルの選択範囲（5 列 X 9 行）に対し、2 番目の列を最初の条件として昇順に、3番目の列を第2 の条件として降順に並び替えます。

```

ARRAY LONGINT ($Keys;2) `並び替える値を含む列、または行
ARRAY INTEGER ($Orders;2) `0= 昇順 / 1= 降順

`Initialization
$Keys{1}:=3 `選択範囲の2番目の列を並び替えの第1条件として使用
$Keys{2}:=4 `選択範囲の3番目の列を並び替えの第2条件として使用
$Orders{1}:=0 `選択範囲の2番目の列との関係で行を昇順に並び替える
$Orders{2}:=1 `選択範囲の3番目の列との関係で行を降順に並び替える

PV SORT MANY(Area;2;2;6;10;1;$Keys;$Orders)

```

PV SORT ONE (area ; left ; top ; right ; bottom ; direction ; key ; order)

引数	型		説明
area	倍長整数	⇒	4D View エリア
left	倍長整数	⇒	左側の列番号
top	倍長整数	⇒	上側の行番号
right	倍長整数	⇒	右側の列番号
bottom	倍長整数	⇒	下側の行番号
direction	整数	⇒	0=行; 1=列
key	倍長整数	⇒	並び替える値を含む列、または行
order	整数	⇒	0=昇順; 1=降順

説明

PV SORT ONEコマンドは、*left*, *top*, *right*, および *bottom* で区切られた選択範囲の内容を、*key* となる行または列内の値との関連で、*order* で指定された順序に並び替えます。

direction には、行または列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、*key* 行の値に応じて列を並び替えます。
- 1 を渡した場合、*key* 列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列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、および計算結果の列（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列）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、列ヘッダのクリックによって、スタティックな列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列だけが含まれています。

- 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、列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) \クリックがヘッダで行われた
    $0:=True \イベントをブロック
    PV SORT ONE(area;$4;1;$4;25;1;$4;0) \選択した列を昇順ソート
End if
```

PV SPECIAL CLEAR

PV SPECIAL CLEAR (area ; value ; formula ; format ; borders)

引数	型		説明
area	倍長整数	⇒	4D View エリア
value	整数	⇒	0= クリアしない ; 1= クリア
formula	整数	⇒	0= クリアしない ; 1= クリア
format	整数	⇒	0= クリアしない ; 1= クリア
borders	整数	⇒	0= クリアしない ; 1= クリア

説明

PV SPECIAL CLEAR コマンドは、*area* におけるカレントセルの選択範囲から引数 *value*、*formula*、*format*、*borders* 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ミュラ、フォーマット、罫線を消去し、フォーミ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) `フォーミュラ、フォーマット、罫線を消去
```

PV SPECIAL CUT (area ; value ; formula ; format ; borders)

引数	型		説明
area	倍長整数	⇒	4D View エリア
value	整数	⇒	0= カットしない; 1= カット
formula	整数	⇒	0= カットしない; 1= カット
format	整数	⇒	0= カットしない; 1= カット
borders	整数	⇒	0= カットしない; 1= カット

説明

PV SPECIAL CUT コマンドは、*area* におけるカレントセルの選択範囲に対して、引数 *value*、*formula*、*format*、*borders* 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 コマンドの例題参照

PV SPECIAL PASTE (area ; value ; formula ; format ; borders)

引数	型		説明
area	倍長整数	⇒	4D View エリア
value	整数	⇒	0= ペーストしない; 1= ペースト
formula	整数	⇒	0= ペーストしない; 1= ペースト
format	整数	⇒	0= ペーストしない; 1= ペースト
borders	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value、formula、format、bordersで指定された情報をarea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
```

PVセレクション

セレクションコマンドについて

 PV GET SELECTED RANGES LIST

 PV Is all selected

 PV Is cell selected

 PV Is column selected

 PV Is range selected

 PV Is row selected

 PV SELECT ALL

 PV SELECT CELL

 PV SELECT COLUMNS

 PV SELECT RANGE

 PV SELECT RANGES LIST

 PV SELECT ROWS

🌱 セレクションコマンドについて

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 GET SELECTED RANGES LIST

PV GET SELECTED RANGES LIST (area ; left ; top ; right ; bottom)

引数	型		説明
area	倍長整数	⇒	4D View エリア
left	倍長整数配列	⇐	左側のセルの列番号の配列
top	倍長整数配列	⇐	上側のセルの行番号の配列
right	倍長整数配列	⇐	右側のセルの列番号の配列
bottom	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、*area*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

PV Is all selected

PV Is all selected (area) -> 戻り値

引数	型		説明
area	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*は、*area*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。

PV Is cell selected

PV Is cell selected (area ; column ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	セルの列番号
row	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*は、*area*中で*column*と*row*で指定されたセルが、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コマンドの例題参照

PV Is column selected

PV Is column selected (area ; column) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	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*は、*area*内の列番号である*column*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の例題参照](#)

PV Is range selected

PV Is range selected (area ; left ; top ; right ; bottom) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
left	倍長整数	→	左側のセルの列番号
top	倍長整数	→	上側のセルの行番号
right	倍長整数	→	右側のセルの列番号
bottom	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*は、引数*left*、*top*、*right*、*bottom*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](#)コマンドの例題参照

PV Is row selected

PV Is row selected (area ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
row	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*は、*area*の行番号である*row*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](#)コマンドの例題参照

PV SELECT ALL

PV SELECT ALL (area ; selection)

引数	型		説明
area	倍長整数	⇒	4D View エリア
selection	整数	⇒	選択オプション: 0 = 選択、1 = 選択解除

説明

PV SELECT ALL コマンドは、*area* のすべてのセルを選択あるいは選択解除します。

selection 引数が省略されるか0が渡されると、エリアのすべてのセルが選択されます。*selection* に1を渡すとすべてのセルの選択が解除されます。

PV SELECT CELL (area ; column ; row ; action)

引数	型		説明
area	倍長整数	➡	4D View エリア
column	倍長整数	➡	セルの列番号
row	倍長整数	➡	セルの行番号
action	整数	➡	選択動作

説明

PV SELECT CELLコマンドは、column とrowが交差する位置にあるセルを選択します。

引数action を使用して、既に選択されているセルがある場合の選択動作を定義できます。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。action の指定は**PV Selection action**テーマの定数を使用して行います

定数	型	値	コメント
pv selection add	倍長整数	1	新しいセレクションは既存のセレクションに追加される
pv selection reduce	倍長整数	2	既存のセレクションから選択したセレクションが取り除かれる
pv selection set	倍長整数	0	新しいセレクションが既存のセレクションを置き換える

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

PV GET SELECTED RANGES LIST(Area;$Left;$Top;$Right;$Bottom) `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

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

```

PV SELECT COLUMNS (area ; first ; last ; action)

引数	型		説明
area	倍長整数	➡	4D View エリア
first	倍長整数	➡	最初の選択列
last	倍長整数	➡	最後の選択列
action	整数	➡	選択動作

説明

PV SELECT COLUMNSコマンドは、*first* と*last*で指定された列番号の間に存在する*area* の列を選択します。

引数*action* 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。*action* の定義は、**PV Selection action** 定数を使用して行います:

定数	型	値	コメント
pv selection add	倍長整数	1	新しいセレクションは既存のセレクションに追加される
pv selection reduce	倍長整数	2	既存のセレクションから選択したセレクションが取り除かれる
pv selection set	倍長整数	0	新しいセレクションが既存のセレクションを置き換える

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; $RowSelect) `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
    
```

🔧 PV SELECT RANGE

PV SELECT RANGE (area ; left ; top ; right ; bottom ; action)

引数	型		説明
area	倍長整数	⇒	4D View エリア
left	倍長整数	⇒	左側のセルの列番号
top	倍長整数	⇒	上側のセルの行番号
right	倍長整数	⇒	右側のセルの列番号
bottom	倍長整数	⇒	下側のセルの行番号
action	整数	⇒	選択動作

説明

PV SELECT RANGE コマンドは、*left*、*top*、*right*、*bottom* で定義された範囲のセルを選択します。

引数 *action* 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。*action* の定義は **PV Selection action** 定数を使用しています。

定数	型	値	コメント
pv selection add	倍長整数	1	新しいセレクションは既存のセレクションに追加される
pv selection reduce	倍長整数	2	既存のセレクションから選択したセレクションが取り除かれる
pv selection set	倍長整数	0	新しいセレクションが既存のセレクションを置き換える

例題 1

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

PV GET SELECTED RANGES LIST(Area;$Left;$Top;$Right;$Bottom) ´ 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

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
```

例題 2

この例題は **Alt+clicked** (Windows) や **Option+clicked** (Mac OS) 時に、そのセルが選択範囲中にあるかどうかによって動作を切り替えます：

```
´カレントの選択範囲を指定
PV SELECT RANGE(area;1;5;2;9;pv_selection_set)

´エリアがクリックされたらメソッドを呼び出す
PV ON EVENT(area;pv_on_clicked;"ExampleView")

´ExampleView メソッド
C_LONGINT ($1;$2;$3;$4;$5)
If (($2=pv_on_clicked) & ($3=2048)) ´Alt + click または Option + click
    If (PV Is cell selected(area;$4;$5)=1)
```

セルが選択範囲内なら、選択から取り除く

```
PV SELECT RANGE(area;$4;$5;$4;$5;pv_selection_reduce)
```

Else

セルが選択範囲外なら、そこに加える

```
PV SELECT RANGE(area;$4;$5;$4;$5;pv_selection_add)
```

End if

End if

🔧 PV SELECT RANGES LIST

PV SELECT RANGES LIST (area ; left ; top ; right ; bottom ; action)

引数	型		説明
area	倍長整数	⇒	4D View エリア
left	配列	⇒	左側のセルの列番号の配列
top	配列	⇒	上側のセルの行番号の配列
right	配列	⇒	右側のセルの列番号の配列
bottom	配列	⇒	下側のセルの行番号の配列
action	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left、top、right、bottom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。

actionの定義は、PV Selection action定数を使用して行います:

定数	型	値	コメント
pv selection add	倍長整数	1	新しいセレクションは既存のセレクションに追加される
pv selection reduce	倍長整数	2	既存のセレクションから選択したセレクションが取り除かれる
pv selection set	倍長整数	0	新しいセレクションが既存のセレクションを置き換える

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($Index) `Loop index
C_INTEGER($Number) `Number of ranges

$Number:=5 `The group of five in the range
ARRAY LONGINT($Left;5)
ARRAY LONGINT($Top;5)
ARRAY LONGINT($Right;5)
ARRAY LONGINT($Bottom;5)

`Initialization
For($Index;1;$Number)
    $Left{$Index}:=$Index*3 `Left limits
    $Top{$Index}:=$Index*6 `Top limits
    $Right{$Index}:=$Index*4 `Right limits
    $Bottom{$Index}:=$Index*7 `Bottom limits
End for

PV SELECT RANGES LIST(Area;$Left;$Top;$Right;$Bottom;pv_selection add)
```

PV SELECT ROWS (area ; first ; last ; action)

引数	型		説明
area	倍長整数	→	4D View エリア
first	倍長整数	→	最初の選択行
last	倍長整数	→	最後の選択行
action	整数	→	選択動作

説明

`PV SELECT ROWS` コマンドは、`first` と `last` で指定された行番号の間に存在する `area` の行を選択します。

引数 `action` 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。`action` の定義は、**PV Selection action** 定数を使用して行います:

定数	型	値	コメント
pv selection add	倍長整数	1	新しいセレクションは既存のセレクションに追加される
pv selection reduce	倍長整数	2	既存のセレクションから選択したセレクションが取り除かれる
pv selection set	倍長整数	0	新しいセレクションが既存のセレクションを置き換える

例題

[PV SELECT COLUMNS](#) コマンドの例題参照

PVツール

-  ツールコマンドについて
-  PV Color to index
-  PV COLOR TO RGB
-  PV Get window title
-  PV Index to color
-  PV RGB to color
-  PV SET WINDOW TITLE

ツールコマンドについて

このテーマではさまざまな方法でのカラー参照を管理する各種ツール、およびプラグインウィンドウのタイトルの取得や設定を行うコマンドについて説明しています。

PV Color to index (color) -> 戻り値

引数	型		説明
color	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*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 *color* 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。

4D で使用されるRGB カラーに関する詳細は、以下の4Dコマンドの説明を参照してください。

- **OBJECT SET RGB COLORS** : 4D で使用されるRGBカラーシステム用。
- **OBJECT SET COLOR** : 4D のカラーパレット用。

注: 4D View カラーの内部コーディングはBGR 型で、4DのRGBコーディングに合わせるために反対にされます。しかしながら、使用する際の原理は同じです

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($Color) `A1 セルの背景色(RGB)後にインデックス後のカラー
C_INTEGER($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;"Color") `ラベル
PV SET CELL STRING VALUE(Area;2;2;"Red")
PV SET CELL STRING VALUE(Area;2;3;"Green")
PV SET CELL STRING VALUE(Area;2;4;"Blue")

$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)
PV SET CELL PROPERTY(Area;1;1;pv_style_color_back_odd;$Color) `A1 の背景色を割り当てる

$Index:=PV Color to index($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
PV SET CELL PROPERTY(Area;3;1;pv_style_color_back_odd;$Color) `C1 セルの背景色を割り当てる

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)

```

PV COLOR TO RGB (color ; red ; green ; blue)

引数	型		説明
color	倍長整数	➡	カラー
red	整数	←	赤の成分 (0 ~ 255)
green	整数	←	緑の成分 (0 ~ 255)
blue	整数	←	青の成分 (0 ~ 255)

説明

PV COLOR TO RGB コマンドは、RGB *color*の3つの成分を取得し、引数*red*、*green*、*blue*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、4Dコマンド**OBJECT SET RGB COLORS**を参照してください。

注: 4D View カラーの内部コーディングはBGR 型で、4DのRGBコーディングに合わせるために反対にされます。しかしながら、使用する際の原理は同じです。

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; "Color") `ラベル
PV SET CELL STRING VALUE (Area; 2; 2; "Red")
PV SET CELL STRING VALUE (Area; 2; 3; "Green")
PV SET CELL STRING VALUE (Area; 2; 4; "Blue")

$Color := PV Get cell property (Area; 1; 1; pv style color back odd)
PV COLOR TO RGB ($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)
```

PV Get window title

PV Get window title (area) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
戻り値	文字	↩	プラグインウィンドウのタイトル

説明

*PV Get window title*は、4D View のプラグインarea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

PV Index to color

PV Index to color (index) -> 戻り値

引数	型		説明
index	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*は、*index*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。

index 引数に対しては、**Colors** テーマ内の4Dランゲージの定数を、最初の16色に対して使用することができます(256を含む、パレットの1列目)。

4Dで使用されるカラーに関する詳細は、以下の4Dコマンドの説明を参照してください。

- **OBJECT SET RGB COLORS** : 4Dで使用されるRGBカラーシステムについて。
- **OBJECT SET COLOR** : 4Dのカラーパレットについて。

注: 4D View カラーの内部コーディングはBGR 型で、4DのRGBコーディングに合わせるために反対にされます。しかしながら、使用する際の原理は同じです。

例題

*PV SET RANGE BORDER*および*PV Color to index* コマンドの例題参照

PV RGB to color

PV RGB to color (red ; green ; blue) -> 戻り値

引数	型		説明
red	整数	→	赤の成分 (0 ~ 255)
green	整数	→	緑の成分 (0 ~ 255)
blue	整数	→	青の成分 (0 ~ 255)
戻り値	倍長整数	↩	カラー

説明

*PV RGB to color*は、RGB カラーを定義する倍長整数タイプの値を返します。この値は*red*、*green*、*blue*の成分に基づいて決定されます。

4Dで使用されるカラーシステムに関する詳細は、4Dコマンドである[OBJECT SET RGB COLORS](#)を参照してください。

注: 4D View カラーの内部コーディングはBGR 型で、4DのRGBコーディングに合わせるために反対にされます。しかしながら、使用する際の原理は同じです。

例題

PV Color to index コマンドの例題を参照

PV SET WINDOW TITLE

PV SET WINDOW TITLE (area ; title)

引数	型		説明
area	倍長整数	→	4D View エリア
title	文字	→	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 *area* のウィンドウタイトルを *title* に設定します。

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)+") ")
```

PVドキュメント

ドキュメントコマンドについて

 PV EXPORT

 PV GET DOCUMENT INFO

 PV Get document property

 PV OPEN DOCUMENT

 PV SAVE DOCUMENT

 PV SET DOCUMENT INFO

 PV SET DOCUMENT PROPERTY

ドキュメントコマンドについて

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列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 EXPORT (area ; document ; replace ; format)

引数	型	説明
area	倍長整数	→ 4D View エリア
document	文字	→ ドキュメント名または空の文字列
replace	整数	→ 0= 置換しない ; 1= 置換
format	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 area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。

document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replace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、replace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（documentに空の文字列を指定）、引数replace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：

定数	型	値
pv html	倍長整数	3
pv sylk	倍長整数	2
pv tab tab return	倍長整数	1

Note : 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 ($ALleft;0) `左側のセルの列番号
ARRAY LONGINT ($ALtop;0) `上側のセルの行番号
ARRAY LONGINT ($ALright;0) `右側のセルの列番号
ARRAY LONGINT ($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)

```


PV GET DOCUMENT INFO

PV GET DOCUMENT INFO (area ; title ; subject ; author ; company ; comment ; creationDate ; creationTime ; modificationDate ; modificationTime)

引数	型		説明
area	倍長整数	→	4D View エリア
title	文字	←	ドキュメントのタイトル
subject	文字	←	ドキュメントの題名
author	文字	←	ドキュメントの作成者
company	文字	←	会社名
comment	テキスト	←	コメント
creationDate	日付	←	作成日
creationTime	時間	←	作成時刻
modificationDate	日付	←	最終変更日
modificationTime	時間	←	最終変更時刻

説明

PV GET DOCUMENT INFO コマンドは、4D View *area* に表示されたドキュメントの情報を、引数 *title*、*subject*、*author*、*company*、*comment* 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 *creationDate*、*creationTime*、*modificationDate*、*modificationTime* 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 コマンドの例題参照


```
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)+".")
```

PV OPEN DOCUMENT (area ; document ; template)

引数	型	説明
area	倍長整数	⇒ 4D View エリア
document	文字	⇒ ドキュメントの名前
template	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 *document* を *area* 内で開きます。

document 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 *template* により、ドキュメントを標準ドキュメントとして開くか (*template* =0)、またはテンプレートとして開くか (*template* =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。*template* 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。

Note: “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) `ドキュメント選択
```

システム変数およびセット

Document システム変数には最後に開かれたディスクファイルのドキュメント名あるいはドキュメントパスが格納されます (4D Language Reference マニュアルの **システム変数** を参照)。

OK システム変数には *document* が正しく開かれると1が設定されます。

PV SAVE DOCUMENT (area ; document ; template ; replace ; format)

引数	型	説明
area	倍長整数	→ 4D View エリア
document	文字	→ ドキュメントの名前
template	整数	→ 0= ドキュメント; 1= テンプレート
replace	整数	→ 0= 置換しない; 1= 置換
format	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 *area* をディスクドキュメントとして保存します。

document 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*template*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 (*template* =0)、またはテンプレートとして保存するか (*template* =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*erase* 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、*erase* 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 (*document*に空の文字列を渡す) には、*erase* 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*format* 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:

定数	型	値
pv html	倍長整数	3
pv sylk	倍長整数	2
pv tab tab return	倍長整数	1
pv view	倍長整数	0

例題

[PV OPEN DOCUMENT](#)コマンドで外部ドキュメントを開いた後、まずはじめにアクティブセルに対する変更をすべて検出するコールバックメソッドを設定します。

```
C_BOOLEAN (FlagModifiedArea)
PV ON EVENT (Area;pv_on_active_cell_changed;"EventMethod")
```

プロジェクトメソッド**EventMethod**のコードは以下の通りです:

```
C_LONGINT ($1) `4D View area reference
C_LONGINT ($2) `Type of event
C_LONGINT ($3) `Modification key code
C_LONGINT ($4) `Column number
C_LONGINT ($5) `Row number
C_LONGINT ($6) `Ascii code of the key
C_BOOLEAN ($0) `Value to return

$0:=False
FlagModifiedArea:=True `Modified area
```

ドキュメントが変更されたら、ユーザは変更後のドキュメントを保存し、好きな名前を指定することができます。

```
If(FlagModifiedArea) `Document modified?  
  CONFIRM("Do you want to save this document as a template??";"Template";"Document")  
  `This will be a template if the dialog box is confirmed (OK=1)  
  PV SAVE DOCUMENT(Area;"";OK;1;pv_view)  
End if
```

システム変数およびセット

ドキュメントが正常に保存された場合、システム変数OK には1 が代入されます。

PV SET DOCUMENT INFO (area ; title ; subject ; author ; company ; comment)

引数	型		説明
area	倍長整数	➡	4D View エリア
title	文字	➡	ドキュメントのタイトル
subject	文字	➡	ドキュメントの題名
author	文字	➡	ドキュメントの作成者
company	文字	➡	会社名
comment	テキスト	➡	コメント

説明

PV SET DOCUMENT INFOコマンドは、引数title、subject、author、company、comment に渡された情報をarea 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
C_DATE ($CreationDate) `ドキュメントの作成日
C_TIME ($CreationTime) `ドキュメントの作成時刻
C_DATE ($ModificationDate) `ドキュメントの最終更新日
C_TIME ($ModificationTime) `ドキュメントの最終更新時刻

`ドキュメントに関する情報を取得
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?"; $Title)
$Subject :=Request ("ドキュメントの題名?"; $Subject)
$Author :=Request ("作成者の名前?"; $Author)
$Company :=Request ("会社名?"; $Company)
$Comment :=Request ("コメント?"; $Comment)

PV SET DOCUMENT INFO(Area; $Title; $Subject; $Author; $Company; $Comment) `情報を更新

```

PV SET DOCUMENT PROPERTY (area ; option ; value)

引数	型		説明
area	倍長整数	→	4D View エリア
option	倍長整数	→	プロパティの番号
value	倍長整数	→	プロパティの値

説明

PV SET DOCUMENT PROPERTYコマンドは、4D View ドキュメントareaに対し、option で指定されたプロパティのvalueを設定します。

PV Document propertiesテーマの定数を使用してoption 引数を指定できます：

定数	型	値	コメント
pv column count	倍長整数	0	<p>エリアに表示される列数の設定と読み込みを許可</p> <p>エリアの"更新"属性の読み書きを許可。使用可能な値: <code>pv value on</code>または<code>pv value off</code></p>
pv document modified	倍長整数	4	<ul style="list-style-type: none"> 書き込みモードでこの定数が使用されると(PV SET DOCUMENT PROPERTYコマンド)、value引数に<code>pv value on</code>を渡すとエリアが閉じられるときに、エリアが更新されたことを通知する警告ダイアログが表示されます。value引数に<code>pv value off</code>を渡し、ユーザーもプログラムもドキュメントを更新していなければこのダイアログボックスは表示されません。 この定数がPV Get document propertyコマンドで読み取りのために使用されると、ドキュメントが更新されていた場合に1が、そうでなければ0が返されます。 <p>エリアのフォーミュラ中で、4D変数やメソッド、コマンドの呼び出しを禁止します。利用可能な値: <code>pv value on</code>または<code>pv value off</code></p>
pv no external call	倍長整数	3	<ul style="list-style-type: none"> <code>pv value on</code>: フォーミュラ中で、4D変数やメソッド、コマンドの呼び出しが禁止されます。この場合、PV入力認可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。 <code>pv value off</code>: フォーミュラ中で、すべての4D変数やメソッド、コマンドの呼び出しが許可されます (デフォルト)。
pv row count	倍長整数	1	<p>エリアに表示される行数の設定と読み込みを許可</p>

例題 1

このメソッドを使用して新規4D Viewエリアの列数や行数を指定できます。

```
PV SET DOCUMENT PROPERTY(Area;pv column count;10) `10 列
PV SET DOCUMENT PROPERTY(Area;pv row count;20) `20行
```

例題 2

このメソッドは例えば4D Viewドキュメントの閉じるボタンに割り当てられて、エリアが更新された旨のダイアログを表示しないようにします：

```
If (PV Get document property(Area;pv document modified)#0)
  PV SET DOCUMENT PROPERTY(Area;pv document modified;0) `0 = 変更しない, 1 = 変更する
End if
```


PVドラッグアンドドロップ

 ドラッグアンドドロップコマンドについて

 PV GET DRAG SIGNATURES

 PV GET DRAG SOURCE

 PV Get drop info

 PV GET DROP SIGNATURES

 PV GET DROP TARGET

 PV SET DRAG SIGNATURES

 PV SET DROP SIGNATURES

🌱 ドラッグアンドドロップコマンドについて

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）
- 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する署名

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。

4Dオブジェクトのドラッグ&ドロップ

4D Viewでは4Dオブジェクトをセル間でドラッグおよびドロップできます。BLOBを除き、すべての4Dフィールドおよび変数タイプを4D Viewエリアにドロップできます。

- 4Dではドラッグされるオブジェクトに“ドラッグ可”プロパティが選択されていなければなりません。
- 4D Viewでは、**PV Drag drop allowed**テーマの定数`pv_DD_4D_objects`を**PV SET AREA PROPERTY**コマンドに渡さなければなりません。

4Dオブジェクトの署名は `__OBJECT4D__` (`__` はアンダーライン2つ) です。この内部的な署名は変更できません。この署名を**PV SET DROP SIGNATURES**コマンドに渡して4Dオブジェクトのドロップを許可します。

PV GET DRAG SIGNATURES

PV GET DRAG SIGNATURES (area ; signatures)

引数	型		説明
area	倍長整数	➡	4D View エリア
signatures	文字配列	↔	署名の配列

説明

PV GET DRAG SIGNATURES コマンドは、*area* のドラッグ署名を取得し、*signatures* 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
```

PV GET DRAG SOURCE

PV GET DRAG SOURCE (area ; source ; signatures)

引数	型		説明
area	倍長整数	→	4D View エリア
source	ポインター	←	ドラッグ対象のソースオブジェクトへのポインタ
signatures	文字配列	←	署名配列

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 *source* に返します。
signatures 配列には移動中のオブジェクトの署名が返されます。ドラッグされ、そしてドロップ中に適切なコードが実行されたときに、この引数を使用することで4Dオブジェクトと他の4D Viewオブジェクトとを区別できます。4Dオブジェクトの署名は `__OBJECT4D__` (`__` は2つの下線) です。kの内部的な署名は変更できません。詳細は [ドラッグアンドドロップコマンドについて](#) を参照してください。

例題

PV SET DRAG SIGNATURES コマンドの例題参照

🔧 PV Get drop info

PV Get drop info (area ; option) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
option	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 drop infoコマンドは、指定されたoptionに対するドラッグ&ドロッププロパティの値を返します。

optionを指定するには**PV Drop info**定数を使用します：

定数	型	値	コメント
pv drag process	倍長整数	0	ドラッグ元エリアのプロセス番号を返す。
pv drag plugin	倍長整数	1	ドラッグされた4D Viewエリアの番号を返す。
pv drag column	倍長整数	2	ドラッグ元の列番号を返す。
pv drag row	倍長整数	3	ドラッグ元の行番号を返す。
pv drag X offset	倍長整数	4	(左上隅のセルを基点として)ドラッグアクションが行われたセルのX座標を返す。
pv drag Y offset	倍長整数	5	(左上隅のセルを基点として)ドラッグアクションが行われたセルのY座標を返す。
pv drag content	倍長整数	6	
pv drop process	倍長整数	7	ドロップ先エリアのプロセス番号を返す。
pv drop plugin	倍長整数	8	ドロップされたエリアの番号を返す。
pv drop column	倍長整数	9	ドロップ先の列番号を返す。
pv drop row	倍長整数	10	ドロップ先の行番号を返す。
pv drop X offset	倍長整数	11	(左上隅のセルを基点として)ドロップアクションが行われたセルのX座標を返す。
pv drop Y offset	倍長整数	12	(左上隅のセルを基点として)ドロップアクションが行われたセルのY座標を返す。
pv drop content	倍長整数	13	
pv drop action	倍長整数	14	ユーザーにより行われたドロップアクションを取得できるようにする。 PV Drop action テーマの定数値を返します。

PV Drop actionテーマの定数は以下の通りです：

定数	型	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

例題

*PV SET DRAG SIGNATURES*コマンドの例題参照

PV GET DROP SIGNATURES

PV GET DROP SIGNATURES (area ; signatures)

引数	型		説明
area	倍長整数	→	4D View エリア
signatures	文字配列	←	署名の配列

説明

PV GET DROP SIGNATURES コマンドは、*area* のドロップ署名を取得して *signatures* 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 コマンドの例題参照

PV GET DROP TARGET

PV GET DROP TARGET (area ; target)

引数	型		説明
area	倍長整数	→	4D View エリア
target	ポインター	←	ドロップ対象のターゲット

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 *target* に返します。

例題

PV SET DRAG SIGNATURES コマンドの例題参照

🔧 PV SET DRAG SIGNATURES

PV SET DRAG SIGNATURES (area ; signatures)

引数	型		説明
area	倍長整数	⇒	4D View エリア
signatures	文字配列	⇒	署名の配列

説明

PV SET DRAG SIGNATURES コマンドは、配列 *signatures* の内容を *area* 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"

PV SET AREA PROPERTY(SourceArea;pv drag trigger;pv trigger on alt click) `alt-clickでドラッグ
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) `列番号
C_LONGINT($5) `行番号
C_LONGINT($6) `キーのASCII コード

C_POINTER(SourceAreaPointer;TargetAreaPointer) `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_BLOB($blob) `一時的なドラッグ&ドロップノートパッド
C_LONGINT($currentColumn;$currentRow) `カレントセルの座標
C_LONGINT($destinationColumn;$destinationRow) `ドロップが行われるセルの座標

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 drop info(TargetAreaPointer->;pv_drop_column)
`コピー先....
  $destinationRow:=PV Get drop info(TargetAreaPointer->;pv_drop_row) `...coordinates
  PV GET CURRENT CELL(TargetAreaPointer->;$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;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
```

PV SET DROP SIGNATURES

PV SET DROP SIGNATURES (area ; signatures)

引数	型		説明
area	倍長整数	⇒	4D View エリア
signatures	文字配列	⇒	署名の配列

説明

PV SET DROP SIGNATURES コマンドは、配列 *signatures* の内容を *area* 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 コマンドの例題参照

PVピクチャ

ピクチャコマンドについて

-  PV Add picture
-  PV Create picture
-  PV Get picture
-  PV Get picture property
-  PV REMOVE PICTURE
-  PV SET PICTURE PROPERTY

ピクチャコマンドについて

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。ピクチャープロパティを変更することで、アピランスや透過、サイズ、位置などを変更できます。

ピクチャーの位置

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るのです。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*PicNum*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 (area ; picture ; expression ; tableNum ; fieldNum) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
picture	ピクチャー	→	4Dピクチャ
expression	文字	→	4D ピクチャを返す任意の式
tableNum	整数	→	テーブル番号
fieldNum	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 コマンドは、*area* 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ピクチャ**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*picture* 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*expression* に式の名前を渡します（引数*picture* 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（"[Table]PictureField"）を返す4D メソッド名を引数*expression* 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*tableNum* および*fieldNum* にテーブルおよびフィールドの番号を渡します（引数*picture* および*expression* 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 ($PicRef) `追加されるピクチャの参照  
C_PICTURE ($Picture) `空のピクチャ（無視される）
```

```
$PicRef:=PV Add picture(Area;$Picture;"";Table(->[Clients]);Field(->[Clients]Photo))
```

PV Create picture

PV Create picture (area ; left ; top ; right ; bottom ; ignoreEmptyCells) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
left	倍長整数	→	左側セルの列番号
top	倍長整数	→	一番上のセルの行番号
right	倍長整数	→	右側セルの列番号
bottom	倍長整数	→	一番下のセルの行番号
ignoreEmptyCells	整数	→	0= 空のセルを無視しない ; 1= 空のセルを無視
戻り値	ピクチャー	↻	領域内のセルのピクチャ

説明

`PV Create picture` コマンドは、引数 `left`、`top`、`right`、`bottom` で指定されたセル範囲のピクチャを返します。

引数 `ignoreEmptyCells` 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 `right` と `bottom` よりも小さい場合、引数 `left`、`top`、`right`、`bottom` で指定された枠の大きさが縮小されます。

注: 作成されたピクチャは 2048 x 2048 ピクセルより大きくすることはできません。それより大きくした場合、ピクチャは自動的に切り落とされます。

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。

```
[Templates]ReducedView:=PV Create picture(Area;2;2;5;5;0) `ピクチャを取得してフィールドに代入
```

PV Get picture

PV Get picture (area ; picNum) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
picNum	倍長整数	→	ピクチャ番号
戻り値	ピクチャー	↩	ピクチャー

説明

PV Get pictureコマンドはarea内のpicNumで指定した番号のピクチャーを返します。

例題

ピクチャ番号1 のピクチャーをカレントセルにコピーします。

```
C_LONGINT ($PicRef) `追加されるピクチャーの参照番号
C_PICTURE ($Picture) `コピーするピクチャー

$Picture:=PV Get picture(Area;1) `ピクチャー番号1
PV REMOVE PICTURE(Area;1)
$PicRef:=PV Add picture(Area;$Picture) `カレントセルにコピーする
```

エラー管理

ピクチャーが空かピクチャー番号が無効な場合、area内でエラーが返されます。

🔧 PV Get picture property

PV Get picture property (area ; picNum ; property) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
picNum	倍長整数	→	ピクチャ番号
property	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` コマンドは、`area` 内で `picNum` で指定されたピクチャに関するプロパティの値を返します。

引数 `properties` を定義するには、**PV Picture properties** テーマの定数を使用します：

定数	型	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 hor 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

また、引数 `properties` 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：

定数	型	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

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"  
  
$PictureInfo:="ピクチャ番号1の情報:"+Char(Carriage_return)  
For($Index;1;12) `異なるプロパティを調査  
    $Value:=PV Get picture property(Area;1;$PropertiesCodes{$Index}) `プロパティの読み込み  
    $PictureInfo:=$PictureInfo+$PropertiesLabels{$Index}+" : "+String($Value)+". " `情報を更新  
End for  
  
ALERT($PictureInfo) `情報を表示
```

PV REMOVE PICTURE

PV REMOVE PICTURE (area ; picNum)

引数	型		説明
area	倍長整数	→	4D View エリア
picNum	倍長整数	→	ピクチャ番号

説明

PV REMOVE PICTURE コマンドは、*picNum* で指定した番号のピクチャを *area* 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す。詳細については[ピクチャコマンドについて](#)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)
```

PV SET PICTURE PROPERTY

PV SET PICTURE PROPERTY (area ; picNum ; property ; value)

引数	型		説明
area	倍長整数	→	4D View エリア
picNum	倍長整数	→	ピクチャ番号
property	倍長整数	→	プロパティ番号
value	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propertyのvalueをpicNumのピクチャに設定します。

引数propertyを指定するには**PV Picture properties**テーマの定数を使用します：

定数	型	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 hor 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

また、引数valueの指定には**PV Picture mapping mode**テーマ内の定数を使用することができます：

定数	型	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

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)
```

PVプラグインプロパティ

 プラグインプロパティコマンドについて

 PV Get plugin property

 PV SET PLUGIN PROPERTY

プラグインプロパティコマンドについて

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列数、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 Get plugin property

PV Get plugin property (property) -> 戻り値

引数	型		説明
property	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 コマンドは、汎用的な4D View プラグインの *property* の現在値を返します。

引数 *property* の指定には **PV Plugin properties** テーマ内の定数を使用します。こ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。")
```

PV SET PLUGIN PROPERTY (property ; value)

引数	型	説明
property	倍長整数 →	プロパティ番号
value	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propertyのvalue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propertyの指定には、**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value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propertyおよびvalueに使用できる定数について、以下のリストで説明します。

定数	型	値	コメント
pv button height	倍長整数	5	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 button width	倍長整数	4	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 confirm convert dialog	倍長整数	6	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。 <ul style="list-style-type: none"> ● 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。
pv default columns count	倍長整数	2	新しい4D View ドキュメント内のデフォルトの列数を定義します。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列が含まれます。関連する値：列数。
pv default rows count	倍長整数	3	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 load template on server	倍長整数	1	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。 <ul style="list-style-type: none"> ● 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 write template on server	倍長整数	0	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。 <ul style="list-style-type: none"> ● 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

例題

すべての4D Viewエリアが100列50行で初期化されて作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)
```

PVプリント

-  プリントコマンドについて
-  PV BLOB TO PRINT SETTINGS
-  PV Get header
-  PV Get print property
-  PV PRINT
-  PV PRINT FORMULAS
-  PV Print settings to blob
-  PV SET HEADER
-  PV SET PRINT PROPERTY

プリントコマンドについて

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 BLOB TO PRINT SETTINGS

PV BLOB TO PRINT SETTINGS (area ; printSettings)

引数	型		説明
area	倍長整数	→	4D Viewエリア
printSettings	BLOB	→	印刷設定を格納したBLOB

説明

PV BLOB TO PRINT SETTINGS コマンドは *area* で指定した 4D View エリアの現在の印刷設定を、*printSettings* BLOB で置き換えます。この BLOB は *PV Print settings to blob* コマンドで生成されなければなりません。

printSettings 引数には印刷に使用されるすべての設定が含まれます:

- 用紙設定 (用紙, 向き, 倍率);
- 印刷設定 (印刷枚数, トレイ等).

Note: 印刷設定は Windows と Mac OS で異なるフォーマットが使用されています。結果 2 プラットフォーム間の *printSettings* BLOB の互換性は保証されません。

printSettings BLOB に有効な印刷設定が含まれていない場合、コマンドはエラーを生成します。

PV Get header

PV Get header (area ; header) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
header	倍長整数	→	ヘッダの位置
戻り値	文字	↺	ヘッダの文字列

説明

PV Get header コマンドは、*header* 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 *header* の指定には **PV Headers & footers** 定数を使用します:

定数	型	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

例題

ヘッダ中央からフッターにテキストを移します:

```
C_TEXT($Header) `ヘッダ中央のテキスト
```

```
$Header:=PV Get header(Area;pv_header_center) `ヘッダ中央のテキストを取得
```

```
PV SET HEADER(Area;pv_header_center; "") `ヘッダを空にする
```

```
PV SET HEADER(Area;pv_footer_center;$Header) `フッターに渡す
```

```
PV PRINT(Area)
```

⚙️ PV Get print property

PV Get print property (area ; property ; value2) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
property	倍長整数	→	プロパティ番号
value2	文字	←	追加のプロパティの値
戻り値	倍長整数	↩	プロパティの値

説明

`PV Get print property` コマンドは、`area` で指定された 4D View エリアの `property` の現在値を返します。特定の印刷プロパティでは `value2` に追加の情報が返されることがあります。

引数 `property` を指定するには、**PV Print properties** テーマの定数を使用します。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。

Note : “pv print dead...” で始まる 4 つの定数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て:

```
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)
```

PV PRINT (area)

引数	型	説明
area	倍長整数	4D View エリア

説明

PV PRINT コマンドは、*area* で指定された 4D View エリアを印刷します。

PV EXECUTE COMMAND コマンドで *pv cmd print page setup* および *pv cmd print preview* 定数を使用すると、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

フッター設定をして印刷します。

```
PV SET HEADER(Area;pv_footer_center;"#D") `ページフッターを設定
CONFIRM("フォーミュラの値を印刷しますか?";"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
```

PV PRINT FORMULAS

PV PRINT FORMULAS (area)

引数	型		説明
area	倍長整数	⇒	4D View エリア

説明

PV PRINT FORMULAS コマンドは、*area* で指定された 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 コマンドの例題参照](#)

PV Print settings to blob

PV Print settings to blob (area) -> 戻り値

引数	型		説明
area	倍長整数	→	4D Viewエリア
戻り値	BLOB	↻	印刷設定

説明

PV Print settings to blob コマンドは *area* で指定した 4D View エリアの現在の印刷設定を BLOB に格納して返します。BLOB には印刷に使用されるすべての設定が含まれます:

- 用紙設定 (用紙, 向き, 倍率)
- 印刷設定 (印刷枚数, トレイ等)

他方、コマンドは印刷オプションダイアログ中の 4D View に特有の印刷オプション (ページヘッダーや行の繰り返し) などは格納しません。

このコマンドを使用して、プリンターのモデルやアクセス可能な印刷設定に関わらず、4D View エリアの印刷設定を保存できます。返される BLOB をプログラムで変更してはいけません。この BLOB は *PV BLOB TO PRINT SETTINGS* コマンドでのみ利用できます。

PV Print settings to blob コマンドは例えば *PV SET PRINT PROPERTY* コマンドを使用して一時的に印刷設定を変更する前に、現在の設定を保存するために使用できます。印刷終了後、*PV BLOB TO PRINT SETTINGS* コマンドを使用して現在の印刷設定を復帰できます。

PV SET HEADER (area ; header ; string)

引数	型		説明
area	倍長整数	→	4D View エリア
header	倍長整数	→	ヘッダの位置
string	文字	→	ヘッダに設定する文字列

説明

PV SET PRINT HEADERコマンドは、*header*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*string*を設定します。

引数*header*の指定には**PV Headers & footers**テーマの定数を使用します:

定数	型	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

引数*string*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およびPV Get headerコマンドの例題参照

PV SET PRINT PROPERTY

PV SET PRINT PROPERTY (area ; property ; value ; value2)

引数	型		説明
area	倍長整数	⇒	4D Viewエリア
property	倍長整数	⇒	プロパティ番号
value	倍長整数	⇒	プロパティの値
value2	文字	⇒	追加のプロパティ値

説明

PV SET PRINT PROPERTY コマンドは、*area* で指定された 4D View エリアに対して *property* の *value*、および追加の *value2* を設定します。

引数 *property* を指定するには **PV Print properties** テーマの定数を使用します。引数 *property* および *value* 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。

定数	型	値	コメント
pv print adjust area	倍長整数	10	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： <ul style="list-style-type: none"> 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 binding	倍長整数	26	印刷が両面モードで行われる際の、綴じしろの場所を設定あるいは取得するために使用します。関連する値: PV Print values テーマの以下の値: <ul style="list-style-type: none"> pv left binding: 左綴じ (デフォルト値) pv top binding: 上綴じ
pv print bottom margin	倍長整数	3	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
pv print centered	倍長整数	9	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： <ul style="list-style-type: none"> 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 color	倍長整数	23	カラー処理のモードを取得あるいは設定するために使用します。このプロパティはカラープリンターでのみ有効です。関連する値: PV Print values テーマの以下の定数: <ul style="list-style-type: none"> pv black and white: 白黒印刷 (モノクロ) pv color: カラー印刷
pv print dead bottom margin	倍長整数	18	Note: このプロパティはWindowsでのみ使用できます。 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。 Note: 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	倍長整数	15	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。 Note: 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 right margin	倍長整数	17	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。 Note: 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 top margin	倍長整数	16	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。 Note: 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 destination	倍長整数	24	印刷の出力先を取得あるいは設定するために使用します。関連する値: PV Print values テーマの以下の定数: <ul style="list-style-type: none"> pv destination printer: プリントジョブはプリンターに送られます。 pv destination file (Windowsのみ): プリントジョブはファイルに送られます。この定数が使用されると、value2には作成されたドキュメントのパスが返されます。value2に空の文字列を渡すかこの引数を省略すると、印刷時にファイルを保存ダイアログが表示されます。 pv destination PDF file (Mac OSのみ): プリントジョブはPDFファイルに送られます。この定数が使用されると、value2には作成されたPDFドキュメントのパスが返されます。value2に空の文字列を渡すかこの引数を省略すると、印刷時にファイルを保存ダイアログが表示されます。 pv destination EPS file (Mac OSのみ): プリントジョブはEPSファイルに送られます。この定数が使用されると、value2には作成されたEPSドキュメントのパスが返されます。value2に空の文字列を渡すかこの引数を省略すると、印刷時にファイルを保存ダイアログが表示されます。

pv print document name	10 倍 長 整 数 27	<p>スプーラーに表示されるプリントジョブ名を設定あるいは取得するために使用します。この定数が使用されると、value2はプリントジョブ名を含みます。valueには0を渡します。標準の処理 (名前として“4D View”) を使用するには、value2に空の文字列を渡します。</p> <p>片面あるいは両面印刷どちらを使用するか指定します。関連する値:</p>
pv print double sided	25	<ul style="list-style-type: none"> • pv value on: 両面印刷 • pv value off: 片面印刷 (デフォルト値) <p>Note: このプロパティはWindowsでのみ利用できます。</p>
pv print frame each page	11	<p>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:</p> <ul style="list-style-type: none"> • 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	12	<p>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:</p> <ul style="list-style-type: none"> • 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 headers	8	<p>行ヘッダと列ヘッダの印刷を行います。関連する値:</p> <ul style="list-style-type: none"> • 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 left margin	0	<p>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</p>
pv print number copies	21	<p>印刷コピー数を設定あるいは取得するために使用します。関連する値: コピー数 (デフォルトで1)</p>
pv print orientation	19	<p>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。</p> <ul style="list-style-type: none"> • 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 pages from	28	<p>印刷の開始ページを設定あるいは取得するために使用します。関連する値: ページ番号</p>
pv print pages to	29	<p>印刷の終了ページを設定あるいは取得するために使用します。関連する値: ページ番号</p>
pv print paper height	14	<p>用紙の高さを返します。関連する値: ピクセル単位の高さ</p>
pv print paper source	22	<p>使用する用紙トレイを設定あるいは取得するために使用します。関連する値: 4DのPRINT OPTION VALUESコマンドから返されるnamesArrayの要素に対応するinfo1Array要素の値。この配列には使用される用紙トレイの名前が含まれます。注: このプロパティはWindowsでのみ使用できます。</p>
pv print paper width	13	<p>用紙の幅を返します。関連する値: ピクセル単位の幅</p>
pv print	11	<p>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</p>

repeat first column	整数 数	4	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	倍長 整数 数	6	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 column	倍長 整数 数	5	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 last row	倍長 整数 数	7	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 right margin	倍長 整数 数	2	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 scale	倍長 整数 数	20	印刷倍率を設定あるいは取得するために使用します。プリンターによっては倍率の変更が許可されない場合があることに留意してください。無効な値を渡すと、プロパティは印刷時に100%にリセットされます。関連する値：印刷倍率
pv print top margin	倍長 整数 数	1	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

例題

Windowsで用紙トレイを選択する例:

```
ARRAY TEXT ($arrNames;0)
ARRAY LONGINT ($arrInfo1;0)

`利用可能なトレイを取得
PRINT OPTION VALUES (Paper source;$arrNames;$arrInfo1)
```

以下は\$arrNamesと\$arrInfo1配列に返される値の一例です:

\$arrNames	\$arrInfo1
Automatic Selection	15
Tray 1	257
Tray 1 (Manual)	258
Tray 2	259
Tray 3	260
Tray 4	261
Envelope Feeder	262

“Tray 1 (Manual)”を使用したい場合、このトレイに対応するインデックスを渡します:

```
PV SET PRINT PROPERTY(area;pv print paper source;3)
```

PVペイン

ペインコマンドについて

-  PV ADD HOR SPLITTER
-  PV ADD VERT SPLITTER
-  PV FREEZE PANES
-  PV Get hor pane property
-  PV Get vert pane property
-  PV REMOVE HOR SPLITTER
-  PV REMOVE VERT SPLITTER
-  PV SET HOR PANE PROPERTY
-  PV SET VERT PANE PROPERTY
-  PV UNFREEZE PANES

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“ペイン”と呼びます。

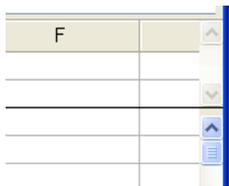
水平ペインは、表示される2つの境界線の間位置するスペースです。例えば、スプレッドシートの上部分、縦スクロールバー上のセパレータ、スプレッドシートの下部分に分かれます。

垂直ペインは、表示される2つのターゲットの間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ペインと垂直ペインがあり、それぞれエリア全体を対象としています。

固定ペインモードを除き、縦横それぞれ同時に複数のペインを持つことができます (後述)。

エリアの**ペインを固定**することができます。ペインが固定化されると、ユーザーがどのようにスプレッドシートをブラウズするかに関わらず、そのエリアは常にスクリーンに表示されます。ユーザーはペインをリサイズしたり削除したりすることはできません。スプレッドシートの他のペインではスクロールしてもその領域にアクセスすることはできません。内容やフォーマットを更新することは可能です。1つの4D Viewエリアは最大2つの固定ペインを持つことができます。エリアの上部に水平ペイン、エリアの左側に垂直ペインです。同じドキュメント内に標準のペインと固定ペインを同時に持つことはできません。1つの4D Viewエリアは"標準ペイン"モードまたは"固定ペインモード"いずれかで動作します。"固定ペイン"モードで動作しているとき、ペインを追加することはできません。新しいスプリッターを追加するには、固定化を解除しなければなりません：



このテーマ内のコマンドを使用すると、4D View エリアの各ペインを操作することができます。例えばペイン (水平または垂直) の追加や削除、ペインのプロパティの取得や設定、エリアのペインを固定化したり解除したりすることなどを行うことができます。

PV ADD HOR SPLITTER

PV ADD HOR SPLITTER (area ; splitter ; position ; locked)

引数	型		説明
area	倍長整数	➡	4D View エリア
splitter	整数	➡	水平方向のセパレータの番号
position	整数	➡	最後のセパレータからの、セパレータの相対位置 (ピクセル単位)
locked	整数	➡	0= ロックしない; 1= ロック

説明

PV ADD HOR SPLITTER コマンドは、*splitter* に指定した番号の水平分割ボックスを新しく *area* 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 *position* (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 *locked* に 1 を指定すると、ペインのサイズは変更できません。0 を指定した場合は、ユーザが自由にペインのサイズを変更することができます。

Notes:

- ペインの位置は列ヘッダーの高さを含みます。列ヘッダーの高さは *PV Get area property* コマンドの二番目の引数に *pv column headers height* 定数を渡すことで取得できます。
- 水平ペインの最小の高さは 8 ピクセルです。
- PV Get area property* コマンドの二番目の引数に *pv hor pane count* 定数を渡すと、水平ペインの数を知ることができます。水平ペインが存在しない場合、**PV Get area property** は 1 を返します。つまり 1 つのペインがエリア全体ということです。

例題

エリアに既存するペインに続けて、30 ピクセルの高さの水平ペインを追加します。

```
C_LONGINT ($HorPaneNum) `既存の水平ペイン数
C_LONGINT ($Position) `ペインの位置

`水平ペイン数
$HorPaneNum:=PV Get area property(Area;pv_hor_pane_count)
$Position:=30 `高さ30ピクセル
PV ADD HOR SPLITTER(Area;$HorPaneNum;$Position;0) `リサイズ可能
```

エラー管理

エリアが固定ペインモードのときに *PV ADD HOR SPLITTER* コマンドが実行されると、エラー 92 (固定ペインにはスプリッターを追加できません) が生成されます。

PV ADD VERT SPLITTER

PV ADD VERT SPLITTER (area ; splitter ; position ; locked)

引数	型		説明
area	倍長整数	➡	4D View エリア
splitter	整数	➡	垂直方向のセパレータの番号
position	整数	➡	最後のセパレータからの、セパレータの相対位置 (ピクセル単位)
locked	整数	➡	0= ロックしない; 1= ロック

説明

PV ADD VERT SPLITTER コマンドは、*splitter* に指定した番号の垂直分割ボックスを新しく *area* に作成します。分割ボックスはエリアの左から *position* (ピクセル単位で指定) に作成されます。

引数 *locked* に 1 を指定すると、ペインのサイズは変更できません。0 を指定した場合は、ユーザが自由にペインのサイズを変更することができます。

Notes:

- ペインの位置は行ヘッダーの幅を含みます。行ヘッダーの幅は **PV Get area property** コマンドの二番目の引数に *pv row headers width* 定数を渡すことで取得できます。
- 垂直ペインの最小の幅は8ピクセルです。
- PV Get area property** コマンドの二番目の引数に *pv vert pane count* 定数を渡すと、垂直ペインの数を知ることができます。垂直ペインが存在しない場合、**PV Get area property** は1を返します。つまり1つのペインがエリア全体ということです。

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を表示する垂直ペインを作成してみましょう。

```
C_LONGINT($ColumnWidth) `A列の幅 (ピクセル)
$ColumnWidth:=PV Get column width(Area;1) `A列
PV ADD VERT SPLITTER(Area;1;$ColumnWidth;0) `リサイズ可能
```

エラー管理

エリアが固定ペインモードのときに **PV ADD VERT SPLITTER** コマンドが実行されると、エラー92 (固定ペインにはスプリッターを追加できません) が生成されます。

PV FREEZE PANES (area ; mode)

引数	型	説明
area	倍長整数	→ 4D Viewエリア
mode	倍長整数	→ 0=スクロールをロック, 1=スクロールとヘッダーの変更をロック

説明

PV FREEZE PANES コマンドは *area* で指定された 4D View エリアの最上部の水平ペインおよび最左部の垂直ペインを固定化します。コマンドが動作するためには、*area* は 1 つの縦スプリッターあるいは横スプリッター、またはその両方があり、かつ非固定モードでなければなりません。

mode 引数を使用して、エリアに適用するロックのタイプを指定します：

- *mode* = 0: ペイン中のスクロールだけがロックされます。これは表示メニューからペインをフリーズコマンドを使用して実行されるロックです。
- *mode* = 1: ロックが拡張されます。スクロールに加えヘッダーの更新 (スタイル、フォントサイズ等) もロックされ、ペインのセルをクリックするとすべての列/行が選択されます。

area にペインスプリッターが含まれていないか、2 つ以上の水平または垂直ペインスプリッターが含まれている場合、コマンドはなにも行わず、OK システム変数が 0 に設定されます。

コマンドが正しく実行されると OK システム変数に 1 が設定され、表示メニューのペインをフリーズコマンドが使用不可になり、ペインのフリーズ解除コマンドが有効になります。

エリアを標準ペインモードに戻すには、*PV UNFREEZE PANES* コマンドを実行するか、ペインのフリーズ解除メニューコマンドを選択します。

例題

以下のコードはエリアに 2 つの水平ペインが含まれているかどうかを確認し、そうであれば拡張モードで固定化します：

```
$nbpanes:=PV Get area property(area;pv hor pane count)
If ($nbpanes=2)
    PV FREEZE PANES(area;1)
End if
```

システム変数およびセット

area にペインスプリッターが含まれていないか、2 つ以上の水平または垂直ペインスプリッターが含まれている場合、コマンドはなにも行わず、OK システム変数が 0 に設定されます。コマンドが正しく実行されると OK システム変数に 1 が設定されます。

🔧 PV Get hor pane property

PV Get hor pane property (area ; pane ; property) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
pane	整数	→	水平ペインの番号
property	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 コマンドは、*area* 中で *pane* に渡された番号の水平ペインに関する *property* の現在値を返します。
property を指定するには **PV Pane properties** テーマの定数を使用します。

定数	型	値	コメント
pv pane lock scrollbar	倍長整数	0	このプロパティは固定ペインモードのエリアでは無効です。
pv pane lock splitter	倍長整数	1	このプロパティは固定ペインモードのエリアでは無効です。
pv pane view splitter cursor	倍長整数	2	このプロパティは固定ペインモードのエリアでは無効です。
pv pane size in pixels	倍長整数	3	このプロパティは固定ペインモードのエリアでは無効です。
pv pane first row	倍長整数	4	
pv pane rows count	倍長整数	5	
pv pane true scroll	倍長整数	6	スクロールカーソルの現在位置に関わらず、エリアの基点 (例えば最初のセル) からピクセル値を使用してペインの内容をスクロールするために使用します。このプロパティはエリアが固定ペインモードのときは無効です。

例題

PV SET HOR PANE PROPERTY コマンドの例題参照

🔧 PV Get vert pane property

PV Get vert pane property (area ; pane ; property) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
pane	整数	→	垂直ペインの番号
property	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 コマンドは、*area* 中で *pane* に渡された番号の水平ペインに関する *property* の現在値を返します。

property を指定するには **PV Pane properties** テーマの定数を使用します:

定数	型	値	コメント
pv pane lock scrollbar	倍長整数	0	このプロパティは固定ペインモードのエリアでは無効です。
pv pane lock splitter	倍長整数	1	このプロパティは固定ペインモードのエリアでは無効です。
pv pane view splitter cursor	倍長整数	2	このプロパティは固定ペインモードのエリアでは無効です。
pv pane size in pixels	倍長整数	3	このプロパティは固定ペインモードのエリアでは無効です。
pv pane first column	倍長整数	4	
pv pane columns count	倍長整数	5	
pv pane true scroll	倍長整数	6	スクロールカーソルの現在位置に関わらず、エリアの基点 (例えば最初のセル) からピクセル値を使用してペインの内容をスクロールするために使用します。このプロパティはエリアが固定ペインモードのときは無効です。

例題

PV SET VERT PANE PROPERTY コマンドの例題参照

PV REMOVE HOR SPLITTER

PV REMOVE HOR SPLITTER (area ; splitter)

引数	型		説明
area	倍長整数	⇒	4D View エリア
splitter	整数	⇒	水平ペインの番号

説明

PV REMOVE HOR SPLITTER コマンドは、*splitter* に渡した番号の水平スプリッターを *area* 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ペインを削除することができます。

Note: フリーズされたエリアのコンテキストで *PV REMOVE HOR SPLITTER(area;1)* が実行されると、固定ペインは取り除かれ、エリアは標準ペインモードとなります。

例題

最後の水平ペインを削除します:

```
C_LONGINT ($HorPaneNum) `水平ペイン数
`水平ペイン数
$HorPaneNum:=PV Get area property(Area;pv_hor_pane_count)
PV REMOVE HOR SPLITTER(Area;$HorPaneNum)
```


PV SET HOR PANE PROPERTY

PV SET HOR PANE PROPERTY (area ; pane ; property ; value)

引数	型		説明
area	倍長整数	→	4D View エリア
pane	整数	→	水平ペインの番号
property	倍長整数	→	プロパティ番号
value	倍長整数	→	プロパティの値

説明

`PV SET HOR PANE PROPERTY` コマンドは、`area` において `pane` に渡された番号の水平ペインに関する `property` の値を `value` に設定します。

`property` を指定するには **PV Pane properties** テーマの定数を使用します：

定数	型	値	コメント
pv pane lock scrollbar	倍長整数	0	このプロパティは固定ペインモードのエリアでは無効です。
pv pane lock splitter	倍長整数	1	このプロパティは固定ペインモードのエリアでは無効です。
pv pane view splitter cursor	倍長整数	2	このプロパティは固定ペインモードのエリアでは無効です。
pv pane size in pixels	倍長整数	3	このプロパティは固定ペインモードのエリアでは無効です。
pv pane first row	倍長整数	4	
pv pane rows count	倍長整数	5	
pv pane true scroll	倍長整数	6	スクロールカーソルの現在位置に関わらず、エリアの基点 (例えば最初のセル) からピクセル値を使用してペインの内容をスクロールするために使用します。このプロパティはエリアが固定ペインモードのときは無効です。
pv pane relative scroll	倍長整数	7	プロパティは "PV SET..." コマンドでのみ使用できます。スクロールカーソルの現在位置を基点とし、ピクセル値を使用してペインの内容をスクロールするために使用します。一番上の行が完全に表示されるように、ピクセルでのスクロールは調整されます。このプロパティはエリアが固定ペインモードのときは無効です。

例題

先頭の水平ペインの高さを30ピクセルだけ上げます。

```
C_LONGINT ($Size) ` 最上部の水平ペインの高さ
```

```
$Size:=PV Get hor pane property(Area;1;pv pane size in pixels)
PV SET HOR PANE PROPERTY(Area;1;pv pane size in pixels;$Size+30) `30ピクセル加える
```

PV SET VERT PANE PROPERTY

PV SET VERT PANE PROPERTY (area ; pane ; property ; value)

引数	型		説明
area	倍長整数	⇒	4D View エリア
pane	整数	⇒	垂直ペインの番号
property	倍長整数	⇒	プロパティ番号
value	倍長整数	⇒	プロパティの値

説明

PV SET VERT PANE PROPERTY コマンドは、*area* において *pane* に渡された番号の垂直ペインに関する *property* の値を *value* に設定します。

property を指定するには **PV Pane properties** テーマの定数を使用します:

定数	型	値	コメント
pv pane lock scrollbar	倍長整数	0	このプロパティは固定ペインモードのエリアでは無効です。
pv pane lock splitter	倍長整数	1	このプロパティは固定ペインモードのエリアでは無効です。
pv pane view splitter cursor	倍長整数	2	このプロパティは固定ペインモードのエリアでは無効です。
pv pane size in pixels	倍長整数	3	このプロパティは固定ペインモードのエリアでは無効です。
pv pane first column	倍長整数	4	
pv pane columns count	倍長整数	5	
pv pane true scroll	倍長整数	6	スクロールカーソルの現在位置に関わらず、エリアの基点 (例えば最初のセル) からピクセル値を使用してペインの内容をスクロールするために使用します。このプロパティはエリアが固定ペインモードのときは無効です。
pv pane relative scroll	倍長整数	7	プロパティは "PV SET..." コマンドでのみ使用できます。スクロールカーソルの現在位置を基点とし、ピクセル値を使用してペインの内容をスクロールするために使用します。一番上の行が完全に表示されるように、ピクセルでのスクロールは調整されます。このプロパティはエリアが固定ペインモードのときは無効です。

例題 1

先頭の垂直ペインの幅を30ピクセルだけ広げます。

```
C_LONGINT($Size) `先頭の垂直ペインの幅
$Size:=PV Get vert pane property(Area;1;pv pane size in pixels)
```

```
PV SET VERT PANE PROPERTY(Area;1;pv pane size in pixels;$Size+30) `30ピクセル追加
```

例題 2

先頭のペインを基点から50ピクセルスクロールします。

```
PV SET VERT PANE PROPERTY(Area;1;pv pane true scroll;50)
```

PV UNFREEZE PANES (area)

引数	型	説明
area	倍長整数	4D View エリア

説明

PV UNFREEZE PANES コマンドは *area* で指定した 4D View エリアを、ペインの追加や削除およびリサイズが可能な標準ペインモードに変更します。

このコマンドが動作するためには、エリアのペインが *PV FREEZE PANES* コマンドあるいは表示メニューのペインをフリーズを使用して、固定されていなければなりません。

area にペインブリッターが含まれていないか固定されていない場合、コマンドはなににも行わず、OKシステム変数が0に設定されます。コマンドが正しく実行されるとOKシステム変数に1が設定され、表示メニューのペインのフリーズ解除コマンドが使用不可になり、ペインをフリーズコマンドが有効になります。

例題

以下の例題ではエリアが固定されていれば、固定を解除します：

```
PV GET COMMAND STATUS(area;pv_cmd_unfreeze_panes;$status;$check;$name)
If($status=1)
    PV UNFREEZE PANES(area)
End if
```

システム変数およびセット

area にペインブリッターが含まれていないか固定されていない場合、コマンドはなににも行わず、OKシステム変数が0に設定されます。コマンドが正しく実行されるとOKシステム変数に1が設定されます。

PV入力認可

 入力認可コマンドについて

 PV GET ALLOWED COM LIST

 PV GET ALLOWED MET LIST

 PV GET ALLOWED VAR LIST

 PV SET ALLOWED COM LIST

 PV SET ALLOWED MET LIST

 PV SET ALLOWED VAR LIST

✚ 入力認可コマンドについて

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 GET ALLOWED COM LIST

PV GET ALLOWED COM LIST (area ; arrayCom)

引数	型		説明
area	倍長整数	→	4D View エリア
arrayCom	文字配列	←	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*arrayCom*に返します。

PV GET ALLOWED MET LIST

PV GET ALLOWED MET LIST (area ; arrayMet)

引数	型		説明
area	倍長整数	⇒	4D View エリア
arrayMet	文字配列	⇐	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 *arrayMet* に返します。

PV GET ALLOWED VAR LIST

PV GET ALLOWED VAR LIST (area ; arrayVar)

引数	型		説明
area	倍長整数	→	4D View エリア
arrayVar	文字配列	←	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 *arrayVar* に返します。

PV SET ALLOWED COM LIST

PV SET ALLOWED COM LIST (area ; arrayCom)

引数	型		説明
area	倍長整数	→	4D View エリア
arrayCom	文字配列	→	4Dコマンド名の配列

説明

PV SET ALLOWED COM LIST コマンドは、*area*内のフォーミュラで許可される4Dコマンドのリストを*arrayCom*に設定します。

Note : このコマンドを有効にするためには、入力認可システムがアクティブでなければなりません。[入力認可コマンドについて](#)を参照してください。

PV SET ALLOWED MET LIST

PV SET ALLOWED MET LIST (area ; arrayMet)

引数	型		説明
area	倍長整数	⇒	4D View エリア
arrayMet	文字配列	⇒	メソッド名の配列

説明

PV SET ALLOWED MET LIST コマンドは、*area*内のフォーミュラで許可される4Dメソッドのリストを*arrayMet*に設定します。

Note : このコマンドを有効にするためには、入力認可システムがアクティブでなければなりません。[入力認可コマンドについて](#)を参照してください。

PV SET ALLOWED VAR LIST

PV SET ALLOWED VAR LIST (area ; arrayVar)

引数	型		説明
area	倍長整数	→	4D View エリア
arrayVar	文字配列	→	変数名の配列

説明

PV SET ALLOWED VAR LIST コマンドは、*area*内のフォーミュラで許可される変数 (プロセスおよびインタープロセス変数のみ) のリストを*arrayVar*に設定します。

Note : このコマンドを有効にするためには、入力認可システムがアクティブでなければなりません。[入力認可コマンドについて](#)を参照してください。

PV列と行

列と行コマンドについて

-  PV DELETE CELLS
-  PV DELETE COLUMNS
-  PV DELETE ROWS
-  PV GET COLUMN HEADER
-  PV Get column width
-  PV Get row header
-  PV Get row height
-  PV INSERT CELLS
-  PV INSERT COLUMNS
-  PV INSERT ROWS
-  PV SET COLUMN HEADER
-  PV SET COLUMNS WIDTH
-  PV SET ROW HEADER
-  PV SET ROWS HEIGHT

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列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）
- 列と行のヘッダの読み込みと設定（行タイトルと列タイトル、および列の並び替えオプション）

行とカラムの名前

4D View エリアの行や列に名前を関連付けることができ、この名前を使用するとドキュメント内での行と列の取り扱いがより容易になります。デフォルトでそれぞれの行と列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列は自動的に命名されます：

- **行:** 名前は実際の行番号に対応します。
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。
- **列:** 文字を使用して列名が付けられます。エリアのプロパティに応じ、列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）。
列番号を呼び出すコマンドに対して、列番号と列名との取り決めはデフォルトで以下の通りになります。：

列名	列番号
A	1
B	2
C	3
[...]	
Y	25
Z	26
AA	27
AB	28
AC	29
etc.	

PV DELETE CELLS

PV DELETE CELLS (area ; column ; row ; number ; direction)

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	列番号
row	倍長整数	→	行番号
number	倍長整数	→	削除するセル数
direction	倍長整数	→	セルをシフトする方向

説明

PV DELETE CELLSコマンドは、*column* と*row*で指定したセルより開始して、*area* 内のセルを*number* の数だけ削除します。

引数*direction* 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 または を使用します：

定数	型	値
pv to the left	倍長整数	2
pv to the top	倍長整数	3

Note: このコマンドを動的エリアで呼び出してはいけません。これを行うとエラー86が生成されます。

例題

次の例題では、2 列目の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(area;2;2;1;pv to the top)
```

PV DELETE COLUMNS

PV DELETE COLUMNS (area ; start ; number)

引数	型		説明
area	倍長整数	⇒	4D View エリア
start	倍長整数	⇒	開始列番号
number	倍長整数	⇒	列数

説明

PV DELETE COLUMNSコマンドは、*start*列番号から始めて、*area* の*number* の数の列を削除します。

Note: このコマンドを動的エリアで呼び出してはいけません。これを行うとエラー86が生成されます。

例題

表の1 番目の列 (A) を削除し、残りの列エリアの内容を左側にシフトします:

```
PV DELETE COLUMNS(Area;1;1)
```

PV DELETE ROWS

PV DELETE ROWS (area ; start ; number)

引数	型		説明
area	倍長整数	→	4D View エリア
start	倍長整数	→	開始行番号
number	倍長整数	→	行数

説明

PV DELETE ROWSコマンドは、*start*行番号から始めて、*area* の*number* の数の行を削除します。

Note: このコマンドを動的エリアで呼び出してはいけません。これを行うとエラー86が生成されます。

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)
```

PV GET COLUMN HEADER

PV GET COLUMN HEADER (area ; column ; title)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	列番号
title	文字	⇐	列名

説明

PV GET COLUMN HEADERコマンドは、指定された`column`の現在の`title`を取得します。
行と列のデフォルト名に関する詳細は[列と行コマンドについて](#)を参照してください。

例題

*PV SET COLUMN HEADER*コマンドの例題参照

PV Get column width

PV Get column width (area ; column) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
column	倍長整数	→	列番号
戻り値	整数	↩	列幅 (ピクセル単位)

説明

PV Get column width コマンドは、指定された *column* の幅 (ピクセル単位) を返します。

例題

PV SET COLUMNS WIDTH および *PV ADD VERT SPLITTER* コマンドの例題参照

PV Get row header

PV Get row header (area ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
row	倍長整数	→	行番号
戻り値	文字	↩	行の名前

説明

PV Get row header コマンドは、指定された *row* の現在の名前を返します。
行と列のデフォルト名に関する詳細は [列と行コマンドについて](#) を参照してください。

例題

PV SET COLUMN HEADER コマンドの例題参照

🔧 PV Get row height

PV Get row height (area ; row) -> 戻り値

引数	型		説明
area	倍長整数	→	4D View エリア
row	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コマンドは、指定されたrowの高さ (ピクセル単位) を返します。

例題 1

PV SET COLUMNS WIDTHコマンドの例題参照

例題 2

4D バージョン 2004.5より、**Print form**コマンドを使用して4D Viewエリアを印刷できるようになりました。一般的にこれらのエリアは固定サイズで印刷されます。以下の例題では、4Dの印刷コマンドとPV Get row heightコマンドを使用して、内容に応じ、4D Viewエリアの印刷の高さを変更する方法を示します。

- これは **Print form** コマンドで呼び出されるフォームのフォームメソッドです:

```
If (Form event=On Printing Detail)
  GET OBJECT RECT (4DViewarea; $left; $top; $right; $bottom)
  $posmarker := Get print marker (Form detail)
  $areaheight := $bottom - $top
  $newheight := Get4DViewSizeCalcul
  ` Get 4DViewSizeCalcul は内容に応じて4D Viewエリアの高さを返します
  $offset := $newheight - $areaheight
  MOVE OBJECT (4DViewarea; 0; 0; 0; $offset)
  SET PRINT MARKER (Form detail; $posmarker + $offset)
End if
```

- Get4DViewSizeCalcul**メソッドは以下の通りです:

```
$area := PV New offscreen area
PV BLOB TO AREA ($area; [Table 1]View_)
PV EXECUTE COMMAND ($area; pv cmd edit go to last cell)
PV GET CURRENT CELL ($area; $column; $row)
$height := 0
For ($i; 1; $row)
  $rowHeight := PV Get row height ($area; $i)
  $height := $height + $rowHeight
End for
PV DELETE OFFSCREEN AREA ($area)
$0 := Trunc ($height * 0.75; 0)
```

PV INSERT CELLS (area ; column ; row ; number ; direction)

引数	型		説明
area	倍長整数	➡	4D View エリア
column	倍長整数	➡	列番号
row	倍長整数	➡	行番号
number	倍長整数	➡	挿入するセル数
direction	倍長整数	➡	挿入する方向

説明

PV INSERT CELLSコマンドは、*column* と*row*で指定したセルより開始して、*area* に*number* の数のセルを挿入します。引数*direction* 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またはを使用します：

定数	型	値
pv to the bottom	倍長整数	1
pv to the right	倍長整数	0

Note: このコマンドを動的エリアで呼び出してはいけません。これを行うとエラー86が生成されます。

例題

次の例題では、1 列目の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)
```

PV INSERT COLUMNS

PV INSERT COLUMNS (area ; start ; number)

引数	型		説明
area	倍長整数	⇒	4D View エリア
start	倍長整数	⇒	新しい列の挿入箇所
number	倍長整数	⇒	列数

説明

PV INSERT COLUMNS コマンドは、*start* 列番号から開始して *area* に *number* の数の列を挿入します。列は、*start* で指定した列の直前に挿入されます。

Note: このコマンドを動的エリアで呼び出してはいけません。これを行うとエラー86が生成されます。

例題

表の1 番目の列 (A) の前に列を1 つ挿入し、列エリアの内容を右側にシフトします:

```
PV INSERT COLUMNS(Area;1;1)
```

PV INSERT ROWS

PV INSERT ROWS (area ; start ; number)

引数	型		説明
area	倍長整数	→	4D View エリア
start	倍長整数	→	新しい行の挿入箇所
number	倍長整数	→	行数

説明

PV INSERT ROWS コマンドは、*start* 行番号から開始して *area* に *number* の数の行を挿入します。行は、*start* で指定した行の直前に挿入されます。

Note: このコマンドを動的エリアで呼び出してはいけません。これを行うとエラー86が生成されます。

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)
```

⚙️ PV SET COLUMN HEADER

PV SET COLUMN HEADER (area ; column ; title)

引数	型		説明
area	倍長整数	⇒	4D View エリア
column	倍長整数	⇒	列番号
title	文字	⇒	列名

説明

`PV SET COLUMN HEADER`コマンドは、指定された`column`の`title`を設定します。
行と列のデフォルト名に関する詳細は[列と行コマンドについて](#)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列と10行に新しい名前を割り当てます。

```
C_INTEGER($Index) `ループのインデックス
C_STRING(80;$Title) `列/行の名前

For($Index;1;10)
  PV GET COLUMN HEADER(Area;$Index;$Title) ` $Index 列目の名前を取得する
  $Title:="Column"+$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
```

🔧 PV SET COLUMNS WIDTH

PV SET COLUMNS WIDTH (area ; first ; last ; width)

引数	型		説明
area	倍長整数	→	4D View エリア
first	倍長整数	→	最初の列番号
last	倍長整数	→	最後の列番号
width	整数	→	列幅 (ピクセル単位)

説明

PV SET COLUMNS WIDTH コマンドを使用して、*area* 内の *first* と *last* の間にある列の幅 (ピクセル単位) を *width* に変更することができます。

Note : 引数 *first* と *last* に 0 (ゼロ) を渡すと、指定した *width* が *area* の全列に対して適用され、この値がそのエリアの列の新しいデフォルト幅になります。ユーザが列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、行と列のサイズを変更する方法を説明します:

```
C_LONGINT ($Column; $Row) ` 主要ループのインデックス
C_LONGINT ($Width; $Height) ` 列と行を拡張するループのインデックス
C_LONGINT ($StartWidth) ` X 番目の列の元の幅
C_LONGINT ($RequestWidth) ` X 番目の列に求められる幅

` 初期化を行う
$StartWidth:=5 ` 元の幅を設定する
$RequestWidth:=$StartWidth+5 ` 最初の列を10 ポイントの幅に設定

For ($Column; 1; 5) ` 最初の5列を処理する
  $Row:=$Column ` 最初の5行だけを処理する
  For ($Width; $StartWidth; $RequestWidth; 2) ` "$Column"... 番目の列
    $Height:=$Width-5 ` 高さを更新
    PV SET ROWS HEIGHT(area; $Row; $Row; PV Get row height(area; $Row)+$Height) ` 1 行
    PV SET COLUMNS WIDTH(area; $Column; $Column; PV Get column width(area; $Row)+$Width) ` 1 列
    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 ` 次の列を拡張する
End for
```

PV SET ROW HEADER

PV SET ROW HEADER (area ; row ; title)

引数	型		説明
area	倍長整数	⇒	4D View エリア
row	倍長整数	⇒	行番号
title	文字	⇒	行の名前

説明

PV SET ROW HEADER コマンドは、指定された *row* の *title* を設定します。
行と列のデフォルト名に関する詳細は [列と行コマンドについて](#) を参照してください。

例題

PV SET COLUMN HEADER コマンドの例題参照

PV SET ROWS HEIGHT

PV SET ROWS HEIGHT (area ; first ; last ; height)

引数	型		説明
area	倍長整数	→	4D View エリア
first	倍長整数	→	最初の行番号
last	倍長整数	→	最後の行番号
height	整数	→	行の高さ (ピクセル単位)

説明

PV SET ROWS HEIGHT コマンドは、*area* 内の *first* と *last* の間にある行の高さ (ピクセル単位) を *height* に設定します。

Note: 引数 *first* と *last* 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 コマンドの例題参照

PV罫線

 罫線コマンドについて

 PV GET BORDER ROW RANGES

 PV GET BORDER COLUMN RANGES

 PV GET BORDER STYLE

 PV SET BORDER STYLE

 PV SET RANGE BORDER

枠線コマンドについて

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[セレクトコマンドについて](#)を参照してください。

PV GET BORDER ROW RANGES (area ; left ; top ; right ; bottom ; borderTypes ; borderColors)

引数	型		説明
area	倍長整数	→	4D View エリア
left	倍長整数配列	←	左端のセルの列番号の配列
top	倍長整数配列	←	上端のセルの行番号の配列
right	倍長整数配列	←	右端のセルの列番号の配列
bottom	倍長整数配列	←	下端のセルの行番号の配列
borderTypes	倍長整数配列	←	境界線型の配列
borderColors	倍長整数配列	←	境界線カラーの配列

説明

PV GET BORDER ROW RANGES コマンドは、上側の境界線スタイルが同じセルの範囲を表すリストを返します(境界線スタイルは型とカラーから構成されます)。

それぞれの範囲は、*left*、*top*、*right* そして *bottom* という4つの同期した配列を通して返されます。各配列の要素は、それぞれ範囲の左端、上端、右端、下端のセル番号を表します。

それぞれに対応する境界線スタイルは、*borderTypes* と *borderColors* 配列引数に返されます:

- borderTypes* 引数の値は、**PV Border style** テーマの定数と比較することが可能です:

定数	型	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 half	倍長整数	14
pv border style none	倍長整数	0
pv border style quarter	倍長整数	13

- borderColors* 引数の値は、BGR型の倍長整数です。詳細な情報に関しては、**PV Color to index** コマンドを参照して下さい。

このコマンドは、**PV GET BORDER COLUMN RANGES** コマンドと合わせて使用する事でエリア内の境界線の完全な定義を取得することができます。これは、4D Viewエリアを例えばMS Excelフォーマットなどに書き出すときに便利です。

注: コマンドによって返される範囲のリストは、その範囲がどのように定義されたかによって異なってきます。例えば、範囲(4A;4E)を選択している状態で水平な線を描画した場合、コマンドはその範囲(4A;4E)に対応した単一の値のみを返します。それに対し、行4に対して水平な線を描画し、それを列A~E に対してループさせた場合、コマンドはそれぞれの描画に対応して5つの値を返します。描画の結果は見た目上では全く同じですが、内部で保存されている情報は異なるという事です。

例題

エリア内に、以下の様に枠線が配置されていたとします:

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

このとき、以下のコードを実行すると:

```
PV GET BORDER ROW  
RANGES(myArea;LeftArray;TopArray;RightArray;BottomArray;BorderTypeArray;BorderColorArray)
```

四つの範囲が検出され、その結果配列内には以下の様な値が格納されて返されます:

LeftArray	TopArray	RightArray	BottomArray	BorderTypeArray	BorderColorArray
2	3	3	3	3	15597568
2	5	3	5	3	15597568
3	8	4	8	9	39168
2	9	4	9	2	255

PV GET BORDER COLUMN RANGES

PV GET BORDER COLUMN RANGES (area ; left ; top ; right ; bottom ; borderTypes ; borderColors)

引数	型		説明
area	倍長整数	→	4D Viewのエリア
left	倍長整数配列	←	範囲の左端のセルの列番号の配列
top	倍長整数配列	←	範囲の上端のセルの行番号の配列
right	倍長整数配列	←	範囲の右端のセルの列番号の配列
bottom	倍長整数配列	←	範囲の下端のセルの行番号の配列
borderTypes	倍長整数配列	←	境界線タイプの配列
borderColors	倍長整数配列	←	境界線カラーの配列

説明

PV GET BORDER COLUMN RANGES コマンドは、左側の境界線スタイルが同じセルの範囲を表すリストを返します(境界線スタイルは型とカラーから構成されます)。

それぞれの範囲は、*left*、*top*、*right* そして *bottom* という4つの同期した配列を通して返されます。各配列の要素は、それぞれ範囲の左端、上端、右端、下端のセル番号を表します。

それぞれに対応する境界線スタイルは、*borderTypes* と *borderColors* 配列引数に返されます:

- *borderTypes* 引数の値は、**PV Border style** テーマの定数と比較することが可能です:

定数	型	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 half	倍長整数	14
pv border style none	倍長整数	0
pv border style quarter	倍長整数	13

- *borderColors* 引数の値は、BGR型の倍長整数です。詳細な情報に関しては、**PV Color to index** コマンドを参照して下さい。

このコマンドは、**PV GET BORDER ROW RANGES** コマンドと合わせて使用する事でエリア内の境界線の完全な定義を取得することができます。これは、4D Viewエリアを例えばMS Excelフォーマットなどに書き出すときに便利です。

注: コマンドによって返される範囲のリストは、範囲の定義の仕方によって異なります。範囲(1B;5B)を選択した状態で垂直の線を引いた場合、このコマンドはその範囲(1B;5B)に対応した単一の値のみを返します。しかし、B列に垂直の線を引き、それを行1~5に繰り返した場合、コマンドはそれぞれの反復に対応した、5つの値を返します。表示される結果は全く同じですが、エリアに保存されている情報は異なります。

例題

エリア内に、以下の様に枠線が配置されていたとします:

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

このとき、以下のコードを実行すると:

```
PV GET BORDER COLUMN
RANGES(myArea;LeftArray;TopArray;RightArray;BottomArray;BorderTypeArray;BorderColorArray)
```

二つの範囲が検出され、その結果配列内には以下の様な値が格納されて返されます:

LeftArray	TopArray	RightArray	BottomArray	BorderTypeArray	BorderColorArray
2	3	2	4	3	15597568
4	3	4	4	3	15597568

PV GET BORDER STYLE

PV GET BORDER STYLE (area ; edge ; style ; color)

引数	型		説明
area	倍長整数	→	4D View エリア
edge	倍長整数	→	枠線の辺
style	倍長整数	←	枠線のスタイル
color	倍長整数	←	枠線の色

説明

PV GET BORDER STYLEコマンドは、4D View areaの枠線のedge に設定されたstyle とcolor を取得します。

引数edge の指定には、**PV Border edge**定数を使用します。

定数	型	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

引数style に返される値は、**PV Border style**定数の値に相当します。

定数	型	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 half	倍長整数	14
pv border style none	倍長整数	0
pv border style quarter	倍長整数	13

詳細はPV SET BORDER STYLEコマンドの説明を参照してください。

例題

4D Viewエリア内の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; $Color) `スタイル属性</p>`スタイル情報を取得
PV GET BORDER STYLE(Area;pv_border_edge_bottom;$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
`枠のスタイルと色
```

```
PV SET BORDER STYLE(Area;pv border edge bottom;pv border style 111;PV Index to color(Light  
blue))  
End if
```

PV SET BORDER STYLE (area ; edge ; style ; color)

引数	型		説明
area	倍長整数	➡	4D View エリア
edge	倍長整数	➡	枠線の辺
style	倍長整数	➡	枠線のスタイル
color	倍長整数	➡	枠線の色

説明

PV SET BORDER STYLEコマンドは、枠線の`edge`に対して`style` と`color` を設定します。

Note: 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`edge` の指定には**PV Border edge**定数を使用します:

```
[#table_kst them="94872"/]
```

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`style` の指定には**PV Border style**定数を使用します:

```
[#table_kst them="94904"/]
```

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イラストで説明します:

	Untouched	None	
<code>pv border style 1</code>			<code>pv border style 2</code>
<code>pv border style 3</code>			<code>pv border style 4</code>
<code>pv border style 5</code>			<code>pv border style 6</code>
<code>pv border style 111</code>			<code>pv border style 211</code>
<code>pv border style 112</code>			<code>pv border style 212</code>
<code>pv border style 222</code>			<code>pv border style 232</code>

- 定数`pv border style quarter` と `pv border style half` はそれぞれ枠線のサイズを0.25ピクセルと0.5ピクセルに設定するために使用されます。これらの値は印刷時にのみ有効です。

引数 `color` は、BGR 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。また4D ランゲージColorsテーマの定数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。

4Dで使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:

- **OBJECT SET RGB COLORS** : 4Dが使用するRGBカラーのシステム用
- **OBJECT SET COLOR** : 4Dパレットのインデックスカラー用

注: 4D View カラーの内部コーディングはBGR型で、4DのRGBコーディングに合わせるために反対にされます。しかしながら、使用する際の原理は同じです。

例題

PV SET RANGE BORDER

PV SET RANGE BORDER (area ; left ; top ; right ; bottom)

引数	型		説明
area	倍長整数	→	4D View エリア
left	倍長整数	→	左側のセルの列番号
top	倍長整数	→	上側のセルの行番号
right	倍長整数	→	右側のセルの列番号
bottom	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 *left*、*top*、*right*、*bottom* で定義します。

		Left=2	Right=3	
	A	B	C	D
	1			
	2			
	3			
Top=4	4			
	5			
	6			
Bottom=8	7			
	8			
	9			
	10			
	11			
	12			

領域に関する詳細は [セクションコマンドについて](#) を参照してください。

例題

4D View エリア上で、列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
```

定数テーマリスト

-  PV Area properties
-  PV Arrow keys
-  PV Border edge
-  PV Border style
-  PV Carriage return
-  PV Cell properties
-  PV Cell value type
-  PV Commands
-  PV Control
-  PV Directions
-  PV Document format
-  PV Document properties
-  PV Drag drop allowed
-  PV Drop action
-  PV Drop info
-  PV Drop mode
-  PV Event
-  PV Header sort
-  PV Headers & footers
-  PV Input enter key mode
-  PV Pane properties
-  PV Picture mapping mode
-  PV Picture properties
-  PV Plugin properties
-  PV Print properties
-  PV Print values
-  PV Report functions
-  PV Select mode
-  PV Selection action
-  PV Style format date time
-  PV Style properties
-  PV Style special values
-  PV Style values
-  PV Triggers

PV Area properties

定数	型	値	コメント
pv allow undo redo	倍長整数	39	<p>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：</p> <ul style="list-style-type: none"> • 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）。
pv arrow keys	倍長整数	9	<p>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、入力が確定されます。関連する値：PV Arrow keysテーマ内の定数。</p> <ul style="list-style-type: none"> • 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 carriage return	倍長整数	8	<p>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。</p> <ul style="list-style-type: none"> • 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 column headers height	倍長整数	21	<p>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）</p>
pv copy hidden	倍長整数	19	<p>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：</p> <ul style="list-style-type: none"> • 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 current cell highlight	倍長整数	3	<p>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：</p> <ul style="list-style-type: none"> • 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 drag allowed	倍長整数	13	<p>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。</p> <ul style="list-style-type: none"> • 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。
			<p>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</p>

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。

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 drag
trigger

倍
長
整
数

12

- [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。

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 drop
allowed

倍
長
整
数

33

- [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。
- [pv DD 4D objects](#): 4Dオブジェクトをドロップできる。4Dフィールド (BLOBとサブテーブルを除く) フィールドと、(BLOBを除く) 変数をドロップできます。

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。

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
mode

倍
長
整
数

14

- [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 field

倍
長

17

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。

tag	整数	17	例: "E1Field1", "E1Field2", "E1Field3"; "E2Field4", "E2Field5"; コンマがフィールド区切りです。
pv field wrapper	倍長整数	18	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。 列ヘッダーがクリックされたときに、(動的あるいは静的なデータの) 標準のソートを行うか禁止するかを設定します。 . 関連する値: PV Header sort テーマの以下の値:
pv headers sort	倍長整数	20	<ul style="list-style-type: none"> • pv sort not allowed (デフォルト値): ユーザーが列ヘッダをクリックした際に、4D Viewは標準のソートを行わない (その代わりに開発者によるカスタマイズされたソートを実行できます) • pv sort allowed: ユーザーが列ヘッダをクリックした際に、4D Viewは標準のソートを実行する。この場合、ソート順を示すマークがヘッダーに表示されます。次に久利いくすると昇順と降順が入れ替わります。 <p>動的な列をソートすると、他の列も同期してソートされます。結果レコードの並びを保つことができます。静的な列のソートはその列のみをソートします。</p>
pv hor pane count	倍長整数	11	エリアの横方向のペイン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のみ読み込まれます。戻り値: ペイン数。 Enter キー (数値キーパッド) 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 input enter key mode	倍長整数	15	<ul style="list-style-type: none"> • 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。 <p>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。 エリアの入力トリガを設定します。データ入力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。</p>
pv input trigger	倍長整数	6	<ul style="list-style-type: none"> • 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 (数値キーパッド) 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 click) 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 record tag	倍長整数 16	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 resizable columns	倍長整数 4	カラムサイズの変更の可、不可を設定します。関連する値： <ul style="list-style-type: none">● <u>pv value on</u> : エリアのカラムサイズを変更できる。● <u>pv value off</u> : エリアのカラムサイズは変更できない。
pv resizable rows	倍長整数 5	行サイズの変更の可、不可を設定します。関連する値： <ul style="list-style-type: none">● <u>pv value on</u> : エリアの行サイズを変更できる。● <u>pv value off</u> : エリアの行サイズは変更できない。
pv row headers width	倍長整数 22	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 saving dialog	倍長整数 37	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： <ul style="list-style-type: none">● <u>pv value on</u> : 確認メッセージが表示される (デフォルト値)。● <u>pv value off</u> : 確認メッセージは表示されない。
pv select highlight	倍長整数 1	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： <ul style="list-style-type: none">● <u>pv value on</u> : エリアの選択範囲を高輝度表示する。● <u>pv value off</u> : 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 mode	倍長整数 0	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。 <ul style="list-style-type: none">● <u>pv select not allowed</u> : 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。● <u>pv select single row</u> : エリアでは一度に1 行だけを選択できる。● <u>pv select adjacent rows</u> : エリアでは隣接する行だけを選択できる。● <u>pv select multiple rows</u> : エリアでは隣接していなくても複数の行を選択できる。● <u>pv select single column</u> : エリアでは一度に1 列だけを選択できる。● <u>pv select adjacent columns</u> : エリアでは隣接するカラムだけを選択できる。● <u>pv select multiple columns</u> : 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。● <u>pv select single cell</u> : エリアでは一度に1 個のセルだけを選択できる。● <u>pv select adjacent cells</u> : エリアでは隣接するセルだけを選択できる。● <u>pv select multiple cells</u> : 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。
pv select null	倍長整数 2	注：選択範囲内でのデータ入力は可能なままです (<u>pv select not allowed</u> 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、 <i>PV SET AREA PROPERTY</i> (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。 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： <ul style="list-style-type: none">● <u>pv value on</u> : 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。● <u>pv value off</u> : エリアの選択は必須。

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 click](#)) 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 show borders toolbar](#) 7 倍長整数 29 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](#) : 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 column headers](#) 7 倍長整数 23 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](#) : 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 formula toolbar](#) 7 倍長整数 30 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](#) : 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](#) 7 倍長整数 31 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](#) : 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 hor scrollbar](#) 7 倍長整数 34 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](#) : 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 menu bar](#) 7 倍長整数 25 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](#) : 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](#) 7 倍長 7 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：

numbers toolbar	調整数	27	<ul style="list-style-type: none"> ● <u>pv value on</u> : 数字ツールバーを表示する。 ● <u>pv value off</u> : 数字ツールバーを表示しない。
pv show row headers	倍長整数	24	<p>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 :</p> <ul style="list-style-type: none"> ● <u>pv value on</u> : 行ヘッダを表示する。 ● <u>pv value off</u> : 行ヘッダを表示しない。
pv show selection	倍長整数	40	<p>フォーカスを持たない4D Viewエリア中でセレクションの表示モードを設定/取得できます。関連する値:</p> <ul style="list-style-type: none"> ● <u>pv value on</u> : 4D Viewエリアにフォーカスがあるかないかに関わらず、エリアのセレクションは常に表示 (ハイライト) される。 ● <u>pv value off</u> : 4D Viewエリアがフォーカスを失うとセレクションも表示されなくなる
pv show standard toolbar	倍長整数	26	<p>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 :</p> <ul style="list-style-type: none"> ● <u>pv value on</u> : 標準ツールバーを表示する。 ● <u>pv value off</u> : 標準ツールバーを表示しない。
pv show style toolbar	倍長整数	28	<p>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 :</p> <ul style="list-style-type: none"> ● <u>pv value on</u> : スタイルツールバーを表示する。 ● <u>pv value off</u> : スタイルツールバーを表示しない。
pv show vert grid	倍長整数	32	<p>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 :</p> <ul style="list-style-type: none"> ● <u>pv value on</u> : 垂直グリッドを表示する。 ● <u>pv value off</u> : 垂直グリッドを表示しない。
pv show vert scrollbar	倍長整数	35	<p>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 :</p> <ul style="list-style-type: none"> ● <u>pv value on</u> : 縦スクロールバーを表示する。 ● <u>pv value off</u> : 縦スクロールバーを表示しない。
pv vert pane count	倍長整数	10	<p>エリアの縦方向のペイン数の読み込みを許可します。この定数は <i>PV Get area property</i> コマンドを使用した時にのみ読み込まれます。戻り値 : ペイン数。 覚書 : ペインとは、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せん)。</p>
pv zoom factor	倍長整数	36	<p>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%。</p>

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できます。

定数	型	値	コメント
pv arrow keys allowed	倍長整数	0	すべての矢印キーの利用を許可
pv arrow keys not allowed	倍長整数	3	データ受け入れの用途では矢印キーの利用を許可しない
pv right and left arrow keys	倍長整数	2	左と右矢印キーのみ利用を許可
pv top and bottom arrow keys	倍長整数	1	上と下矢印キーのみ利用を許可

PV Border edge

このテーマの定数を使用してセルの罫線を設定できます。1つ以上の罫線を設定するために、複数の定数を加算できます。複数のセルを選択したら、最初の4つの定数は範囲の外枠を示します。この場合、選択範囲の内枠は"pv border edge inner hor"と"pv border edge inner vert"定数で設定できます。

定数	型	値	コメント
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

定数	型	値	コメント
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 half	倍長整数	14	
pv border style none	倍長整数	0	
pv border style quarter	倍長整数	13	

PV Carriage return

このテーマの定数を使用して、入力中の改行による受け入れアクションを設定できます。認めるように設定されることで、割り当てられた更新キーストロークに関わらず、改行により入力が受け入れられます。

定数	型	値	コメント
pv cr allowed	倍長整数	1	改行キーを押すと、セル内で新しい行が作成されます。
pv cr allowed with ctrl	倍長整数	2	Ctrl+改行 (Mac OS ではCommand+改行) により、セル内に新しい行が作成される
pv cr allowed with shift	倍長整数	4	Shift + 改行キーを押すと、セル内で新しい行が作成されます。
pv cr not allowed	倍長整数	0	エリア内に複数行のデータ入力は許可されない。

PV Cell properties

定数	型	値	コメント
pv add name	倍長整数	0	セルに既につけられた名称に新しい名前が追加される。
pv cell height	倍長整数	101	セルの高さの設定を許可。関連する値: ピクセル単位の高さ
pv cell width	倍長整数	100	セルの幅の設定を許可。関連する値: ピクセル単位の幅
pv replace name	倍長整数	1	セルに付けられている名称が新しい名前で置き換えられる。

PV Cell value type

定数	型	値	コメント
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

このテーマの定数を使用して、4D Viewの機能にプラグインのインターフェースからアクセスできます。名称は以下のルールでつけられています:

- "cmd" はメニューコマンド (およびツールパレットの対応するアイコン) を表します。
- "pal" はツールパレットアイコンからのみアクセス可能な機能を表します。

これらの定数は *PV SET COMMAND STATUS* と *PV GET COMMAND STATUS* コマンドでのみ利用できます。

定数	型	値	コメント
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	
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 panes	倍長整数	171
pv cmd freeze references	倍長整数	121
pv cmd go to full screen	倍長整数	20
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
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 unfreeze panes	倍長整数	172
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
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

定数	型	値	コメント
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

定数	型	値	コメント
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

定数	型	値	コメント
pv html	倍長整数	3	
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

定数	型	値	コメント
pv column count	倍長整数	0	<p>エリアに表示される列数の設定と読み込みを許可</p> <p>エリアの"更新"属性の読み書きを許可。使用可能な値: <code>pv value on</code>または<code>pv value off</code></p>
pv document modified	倍長整数	4	<ul style="list-style-type: none"> 書き込みモードでこの定数が使用されると(<code>PV SET DOCUMENT PROPERTY</code>コマンド)、<code>value</code>引数に<code>pv value on</code>を渡すとエリアが閉じられるときに、エリアが更新されたことを通知する警告ダイアログが表示されます。<code>value</code>引数に<code>pv value off</code>を渡し、ユーザーもプログラムもドキュメントを更新していなければこのダイアログボックスは表示されません。 この定数が<code>PV Get document property</code>コマンドで読み取りのために使用されると、ドキュメントが更新されていた場合に1が、そうでなければ0が返されます。 <p>エリアのフォーミュラ中で、4D変数やメソッド、コマンドの呼び出しを禁止します。利用可能な値: <code>pv value on</code>または<code>pv value off</code></p>
pv no external call	倍長整数	3	<ul style="list-style-type: none"> <code>pv value on</code>: フォーミュラ中で、4D変数やメソッド、コマンドの呼び出しが禁止されます。この場合、PV入力認可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。 <code>pv value off</code>: フォーミュラ中で、すべての4D変数やメソッド、コマンドの呼び出しが許可されます(デフォルト)。
pv picture count	倍長整数	2	<p>この定数は読み込みのみです (<code>PV Get document property</code>コマンド)。エリアにペーストされたピクチャ数を返します。</p>
pv row count	倍長整数	1	<p>エリアに表示される行数の設定と読み込みを許可</p>

PV Drag drop allowed

このテーマの定数を使用して、ドラッグ&ドロップ時に4D Viewがドラッグ元あるいはドロップ先として受け入れる要素のタイプを指定および取得できます。これらの定数を加算して複数の要素タイプを指定できます。利用方法の例題はPV SET DRAG SIGNATURESコマンドの説明にあります。

定数	型	値	コメント
pv DD 4D objects	倍長整数	1024	4Dオブジェクトをドロップできます。すべての型の4Dフィールド (BLOBとサブテーブルを除く) と変数 (BLOBを除く) をドロップできます。
pv DD adjacent cells	倍長整数	4	隣接する複数のセルあるいは選択した1つのセルをドラッグあるいはドロップできます。
pv DD adjacent columns	倍長整数	256	隣接する複数の列または選択した1つの列をドラッグおよびドロップできます。
pv DD adjacent rows	倍長整数	32	隣接する複数の行または選択した1つの行をドラッグおよびドロップできます。
pv DD multiple cells	倍長整数	8	隣接しているかどうかに関わらず、複数のセルあるいは選択した1つのセルをドラッグあるいはドロップできます。
pv DD multiple columns	倍長整数	512	隣接しているかどうかに関わらず、複数の列あるいは選択した1つの列をドラッグあるいはドロップできます。
pv DD multiple rows	倍長整数	64	隣接しているかどうかに関わらず、複数の行あるいは選択した1つの行をドラッグあるいはドロップできます。
pv DD not allowed	倍長整数	0	ドラッグ&ドロップが許可されていても、エリア中の選択をドラッグもドロップもできません
pv DD single cell	倍長整数	2	個別のセルのセレクションをドラッグおよびドロップできます。
pv DD single column	倍長整数	128	個別の列のセレクションをドラッグおよびドロップできます。
pv DD single row	倍長整数	16	個別の行のセレクションをドラッグおよびドロップできます。

PV Drop action

定数	型	値	コメント
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

定数	型	値	コメント
pv drag column	倍長整数	2	ドラッグ元の列番号を返す。
pv drag content	倍長整数	6	
pv drag plugin	倍長整数	1	ドラッグされた4D Viewエリアの番号を返す。
pv drag process	倍長整数	0	ドラッグ元エリアのプロセス番号を返す。
pv drag row	倍長整数	3	ドラッグ元の行番号を返す。
pv drag X offset	倍長整数	4	(左上隅のセルを基点として)ドラッグアクションが行われたセルのX座標を返す。
pv drag Y offset	倍長整数	5	(左上隅のセルを基点として)ドラッグアクションが行われたセルのY座標を返す。
pv drop action	倍長整数	14	ユーザーにより行われたドロップアクションを取得できるようにする。 PV Drop action テーマの定数値を返します。
pv drop column	倍長整数	9	ドロップ先の列番号を返す。
pv drop content	倍長整数	13	
pv drop plugin	倍長整数	8	ドロップされたエリアの番号を返す。
pv drop process	倍長整数	7	ドロップ先エリアのプロセス番号を返す。
pv drop row	倍長整数	10	ドロップ先の行番号を返す。
pv drop X offset	倍長整数	11	(左上隅のセルを基点として)ドロップアクションが行われたセルのX座標を返す。
pv drop Y offset	倍長整数	12	(左上隅のセルを基点として)ドロップアクションが行われたセルのY座標を返す。

PV Drop mode

定数	型	値	コメント
pv drop insert only	倍長整数	1	ドロップされた値はエリア内に挿入のみ可能
pv drop insert or replace	倍長整数	0	ドロップされた値はエリアに挿入されるか、既存の値を置き換える
pv drop replace only	倍長整数	2	ドロップされた値はエリアの既存の値を置き換えのみ可能

PV Event

特別な動作:

- `pv on cell value changed`はダイナミックエリアでは生成されません。
- `pv on getting focus`と`pv on losing focus`は、セルではなく、4D Viewエリアがフォーカスを得たときあるいは失ったときに生成されます。

定数	型	値	コメント
<code>pv on active cell changed</code>	倍長整数	8	
<code>pv on cell value changed</code>	倍長整数	9	
<code>pv on clicked</code>	倍長整数	2	
<code>pv on column resize</code>	倍長整数	12	
<code>pv on column sort</code>	倍長整数	14	
<code>pv on contextual click</code>	倍長整数	15	
<code>pv on double clicked</code>	倍長整数	4	
<code>pv on drag</code>	倍長整数	10	
<code>pv on drop</code>	倍長整数	11	
<code>pv on getting focus</code>	倍長整数	0	
<code>pv on keyboard</code>	倍長整数	6	
<code>pv on losing focus</code>	倍長整数	1	
<code>pv on right clicked</code>	倍長整数	3	
<code>pv on row resize</code>	倍長整数	13	
<code>pv on scrolled</code>	倍長整数	5	
<code>pv on selection changed</code>	倍長整数	7	

PV Header sort

定数	型	値	コメント
pv ascending sort	倍長整数	2	4D Viewは昇順ソートを実行します。
pv descending sort	倍長整数	3	4D Viewは降順ソートを実行します。
pv sort allowed	倍長整数	1	ユーザーが列ヘッダーをクリックすると、4D Viewは標準のソートを実行します。この場合、ヘッダーにソートの方向を示すマークが表示されます。続くクリックは昇順と降順を切り替えます。
pv sort not allowed	倍長整数	0	(デフォルト値) ユーザーが列ヘッダーをクリックしても、4D Viewは標準のソートを実行しません (開発者はカスタマイズされた方法でソートを管理することができます)。

PV Headers & footers

定数	型	値	コメント
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

このテーマの定数を使用して、入力中の**Enter**キーのアクションを指定できます。"標準"モードで使用されるとき、**Enter**キーは入力のみを有効にします。また更新されたセルの右 (`pv enter key as tab`) や上 (`pv enter key as return`) のセルを有効にすることもできます。

定数	型	値	コメント
<code>pv enter key as return</code>	倍長 整数	2	Enterキーは現在のセルの入力を受け入れ、その下のセルを選択します。Shift+Enterキーを押すと上のセルが選択されます。
<code>pv enter key as tab</code>	倍長 整数	1	Enterキーは現在のセルの入力を受け入れ、その右のセルを選択します。Shift+Enterキーを押すと左のセルが選択されます。
<code>pv enter key standard</code>	倍長 整数	0	Enterキーは現在のセルの入力を受け入れ、そのセルを選択します (カレントセルは変更されません)。

PV Pane properties

`pv pane relative scroll`定数は書き込みコマンド (**PV SET PANE PROPERTY**等) でのみ利用できます。

定数	型	値	コメント
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	このプロパティは固定ペインモードのエリアでは無効です。
pv pane lock splitter	倍長整数	1	このプロパティは固定ペインモードのエリアでは無効です。
pv pane relative scroll	倍長整数	7	プロパティは"PV SET..."コマンドでのみ使用できます。スクロールカーソルの現在位置を基点とし、ピクセル値を使用してペインの内容をスクロールするために使用します。一番上の行が完全に表示されるように、ピクセルでのスクロールは調整されます。このプロパティはエリアが固定ペインモードのときは無効です。
pv pane rows count	倍長整数	5	
pv pane size in pixels	倍長整数	3	このプロパティは固定ペインモードのエリアでは無効です。
pv pane true scroll	倍長整数	6	スクロールカーソルの現在位置に関わらず、エリアの基点 (例えば最初のセル) からピクセル値を使用してペインの内容をスクロールするために使用します。このプロパティはエリアが固定ペインモードのときは無効です。
pv pane view splitter cursor	倍長整数	2	このプロパティは固定ペインモードのエリアでは無効です。

PV Picture mapping mode

定数	型	値	コメント
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`と`pv picture data height`定数は読み込みのみで使用できます。

定数	型	値	コメント
<code>pv picture background</code>	倍長整数	8	
<code>pv picture column</code>	倍長整数	0	
<code>pv picture data height</code>	倍長整数	5	
<code>pv picture data width</code>	倍長整数	4	
<code>pv picture display height</code>	倍長整数	7	
<code>pv picture display width</code>	倍長整数	6	
<code>pv picture fixed size</code>	倍長整数	10	
<code>pv picture hor offset</code>	倍長整数	2	
<code>pv picture locked</code>	倍長整数	11	
<code>pv picture mapping mode</code>	倍長整数	9	
<code>pv picture row</code>	倍長整数	1	
<code>pv picture vert offset</code>	倍長整数	3	

PV Plugin properties

定数	型	値	コメント
pv button height	倍長整数	5	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 button width	倍長整数	4	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 confirm convert dialog	倍長整数	6	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。 <ul style="list-style-type: none"> ● 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。
pv default columns count	倍長整数	2	新しい4D View ドキュメント内のデフォルトの列数を定義します。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列が含まれます。関連する値：列数。
pv default rows count	倍長整数	3	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 load template on server	倍長整数	1	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。 <ul style="list-style-type: none"> ● 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 write template on server	倍長整数	0	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。 <ul style="list-style-type: none"> ● 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

定数	型	値	コメント
pv print adjust area	倍長整数	10	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： <ul style="list-style-type: none"> 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 binding	倍長整数	26	印刷が両面モードで行われる際の、綴じしろの場所を設定あるいは取得するために使用します。関連する値: PV Print values テーマの以下の値: <ul style="list-style-type: none"> pv left binding: 左綴じ (デフォルト値) pv top binding: 上綴じ
pv print bottom margin	倍長整数	3	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
pv print centered	倍長整数	9	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： <ul style="list-style-type: none"> 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 color	倍長整数	23	カラー処理のモードを取得あるいは設定するために使用します。このプロパティはカラープリンターでのみ有効です。関連する値: PV Print values テーマの以下の定数: <ul style="list-style-type: none"> pv black and white: 白黒印刷 (モノクロ) pv color: カラー印刷
pv print dead bottom margin	倍長整数	18	Note: このプロパティはWindowsでのみ使用できます。 この定数は読み取り専用で (<i>PV Get print property</i> 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。 Note: 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	倍長整数	15	この定数は読み取り専用で (<i>PV Get print property</i> 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。 Note: 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 right margin	倍長整数	17	この定数は読み取り専用で (<i>PV Get print property</i> 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。 Note: 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 top margin	倍長整数	16	この定数は読み取り専用で (<i>PV Get print property</i> 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。 Note: 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 destination	倍長整数	24	印刷の出力先を取得あるいは設定するために使用します。関連する値: PV Print values テーマの以下の定数: <ul style="list-style-type: none"> pv destination printer: プリントジョブはプリンターに送られます。 pv destination file (Windowsのみ): プリントジョブはファイルに送られます。この定数が使用されると、value2には作成されたドキュメントのパスが返されます。value2に空の文字列を渡すかこの引数を省略すると、印刷時にファイルを保存ダイアログが表示されます。 pv destination PDF file (Mac OSのみ): プリントジョブはPDFファイルに送られます。この定数が使用されると、value2には作成されたPDFドキュメントのパスが返されます。value2に空の文字列を渡すかこの引数を省略すると、印刷時にファイルを保存ダイアログが表示されます。 pv destination EPS file (Mac OSのみ): プリントジョブはEPSファイルに送られます。この定数が使用されると、value2には作成されたEPSドキュメントのパスが返されます。value2に空の文字列を渡すかこの引数を省略すると、印刷時にファイルを保存ダイアログが表示されます。

pv print document name	10 倍長整数	27	<p>スプーラーに表示されるプリントジョブ名を設定あるいは取得するために使用します。この定数が使用されると、value2はプリントジョブ名を含みます。valueには0を渡します。</p> <p>標準の処理 (名前として“4D View”) を使用するには、value2に空の文字列を渡します。</p> <p>片面あるいは両面印刷どちらを使用するか指定します。関連する値:</p>
pv print double sided	10 倍長整数	25	<ul style="list-style-type: none"> • <u>pv value on</u>: 両面印刷 • <u>pv value off</u>: 片面印刷 (デフォルト値) <p>Note: このプロパティはWindowsでのみ利用できます。</p>
pv print frame each page	10 倍長整数	11	<p>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:</p> <ul style="list-style-type: none"> • <u>pv value on</u> : 各ページに枠を印刷する。 • <u>pv value off</u> : 枠を印刷しない。
pv print grid	10 倍長整数	12	<p>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:</p> <ul style="list-style-type: none"> • <u>pv value on</u> : グリッドを印刷する。 • <u>pv value off</u> : グリッドを印刷しない。
pv print headers	10 倍長整数	8	<p>行ヘッダと列ヘッダの印刷を行います。関連する値:</p> <ul style="list-style-type: none"> • <u>pv value on</u> : 行ヘッダと列ヘッダを印刷する。 • <u>pv value off</u> : 行ヘッダと列ヘッダを印刷しない。
pv print left margin	10 倍長整数	0	<p>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。</p> <p>関連する値: ピクセル単位のマージン</p>
pv print number copies	10 倍長整数	21	<p>印刷コピー数を設定あるいは取得するために使用します。関連する値: コピー数 (デフォルトで1)</p>
pv print orientation	10 倍長整数	19	<p>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。</p> <ul style="list-style-type: none"> • <u>pv portrait orientation</u> : 用紙は縦方向に合わせる。 • <u>pv landscape orientation</u> : 用紙は横方向に合わせる。
pv print pages from	10 倍長整数	28	<p>印刷の開始ページを設定あるいは取得するために使用します。</p> <p>関連する値: ページ番号</p>
pv print pages to	10 倍長整数	29	<p>印刷の終了ページを設定あるいは取得するために使用します。</p> <p>関連する値: ページ番号</p>
pv print paper height	10 倍長整数	14	<p>用紙の高さを返します。関連する値: ピクセル単位の高さ</p>
pv print paper source	10 倍長整数	22	<p>使用する用紙トレイを設定あるいは取得するために使用します。関連する値: 4DのPRINT OPTION VALUESコマンドから返されるnamesArrayの要素に対応するinfo1Array要素の値。この配列には使用される用紙トレイの名前が含まれます。注: このプロパティはWindowsでのみ使用できます。</p>
pv print paper width	10 倍長整数	13	<p>用紙の幅を返します。</p> <p>関連する値: ピクセル単位の幅</p>
pv print	10 倍長整数		<p>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</p>

repeat first column	整数	4	<u>column</u> 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	倍長整数	6	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 <u>pv print repeat last row</u> 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 column	倍長整数	5	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 <u>pv print repeat first column</u> 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 row	倍長整数	7	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 <u>pv print repeat first row</u> 定数と組み合わせて使用します。 関連する値：行番号
pv print right margin	倍長整数	2	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 scale	倍長整数	20	印刷倍率を設定あるいは取得するために使用します。プリンターによっては倍率の変更が許可されないことがあることに留意してください。無効な値を渡すと、プロパティは印刷時に100%にリセットされます。関連する値: 印刷倍率
pv print top margin	倍長整数	1	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 values

定数	型	値	コメント
pv black and white	倍長整数	2	白黒で印刷 (モノクロ)
pv color	倍長整数	3	カラーで印刷
pv destination EPS file	倍長整数	7	印刷ジョブはEPSファイルに送られる (Mac OSのみ)
pv destination file	倍長整数	5	印刷ジョブはファイルに送られる (Windowsのみ)
pv destination PDF file	倍長整数	6	印刷ジョブはPDFファイルに送られる (Mac OSのみ)
pv destination printer	倍長整数	4	印刷ジョブはプリンターに送られる
pv landscape orientation	倍長整数	1	用紙は横向きモード
pv left binding	倍長整数	8	左綴じ (デフォルト値)
pv portrait orientation	倍長整数	0	用紙は縦向きモード
pv top binding	倍長整数	9	上綴じ

PV Report functions

定数	型	値	コメント
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

定数	型	値	コメント
pv select adjacent cells	倍長整数	8	エリア中で隣接するセルのみを選択可能
pv select adjacent columns	倍長整数	5	エリア中で隣接する列のみを選択可能
pv select adjacent rows	倍長整数	2	エリア中で隣接する行のみを選択可能
pv select multiple cells	倍長整数	9	エリア内で、隣接するしないに関わらず、複数のセルを選択可能
pv select multiple columns	倍長整数	6	エリア内で、隣接するしないに関わらず、複数の列を選択可能
pv select multiple rows	倍長整数	3	エリア内で、隣接するしないに関わらず、複数の行を選択可能
pv select not allowed	倍長整数	0	エリア中で選択を行うことはできません (すべてのセルの選択が解除)。データ入力もできません (フォーミュラエディタはロックされます)。データは見ることだけできます。
pv select single cell	倍長整数	7	エリア中で一度に1つのセルのみを選択可能
pv select single column	倍長整数	4	エリア中で一度に1つの列のみを選択可能
pv select single row	倍長整数	1	エリア中で一度に1つの行のみを選択可能

PV Selection action

定数	型	値	コメント
pv selection add	倍長整数	1	新しいセレクションは既存のセレクションに追加される
pv selection reduce	倍長整数	2	既存のセレクションから選択したセレクションが取り除かれる
pv selection set	倍長整数	0	新しいセレクションが既存のセレクションを置き換える

PV Style format date time

システム設定により、表示結果が異なる場合があります。

定数	型	値	コメント
pv Abbr Month Day Year	倍長整数	6	21 Feb, 2002
pv Abbreviated	倍長整数	2	Thu 21 Feb 2002
pv Abbreviated H MM AM PM	倍長整数	13	Thu 21Feb 2002 at 12:30 PM
pv Day Name	倍長整数	7	Thursday
pv Day Number	倍長整数	8	21
pv HH MM	倍長整数	18	12:30
pv HH MM AM PM	倍長整数	21	12:30 PM
pv HH MM SS	倍長整数	17	12:30:00
pv Hour Min	倍長整数	20	12 hours 30 minutes
pv Hour Min Sec	倍長整数	19	12 hours 30 minutes 0 second
pv Long	倍長整数	3	Thursday 21 February 2002
pv Long H MM AM PM	倍長整数	12	Thursday 21 February 2002 at 12:30 PM
pv Month Day Year	倍長整数	5	21 February, 2002
pv Month Day Year H MM AM PM	倍長整数	15	21 February, 2002 at 12:30 PM
pv Month Name	倍長整数	9	February
pv Month Number	倍長整数	10	2
pv Short	倍長整数	1	02/21/02
pv Short HH MM SS	倍長整数	14	02/21/02 at 12:30:00
pv Short2	倍長整数	4	02/21/2002
pv Short2 Hour Min Sec	倍長整数	16	21/02/2002 and 12 hours 30 minutes 0 second
pv Year Number	倍長整数	11	2002

PV Style properties

定数	型	値	コメント
pv style automatic word wrap	倍長整数	33	セルの内容が幅を超えたときの自動ワードラップ機能を設定できます。関連する値: PV Style values テーマの定数 <ul style="list-style-type: none"> pv value on: 必要に応じてセルの内容が自動で次の行に送られる pv value off: 必要に応じてセルの内容が隣接したセルにはみ出して表示される
pv style based on	倍長整数	4	領域内の各セルは、引数value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 color back even	倍長整数	11	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color back odd	倍長整数	12	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color minus even	倍長整数	17	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color minus odd	倍長整数	18	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color text even	倍長整数	13	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color text odd	倍長整数	14	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color zero even	倍長整数	15	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style color zero odd	倍長整数	16	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 (<i>PV RGB to color</i> 関数、および <i>PV Index to color</i> 関数を参照)。
pv style format alpha	倍長整数	6	領域内の各セルは、引数value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 bool	倍長整数	8	領域内の各セルは、引数value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。
			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。 <ul style="list-style-type: none"> pv Short: 02/21/02 pv Abbreviated: Thu 21 Feb 2002 pv Long: Thursday 21 February 2002 pv Short2: 02/21/2002

pv style format date time	倍 長 整 数	9	<ul style="list-style-type: none"> • 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 21Feb 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
pv style format forced text	倍 長 整 数	32	<p>Note: 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。</p> <p>セルに対し生テキストの表示を強制します。セルの内容 (数値、日付、テキスト等) に基づき4D Viewが適用する自動表示フォーマットは使用されません。関連する値: PV Style valuesテーマの定数</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルの内容は自動フォーマットなしで表示される • <u>pv value off</u> (デフォルト): セルの内容は自動フォーマットされて表示される
pv style format num	倍 長 整 数	7	<p>領域内の各セルは、引数value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。</p> <p>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。</p> <p>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。</p>
pv style format picture	倍 長 整 数	10	<ul style="list-style-type: none"> • 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 hidden	倍 長 整 数	1	<p>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できません。関連する値: PV Style valuesテーマ内の定数。</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルはロックされ、隠されている。 • <u>pv value off</u>: セルはロックされておらず、表示されている。
pv style hor alignment	倍 長 整 数	29	<p>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。</p> <ul style="list-style-type: none"> • <u>pv value hor alignment default</u>: セル領域にデフォルトの横揃えを適用する。 • <u>pv value hor alignment left</u>: セル領域に左横揃えを適用する。 • <u>pv value hor alignment center</u>: セル領域に中央横揃えを適用する。 • <u>pv value hor alignment right</u>: セル領域に右横揃えを適用する。
pv style locked	倍 長 整 数	0	<p>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できません。関連する値: PV Style valuesテーマ内の定数。</p> <ul style="list-style-type: none"> • <u>pv value on</u>: セルはロックされる。 • <u>pv value off</u>: セルはロックされない。
倍			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 style rotation	倍長整数 31	<ul style="list-style-type: none"> • 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。
pv style spellcheck	倍長整数 2	<p>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。</p> <ul style="list-style-type: none"> • 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 text bold	倍長整数 22	<p>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。</p> <ul style="list-style-type: none"> • 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 condensed	倍長整数 27	<p>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。</p> <ul style="list-style-type: none"> • 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	倍長整数 28	<p>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。</p> <ul style="list-style-type: none"> • 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 text face	倍長整数 21	<p>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。</p>
pv style text font	倍長整数 19	<p>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 (<i>PV Add font</i> 関数、および <i>PV GET FONT LIST</i> コマンドを参照)。</p>
pv style text italic	倍長整数 23	<p>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。</p> <ul style="list-style-type: none"> • 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 outline	倍長整数 25	<p>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。</p> <ul style="list-style-type: none"> • 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	倍長整数 26	<p>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。</p> <ul style="list-style-type: none"> • 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 size	倍長整数 20	<p>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。</p>
pv style text underline	倍長整数 24	<p>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。</p> <ul style="list-style-type: none"> • 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 use
picture
height

倍
長
整
数

3

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 vert
alignment

倍
長
整
数

30

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 special values

これらの定数を使用して、*PV SET STYLE PROPERTY*と*PV Get style property*コマンドで標準の4D Viewスタイルシートを適用できます ('style' 引数)。

定数	型	値	コメント
pv style cells	倍長整数	-1	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	倍長整数	-2	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	倍長整数	-3	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”)。

PV Style values

"mixed"が名前につけられた定数はプロパティ読み出しコマンドでのみ利用できます。

定数	型	値	コメント
pv value base style mixed	倍長整数	65535	
pv value color 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	セルに横中央揃えを適用
pv value hor alignment default	倍長整数	0	セルに横方向のデフォルト揃えを適用
pv value hor alignment left	倍長整数	1	セルに左揃えを適用
pv value hor alignment mixed	倍長整数	255	セルにミックスした横揃えを適用
pv value hor alignment right	倍長整数	3	セルに右揃えを適用
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 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 none	倍長整数	-3	
pv value off	倍長整数	0	セルに値を適用しない
pv value on	倍長整数	1	セルに値を適用
pv value rotation 0	倍長整数	0	セルに回転を適用しない
pv value rotation 180	倍長整数	2	180度の回転を適用
pv value rotation 270	倍長整数	3	左方向に270度の回転を適用
pv value rotation 90	倍長整数	1	左方向に90度の回転を適用
pv value rotation mixed	倍長整数	255	左方向に、ミックスした回転を適用
pv value vert alignment bottom	倍長整数	3	セルに下揃えを適用
pv value vert alignment center	倍長整数	2	セルに縦中央揃えを適用
pv value vert alignment mixed	倍長整数	255	セルにミックスした縦揃えを適用
pv value vert alignment top	倍長整数	1	セルに上揃えを適用

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 定数を加算できます。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。
- "ctrl"はWindowsでCtrlキーを、Mac OSでCommandキーを表します。

定数	型	値	コメント
pv trigger input key	倍長整数	1	データ入力は任意のキー入力により開始する。この場合、セル間のブラウズはキーボードを使用するのみ可能 (TabおよびShift+Tabで水平移動、EnterおよびShift+Enterで縦移動、または矢印キー)。
pv trigger input on enter	倍長整数	2	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	倍長整数	4	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 none	倍長整数	0	<ul style="list-style-type: none"> ドラッグ: エリア内でドラッグは行えない。 データ入力: データ入力モードでキーが許可されている場合でも (<code>pv input enter key mode</code>定数を参照) データ入力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。 選択: 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 on alt click	倍長整数	32	<ul style="list-style-type: none"> ドラッグ: 選択範囲のドラッグはAlt キー+ クリックを使用して行う。 データ入力: データ入力はセルのAlt キー+ クリックにより開始する。 選択: 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	倍長整数	64	<ul style="list-style-type: none"> ドラッグ: 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。 データ入力: 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。 選択: 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 click	倍長整数	8	<ul style="list-style-type: none"> ドラッグ: 選択範囲のドラッグはマウスクリックを使用して行う。 データ入力: データ入力はセルのクリックにより開始する。 <code>pv trigger input on gain sel</code> 定数とは異なり、入力カーソルは表示されない。 選択: 選択範囲の決定はマウスクリックによって行う。
pv trigger on ctrl click	倍長整数	128	<ul style="list-style-type: none"> ドラッグ: 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)。 データ入力: データ入力はセルのCtrl キー+ クリック (MacOS ではCommand キー+ クリック) により開始する。 選択: 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	倍長整数	256	<ul style="list-style-type: none"> ドラッグ: 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)。 データ入力: データ入力はセルの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。 選択: 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 double click	倍長整数	16	<ul style="list-style-type: none"> ドラッグ: 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。 データ入力: 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。 選択: 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 shift click	倍 長 整 数	512	<ul style="list-style-type: none"> ● ドラッグ: 選択範囲のドラッグはShift キー+ クリックを使用して行う。 ● データ入力: データ入力はセルのShiftキー+ クリックにより開始する。 ● 選択: 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 click	倍 長 整 数	1024	<ul style="list-style-type: none"> ● ドラッグ: 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。 ● データ入力: データ入力はセルのShiftキー+ ダブルクリックにより開始する。 ● 選択: 選択範囲の決定はShiftキー+ ダブルクリックによって行う。
pv trigger select on arrow	倍 長 整 数	1	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 return	倍 長 整 数	4	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 select on tab	倍 長 整 数	2	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。

目録 付録

付録A: 4D Viewエラーコードリスト

付録A: 4D Viewエラーコードリスト

エラー管理メソッド内で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。

No. 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。
- 4 これは古い引数です。
- 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。
- 41 回転が正しくありません。
- 42 廃止された枠線スタイルです。
- 43 無効なヘッダータイプです。
- 44 無効な日付&時間フォーマットです。
- 45 無効なピクチャーフォーマットです。
- 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。

- 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。
- 87 ソートの選択が正しくありません。
- 88 印刷プロパティが正しくありません。
- 89 プロパティ値が正しくありません。
- 90 印刷設定が正しくありません。
- 91 このコマンドを適用するには少なくとも1つの水平スプリッターと1つの垂直スプリッターがなければなりません。
- 92 ペインがフリーズされているときにスプリッターを追加することはできません。

4D View - コマンドリスト (文字順)

A	B	C	D	E	F	G	I	N	O	P	R	S	U	V
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 PV GET BORDER ROW RANGES