










# 4D Write Proリファレンス

-  はじめに
-  4D Write Pro エリアを定義する
-  4D Write Proドキュメントを4Dオブジェクトフィールドに保存する
-  4D Write Pro エリアを使用する
-  4D Write Pro ドキュメントの印刷
-  4D Write Pro ドキュメントに含める式の制限
-  4D Write ドキュメントの読み込み
-  4D Write Proランゲージ
-  新着

# はじめに

## 概要

---

4D Write Proは、4Dユーザーに対して、4Dデータベースに完全に統合した進化したワードプロセストールを提供します。4D Write Proを使用すれば、プリフォーマットされたEメールや文章に画像、スキャン済みの署名、フォーマット済みのテキストやダイナミック変数用のプレースホルダーなどを含めることができます。また請求書やレポートを動的に作成し、フォーマット済みのテキストや画像を含めることができます。

この製品の主な機能は以下の通りです：

- 4D Write との互換性: 4D Write Proオブジェクトは、以前の4D Writeドキュメントを開いたり変換したりでき、また4D Writeのプロパティの大部分をサポートしています。
- ワードプロセス: フォームに埋め込まれた4D Write Proオブジェクトは、テキストとスタイルの操作、画像の挿入、読み込みと書き出しなどを含めた様々な標準のワープロ機能を持ち合わせます。
- データベースとの統合:
  - 4D Write Pro オブジェクトはデータベースからのデータや、4Dが計算をしたデータなどを表示する変数部分を表示することができます。
  - 4D Write Pro ドキュメントはデータベースフィールド内や、ディスク上に保存することもできます。

## インストールとアクティベート

---

4D Write Pro はプラグインではなく、4D自身に完全に統合されており、配布と管理がより簡単になっています。追加のインストールは何も必要なく、フォーム内に4D Write Proエリアを追加するだけで、4Dアプリケーション内において4D Write Pro変数を直接管理することができます。

しかしながら、4D Write Proは4D Writeと同じライセンスを使用することに注意して下さい。機能を有効化するためには、アプリケーションにそのライセンスをインストールする必要があります。

**必要要件:** Windows上では、4D Write Proの機能はDirect2Dに依存しています。Windows 7またはWindows Server 2008以降のマシンでは、必要なバージョンのDirect2Dを利用可能にするために、Windowsプラットフォーム更新プログラムがインストールされていることを確認して下さい。

## このマニュアルについて

---

このマニュアルは4D Write Pro リファレンスガイド です。ここでは全ての4D Write Proの機能に加え、ユーザーインターフェースとランゲージコマンドについてカバーします。

4D Write Proオブジェクトは、特定のコマンド("4D Write Pro" テーマ)に加えて、4Dランゲージリファレンス内にある他の4Dテーマ(**オブジェクト(フォーム)** テーマと**スタイル付きテキスト** テーマ)のコマンドを使用しても管理する事ができるという点に注意して下さい。



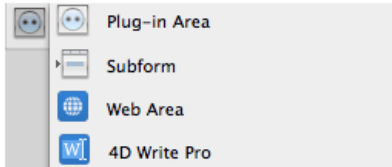
# 4D Write Pro

- WP CREATE BOOKMARK New 16.0
- WP DELETE BOOKMARK New 16.0
- WP EXPORT DOCUMENT
- WP EXPORT VARIABLE
- WP GET ATTRIBUTES
- WP Get bookmark range New 16.0
- WP GET BOOKMARKS New 16.0
- WP Get page count New 16.0
- WP Get paragraphs
- WP Get pictures
- WP Get range
- WP Get selection
- WP Import document
- WP INSERT BREAK New 16.0
- WP INSERT DOCUMENT New 16.0
- WP INSERT PICTURE New 16.0
- WP Is font style supported
- WP New Updated 16.0
- WP PRINT Updated 16.0
- WP RESET ATTRIBUTES
- WP SELECT
- WP SET ATTRIBUTES
- WP USE PAGE SETUP

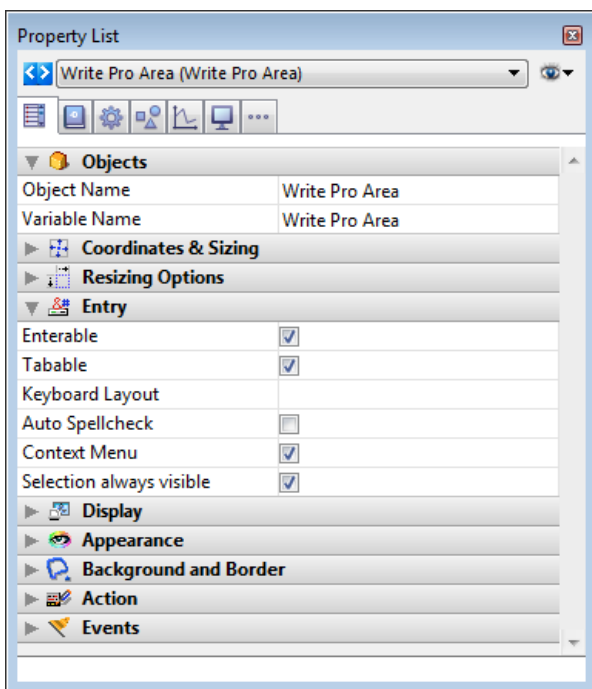
## 4D Write Pro エリアを定義する

### エリアの作成

4Dでは、4D Write Proドキュメントは**Write Pro**という4D フォームオブジェクト内にて手動で表示・編集することができます。このオブジェクトはオブジェクトバーの最終ツールの一部として提供されています:



4D Write Proフォームエリアはプロパティリスト内の標準のプロパティを通して設定することができます。すなわし、**オブジェクト名** そして **変数名**、**座標**、**入力**、**表示**、**外見**、**イベント**です。



**変数名**のプロパティはランゲージにおいて4D Write Proエリアの参照として使用します。変数は`object`型でなければならないという点に注意して下さい(詳細については**C\_OBJECT** コマンドを参照して下さい)。

"入力" プロパティはテキスト入力の基本的な機能を管理します:

- **入力可能:** エリアをロック・アンロックすることによって編集を許可または許可しないことを選択できます。
- **自動スペルチェック:** 4D Write Pro において利用可能です。
- **コンテキストメニュー:** アプリケーションモードにおいてコンテキストメニューを有効化・無効化します(**4D Write Pro エリアを使用する** の章を参照して下さい)。
- **選択を常に表示:** 標準のテキストエリアにおいてテキストの選択を管理します。

### オブジェクトライブラリーの4D Write Proウィジェットの使用

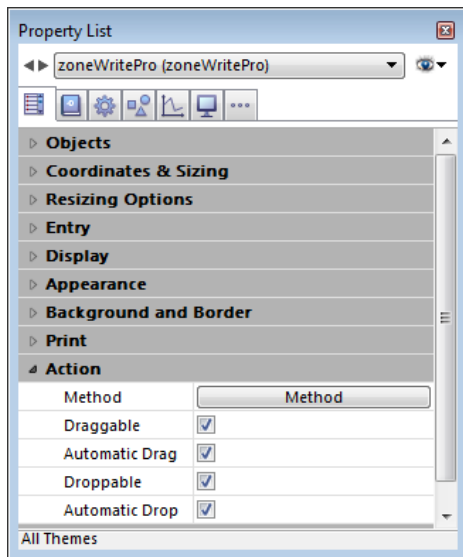
オブジェクトライブラリー内("入力エリア"テーマ内)にある**4D Write Pro** オブジェクトを仕様することで設定済みの4D Write Proエリアを作成することができます:

このエリアには、エリアの全ての属性(フォント、カラー、スタイル等)を管理することができるコントロールパネルが付属します:

より詳細な情報については、**4D Write Pro エリア**の章を参照してください。

## ドラッグ&ドロップの設定

ご自分の4D Write Proエリアでのドラッグ&ドロップ機能の設定をするためには、プロパティリストの"アクション"テーマ内にて適切なオプションを選択する必要があります:



4D Write Proエリアは、二つのドラッグ&ドロップモードをサポートします:

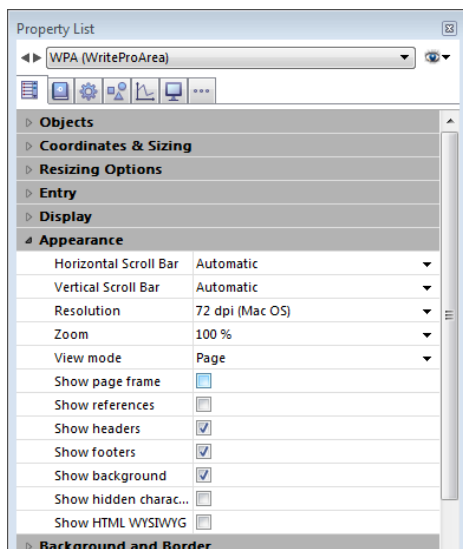
- **カスタムモード:** "ドラッグ可"と"ドロップ可"のオプションのみチェックされています。  
このモードでは、テキストを選択し、移動させる事ができます。 On Begin Drag Over イベントのオブジェクトメソッドが呼び出され、このカスタムモードを使用してドロップ時の挙動を定義する事ができます。
- **自動モード:** "ドラッグ可"、"ドロップ可"、そして"自動ドラッグ"と"自動ドロップ"のオプションがチェックされています。  
このモードでは、選択したテキストを移動または(**Alt/Option**キーを押しながら)コピーすることができます。 On Begin Drag Over イベントはトリガーされません。

**注:** "自動ドラッグ"と"自動ドロップ"のオプションしか選択していなかった場合、4D Write Proエリアには何の影響もありません。

## ビュープロパティの設定

4D Write Pro エリアに 4D Write Pro ドキュメントがデフォルトでどのように表示されるかを指定するには、プロパティリストで提供されているドキュメントビュープロパティを設定します。これらのプロパティを使って、例えば 4D Write Pro ドキュメントを印刷されるとおりに表示するか、それともブラウザで見たときと同じように表示するかを設定することができます。同一のフォーム上で、同じ 4D Write Pro ドキュメントの異なるビューを表示させることも可能です。

ドキュメントビューの設定は、4D Write Pro フォームオブジェクトのプロパティリストにおいて、**アピアランス**テーマの項目で指定します:



- **解像度:** 4D Write Pro エリアの画面解像度を設定します。デフォルト値は **72 dpi (Mac OS)** で、これはすべてのプラットフォームにおける 4D フォームの標準解像度です。この項目を**自動**に設定すると、Mac OS と Windows 間でドキュメントのレンダリングが異なることとなります。特定の dpi 値を指定すると、両プラットフォームでのレンダリングを同じになります。
- **ズーム:** 4D Write Pro エリアのコンテンツ表示に使用するズーム率を設定します。デフォルト値は 100% です。
- **ビューモード:** フォームエリア内の 4D Write Pro ドキュメントの表示モードを設定します。次の値が提供されています:
  - **ページ** (デフォルト): もっとも完全といえるビューモードで、ページの枠、余白、改ページ、ヘッダー & フッターなどを含みます。詳細については **ページビュー機能** を参照ください。
  - **下書き:** 基本のドキュメントプロパティを含む下書きモードです。
  - **埋め込み:** 埋め込みエリアに適切なビューモードです。余白や、ヘッダー & フッター、ページフレームなどは表示されません。  
このモードは Web に似た出力をするのに使用することもできます (この場合には、解像度を 96 dpi に設定のうえ、**HTML WYSIWYG を表示** オプションを有効にします)。

**注:** ビューモードプロパティは画面上のレンダリングにのみ使用されます。印刷設定については、特定のレンダリングルールが自動的に適用されます (**4D Write Pro ドキュメントの印刷** 参照)。

- **ページ枠を表示:** ページビューモードが "ページ" に設定されている場合に、ページのフレームを表示 / 非表示にします。デフォルトは非表示です。
- **参照を表示:** ドキュメントに参照として挿入された 4D 式をすべて表示します。このオプションが無効 (デフォルト) になっていると、4D Write Pro は挿入された 4D 式をカレント値で評価して、その値を表示します。もとの 4D 式を確認したいときには、このオプションを有効にします。すると、これらの参照は灰色の背景色とともにドキュメント内に表示されます。  
例えば、フォーマットを指定したカレント日付を挿入していると、デフォルトでは次の表示になります:

July 11, 2016

**参照を表示** オプションを有効にすると、代わりにもとの参照が表示されます:

String(Current date;Internal date long)

**注:** 4D 式を挿入するには、**ST INSERT EXPRESSION** コマンドを使います。

- **ヘッダー / フッターを表示:** ページビューモードが "ページ" に設定されている場合に、ページのヘッダー & フッターを表示 / 非表示にします。デフォルトは表示です。ヘッダー & フッターについての詳細は **を参照** ください。
- **背景を表示:** ページの背景画像および背景色を表示 / 非表示にします。デフォルトは表示です。
- **非表示文字を表示:** 非表示の文字を表示 / 非表示にします。デフォルトは非表示です。
- **HTML WYSIWYG を表示:** HTML WYSIWYG ビューを有効 / 無効にします。このビューでは、すべてのブラウザーに対応していない 4D Write Pro の属性が取り除かれます (デフォルトは無効)。

**互換性に関する注記:** 4D v15 R5 までのバージョンで作成された 4D Write Pro ドキュメントは、これらのプロパティはデフォルト値に設定されますが、例外として解像度のみ自動に設定されます。

```
C_OBJECT ( {method ;} variable {; variable2 ; ... ; variableN} )
```

引数	型		説明
method	メソッド	→	メソッド名
variable	変数	→	宣言する変数の名前、または引数\${...}

## 説明

**C\_OBJECT** は、指定されたそれぞれの変数をランゲージオブジェクト型としてタイプ定義します。

*Object* 型はv14より4Dランゲージでサポートされるようになりました。この型のオブジェクトは **オブジェクト(ランゲージ)** テーマのコマンドで管理されます。

プロセス、インタープロセスまたはローカル変数を宣言・タイプするためには( *method* 引数が渡されない)第一記法を使用します。この記法はインタープリテッドデータベースにて使用可能です。

メソッドの結果と引数 (\$0, \$1, \$2, 等)をコンパイラへと先に宣言するためには( *method* 引数が渡される)第二記法を使用します。データベースがコンパイルされる際に変数タイピングフェーズを省略して時間を節約したいときにはこの記法を使用しなければなりません。

**警告:** 第二記法はインタープリタモードで実行することはできません。このため、この記法を使用する際には、この記法をインタープリタモードでは実行されないメソッド(名前が"COMPILER"で始まる必要があります)に保存する必要があります。

**特殊使用法:** **C\_OBJECT**(\${...}) の記法を使用すると、メソッドの最後の引数が同じ型であればメソッドに対して同じ型の不定数の引数を指定することができます。例えば、**C\_OBJECT**(\${5}) は5番目の引数以降、メソッドはこの型の引数を不定数受け取ることができるということをコンパイラに意味します。

## 例題

**コンパイラコマンド** セクションを参照してください。

## 4D Write Proドキュメントを4Dオブジェクトフィールドに保存する

4D Write Proドキュメントを自動的に4Dデータファイルに保存できるようになりました。フォーム内に4D Write Proエリアを作成し、そのエリアの中身を保存するオブジェクトフィールドを作成すれば、エリア内に入力したテキストはレコードが評価されるごとにそれぞれのレコードへと自動的に保存されます。その後**QUERY BY ATTRIBUTE**コマンドを使用して内部属性の値に基づいてレコードを選択する事ができます。また独自の属性を4D Write Proエリアに追加しクエリすることができます。

この章では以下の機能について説明しています：

- フォーム内の4D Write Proエリアを4D オブジェクトフィールドへと結びつける
- 標準のオブジェクトコマンド**OB SET**、**OB Get**および**QUERY BY ATTRIBUTE**を使用して、保存されている4D Write Proドキュメントのカスタムの属性を設定、取得、そしてクエリする

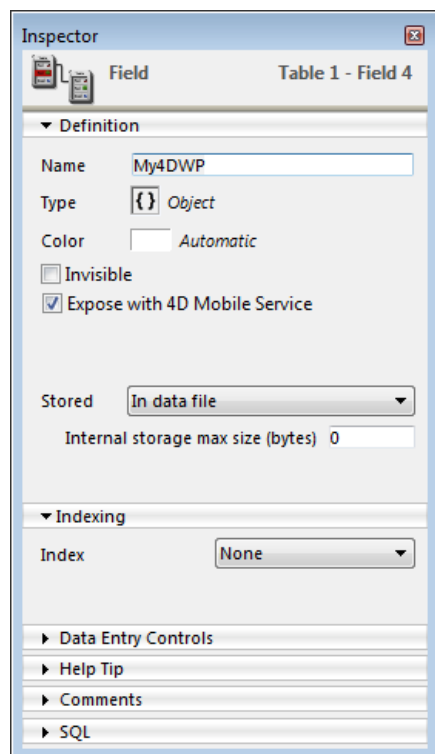
### 4D オブジェクトフィールドを4D Write Proエリアへと割り当てる

4D Write Proエリアを4Dオブジェクトフィールドに割り当てるためには、エリアの変数名プロパティにフィールド名を入力するだけです。

#### ストラクチャー内にオブジェクトフィールドを作成

データベースのストラクチャー内において、4Dオブジェクトフィールドであればどれも4D Write Proドキュメントを保存するのに使用する事ができます。他のオブジェクトフィールドと同様、必要に応じて以下のフィールドの標準のプロパティを定義する必要があります：

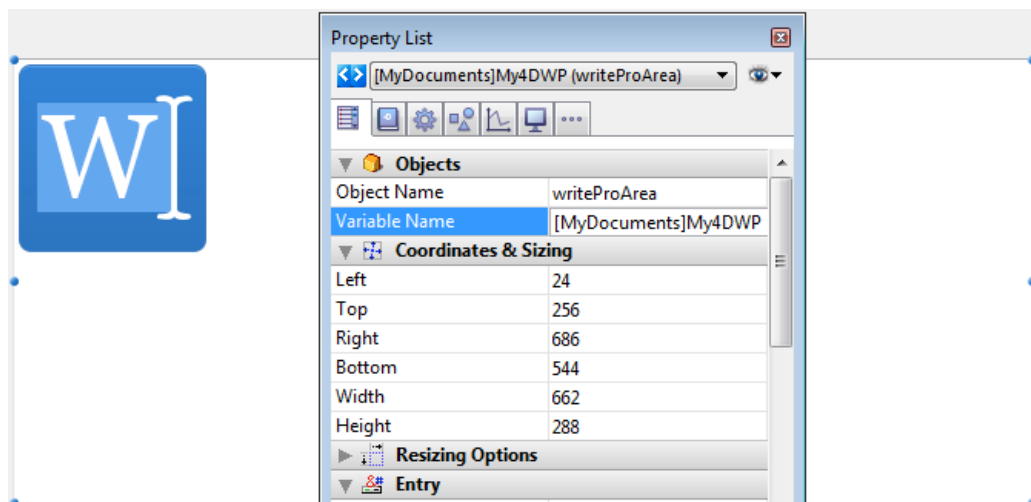
- フィールド名
- "4D Mobile Serviceに公開"などの属性とインデックス
- 保存オプション(詳細についてはを[データをデータファイル外に保存参照](#)して下さい)



#### オブジェクトフィールドを4D Write Proエリアに割り当て



4D Write Proドキュメントを保存するオブジェクトフィールドを決めたら、あとはそのエリアを含んでいるフォーム内で参照するだけです。どのようなテーブルフォームもプロジェクトフォームも使用する事ができます。フォームエディター内において、4D Write Proエリアのプロパティリスト内の、**変数名**の欄に標準の"[Table]Field"表記を使用してフィールド名を入力して下さい:



これで4D Write Proエリアはフィールドと関連付けがなされ、エリアの中身はレコード毎に自動的に保存されるようになりました。4D自動アクションボタンを使用しない場合、4Dコマンドを使用してエリアを手動で保存しなければならない点に注意して下さい。

## カスタムの属性を使用

4D Write Proエリアがオブジェクトフィールド内に保存されているとき、4D Write Proドキュメントにはカスタムの属性を保存または読み出すことができます。例えば作者名、ドキュメントのカテゴリなど、どんな追加情報でも有用だと思えるものは使用する事ができます。そしてカスタムの属性をクエリし、条件に合致したレコードを選択することができます

カスタムの属性は **WP EXPORT DOCUMENT** または **WP EXPORT VARIABLE** コマンドで書き出されます。カスタムの属性は**JSON Stringify**コマンドを使用して4D Write Pro オブジェクトフィールドをJSONに変換する際にも書き出されます(同時に4D Write Pro のメインドキュメント属性も書き出されます)。

カスタムの属性を設定または取得するためには、標準の**OB Get** と **OB SET**コマンドを使用するだけです。

たとえばフォームメソッドにおいて、以下のように書くことがカスタムの属性を設定できます:

```
If(Form event=On Validate)
  OB SET([MyDocuments]My4DWP;"myatt_Last edition by";Current user)
  OB SET([MyDocuments]My4DWP;"myatt_Category";"Memo")
End if
```

また、以下のように書いて、ドキュメントのカスタムの属性を読み出すことができます:

```
vAttrib:=OB Get([MyDocuments]My4DWP;"myatt_Last edition by")
```

カスタムの4D Write Pro属性をデータファイルに保存していた場合、これらの属性をクエリして適切な属性の値を含むレコードのセレクションを作成することができます。以下の例では、レコードを選択するためにオブジェクトフィールドを含んでいるテーブルをクエリします:

```
QUERY BY ATTRIBUTE([MyDocuments];[MyDocuments]My4DWP;"myatt_Category";="Memo")
//MyDocuments内の、(4D Write Proエリアに割り当てられている)My4DWPオブジェクトフィールド内で
//"myatt_Category"というカスタム属性が"Memo"という値を含んでいるレコードを全て選択します
```

**カスタム属性の名前についての注意:** カスタム属性は4D Write Pro内部属性と同じ名前空間を共有するため、内部属性とカスタム属性との衝突を避けるために、独自の属性を定義するときには必ず接頭辞をつけた名前を定義する事が強く推奨されます。接頭辞がついていない名前は4D Write Proの内部属性のために予約されているからです。接頭辞であればどのような独自の接頭辞であっても使用する事ができます(上記の例では"myatt\_"を接頭辞として使用しました)。

**注:** 4D v15 R4以降、4D Write Pro内部属性は、標準の**QUERY BY ATTRIBUTE**と**OB Get**と**OB SET**コマンドだけではなく、**WP SET ATTRIBUTES**、**WP GET ATTRIBUTES**、**WP RESET ATTRIBUTES**コマンドを使用してもプログラムでアクセスする事ができるようになりました。詳細な情報については、**4D Write Pro属性**の章を参照して下さい。



QUERY BY ATTRIBUTE ( {aTable}{;}{conjOp ;} objectField ; attributePath ; queryOp ; value {; \*} )

引数	型	説明
aTable	テーブル	⇒ セレクションまたはレコードを返すテーブル 省略時:デフォルトテーブル
conjOp	演算子	⇒ 複数のクエリ(あれば)を連結する際に使用する結合演算子
objectField	フィールド	⇒ 属性をクエリするオブジェクトフィールド
attributePath	文字	⇒ 属性の名前またはパス
queryOp	演算子, 文字	⇒ クエリ演算子(比較演算子)
value	テキスト, Number, 日付, 時間	⇒ 比較する値
*	演算子	⇒ クエリ継続フラグ

## 説明

**QUERY BY ATTRIBUTE** は *objectField*、*attributePath*、*queryOp* そして *value* 引数を使用して定義されたクエリ文字列に合致するレコードを検索し、*aTable* に対しレコードのセレクションを返します。

**注:** オブジェクトフィールド(4D v15より新たに導入)についての詳細な情報に関しては、[オブジェクトフィールドデータ型](#)のセクションを参照して下さい。

**QUERY BY ATTRIBUTE** はカレントプロセスにおいて *aTable* で指定されたテーブルのカレントセレクションを変更し、新しいセレクションの第一レコードをカレントレコードとします。*aTable* 引数が省略されていた場合、コマンドはデフォルトのテーブルへと適用されます。デフォルトテーブルが設定されていない場合、エラーが発生します。

任意の *conjOp* 引数を使用すると、**QUERY BY ATTRIBUTE** の呼び出しを複数のクエリ定義と組み合わせることができます。使用可能な接続演算子は **QUERY** コマンドに対して使用できるものと同じです:

接続子	QUERY BY ATTRIBUTEで使用する記号
AND	&
OR	
Except	#

*conjOp* 引数は、複数のクエリの最初の **QUERY BY ATTRIBUTE** の呼び出しには使用しません。また単一のクエリの場合にも使用しません。複数のクエリにおいて省略された場合、AND (&) 演算子がデフォルトで使用されます。

*objectField* 引数には、クエリしたい属性のオブジェクトフィールドを渡します。そのオブジェクトフィールドが *aTable* 引数で指定したテーブルに自動または手動でリレートした1テーブルに属していた場合、*objectField* には他のテーブルに属するフィールドを指定することもできます。

**QUERY BY ATTRIBUTE** はドキュメントがオブジェクトフィールドに保存されていた場合、4D Write Proのカスタム属性をサポートします。この点についてのより詳細な情報については、[4D Write Proドキュメントを4Dオブジェクトフィールドに保存する](#) の章を参照して下さい。

*attributePath* 引数にはレコード毎に値を比較したい属性のパスを渡します (例: "children.girls.age")。単一の属性名 (例えば "place") を渡した場合には、オブジェクトフィールドの第一レベルでその名称に合致する属性を指定したことになります。属性"x"が配列であった場合、**QUERY BY ATTRIBUTE** コマンドは、少なくとも一つの要素が条件に合致する属性"x"を含むレコードを検索します。配列内の属性を検索するためには、*attributePath* 引数内において属性"x"の名前に"."[]を付与することにより、**QUERY BY ATTRIBUTE** コマンドに対し、属性"x"が配列であるという事を指示する必要があります(例3を参照して下さい)。

**注:**

- 属性名は大文字と小文字が区別されるという点に注して下さい。つまり同じレコード内にて "MyAtt" と "myAtt" という、異なる属性名を持つことができるということです。
- 属性名は不要な空白を取り除くために短縮されます。例えば、" my first attribute .my second attribute " は、"my first attribute.my second attribute"として解釈されます。

*queryOp* 引数は、*objectField* 引数と *value* 引数の間に適用される比較演算子です。以下の記号のどれか一つを渡す事ができます:

比較	QUERY BY ATTRIBUTEで使用する記号
同値である	=
同値でない	#
未満	<
を超える	>
以下	<=
以上	>=

**注:** 比較演算子を記号ではなく、テキスト表現で指定することも可能です。詳細な情報に関しては、**QUERY** コマンドの説明を参照して下さい。

*value* 引数は、*attributePath* 引数と比較するためのデータです。この値は*attributePath*引数と同じデータ型として評価されるものであればどんな表現も可能です。値は一度だけ、クエリの最初に評価されます。値はそれぞれのレコードに対して毎回評価されることはありません。文字列内に含まれる文字列をクエリする("を含む"クエリ)ためには、ワイルドカード記号(@)を*value* 引数に使用して検索したい文字列を隔離します(例:"@Smith@")。この場合、インデックスの利点を一部しか享受しないという点に注して下さい(データ保存のコンパクト化)。

属性によるクエリのストラクチャーは以下のようになります:

```
QUERY BY ATTRIBUTE ([Table] ; [Table]ObjectField ; "attribute1.attribute2"; =; value)
```

**注:** 全ての演算子(ただし"#"は除く)に対して、オブジェクトフィールドには属性が含まれている、というのが暗示的な前提条件になります。しかしながら、"#"演算子に対しては、未定義の属性も使用可能です(以下を参照して下さい)。

"#"演算子を使用した属性のクエリは、そのオブジェクトフィールドに対してプロパティがチェックされているかどうかで結果が異なります:

- **ヌル値を空の値にマップ**プロパティがチェックされている(デフォルトのオプション、多くの場合において推奨)  
この場合、"#"演算子はフィールド内にて検索した値がどの属性にも存在しないレコードを検索します。このコンテキストにおいては、4Dは以下の場合においても同じように対応します:
  - 属性の値が検索した値とは異なるフィールド
  - 属性が存在しない(あるいはヌル値を含む)フィールド

例えば、以下のクエリは、Rexという名前ではない犬を飼っている人のレコードに加えて、犬を飼っていない人、あるいは名前のない犬を飼っている人のレコードも返します:

```
QUERY BY ATTRIBUTE ([People] ; [People]Animals ; "dog.name" ; # ; "Rex")
```

その他の例:以下のクエリは、*[Table]ObjectField* 内で値が*value* ではない*attribute2* 属性を含むオブジェクトである*attribute1* 属性を含んでいるオブジェクトを含んでいる全てのレコードに加え、*attribute1* も *attribute2* も含まないオブジェクトフィールドのレコードを返します:

```
QUERY BY ATTRIBUTE ([Table] ; [Table]ObjectField ; "attribute1.attribute2" ; # ; value)
```

この原則は配列属性にも適用されます。例えば:

```
QUERY BY ATTRIBUTE ([People] ; [People]OB_Field ; "locations[].city" ; # ; "paris")
```

上記のクエリは、Parisに住所を持っていないPeopleのレコードを返します。

属性が未定義であるレコードだけを特定して取得したい場合、空のオブジェクトを使用して下さい(例2を参照して下さい)。しかしながら配列内にてヌル値を検索することはサポートされていない点に注意してください。

- **ヌル値を空の値にマップ**プロパティがチェックされていない("SQL"モード)  
この場合、未定義の属性(フィールド内に存在しない属性、あるいは値がヌル)はデフォルトで空の値と同じではないと判断されます。結果として、"属性Aが属性Bと異なる"というタイプのクエリは、属性Aが未定義のレコードを返しません。

[People]Animalsフィールドに対して**ヌル値を空の値にマップ**オプションがチェックされていない場合に、上記と同じ例を使用した場合、以下のクエリは名前("name"属性)が"Rex"ではない犬を持っている人のレコードのみを返します。

この場合、犬を持っていない人、あるいは名前のない犬を持っている人のレコードは返されません。

```
QUERY BY ATTRIBUTE ([People]; [People]Animals; "dog.name"; #; "Rex")
```

このSQLロジックに近いオペレーションは、特定の用途のために準備されている者です。

## 複数のクエリの作成

属性によるクエリを複数組み合わせる際には、以下のルールが適用されます:

- 最初のクエリ文字列は接続子を含んではいけません。
- その後のそれぞれのクエリ文字列は接続子から始まります。省略した場合AND(&)演算子がデフォルトで使用されます。
- 最初のクエリと、最後を除く他の全てのクエリは、\* 演算子を使用しなければなりません。
- **QUERY BY ATTRIBUTE** コマンドは**QUERY** コマンドと組み合わせて使用することができます(例を参照して下さい)。
- クエリを実行するためには、最後の**QUERY BY ATTRIBUTE** コマンドにおいて \* 引数を指定してはいけません。その代り、**QUERY** コマンドをテーブルのみで(他の引数を必要とせず)実行する事ができます。

**注:** それぞれのテーブルは現在ビルトされたクエリを維持します。これはつまり、それぞれのテーブルに対して一つずつ、複数のクエリを同時に作成できるという事です。

どのように定義されたとしても、クエリには以下の制限が付きまますNo matter which way a query has been defined:

- 実際のクエリオペレーションが実行にある程度の時間が必要になる場合、4Dは自動的に進捗バー(サーモメーター)を含めたメッセージを表示します。このメッセージは**MESSAGES ON** や **MESSAGES OFF** コマンドを使用することによってオン・オフを切り替えることができます。進捗バーが表示されているとき、ユーザーはストップボタンを押すことによってクエリを中断することができます。クエリが完了すると、OK変数が1に設定されます。それ以外の場合、例えばクエリが中断されたばあなどには、OK変数は0(ゼロ)に設定されます。
- インデックスされたオブジェクトフィールドが指定された場合、クエリは毎回可能な限り最適化されるので(インデックスされたフィールドから先に検索されます)、結果として可能な限り最小限の時間でクエリを終えることができます。

## オブジェクト内の日付の値

日付はオブジェクト内において、データベース設定に沿った形で保存されています。デフォルトでは、タイムゾーンは考慮されません(**SET DATABASE PARAMETER** コマンドのJSON use local time を参照して下さい)。

```
!1973-05-22! -> "1973-05-21T23:00:00.000Z"
```

この設定はクエリにおいても影響するので、データベースを常に毎回同じ場所で使用し、データにアクセスする全てのマシンの設定が同じであれば何も心配する必要がありません。この場合、以下のクエリは、Birthday属性が!1973-05-22!( "1973-05-21T23:00:00.00Z"として保存されている)に一致するレコードを正確に返します:

```
QUERY BY ATTRIBUTE ([Persons]; [Persons]OB_Info; "Birthday"; =; !1973-05-22!)
```

GMT設定を使用したくない場合、これらの設定を以下の様にして変更する事ができます:

```
SET DATABASE PARAMETER (JSON use local time;0)
```

ただし、この設定の範囲はプロセスのみであるという点に注意して下さい。設定をこのように変更した場合、1965年10月1日は"1965-10-01T00:00:00.000Z"として保存されますが、クエリを実行する前に同じ引数を設定する必要が出てきます:

```
SET DATABASE PARAMETER (JSON use local time;0)
QUERY BY ATTRIBUTE ([Persons]; [Persons]OB_Info; "Birthday"; =; !1976-11-27!)
```

## 仮想的長さプロパティの使用

このコマンドでは、仮想的な"長さ"プロパティを使用する事ができます。このプロパティは全ての配列型属性に対して自動的に利用可能で、配列のサイズ、つまり含まれる要素の数を返します。**QUERY BY ATTRIBUTE**コマンドを実行するコンテキストにおいて使用する事ができます(例第4を参照)。

## 例題 1

この例においては、"age"属性は文字列か整数の値であり、年齢が20歳から29歳の人を検索したい場合を考えます。最初の2行は属性を整数としてクエリし(>=20 かつ < 30)、最後の1行はフィールドを文字列としてクエリします("2" で始まるが、"2"ではない値)。

```
QUERY BY ATTRIBUTE ([Persons]; [Persons]OB_Info; "age"; >=; 20; *)
QUERY BY ATTRIBUTE ([Persons]; & ; [Persons]OB_Info; "age"; <; 30; *)
QUERY BY ATTRIBUTE ([Persons]; |; [Persons]OB_Info; "age"; =; "2@"; *)
QUERY BY ATTRIBUTE ([Persons]; & ; [Persons]OB_Info; "age"; #; "2") //実行したいので、ここでは最後の * は無い
```

## 例題 2

**QUERY BY ATTRIBUTE** コマンドを使用すると、特定の属性が定義されている(あるいは定義されていない)レコードを探す事ができます。そのためには、空のオブジェクトを使用します。

```
//オブジェクトフィールド内にてEメールが定義されているレコードを探す
C_OBJECT($undefined)
QUERY BY ATTRIBUTE ([Persons]; [Persons]Info; "e-mail"; #; $undefined)
```

```
//オブジェクトフィールド内にてZIPコード(郵便番号)が定義されていないレコードを探す
C_OBJECT($undefined)
QUERY BY ATTRIBUTE ([Persons]; [Persons]Info; "zip code"; =; $undefined)
```

**注:** この特定のシンタックスは配列型属性ではサポートされていません。配列要素内でNULL値を検索した場合には不正な結果が返されます。

## 例題 3

配列の属性を含むフィールドを検索したい場合を考えます。以下の二つのレコードがあった時、:

レコード1:

```
[People]name: "martin"
[People]OB_Field:
  "locations" : [ {
    "kind": "office",
    "city": "paris"
  } ]
```

レコード2:

```
[People]name: "smith"
[People]OB_Field:
  "locations" : [ {
    "kind": "home",
    "city": "lyon"
  }, {
    "kind": "office",
    "city": "paris"
  } ]
```

... **QUERY BY ATTRIBUTE** コマンドに対して以下の宣言をすると、locationが"paris"である人を探します:

```
//配列の属性を"."[]" シンタックスでフラグ付けする
QUERY BY ATTRIBUTE ([People]; [People]OB_Field; "locations.[] .city"; =; "paris")
// "martin" と "smith" を返す
```

**注:** 同じ配列の属性に対し複数の条件を定義した場合、合致した条件は必ずしも同じ配列の要素に対して適用されるとは限りません。以下の例においては、"kind"が"home"である"locations"要素と、"city"が"paris"である"locations"要素を持ってい

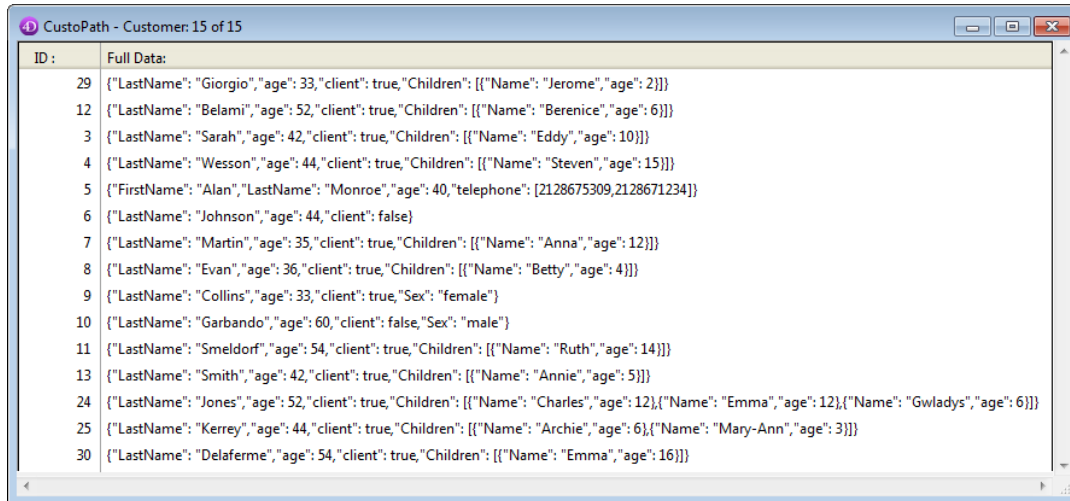


るために"smith"が返されますが、これら二つは同じ要素ではありません:

```
QUERY BY ATTRIBUTE ([People]; [People]OB_Field;"locations.[] .kind";="home";*)
QUERY BY ATTRIBUTE ([People]; & ; [People]OB_Field;"locations.[] .city";="paris")
// "smith"を返す
```

## 例題 4

この例題では、仮想的な"長さ"プロパティの使用について説明します。データベースには[Customer]full\_Dataというオブジェクトフィールドがあり、以下のデータが入っているものとします:



ID	Full Data:
29	{"LastName": "Giorgio", "age": 33, "client": true, "Children": [{"Name": "Jerome", "age": 2}]}
12	{"LastName": "Belami", "age": 52, "client": true, "Children": [{"Name": "Berenice", "age": 6}]}
3	{"LastName": "Sarah", "age": 42, "client": true, "Children": [{"Name": "Eddy", "age": 10}]}
4	{"LastName": "Wesson", "age": 44, "client": true, "Children": [{"Name": "Steven", "age": 15}]}
5	{"FirstName": "Alan", "LastName": "Monroe", "age": 40, "telephone": [2128675309, 2128671234]}
6	{"LastName": "Johnson", "age": 44, "client": false}
7	{"LastName": "Martin", "age": 35, "client": true, "Children": [{"Name": "Anna", "age": 12}]}
8	{"LastName": "Evan", "age": 36, "client": true, "Children": [{"Name": "Betty", "age": 4}]}
9	{"LastName": "Collins", "age": 33, "client": true, "Sex": "female"}
10	{"LastName": "Garbando", "age": 60, "client": false, "Sex": "male"}
11	{"LastName": "Smeldorf", "age": 54, "client": true, "Children": [{"Name": "Ruth", "age": 14}]}
13	{"LastName": "Smith", "age": 42, "client": true, "Children": [{"Name": "Annie", "age": 5}]}
24	{"LastName": "Jones", "age": 52, "client": true, "Children": [{"Name": "Charles", "age": 12}, {"Name": "Emma", "age": 12}, {"Name": "Gwladys", "age": 6}]}
25	{"LastName": "Kerrey", "age": 44, "client": true, "Children": [{"Name": "Archie", "age": 6}, {"Name": "Mary-Ann", "age": 3}]}
30	{"LastName": "Delaferme", "age": 54, "client": true, "Children": [{"Name": "Emma", "age": 16}]}

子供が二人以上いる顧客のレコードを取得したい場合を考えます。この場合、以下のように書くことができます:

```
QUERY BY ATTRIBUTE ([Customer]; [Customer]full_Data;"Children.length";>=;2)
```

## システム変数およびセット

クエリが正しく実行されると、OKシステム変数が1に設定されます。以下の場合は0に設定されます:

- クエリ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した。
- "QUERY and LOCK"モード (**SET QUERY AND LOCK**コマンド参照)、クエリが一つ以上のロックされたレコードを見つけた。この場合、LockedSetシステムセットが更新されます。

WP EXPORT DOCUMENT ( wpDoc ; filePath {; format {; options}} )

引数	型		説明
wpDoc	Object	→	4D Write Pro変数
filePath	文字	→	書き出したファイルのパス
format	倍長整数	→	ドキュメント出力フォーマット
options	倍長整数	→	出力オプション

### 説明

---

**WP EXPORT DOCUMENT** コマンドは *wpDoc* 引数で指定した4D Write Pro オブジェクトを *filePath* 引数や他の任意の引数で定義されたディスク上のドキュメントへと書き出します。

より詳細な情報に関しては、*4D Write Pro* リファレンスマニュアル内の **4D Write Proランゲージ** の章内でのこのコマンドの詳細を参照して下さい。



OB Get ( object ; property {; type} ) -> 戻り値

引数	型	説明
object	Object, Object Field	→ 構造化されたオブジェクト
property	テキスト	→ 情報を取得したいプロパティ名
type	倍長整数	→ 値を変換したい型
戻り値	Object, テキスト, ブール, ポインター, 実数, 日付	→ プロパティのカレントの値

## 説明

**OB Get** コマンドは、*object* 引数で指定したオブジェクトの *property* のカレントの値を返します。任意の *type* 引数で指定した型へと変換することもできます。

*object* で指定するオブジェクトは、**C\_OBJECT** コマンドを使用して作成されている、あるいはオブジェクトフィールドが選択されている必要があります。

**注:** このコマンドは (**WP GET ATTRIBUTES** コマンドと同様に) 4D Write Pro *objects* に定義されている属性をサポートします (例題9を参照ください)。しかしながら、**WP GET ATTRIBUTES** とは異なり、**OB Get** はピクチャー変数やフィールドを属性値として直接扱うことはできません。

*property* 引数には、情報を取得したいプロパティのラベルを渡します。*property* 引数では、大文字と小文字は区別されることに注意して下さい。

特に指定がなければ、4D はプロパティの値を本来の型のまま返します。このとき、*type* 引数を使用することによって返ってくる値の型を強制的に変換することができます。この場合、*type* 引数には以下の定数のどれか一つを渡して下さい。これらの定数は **Field and Variable Types** テーマ内にあります。

定数	型	値
Is Boolean	倍長整数	6
Is date	倍長整数	4
Is longint	倍長整数	9
Is object	倍長整数	38
Is pointer	倍長整数	23
Is real	倍長整数	1
Is text	倍長整数	2
Is time	倍長整数	11

コマンドは *property* 引数で指定されたプロパティの値を返します。いくつかのデータの型がサポートされています。以下のことに注意して下さい:

- ポインターはそのまま返されますが、**JSON Stringify** コマンドを使用することによってポインターが指してる値に変換することができます。
- 日付は "YYYY-MM-DDTHH:mm:ss.SSSZ" というフォーマットで返されます。
- 実数の値において、小数点はかならずピリオド (".") になります。
- 時間は数字で表現されます。4Dは時間を秒単位での数字を受け取る一方、**OB SET** はJavaScriptスタンダードに合致するためにミリ秒単位で保存します。**OB Get** コマンドが保存された時間を正確に解釈するためには、Is time 定数を使用する必要があります。

## 例題 1

テキスト型の値を取得する場合:

```
C_OBJECT($ref)
C_TEXT($FirstName)
OB SET($ref;"FirstName";"Harry")
$FirstName:=OB Get($ref;"FirstName") // $FirstName = "Harry" (text)
```

## 例題 2

実数型の値を取得して、倍長整数へと変換する場合:

```
OB SET($ref ;"age";42)
$age:=OB Get($ref ;"age") // $age は実数型 (デフォルト)
$age:=OB Get($ref ;"age";Is_longint) // $age を倍長整数型に変換
```

## 例題 3

オブジェクトの中の値を取得する場合:

```
C_OBJECT($ref1;$ref2)
OB SET($ref1;"LastName";"Smith") // $ref1={"LastName":"Smith"}
OB SET($ref2;"son";$ref1) // $ref2={"son":{"LastName":"Smith"}}
$son:=OB Get($ref2;"son") // $son={"LastName":"john"} (object)
$sonsName:=OB Get($son ;"name") // $sonsName="john" (text)
```

## 例題 4

従業員の年齢を二度修正したい場合:

```
C_OBJECT($ref_john;$ref_jim)
OB SET($ref_john;"name";"John";"age";35)
OB SET($ref_jim;"name";"Jim";"age";40)
APPEND TO ARRAY($myArray;$ref_john) // オブジェクト配列を作成
APPEND TO ARRAY($myArray;$ref_jim)
// John の年齢を35から25 へと修正
OB SET($myArray{1};"age";25)
// 配列内で"John" の年齢を修正
For($i;1;Size of array($myArray))
  If(OB Get($myArray{$i};"name")="John")
    OB SET($myArray{$i};"age";36) // 25 から 36 へと修正
  // $ref_john={"name":"John","age":36}
  End if
End for
```

## 例題 5

ISO 形式のデータを日付型で取り出す場合:

```
C_OBJECT($object)
C_DATE($birthday)
C_TEXT($birthdayString)
OB SET($object;"Birthday";"1990-12-25T12:00:00Z")
$birthdayString:=OB Get($object;"Birthday")
// $birthdayString="1990-12-25T12:00:00Z"
$birthday:=OB Get($object;"Birthday";Is_date)
// $birthday=1990/12/25
```

## 例題 6

入れ子にされたオブジェクトを使用することもできます:

```
C_OBJECT($ref1;$child;$children)
```

```

C_TEXT($childName)
OB SET($refl;"firstname";"John";"lastname";"Monroe")
  //{"firstname":"john","lastname":"Monroe"}
OB SET($children;"children";$refl)
$child:=OB Get($children;"children")
  //$son = {"firstname":"John","lastname":"Monroe"} (object)
$childName:=OB Get($child;"lastname")
  //$childName = "Monroe" (text)
  // または
$childName:=OB Get(OB Get($children;"children");"lastname")
  // $childName = "Monroe" (text)

```

## 例題 7

オブジェクトに保存された時間を4Dに復元する例を考えます:

```

C_OBJECT($obj_o)
C_TIME($set_h;$get_h)

$set_h:=?01:00:00?+1
OB SET($obj_o;"myHour";$set_h)
  // $obj_o == {"myHour":3601000}
  // 時間はミリ秒単位で保存されています
$get_h:=OB Get($obj_o;"myHour";Is_time)
  // $get_h == ?01:00:01?
  // 時間は正確に解釈されるようになりました

```

## 例題 8

4Dオブジェクトフィールドの使用例です:

```

// 値を定義する
OB SET([People]Identity_OB;"First name";$firstName)
OB SET([People]Identity_OB;"Last name";$lastName)

// 値を取得する
$firstName:=OB Get([People]Identity_OB;"First name")
$lastName:=OB Get([People]Identity_OB;"Last name")

```

## 例題 9

4D Write Proエリアを含むフォームメソッド内に、以下のように書く事ができます:

```

If(Form event=On_Validate)
  OB SET([MyDocuments]My4DWP;"myatt_Last edition by";Current user)
  OB SET([MyDocuments]My4DWP;"myatt_Category";"Memo")
End if

```

ドキュメントのカスタム属性を読み出す事もできます:

```

vAttrib:=OB Get([MyDocuments]My4DWP;"myatt_Last edition by")

```

## 4D Write Pro エリアを使用する

### 4D Write Pro エリア内のドキュメントを管理する

4D アプリケーションでは、4D Write Pro テーマ内にある特定のコマンドを使用することによって、4D Write Pro ドキュメントを作成、読み込み、書き出しすることができます (**WP EXPORT DOCUMENT**、**WP EXPORT VARIABLE**、**WP Import document**、**WP New**)。

4D Write Pro エリアをデータベースのフォーム内のオブジェクトフィールドと関連づけることができます。これにより、各 4D Write Pro ドキュメントはレコードとともに自動的に保存され、データベースのデータ内に保存されます (**4D Write Pro ドキュメントを4Dオブジェクトフィールドに保存する** 参照)。

### .4wp ドキュメントフォーマット

**.4wp** を使用する事によって 4D Write Pro ドキュメントをロスなくディスク上に保存したりディスクから開いたりすることができます。

**.4wp** フォーマットはドキュメント名と同じ名前を持つ ZIP フォルダーと、その中に格納された HTML テキストと画像から構成されます：

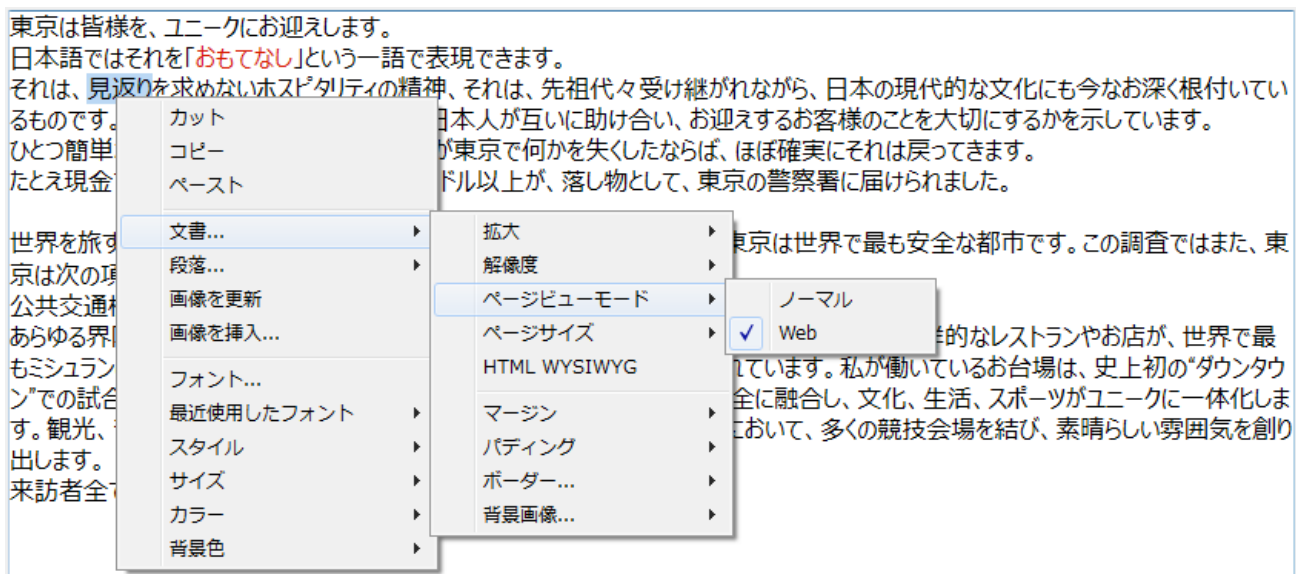
- HTML テキストは、通常の HTML と (計算されていない) 4D 式に加え、4D 特有のタグを組み合わせます。
- 画像は、HTML ファイルの隣にある、ドキュメント名と同じ名前を持つフォルダー内に保存されています。

.4wp ドキュメントは HTML に基づいているので、HTML をサポートしているものであればどんな外部アプリケーションでもそれを読み込んだり開いたりすることができます。

**注:** 4D Write Pro の内部ドキュメントは HTML に 4D 独自の拡張を加えたもので、HTML5/XHTML5 に準拠していながら、このマニュアルで説明されている独自の HTML/CSS 属性のサブセットとタグもサポートしています。したがってデータ損失のリスクなく 4D Write Pro で開けるのは、4D Write Pro で書き出された HTML ドキュメントに限られます。外部で作成された HTML ドキュメントの読み込みはエラーを生成する可能性があります。

### ユーザーインターフェース

4D Write Pro エリア内にて**コンテキストメニュー**プロパティがチェックされている場合、(**4D Write Pro エリアを定義する**を参照して下さい)、アプリケーションモードにおいてユーザーは包括的なコンテキストメニューを使用することができます：



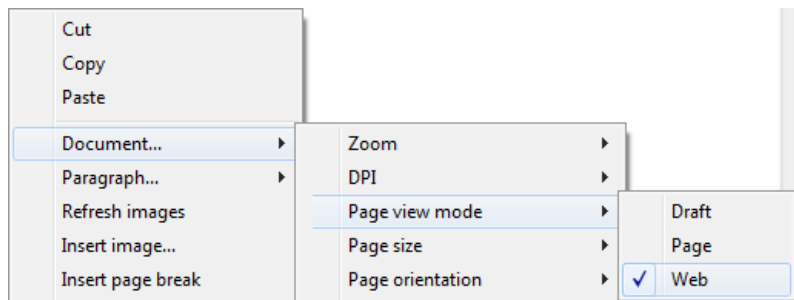
このメニューを通して、提供されているすべての 4D Write Pro 機能へとアクセスすることができます。

### ビューモードの選択

4D Write Proドキュメントは三種のページビューモードで表示することができます:

- **下書き**: 基本的なプロパティを持つ下書きモード
- **ページ** (デフォルト): "印刷ビュー" モード
- **埋め込み**: 埋め込みエリアに最適なビューモードです。このモードでは余白、フッター、ヘッダー、ページフレーム等は表示されません。  
このモードはWeb風のビュー出力を作成するのにも使用できます (96 dpi解像度と **HTML WYSIWYG** オプションを選択した場合)。

ページビューモードはエリアポップアップメニューを使用することで設定が可能です:



**注:** ページビューモードはドキュメントには保存されません。

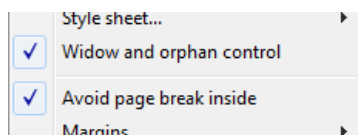
4Dフォームに埋め込まれたエリアについては、プロパティリストを使用してデフォルトのビューモードを設定することができます。この場合、ビューモードは 4D Write Proフォームオブジェクトのプロパティとして保存されます (より詳細な情報については、[ビュープロパティの設定](#) の章を参照してください)。

## ページビュー機能

ドキュメントがページビューモードのとき、以下のドキュメントプロパティがユーザーに表示されます:

- 印刷範囲を示すページのアウトライン
- ページ幅とページの高さ (デフォルトは21x29.7 cm)
- ページの向き (デフォルト: 縦向き)
- ページの余白 (デフォルト: 2.5cm)

これに加えて、**段落...**サブメニューでは新しい段落プロパティが利用可能です:



- **ウィドウ&オーファンコントロール**: このオプションが段落に対して設定されているとき、4D Write Pro はドキュメント内にてウィドウ (段落の最終行がページ上部に取り残される状態) とオーファン (段落の先頭行がページ下部に取り残される状態) を許可しません。前者の場合には、最終行の一つ前の行を加えた 2行がページ上部に表示されます。後者の場合には、単一の先頭行は次ページへと送られます。
- **段落内の改ページを避ける**: このオプションが段落に対してチェックされている場合、4D Write Pro はその段落が 2枚以上のページに分割されないようにします。

また、コンテキストメニューから以下のコマンドを使用する事もできます:

- **改ページを挿入**: カーソルの位置に改ページを追加します。テキストがいくつか選択されていた場合、それらは改ページで置換されます。
- **文書.../ページサイズ**: ページサイズの選択ができます。様々な標準のページサイズが提示されます。
- **文書.../ページの向き**: 標準の向きプロパティ (縦方向 / 横方向)

**注:** ドキュメントが **埋め込み**モードまたは**下書き**モードのときにページプロパティを変更しても見た目上は変化ありませんが、それでも設定は可能です。以下の段落プロパティは、**下書き**モードでも見た目上の変化があります:

- ページの高さ制限 (線が引かれます)
- 内側での改ページを避けるプロパティ
- ウィドウ&オーファンコントロール

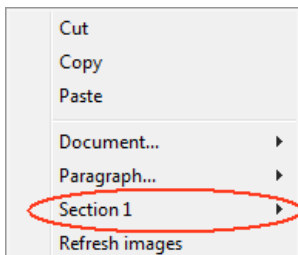
## ヘッダー、フッター、セクションの管理

4D Write Pro ドキュメントはヘッダーとフッターをサポートします。ヘッダーとフッターはセクションと関連しています。セクションとは、ページレンジによって定義されたドキュメントの一部で、ページングと属性の設定を独自に持つことができます。ドキュメントには複数のセクションを含めることができます (1から合計のページ数まで)。それぞれのページは一つのセクションにのみ含めることができます。各セクションに対して、ヘッダーとフッターのセットを定義することができます。

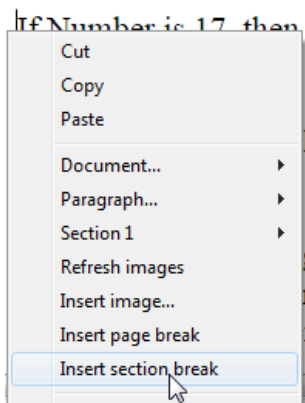
### セクションの定義

セクションとは4D Write Proドキュメント内の継続したページのサブセットです。ドキュメントは1つ以上のセクションを含むことができます。ひとつのセクションには何ページでも (単一のページからドキュメント内の総ページ数まで) 含めることができます。

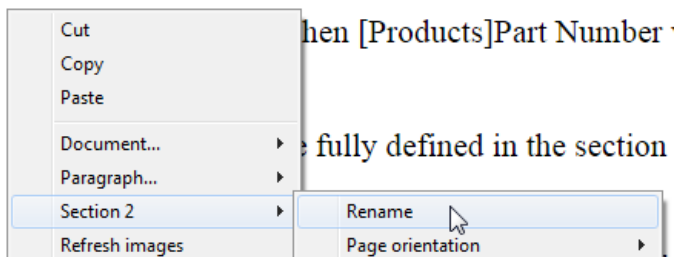
デフォルトで、ドキュメントは**セクション1**と名付けられたひとつのセクションを持ちます。ドキュメント内のどこをクリックしても、4D Write Proコンテキストメニューにそのセクション番号を表示します:



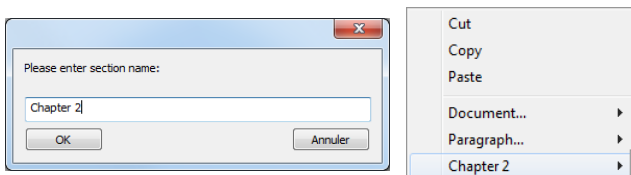
テキストの中にセクションブレイクを追加することで新しいセクションを作成することができます:



セクションブレイクが追加されたとき、コンテキストメニューはそれぞれのセクションに対しインクリメントされた数字を表示します。セクション名は任意に改名することも可能です:



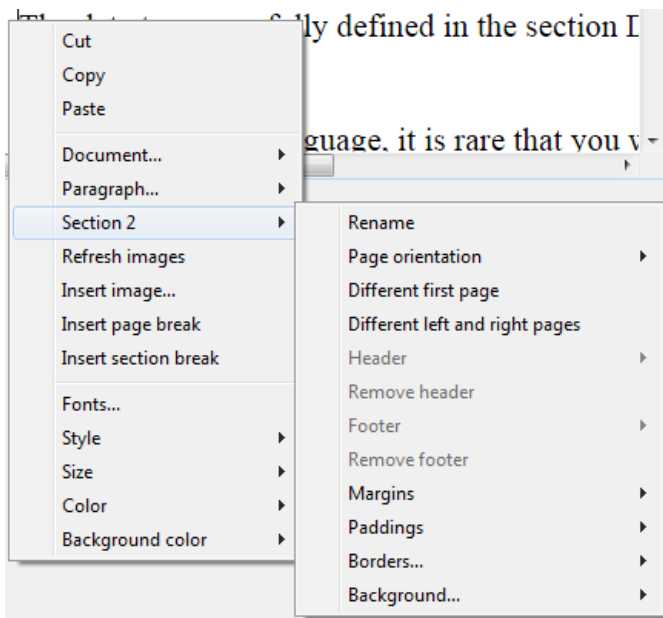
入力した名前はそのセクションの名前として、ドキュメント全体で使用されます:



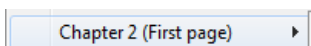
セクションに対し、異なる最初のページ、異なる左/右ページを定義していた場合、そのページタイプもまたメニューに表示されるといふ点に注意して下さい (後述参照)。

### セクション属性

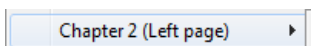
セクションはドキュメント属性を継承しますが、ヘッダーとフッターを含むドキュメント属性は、セクションごとに個別に編集することも可能です。コンテキストポップアップメニューは、セクションレベルで利用可能なプロパティと属性を表示します:



- **ページの向き:** セクションごとに特定のページの向き (横向き/縦向き)を設定することができます。
- **最初のページを個別に設定:** セクションの最初のページに対して異なる属性を設定することができます。この機能は例えば、余白を作成するのに使用することができます。この属性がチェックされている場合、セクションの最初のページはセクションのサブセクションとして管理され、独自の属性を持つことができます。



- **左右のページを個別に設定:** セクションの左ページと右ページで異なる属性を設定することができます。この属性がチェックされていると、セクションの左ページ/右ページはサブセクションとして管理され、それぞれ独自の属性を持つことができます。



- **ヘッダー と フッター コマンド:** これらのオプションを使用すると個別のヘッダーとフッターを定義することができます。これらのオプションは以下に詳細な説明があります。
- **マージン / パディング / 境界線 / 背景:** これらの属性は各セクションに対して個別に設定が可能です。これらの属性のより詳細な情報については、[4D Write Pro属性](#)の記事を参照してください。

## ヘッダーとフッターの挿入

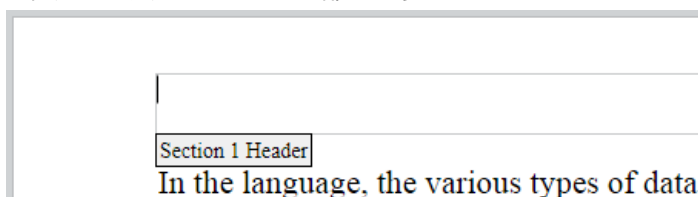
それぞれのセクションには固有のヘッダーとフッターをつけることができます。ヘッダーとフッターはページビューモードがページのときのみ表示されます。

有効化されたオプションによって、ひとつのセクションにつき最大3つまでヘッダーとフッターを定義することが可能です:

- 最初のページ
- 左ページ
- 右ページ

ヘッダーまたはフッターを作成する手順は以下の通りです:

1. まずドキュメントが**ページビューモード**であることを確認してください。
2. 目的のセクションのヘッダーまたはフッターエリアをダブルクリックして、**編集モード**に入ります。
  - ヘッダーエリアはページの上部にあります:



- フッターエリアはページの下部にあります:



date, 4D will assume that you want to

Section 1 Footer

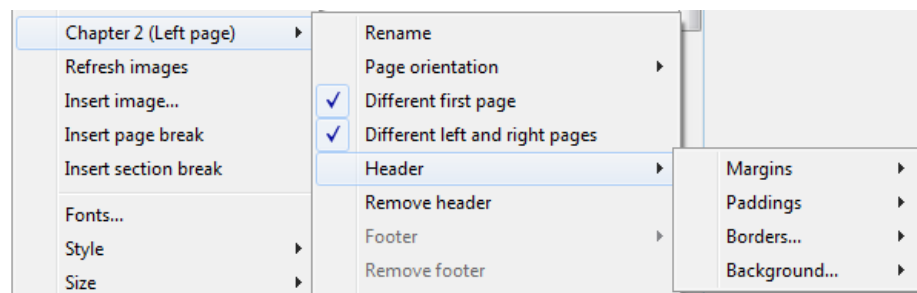
スタティックなコンテンツが入力すると、その入力した内容は各ページのセクションにて自動的に繰り返されます (ただし最初のページが別設定の場合には、そこには反映されません)。

Chapter 2 Footer (Left page)

Chapter 2 - How to become famous

**ST INSERT EXPRESSION** コマンドを使用することで、ページ番号やページ数などの動的なコンテンツを挿入することもできます (より詳細な情報については、**式の挿入**の段落を参照してください)。

セクションのヘッダーとフッターが定義できたら、コンテキストメニューを使用してそれらの属性を設定することができます:



**マージン**、**パディング**、**境界線**、そして**背景**属性についてのより詳細な情報については、**4D Write Pro属性**の章を参照してください。

コンテキストメニューの**ヘッダーを削除**あるいは**フッターを削除**コマンドを選択すると、ヘッダーまたはフッターの定義全体 (コンテンツと属性) を削除することができます。

## 互換性

4D Write Proは、4D Writeプラグインから変換されたドキュメントのヘッダーとフッターも扱うことができます。

4D Writeプラグインのヘッダーとフッターから引き続きサポートされ変換される式とプロパティは、以下の通りです:

- ページ番号とページ数変数
- 個別の最初のページ
- 個別の左/右ページ



WP EXPORT DOCUMENT ( wpDoc ; filePath {; format {; option}} )

引数	型		説明
wpDoc	Object	→	4D Write Pro 変数
filePath	文字	→	書き出されたファイルへのパス
format	倍長整数	→	ドキュメントの出力フォーマット
option	倍長整数	→	書き出しオプション

## 説明

**WP EXPORT DOCUMENT** コマンドは *wpDoc* 引数に渡したディスク上の4D Write Proオブジェクトを、*filePath* 引数で指定した場所に、他の任意の引数に基づいて書き出します。

*wpDoc* 引数には、書き出したい4D Write Proオブジェクトを渡します。

*filePath* 引数には、書き出し先の場所と、書き出したドキュメント名を渡します。ドキュメント名のみを渡した場合、4Dストラクチャーファイルと同階層に保存されます。デフォルトで、*format* 引数を省略した場合、コマンドはドキュメントの拡張子を見て書き出しのフォーマットを選択します。

*format* 引数には**4D Write Pro** テーマからの定数を渡す事もできます。この場合、4Dは必要に応じて適切な拡張子をファイル名に追加します。サポートされるフォーマットは以下の通りです:

定数	型	値	コメント
wk 4wp	倍長整数	4	4D Write Proドキュメントは、ネイティブなアーカイブフォーマット(zipになったHTMLと、別のフォルダに保存された画像)にて保存されています。4D特有のタグも含まれ、4D式は計算はされていません。このフォーマットは、特に4D Write Proドキュメントをディスク上に損失なく保存してアーカイブする事に適しています。
wk web page complete	倍長整数	2	.htm または .html 拡張子。このドキュメントは標準HTMLとして保存され、そのリソースは別に保存されます。4Dタグは除去され、式は値が計算されます。このフォーマットは特に4D Write ProドキュメントWeb ブラウザで表示したい場合に特に適しています。

### 注:

- "4D 特有のタグ"とは、4Dネームスペースと4D CSSスタイルを含めた4D XHTMLのことです。
- 式は、**ST FREEZE EXPRESSIONS**コマンドを使用することで、書き出す前でもいつでも値を確定させることができます。
- 4D Write Proフォーマットに関するより詳細な情報については、**.4wp ドキュメントフォーマット**を参照して下さい。

*options* 引数には、書き出しの設定を指定するオプションを渡します。倍長整数 の値を渡す事で、HTMLコードのスタイルを定義することができます。使用できる定数は以下の通りです:

定数	型	値	コメント
wk html debug	倍長整数	1	フォーマット済みのHTMLコード("整形済みフォーマット")。デバッグが容易。
wk normal	倍長整数	0	標準のHTML コード

- **HTML デバッグオプションなし(デフォルト):**

```
<html xmlns="http://www.w3.org/1999/xhtml"> <head> <title> New 4D Write Pro Document</title> <style type="text/css"> body { background-color:#FFFFFF }ul, ol { margin:0;padding:0 }p,li { white-space:pre-wrap;margin:0pt;padding:0pt;font-family:'Times New Roman' }p.Normal,li.Normal { text-align:left }p_p1,li_p1 { font-family:'Arial';font-size:18pt;color:#1D1D1D }img_img1 { width:51pt;height:51pt }</style> </head> <body> <p class="Normal_p1"> <span>23/12/14</span>,</p> <p class="Normal_p1">Dear <span style="color:#BE0E00">Shaun Stoltz</span> <span style="color:#800062"> <span style="color:#000000">from
```

- **HTML デバッグオプションあり:**

```
<html xmlns="http://www.w3.org/1999/xhtml"> <head> <title> New 4D Write Pro Document</title> <style type="text/css"> body { background-color:#FFFFFF }ul, ol { margin:0;padding:0 }
```

## 例題 1

---

myArea という4D Write Proオブジェクトの中身をデータベースフォルダー内のドキュメントへと書き出したい場合を考えます。'pprint'ボタンを使用してデバッグオプションを設定することができます:

```
C_TEXT($filePath)
$filePath:=Get 4D folder(Database folder)+"Exported files"+Folder
separator+"WriteProExport.html"
If(pprint=0) //デバッグオプションがoffの場合
    WP EXPORT DOCUMENT(myArea;$filePath;wk_web_page_complete;wk_normal)
Else
    WP EXPORT DOCUMENT(myArea;$filePath;wk_web_page_complete;wk_html_debug)
End if
```

## 例題 2

---

myArea 4D Write Proオブジェクト内のコンテンツを、.4wpフォーマットで書き出したい場合を考えます:

```
C_TEXT($path)
C_LONGINT($docRef)

Case of
: (Form event=On Clicked)

    $path:=Get 4D folder(Database folder)+"Export"+Folder separator
    $path:=Select document($path;".4wp";" title";File name entry)

    If($path#"")
        WP EXPORT DOCUMENT(myArea;document;wk_4wp;wk_normal)
    Else
        ALERT("An error occurred.")
    End if
End case
```

ST INSERT EXPRESSION ( { \* ; } object ; expression { ; startSel { ; endSel } } )

引数	型	説明
*	演算子	⇒ 指定時: object はオブジェクト名(文字列) 省略時: object は変数またはフィールド
object	Object	⇒ オブジェクト名( * 指定時)、または変数やフィールド( * 省略時)
expression	テキスト	⇒ 挿入したい4D式と(任意の)フォーマット
startSel	倍長整数	⇒ 選択範囲の開始地点
endSel	倍長整数	⇒ 選択範囲の終了地点

### 説明

**ST INSERT EXPRESSION** コマンドは、*object* 引数で指定したスタイル付テキストフィールドまたはテキスト変数内に、4D 式への参照を挿入します。

任意の \* 演算子を渡した場合、*object* 引数でオブジェクト名を文字列で指定します。省略時には *object* 引数でフィールドまたは変数を指定します。

*expression* 引数には、オブジェクト内にて評価したい4D式を渡します。有効な4D式とは、値を返す文字列です。*expression* 引数には、フィールド、変数、4Dコマンド、値を返す宣言、プロジェクトメソッド等を指定することができます。

4D式は、引用符(" ")で囲まれている必要があります。

**注:** *expression* 引数にピクチャ型の変数を渡すことはできません。

*expression* で指定した4D式から返ってきた値がキャリッジリターンとタブを含んでいた場合、4D式が入っているオブジェクトに合わせてそのテキストを表示します。例えばこのとき、キャリッジリターンは改行として扱われます。

*expression* 引数に、任意のフォーマット情報を渡すことによって、4D式のフォーマット形式を指定することが出来ます。この場合、以下の書式で指定する必要があります:

```
"String(value;format)"
```

*value* 引数には4D式そのものを渡し、*format* 引数には適用したい書式を渡します。*format* には以下のどれかを指定することができます:

- 数字に対しては、(存在するかしないかに関わらず)いかなる表示形式も指定することができます。例えば、"###,##"のような形です。
- 日付に対しては、存在する日付のフォーマットを指定する数字を渡すことができます。このとき、"**Date Display Formats**" テーマ内にある定数([System date short](#) 等)を使用することができます。
- 時刻に対しては、存在する時刻のフォーマットを指定する数字を渡すことができます。このとき、"**Time Display Formats**" テーマ内にある定数([System time short](#) 等)を使用することができます。

例えば、以下の様な形になります:

```
"String([Table_1]Field_1;System date short)"
```

特に何も指定しなければ、*expression* の **値** がマルチスタイルテキストエリアに表示されます。また、**ST SET OPTIONS** コマンドを使用することによって強制的に**参照**を表示させることもできます。

任意の *startSel* 引数と *endSel* 引数は *object* 内のテキストの選択範囲を指定します。*startSel* と *endSel* は標準テキストのみをカウントし、スタイルタグは文字数としてはカウントされません。参照は一文字としてカウントされるということに注意して下さい。

- *startSel* のみを指定した場合、式の結果が指定された位置に挿入されます。
- *startSel* と *endSel* の両方を省略した場合、式の結果がカーソルの位置に挿入されます。
- *startSel* と *endSel* の両方を渡した場合、**ST INSERT EXPRESSION** コマンドは指定された範囲内のテキストを *expression* からの結果で置換します。*endSel* の値が *object* 内の総文字数より大きい場合は、*startSel* の位置からテキストの終わりまでの全てのテキストが *expression* からの結果で置換されます。

4D では、選択範囲を自動的に指定するために *startSel* 引数と *endSel* 引数に渡す定数をご用意しています。これらの定数は "Multistyle Text" テーマ内にあります:

定数	型	値	コメント
ST End highlight	倍長整数	- 1001	オブジェクト内のカレントのテキストセレクションの最後の文字を指定します(*)
ST End text	倍長整数	0	オブジェクトに含まれるテキストの最後の文字を指定します。
ST Start highlight	倍長整数	- 1000	オブジェクト内のテキストのカレントセレクションの最初の文字をを指定します (*)
ST Start text	倍長整数	1	オブジェクト内に含まれるテキストの最初の文字を指定します。

(\*) これら二つの定数を使用する際には、*object* 引数にオブジェクト名を渡してあげる必要があります。フィールドへの参照または変数を渡した場合、コマンドはオブジェクト内の全てのテキストに適用されます。

**注:** もし *startSel* 引数が *endSel* 引数より大きい場合、コマンドは何も行わず、OK変数は0に設定されます(ただし *endSel* が0である場合を除く)。

## 例題

選択したテキストをプロジェクトメソッドの結果で置き換える場合を考えます:

```
ST INSERT EXPRESSION (*;"myText";"MyMethod";ST Start highlight;ST End highlight)
```

## ☰ 4D Write Pro属性

---

4D Write Pro属性を使用するとドキュメント内に保存されているテキストと画像の全てのグラフィックな要素を管理する事ができます。これらの属性は以下のコマンドによって管理されます:

- **WP SET ATTRIBUTES**
- **WP GET ATTRIBUTES**
- **WP RESET ATTRIBUTES**

**注:** 一般的な4Dコマンド**OB SET** と **OB Get** を使っても4D Write Proエリア属性を扱うことはできますが、ピクチャーを直接使用することに関しては制約が付きまます(より詳細な情報については、これらのコマンドの説明を参照して下さい)。

## 背景

---

背景属性を使用すると、ドキュメント内の背景画像エフェクトを定義する事ができます。この適用対応表は以下のようになります:

ドキュメント	段落	文字	ピクチャー
○	○	○	○

定数	コメント
	背景の塗りエリアを指定します。取り得る値:
wk background clip	<ul style="list-style-type: none"> <li>• <a href="#">wk border box</a> (デフォルト): 背景は境界線の外側の端まで塗られます。</li> <li>• <a href="#">wk content box</a>: 背景はコンテンツボックスの内部まで塗られます。</li> <li>• <a href="#">wk padding box</a>: 背景はパディングの橋野外側まで塗られます(もしくはあった場合には境界線の内側まで)。</li> </ul>
	要素の背景色を指定します。取り得る値:
wk background color	<ul style="list-style-type: none"> <li>• CSS カラー("#010101" または "#FFFFFF" または "red")</li> <li>• 4D カラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>• R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列</li> </ul> <p>ドキュメントのデフォルトは"#FFFFFF" と <a href="#">wk transparent</a>で、段落と画像に対しては"transparent"がデフォルトです。</p> <p>背景として使用する画像を指定します。設定可能な値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 画像URL(文字列)。絶対パスまたはストラクチャーファイルあらの相対パスが可能です。</li> <li>• ピクチャー変数またはフィールド</li> </ul> <p>(<b>WP GET ATTRIBUTES</b>によって)返された値: URI(ネットワークURLまたはデータURI)。ネットワークURLで参照されない画像については、最初のURLとは同じでないかもしれません(ネットワークURLのみが保持されます)。ローカルファイルのURLについては、画像ストリームそのものがドキュメント内に保存されるので、返されたURLはデータURIとbas64でエンコードされた画像ストリームです。</p> <p>背景画像どのような位置におかれるかを指定します。取りうる値は以下の通りです:</p>
wk background image	
wk background origin	<ul style="list-style-type: none"> <li>• <a href="#">wk padding box</a> (デフォルト): 背景画像はパディング(または内側の境界線の端)からスタートします。</li> <li>• <a href="#">wk border box</a>: 背景画像は境界線(外側の境界線の端)の四角形からスタートします。</li> <li>• <a href="#">wk content box</a>: 背景画像はコンテンツ四角形からスタートします。</li> </ul> <p>背景画像の水平方向のスタート位置を指定します。取り得る値は以下の通りです:</p>
wk background position h	<ul style="list-style-type: none"> <li>• <a href="#">wk left</a> (デフォルト): 背景画像は水平方向に対して要素の左側からスタートします。</li> <li>• <a href="#">wk center</a>: 背景画像は水平方向に対して要素の中央からスタートします。</li> <li>• <a href="#">wk right</a>: 背景画像は水平方向に対して要素の右側からスタートします。</li> </ul> <p>背景画像の垂直方向のスタート位置を指定します。取り得る値は以下の通りです:</p>
wk background position v	<ul style="list-style-type: none"> <li>• <a href="#">wk top</a> (デフォルト): 背景画像は垂直方向に対して要素の上側からスタートします。</li> <li>• <a href="#">wk middle</a>: 背景画像は垂直方向に対して要素の中央からスタートします。</li> <li>• <a href="#">wk bottom</a>: 背景画像は垂直方向に対して要素の下側からスタートします。</li> </ul> <p>背景画像をどのように繰り返すか(あるいは繰り返さないか)を指定します。取り得る値は以下の通りです:</p>
wk background repeat	<ul style="list-style-type: none"> <li>• <a href="#">wk repeat</a> (デフォルト): 背景画像は垂直方向にも水平方向にも繰り返されます。</li> <li>• <a href="#">wk no repeat</a>: 背景画像は繰り返されません。</li> <li>• <a href="#">wk repeat x</a>: 背景画像は水平方向にのみ繰り返されます。</li> <li>• <a href="#">wk repeat y</a>: 背景画像は垂直方向にのみ繰り返されます。</li> </ul> <p>背景画像の水平方向のサイズを指定します。取り得る値は以下の通りです:</p>
wk background size h	<ul style="list-style-type: none"> <li>• <a href="#">wk auto</a> (デフォルト): 背景画像はその幅を保ちます。</li> <li>• <a href="#">wk contain</a>: 背景画像のアスペクト比を保ったまま、画像全体が表示される範囲で最大のサイズへと拡大/縮小します。このオプションは他のサイズ属性の値も変更します。</li> <li>• <a href="#">wk cover</a>: 背景画像のアスペクト比を保ったまま、背景全体を画像でカバーできる大きさまで拡大/縮小します。このとき背景画像の一部は見切れる可能性があります。このオプションは他のサイズ属性の値も変更します。</li> <li>• 定義されたサイズ: 背景画像の水平方向のサイズは実数値あるいは文字列値を使用して表現されます:</li> </ul>

- 実数: サイズは`wk layout unit`で指定します。
- 文字列: 連結された値と単位のCSS文字列。例: `12pt`は12ポイント、`1.5cm`は1.5センチを意味します。  
最小値: `0pt`、最大値: `1000pt`。相対値(パーセンテージ、%)はサポートされています。

背景画像の垂直方向のサイズを指定します。取り得る値は以下の通りです:

`wk  
background  
size v`

- `wk auto` (デフォルト): 背景画像はその高さを保ちます。
- `wk contain`: 背景画像のアスペクト比を保ったまま、画像全体が表示される範囲で最大のサイズへと拡大/縮小します。このオプションは他のサイズ属性の値も変更します。
- `wk cover`: 背景画像のアスペクト比を保ったまま、背景全体を画像でカバーできる大きさまで拡大/縮小します。このとき背景画像の一部は見切れる可能性があります。このオプションは他のサイズ属性の値も変更します。
- 定義されたサイズ: 背景画像の垂直方向のサイズを実数値あるいは文字列値を使用して表現します:
  - 実数: サイズは`wk layout unit`で指定します。
  - 文字列: 連結された値と単位のCSS文字列。例: `12pt`は12ポイント、`1.5cm`は1.5センチを意味します。  
最小値: `0pt`、最大値: `1000pt`。相対値(パーセンテージ、%)はサポートされています。

## 境界線

---

境界線属性を使用すると、要素の境界線のスタイル、幅、カラーを指定できます。この属性の適用対応表は以下のようになっています:

ドキュメント	段落	文字	ピクチャー
○	○	○	

定数	コメント
	境界線4つともにカラーを設定します。取りうる値は以下の通りです:
wk border color	<ul style="list-style-type: none"> <li>● CSSカラー("#010101"または"#FFFFFF"または"red")。</li> <li>● 4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>● R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul> <p>デフォルト値は"#000000"です(文字列値の場合)。複数のカラーがある場合、<b>WP GET ATTRIBUTES</b>は空の文字列を返します。</p> <p>下の境界線のカラーを設定します。取りうる値は以下の通りです:</p>
wk border color bottom	<ul style="list-style-type: none"> <li>● CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。</li> <li>● 4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>● R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul> <p>左の境界線のカラーを設定します。取りうる値は以下の通りです:</p>
wk border color left	<ul style="list-style-type: none"> <li>● CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。</li> <li>● 4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>● R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul> <p>右の境界線のカラーを設定します。取りうる値は以下の通りです:</p>
wk border color right	<ul style="list-style-type: none"> <li>● CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。</li> <li>● 4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>● R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul> <p>上の境界線のカラーを設定します。取りうる値は以下の通りです:</p>
wk border color top	<ul style="list-style-type: none"> <li>● CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。</li> <li>● 4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>● R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul> <p>角の丸い境界線を指定します。取り得る値は以下の通りです:</p>
wk border radius	<ul style="list-style-type: none"> <li>● <u>wk none</u> (デフォルト): 境界線は丸い角を持ちません。</li> <li>● 整数値または文字列値を使用して表現された半径: <ul style="list-style-type: none"> <li>○ 整数値: <u>wk layout unit</u>での半径</li> <li>○ 文字列値: 値と単位が連結されたCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。</li> </ul> </li> </ul> <p>境界線の4辺全てを指定します。取り得る値は以下の通りです:</p>
wk border style	<ul style="list-style-type: none"> <li>● <u>wk none</u> (デフォルト): 境界線なし</li> <li>● <u>wk hidden</u>: <u>wk none</u>と同様(境界線の競合解決時を除く)</li> <li>● <u>wk solid</u>: 標準の境界線</li> <li>● <u>wk dotted</u>: ドットの境界線</li> <li>● <u>wk dashed</u>: 破線の境界線</li> <li>● <u>wk double</u>: 二重境界線</li> <li>● <u>wk groove</u>: 3D くぼみ境界線(実際の効果は境界線カラーによります)</li> <li>● <u>wk ridge</u>: 3D 隆起境界線(実際の効果は境界線カラーによります)</li> <li>● <u>wk inset</u>: 3D インセット境界線(実際の効果は境界線カラーによります)</li> </ul> <p>下部の境界線を指定します。取り得る値は以下の通りです:</p>
wk border style	<ul style="list-style-type: none"> <li>● <u>wk none</u> (デフォルト): 下部境界線なし</li> <li>● <u>wk hidden</u>: <u>wk none</u>と同様(境界線の競合解決時を除く)</li> <li>● <u>wk solid</u>: 標準の下部境界線</li> <li>● <u>wk dotted</u>: ドットの下部境界線</li> <li>● <u>wk dashed</u>: 破線の下部境界線</li> </ul>



- bottom
- [wk double](#): 二重下部境界線
  - [wk groove](#): 3D くぼみ下部境界線(実際の効果は下部境界線カラーによります)
  - [wk ridge](#): 3D 隆起下部境界線(実際の効果は下部境界線カラーによります)
  - [wk inset](#): 3D インセット下部境界線(実際の効果は下部境界線カラーによります)

左の境界線を指定します。取り得る値は以下の通りです:

- [wk none](#) (デフォルト): 左境界線なし
  - [wk hidden](#): [wk none](#)と同様(境界線の競合解決時を除く)
- wk
- [wk solid](#): 標準の左境界線
- border
- [wk dotted](#): ドットの左境界線
- style
- [wk dashed](#): 破線の左境界線
- left
- [wk double](#): 二重左境界線
  - [wk groove](#): 3D くぼみ左境界線(実際の効果は左境界線カラーによります)
  - [wk ridge](#): 3D 隆起左境界線(実際の効果は左境界線カラーによります)
  - [wk inset](#): 3D インセット左境界線(実際の効果は左境界線カラーによります)

右の境界線を指定します。取り得る値は以下の通りです:

- [wk none](#) (デフォルト): 右境界線なし
  - [wk hidden](#): [wk none](#)と同様(境界線の競合解決時を除く)
- wk
- [wk solid](#): 標準の右境界線
- border
- [wk dotted](#): ドットの右境界線
- style
- [wk dashed](#): 破線の右境界線
- right
- [wk double](#): 二重右境界線
  - [wk groove](#): 3D くぼみ右境界線(実際の効果は右境界線カラーによります)
  - [wk ridge](#): 3D 隆起右境界線(実際の効果は右境界線カラーによります)
  - [wk inset](#): 3D インセット右境界線(実際の効果は右境界線カラーによります)

上部境界線を指定します。取り得る値は以下の通りです:

- [wk none](#) (デフォルト): 上部境界線なし
  - [wk hidden](#): [wk none](#)と同様(境界線の競合解決時を除く)
- wk
- [wk solid](#): 標準の上部境界線
- border
- [wk dotted](#): ドットの上境界線
- style
- [wk dashed](#): 破線の上境界線
- top
- [wk double](#): 二重上部境界線
  - [wk groove](#): 3D くぼみ上部境界線(実際の効果は上部境界線カラーによります)
  - [wk ridge](#): 3D 隆起上部境界線(実際の効果は上部境界線カラーによります)
  - [wk inset](#): 3D インセット上部境界線(実際の効果は上部境界線カラーによります)

四辺全ての境界線の幅を指定します。境界線幅を指定する前に境界線スタイルを指定する必要があります。取り得る値は以下の通りです:

- wk
- 整数値あるいは文字列値を使用して表現された幅:
    - 整数値: [wk layout unit](#)での幅
    - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
  - デフォルト値: 2pt

下部境界線の幅を指定します。取り得る値は以下の通りです:

- wk
- 整数値あるいは文字列値を使用して表現された幅:
    - 整数値: [wk layout unit](#)での幅
    - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
  - デフォルト値: 2pt

左境界線の幅を指定します。取り得る値は以下の通りです:

- wk border width left
  - 整数値あるいは文字列値を使用して表現された幅:
    - 整数値: `wk layout unit`での幅
    - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
  - デフォルト値: 2pt

右境界線の幅を指定します。取り得る値は以下の通りです:

- wk border width right
  - 整数値あるいは文字列値を使用して表現された幅:
    - 整数値: `wk layout unit`での幅
    - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
  - デフォルト値: 2pt

上部境界線の幅を指定します。取り得る値は以下の通りです:

- wk border width top
  - 整数値あるいは文字列値を使用して表現された幅:
    - 整数値: `wk layout unit`での幅
    - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
  - デフォルト値: 2pt

wk inside 選択されたエリアに複数の段落が含まれる場合、属性は、対応して被っている段落のプロパティに対してのみ適用することを指定します(範囲の外側は含まれません)。境界線、パディング、マージン属性に対してのみ適用可能で、指定された属性に追加されなければなりません。[WP SET ATTRIBUTES](#)コマンドの例2を参照して下さい。

wk outside 選択されたエリアが複数の段落を含むとき、ある属性が対応する段落の外側のプロパティにのみ適用されることを指定します(内側には適用されません)。これは境界線、パディング、そしてマージン属性に対してのみ指定可能で、特定の属性に対して追加される必要があります。[WP SET ATTRIBUTES](#)コマンドの例2を参照して下さい。

## ドキュメント

ドキュメント属性を使用すると、ドキュメントの題名、執筆者名、会社名、注記といった標準のドキュメント情報を取得・設定することができます。この適用対応表は以下のようになっています:

ドキュメント 段落 文字 ピクチャー

0

定数	コメント
wk author	ドキュメントの著者名を指定(文字列)
wk company	ドキュメントに関連づけられる会社を指定します(文字列)
wk date creation	ドキュメントの作成日(日付)を返します。この値は読み込みのみで、設定する事はできません。
wk date modified	ドキュメントの最終更新日(日付)を返します。この値は読み込みのみであり、設定する事はできません。
wk dpi	内部でのピクセル<->ポイント変換に使用されるDPI(整数)。常に96です(読み込みのみ) 値が整数あるいは実数として設定・取得された際の一部の属性の次元の単位を指定します。取り得る値は以下の通り:
wk layout unit	<ul style="list-style-type: none"> <li>• <a href="#">wk unit cm</a> (デフォルト): センチメートル</li> <li>• <a href="#">wk unit pt</a>: ポイント</li> <li>• <a href="#">wk unit px</a>: ピクセル</li> <li>• <a href="#">wk unit percent</a> (<a href="#">wk line height</a>、<a href="#">wk background size h</a> / <a href="#">wk background size v</a>に対してのみ使用可能)</li> <li>• <a href="#">wk unit mm</a>: ミリメートル</li> <li>• <a href="#">wk unit inch</a>: インチ</li> </ul>
wk notes	ドキュメントについてのコメントを指定します(文字列)
wk subject	ドキュメントの題名を指定します(文字列)
wk title	ドキュメントのタイトル(文字列)を指定します。デフォルトは"New 4D Write Pro Document"です。
wk version	ドキュメントの内部4DWPバージョンを返します(実数)。この数値は <b>WP GET ATTRIBUTES</b> を使用してのみ読み込み可能です。設定することはできません。

## フォントとテキスト

これらの属性を使用するとテキストのフォントファミリー名、サイズ、スタイルを定義する事ができます。適応対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー
		○	

定数	コメント
wk font	<p><b>FONT STYLE LIST</b>コマンドによって返される、完全なフォント名とスタイル名を指定します。無効なフォント名を設定した場合、コマンドは何もしません。デフォルト値:"Times New Roman"</p> <p>テキストの太さを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● <u>wk true</u> を指定すると、選択された文字を太字フォントスタイルへと設定します。<b>WP GET ATTRIBUTES</b>コマンドを使用するとき、選択された文字のうち少なくとも1文字が太字フォントスタイルをサポートする場合、<u>wk true</u>が返されます。</li> <li>● <u>wk false</u> (デフォルト)を指定すると、選択された文字から太字フォントスタイルを除去します。<b>WP GET ATTRIBUTES</b>コマンドを使用するとき、選択された文字がどれも太字フォントスタイルをサポートしない場合、<u>wk false</u>が返されます。</li> </ul>
wk font bold	
wk font family	<p><u>wk font</u>で定義されたフォントファミリー名を指定します。デフォルトの値は"Times New Roman"です。選択された文字に異なるフォントファミリープロパティが含まれる場合、<b>WP GET ATTRIBUTES</b>コマンドは空の文字列を返します。</p> <p>テキストの、イタリックスタイルを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● <u>wk true</u> は選択された文字をイタリック体あるいは斜体スタイルに設定します。<b>WP GET ATTRIBUTES</b>コマンドは選択された文字のうち少なくとも一つ以上の文字がイタリック体あるいは斜体フォントスタイルをサポートする場合、<u>wk true</u>が返されます。</li> <li>● <u>wk false</u> (デフォルト)は選択された文字のうちイタリック体あるいは斜体フォントスタイルがある場合にはそれらを除去します。コマンドは選択された文字のうちイタリック体あるいは斜体フォントスタイルをサポートする文字がない場合、<u>wk false</u>が返されます。</li> </ul>
wk font italic	
wk font size	<p>テキストのフォントサイズを指定します。取り得る値は以下の通りです(単位はポイントのみ):</p> <ul style="list-style-type: none"> <li>● 実際の値(デフォルト = 12)</li> <li>● 値と単位が連結されたCSS文字列。例: 12pt (12 ポイントを意味する)</li> </ul> <p>テキストのカラーを指定します。取り得る値は以下の通りです:</p>
wk text color	<ul style="list-style-type: none"> <li>● CSSカラー("#010101" や "#FFFFFF" または "red")。デフォルトは文字列の場合"#000000"です。</li> <li>● 4Dカラー倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>● R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列</li> </ul> <p>テキストの打ち消し線のカラーを指定します。取り得る値は以下の通りです:</p>
wk text linethrough color	<ul style="list-style-type: none"> <li>● CSSカラー("#010101"や"#FFFFFF" または "red")</li> <li>● 4Dカラー倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照してください)</li> <li>● R、G、B、コンポーネントそれぞれの要素(0-255)を格納した倍長整数配列</li> </ul> <p>デフォルトは、文字列なら"currentColor"、あるいは倍長整数なら<u>wk default</u> です。</p> <p>テキストの打ち消し線のスタイルを(あれば)指定します。取り得る値は以下の通りです:</p>
wk text linethrough style	<ul style="list-style-type: none"> <li>● <u>wk none</u> (デフォルト): 打ち消し線エフェクトはありません。</li> <li>● <u>wk solid</u>: 選択されたテキストに対して実線を描画します。</li> <li>● <u>wk dotted</u>: 選択されたテキストに対して点線を描画します。</li> <li>● <u>wk dashed</u>: 選択されたテキストに対して破線を描画します。</li> <li>● <u>wk double</u>: 選択されたテキストに対して二重線を描画します。</li> <li>● <u>wk semi transparent</u>: 選択されたテキストに対してグレーの線を描画します。他のスタイルと組み合わせる事が可能です。</li> <li>● <u>wk word</u>: 単語にのみ線を描画します(空白のスペースは描画しません)。他のスタイルと組み合わせる事が可能です。</li> </ul> <p>選択されたテキストのシャドウカラーを指定します。取り得る値は以下の通りです:</p>
wk text shadow	<ul style="list-style-type: none"> <li>● CSSカラー("#010101" や "#FFFFFF" または "red")</li> <li>● 4Dカラー倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> </ul>

- color
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列
  - [wk transparent](#) (デフォルト)

- wk text shadow offset
- シャドウエフェクトのオフセットを指定します。取り得る値:
- ポイント単位で指定されたサイズ。デフォルト値: 1pt

- wk text transform
- テキスト内での大文字と小文字を指定します。取り得る値は以下の通り:
- [wk capitalize](#): 最初の文字は大文字に設定されます。
  - [wk lowercase](#): 文字は小文字に設定されます。
  - [wk uppercase](#): 文字は大文字に設定されます。
  - [wk small uppercase](#): 文字はスモールキャップスに設定されます。
  - [wk none](#) (デフォルト): 変換なし

- wk text underline color
- テキストの下線のカラーを指定します。取り得る値:
- CSS カラー("#010101" や "#FFFFFF" または "red")
  - 4Dカラー倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
  - R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列

- wk text underline style
- デフォルトは、文字列なら"currentColor"、あるいは倍長整数なら [wk default](#) です。  
テキストの下線のスタイルを(あれば)指定します。取り得る値は以下の通りです:
- [wk none](#) (デフォルト): 下線エフェクトはありません。
  - [wk solid](#): 実線の下線を描画します。
  - [wk dotted](#): 点線の下線を描画します。
  - [wk dashed](#): 破線の下線を描画します。
  - [wk double](#): 二重線の下線を描画します。
  - [wk semi transparent](#): グレーの下線を描画します。他の線スタイルと組み合わせる事が可能です。
  - [wk word](#): 単語にのみ下線を描画します(空白のスペースは描画しません)。他の線スタイルと組み合わせる事が可能です。

- wk vertical align
- 要素の垂直揃えを設定します。文字、段落、ピクチャーに対して使用可能です。取り得る値は以下の通りです:
- [wk baseline](#) (デフォルト): 要素のベースラインを親要素のベースラインと揃えます。
  - [wk top](#): 要素の上を、行上で最も高い要素の上と揃えます。
  - [wk bottom](#): 要素の下を、行上で最も低い要素の下と揃えます。
  - [wk middle](#): 要素は親要素の中央に配置されます。
  - [wk superscript](#): 要素を上付き文字として配置します。
  - [wk subscript](#): 要素を下付き文字として配置します。

文字においては、[wk top](#)と[wk bottom](#)は[wk baseline](#)と同じ効果を持ちます。  
段落においては、[wk baseline](#)、[wk superscript](#)と[wk subscript](#) は[wk top](#)と同じ効果を持ちます。

## 高さ/幅

---

高さ/幅属性を使用すると、段落や画像の高さと幅を設定することができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー
	○	○	

定数	コメント
	<p>要素の高さを設定します。高さのプロパティには、パディング、境界線、マージンは含まれません。このプロパティは要素のパディング、境界線、マージンの内側のエリアの高さを設定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <code>wk auto</code> (デフォルト): 高さは要素の中身に基づいて決められます。</li> <li>• 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズです: <ul style="list-style-type: none"> <li>◦ 実数: <code>wk layout unit</code>で設定された単位でのサイズ</li> <li>◦ 文字列: 値と単位が結合されたCSS文字列。例: 12ptは12ポイントを意味し、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10000pt</li> </ul> </li> </ul> <p><code>wk height</code>属性は(定義されていれば)<code>wk min height</code>によって上書きされます。</p> <p><b>注:</b> 現行の実装では、<code>wk height</code>定数はピクチャーに対してのみ使用可能です。</p> <p>要素の最小限の高さを設定します。これは<code>wk height</code>プロパティの値が<code>wk min height</code>プロパティより小さくなることを防ぎます。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <code>wk auto</code> (デフォルト): 最小限の高さは要素の中身に基づいて決められます。</li> <li>• 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>◦ 実数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 1000pt。</li> </ul> </li> </ul> <p><code>wk min height</code>の値は<code>wk height</code>属性を上書きします。</p> <p><b>注:</b> 現行の実装では、これはピクチャーに対してのみ使用可能です。</p> <p>要素の最小限の幅を設定します。これは<code>wk width</code>プロパティの値が<code>wk min width</code>プロパティより小さくなることを防ぎます。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <code>wk auto</code> (デフォルト): 最小限の幅は要素の中身に基づいて決められます。</li> <li>• 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>◦ 実数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 1000pt。</li> </ul> </li> </ul> <p><code>wk min width</code>の値は<code>wk width</code>属性を上書きします。</p> <p><b>注:</b> 現行の実装では、これはピクチャーに対してのみ使用可能です。</p> <p>要素の幅を設定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <code>wk auto</code> (デフォルト): 幅は、要素の中身に基づいて設定されます。</li> <li>• 定義されたサイズ: サイズは実数値あるいは文字列値を使用して表現されたサイズになります: <ul style="list-style-type: none"> <li>◦ 実数: <code>wk layout unit</code>の単位でのサイズ。</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10000pt</li> </ul> </li> </ul> <p><code>wk width</code>属性は、<code>wk min width</code>が定義されていればそれによって上書きされます。</p> <p><b>注:</b> 現行の実装では、<code>wk width</code>はピクチャーに対してのみ使用可能です。</p>
wk height	
wk min height	
wk min width	
wk width	

## 画像

画像属性を使用すると、エリア内に挿入されたピクチャーを管理する事ができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー
			○

定数	コメント
	<p>画像を指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 画像URL(文字列)。絶対パス、あるいはストラクチャーからの相対パスを指定できます。</li> <li>• ピクチャー変数またはフィールド</li> </ul>
wk image	<p>返される値(<b>WP GET ATTRIBUTES</b>): URI(ネットワークURLまたはデータURI)。ネットワークURLで参照されていない画像については最初のURLと同一ではないかもしれません(ネットワークURLのみが保存されます)。ローカルファイルのURLに対しては、画像ストリーム自身がドキュメントに保存されるので、返されるURLはデータURIとbase64にエンコードされた画像ストリームです。</p>
wk image alternative text	<p>画像が表示できない際の、画像の代替テキストを指定します。</p>
	<p>要素の垂直揃えを設定します。文字、段落、ピクチャーに対して使用可能です。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <u>wk baseline</u> (デフォルト): 要素のベースラインを親要素のベースラインと揃えます。</li> <li>• <u>wk top</u>: 要素の上を、行上で最も高い要素の上と揃えます。</li> <li>• <u>wk bottom</u>: 要素の下を、行上で最も低い要素の下と揃えます。</li> <li>• <u>wk middle</u>: 要素は親要素の中央に配置されます。</li> <li>• <u>wk superscript</u>: 要素を上付き文字として配置します。</li> <li>• <u>wk subscript</u>: 要素を下付き文字として配置します。</li> </ul> <p>文字においては、<u>wk top</u>と<u>wk bottom</u>は<u>wk baseline</u>と同じ効果を持ちます。 段落においては、<u>wk baseline</u>、<u>wk superscript</u>と<u>wk subscript</u>は<u>wk top</u>と同じ効果を持ちます。</p>
wk vertical align	

## リスト

4D Write Proは主に二つのタイプのリストをサポートしています:

- 順番付けのないリスト: リスト項目が中点と記されているリスト
- 順番付けされているリスト: リスト項目が番号あるいは文字で記されているリスト

リスト属性を使用するとリストを設定し、異なるリスト項目フォントあるいはマーカーを設定する事ができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー
			○



定数	コメント
wk list font	<b>FONT STYLE LIST</b> コマンドで返される、完全なフォント名を指定してリスト項目マーカを表示するようにします(しかし段落のテキストは表示しません)。システムがフォント名を認識しない場合、その代理を使用します。無効なフォント名を設定した場合、コマンドは何もしません。デフォルト値: "Times"
wk list font family	<code>wk list font</code> によって定義されたフォントファミリー名を使用してリスト項目マーカを表示します(しかし段落のテキストは表示しません)。デフォルト値は"Times New Roman"です。
wk list start number	並べられたリストの始まる値を設定します。取り得る値は以下の通りです: <ul style="list-style-type: none"> <li>● <code>wk auto</code> (デフォルト): 始まる値は、前のリスト項目があればそれに応じます。</li> <li>● 整数値: 始まる値</li> </ul>
wk list string format LTR	左から右への段落方向二対してのリスト項目マーカ文字列フォーマット。定義されると、リストのデフォルトのリスト項目マーカ文字列を上書きします。 <ul style="list-style-type: none"> <li>● 順番に並んでいないリスト: 文字列がリスト項目マーカとして使用されます(通常は単一文字による文字列、例: "-")</li> <li>● 順番に並んでいるリスト: "#"文字を含んだ文字列。"#"は計算された数字や文字のプレースホルダーになります。デフォルトは"#."なので、例えばカレントのリスト項目番号が15でリストスタイルタイプが小数だった場合、リスト項目マーカ文字列は"15."になります。</li> </ul>
wk list string format RTL	右から左への段落方向に対してのリスト項目マーカ文字列フォーマット。定義されると、リストのデフォルトのリスト項目マーカ文字列を上書きします。 <ul style="list-style-type: none"> <li>● 順番に並んでいないリスト: 文字列がリスト項目マーカとして使用されます(通常は単一文字による文字列、例: "-")</li> <li>● 順番に並んでいるリスト: "#"文字を含んだ文字列。"#"は計算された数字や文字のプレースホルダーになります。デフォルトは"#."なので、例えばカレントのリスト項目番号が 15でリストスタイルタイプが小数だった場合、リスト項目マーカ文字列は"15."になります。</li> </ul>
wk list style image	順番に並んでいないリストにおいて、画像をリスト項目マーカとして指定します。取り得る値は以下の通りです: <ul style="list-style-type: none"> <li>● <code>wk none</code> (デフォルト): リスト項目マーカは画像によっては定義されていません。</li> <li>● ローカルファイルが像URL(文字列)。絶対パスあるいはデータベースのリソースディレクトリからの相対パスが指定可能です。</li> </ul>
wk list style image height	返される値( <b>WP GET ATTRIBUTES</b> ): URI(ネットワークURLまたはデータURI)。ネットワークURLで参照されていない画像については最初のURLと同一ではないかもしれませんが(ネットワークURLのみが保存されます)。ローカルファイルのURLに対しては、画像ストリーム自身がドキュメントに保存されるので、返されるURLはデータ URIとbase64にエンコードされた画像ストリームです。 <p>リスト項目マーカとして使用される画像の高さを設定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● <code>wk auto</code> (デフォルト): 高さは画像のサイズに基づいて決められます。</li> <li>● 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 実数: <code>wk layout unit</code>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 1000pt。</li> </ul> </li> </ul>
	順番付けされた、あるいは順番付けされていないリスト項目マーカのタイプを指定します。取り得る値は以下の通りです: <ul style="list-style-type: none"> <li>● <code>wk disc</code> (デフォルト)</li> <li>● <code>wk circle</code></li> <li>● <code>wk square</code></li> <li>● <code>wk decimal</code>: 1 2 3</li> <li>● <code>wk decimal leading zero</code>: 01 02 03</li> <li>● <code>wk lower latin</code>: a b c</li> </ul>

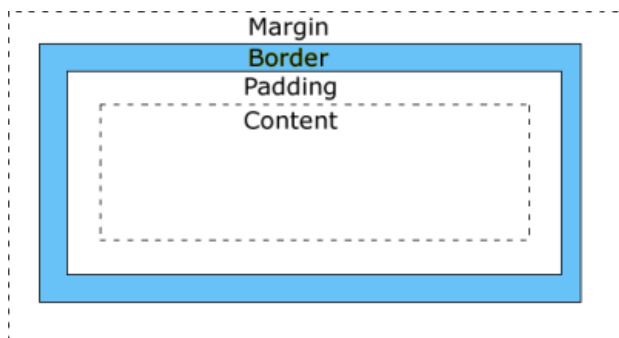


wk list  
style  
type

- [wk lower roman](#): i ii iii iv
- [wk upper latin](#): A B C
- [wk upper roman](#): I II III IV
- [wk lower greek](#): α、β、γ、等
- [wk armenian](#)
- [wk georgian](#)
- [wk hebrew](#)
- [wk hiragana](#)
- [wk katakana](#)
- [wk cjk ideographic](#)
- [wk hollow square](#)
- [wk diamond](#)
- [wk club](#)
- [wk decimal greek](#)
- [wk custom](#): 順番付けされていないリストで "-" をデフォルトのリスト項目マーカーとして使用します。これは標準のリスト項目マーカーを変更する事なく [wk list string format LTR](#) または [wk list string format RTL](#) のリスト項目マーカーをカスタマイズするのに便利なスタイルです。
- [wk none](#)

## マージン

マージンとは、要素の境界線の外側にあるエリアのことで、透明なものです。以下の図は、"ボックス"の要素に対して設定可能な様々な要素の関係性を表しています:



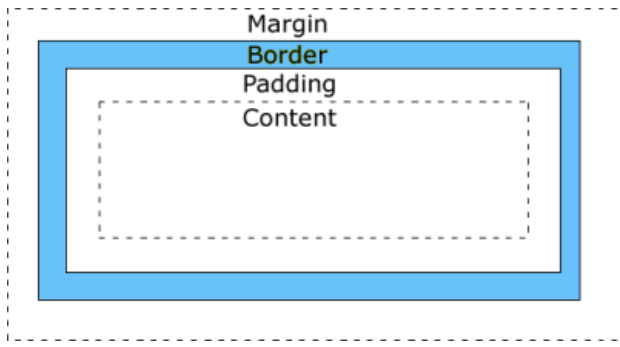
Margin attributes can be applied to:

ドキュメント	段落	文字	ピクチャー
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	

定数	コメント
wk inside	<p>選択されたエリアに複数の段落が含まれる場合、属性は、対応して被っている段落のプロパティに対してのみ適用することを指定します(範囲の外側は含まれません)。境界線、パディング、マージン属性に対してのみ適用可能で、指定された属性に追加されなければなりません。<b>WP SET ATTRIBUTES</b>コマンドの例2を参照して下さい。</p> <p>要素の全てのマージンのサイズを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul>
wk margin bottom	<p>要素の下部のマージンのサイズを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul>
wk margin left	<p>要素の左のマージンのサイズを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul>
wk margin right	<p>要素の右のマージンのサイズを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul>
wk margin top	<p>要素の上部のマージンのサイズを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul>
wk outside	<p>選択されたエリアが複数の段落を含むとき、ある属性が対応する段落の外側のプロパティにのみ適用されることを指定します(内側には適用されません)。これは境界線、パディング、そしてマージン属性に対してのみ指定可能で、特定の属性に対して追加される必要があります。<b>WP SET ATTRIBUTES</b>コマンドの例2を参照して下さい。</p>

## パディング

パディングとは、要素の中身と要素の境界線の間の空白のことを指します。パディングは要素の背景色に影響されます。以下の図は、"ボックス"の要素に対して設定可能な様々な要素の関係性を表しています:



パディング属性は、以下のように適用できます:

ドキュメント	段落	文字	ピクチャー
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

定数	コメント
wk inside	<p>選択されたエリアに複数の段落が含まれる場合、属性は、対応して被っている段落のプロパティに対してのみ適用することを指定します(範囲の外側は含まれません)。境界線、パディング、マージン属性に対してのみ適用可能で、指定された属性に追加されなければなりません。<b>WP SET ATTRIBUTES</b>コマンドの例2を参照して下さい。</p>
wk outside	<p>選択されたエリアが複数の段落を含むとき、ある属性が対応する段落の外側のプロパティにのみ適用されることを指定します(内側には適用されません)。これは境界線、パディング、そしてマージン属性に対してのみ指定可能で、特定の属性に対して追加される必要があります。<b>WP SET ATTRIBUTES</b>コマンドの例2を参照して下さい。</p> <p>要素の全てのサイドのパディングを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>○ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk padding	
wk padding bottom	<p>要素の下部のパディングを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>○ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk padding left	<p>要素の左側のパディングを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>○ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk padding right	<p>要素の右側のパディングを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>○ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk padding top	<p>要素の上部のパディングを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>○ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>

## 段落

段落属性を使用すると段落内でのテキストの配置のためのプロパティを定義する事ができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー
	○		

定数	コメント
wk direction	<p>段落のテキストの方向を指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk left to right</a> (デフォルト)</li> <li>• <a href="#">wk right to left</a></li> </ul>
wk line height	<p>行間スペースを指定します。取り得る値は以下の通り:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk normal</a> (デフォルト): テキストサイズに基づいた値を使用</li> <li>• 整数値あるいは文字列値で表現された高さを使用: <ul style="list-style-type: none"> <li>◦ 整数値: <a href="#">wk layout unit</a> の単位での高さ</li> <li>◦ 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。相対値(パーセンテージ%)はサポートされています。</li> </ul> </li> </ul>
wk tab stop offsets	<p>段落のタブストップを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• スカラー値(デフォルトは35.45pt): 段落全体のデフォルトのオフセットです。 <b>WP GET ATTRIBUTES</b> コマンドは最後に使用されたオフセットを返します(最後の絶対オフセットより前のオフセットについては、デフォルトの相対オフセットになります)。</li> <li>• タブ値の配列: 絶対値の順番付けされたリストで、左のマージンから始まります。最後の値によって定義されたタブオフセットは、段落内で入力される追加のタブ文字にて繰り返し使用されます。タブオフセットが段落の幅より大きい場合、テキストは次の行に移行し、最初のタブ値から開始されます。配列内の値が以前の値より小さい場合、それは無視されます。</li> </ul> <p><b>注:</b> 同じ呼び出しにおいて異なる属性に対して配列とスカラーを同時に使用することはできません。値はCSS文字列(デフォルト)または<a href="#">wk layout unit</a>での単位を使用する実数値で表現されます。最大値は1000ptです。</p> <p>段落でのタブストップタイプを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk left</a> (デフォルト): テキストはタブストップから右側に拡張されます。</li> <li>• <a href="#">wk right</a>: タブスペースが埋まるまで、テキストはタブストップから左側に拡張されます。</li> <li>• <a href="#">wk center</a>: テキストはタブストップにて中央に配置されます。</li> <li>• <a href="#">wk decimal</a>: 小数点より前のテキストは左に拡張され、小数点より後のテキストは右に拡張されます。</li> <li>• <a href="#">wk bar</a>: 指定された位置に垂直の線がおかれます。</li> <li>• タブストップ値の配列(タブストップが配列を通して定義されていた場合)</li> </ul>
wk tab stop types	
wk text align	<p>段落内でのテキストの行揃えを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk left</a> (デフォルト)</li> <li>• <a href="#">wk right</a></li> <li>• <a href="#">wk justify</a></li> <li>• <a href="#">wk center</a></li> </ul>
wk text indent	<p>段落の最初の行のインデントを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 実数: <a href="#">wk layout unit</a> の単位でのサイズ。デフォルトは0です。</li> <li>• 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10000pt</li> </ul>
wk vertical align	<p>要素の垂直揃えを設定します。文字、段落、ピクチャーに対して使用可能です。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk baseline</a> (デフォルト): 要素のベースラインを親要素のベースラインと揃えます。</li> <li>• <a href="#">wk top</a>: 要素の上を、行上で最も高い要素の上と揃えます。</li> <li>• <a href="#">wk bottom</a>: 要素の下を、行上で最も低い要素の下と揃えます。</li> <li>• <a href="#">wk middle</a>: 要素は親要素の中央に配置されます。</li> <li>• <a href="#">wk superscript</a>: 要素を上付き文字として配置します。</li> <li>• <a href="#">wk subscript</a>: 要素を下付き文字として配置します。</li> </ul> <p>文字においては、<a href="#">wk top</a>と<a href="#">wk bottom</a>は<a href="#">wk baseline</a>と同じ効果を持ちます。</p>

段落においては、[wk baseline](#)、[wk superscript](#)と[wk subscript](#) は[wk top](#)と同じ効果を持ちます。

## スタイルシート

---

スタイルシート属性を使用すると指定されたスタイルシートを管理する事ができます。適用の対応表は以下の通りです:

ドキュメント	段落	文字	ピクチャー
	○	○	

定数	コメント
<a href="#">wk new line style sheet</a>	<p>段落に新たな行を追加する際のスタイルシートを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"><li>● 既存のスタイルシート名</li><li>● <a href="#">wk none</a> (デフォルト)</li></ul>
<a href="#">wk style sheet</a>	<p>選択された要素に対してカレントのスタイルシートを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"><li>● <a href="#">wk none</a> (デフォルト)</li><li>● 既存のスタイルシート名</li></ul>

## 4D Write Pro ドキュメントの印刷

4D Write Pro ドキュメントの印刷には次の方法があります:

- 4D フォームの一部として
- 独立したドキュメントとして

### 4D フォーム内ドキュメントの印刷

**PRINT SELECTION** や **PRINT RECORD** などの標準の 4D印刷コマンドを使用して、どのような 4Dフォームでも (プロジェクトフォーム、テーブルフォーム、入力フォーム、出力フォーム)、その中に埋め込まれた 4D Write Pro オブジェクトを印刷することができます。

標準の **印刷時可変** オプションは 4D Write Pro エリアでもサポートされている (\*) ため、印刷時にサイズを操作することができます。このオプションがチェックされているとき、マージン (内側・外側) と上マージンは先頭のページにのみ適用されます。マージン (内側・外側) と下マージンは最終ページにのみ適用されます。ドキュメントのページ付けプロパティは無視されます。ウィドウ・オフアンコントロールは無効化され、改ページは適用されません (これらのプロパティはスクリーン上でのページレンダリングとドキュメントのスタンドアロン印刷においてのみ使用されます)。印刷時可変オプションが選択されていた場合、フォームエリアより上に位置しているオブジェクトのみが印刷されます。このオプションについてのより詳細な情報については、デザインリファレンスマニュアルの "**印刷時可変**" を参照してください。

(\*) **Print object** と **Print form** コマンドは、このオプションと併用できません。

### 印刷とビューモード

4D Write Pro エリアに設定した**ビューモード** (**ビュープロパティの設定** 参照) にかかわらず、**Print form** などの 4D 印刷コマンドをしようすると、常に **埋め込み** モードで印刷されます。つまり、4D Write Pro フォームオブジェクトの次のアピラランス設定が無視されます: ページビューモード (常に **埋め込み**)、ヘッダーを表示、フッターを表示、ページ枠を表示 (常に "No")、非表示の文字を表示 (常に "No")。

### 例題

以下の例は、デフォルト出力フォームに埋め込まれた4D Write Proエリアにおける**印刷時可変**オプションの効果を示したものです。ここでは以下のコードが実行されています:

```
ALL RECORDS ([Movies])
ORDER BY ([Movies]Title)
PRINT SELECTION ([Movies])
```

- 以下は印刷時可変のオプションがチェックされていない (off) 場合の結果です:

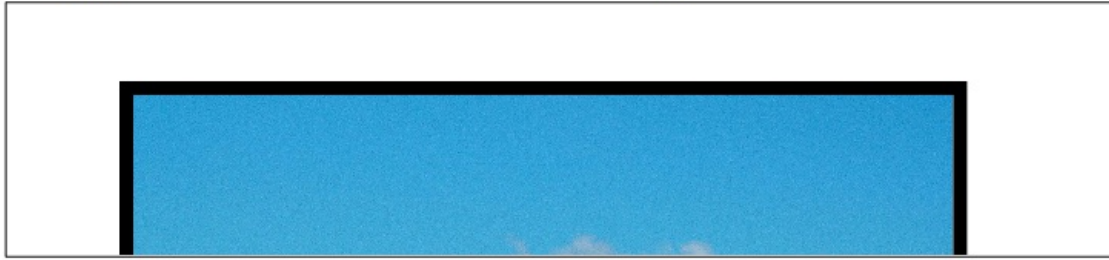
## Mountains of the world

**Name :**

Everest

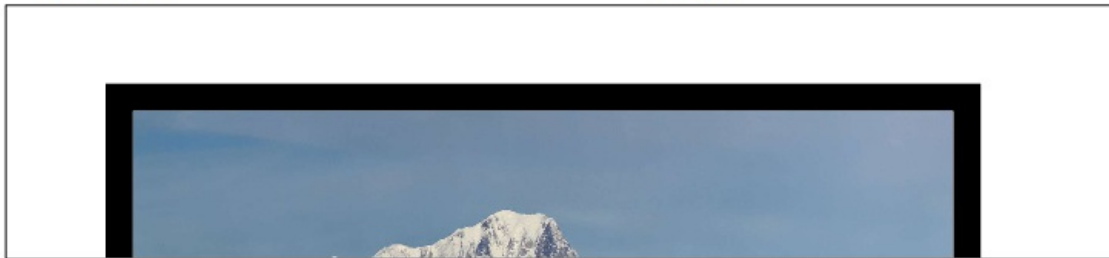
**Continent :**

Asia



Mont Blanc

Europe



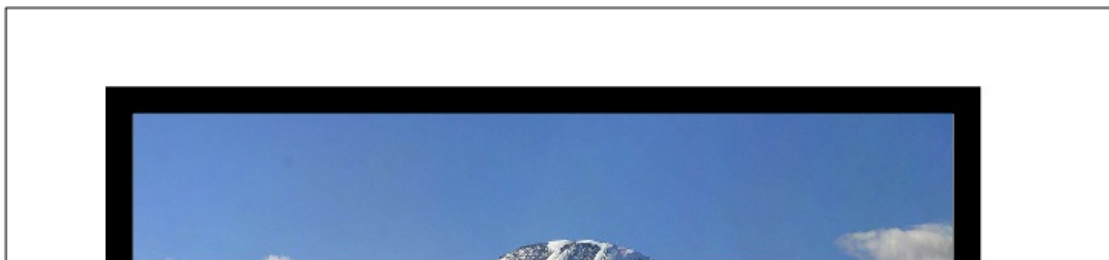
Mount Etna

Europe



Mount Kilimanjaro

Africa



- 以下は印刷時可変のオプションがチェックされている (on) 場合の結果です:



## Mountains of the world

**Name :**

Everest

**Continent :**

Asia



**Mount Everest**, also known in Nepal as Sagarmāthā and in Tibet as Chomolungma, is Earth's highest mountain. It is located in the Mahalangur mountain range in Nepal.[8][9] Its peak is 8,848 metres (29,029 ft) above sea level.[1] It is not the furthest summit from the centre of the Earth. That honour goes to Mount Chimborazo, in the Andes.[10] The international border between China and Nepal runs across Everest's precise summit point. Its massif includes neighbouring peaks Lhotse, 8,516 m (27,940 ft); Nuptse, 7,855 m (25,771 ft) and Changtse, 7,580 m (24,870 ft).

(Sample text source: Wikipedia)

### 独立したドキュメントの印刷

4D v15 R5以降、4D Write Pro に含まれる新機能によって、独立した 4D Write Pro ドキュメントを印刷できるようになったのに加え、フォーマット、ページの向き、ページ番号と言った標準の印刷オプションの操作が可能になりました。

#### 4D Write Pro コマンド

4D Write Proの印刷機能は、主に次の二つのコマンドによって管理されます。**WP PRINT** と **WP USE PAGE SETUP**です。

- **WP PRINT** は 4D Write Pro ドキュメントの印刷ジョブをローンチするか、ドキュメントをカレントの印刷ジョブに追加します。
- **WP USE PAGE SETUP** はカレントプリンターのページ設定を、4D Write Proドキュメント属性のページサイズとページの向きに変更します。

## 設定に関する注意:

- 4D Write Proの印刷コマンドは、64-bit版4D用にデザインされた新しい内部アーキテクチャーに基づいています。これらは32-bit版でもサポートはされますが、**OPEN PRINTING JOB**でローンチされた印刷ジョブ内ではサポートされません(**WP PRINT**を参照してください)。
- Windows 7 あるいは Windows Server 2008 R2のマシン上では、*Platform Update for Windows 7* がインストールされており、印刷機能がサポートされていることを確認してください。

## 標準の 4D コマンド

以下の 4Dコマンドは 4D Write Pro 印刷機能をサポートします:

- **SET PRINT OPTION** と **GET PRINT OPTION**: すべてのオプションが、**WP PRINT** によって印刷される 4D Write Proドキュメントに対してサポートされています。Paper option と Orientation optionに関しては、**WP USE PAGE SETUP** を呼び出してページサイズと向きの属性を 4D Write Pro ドキュメントの設定と同期させる方が効率的な場合もあります。Page range option (15) を使って、印刷するページ範囲を指定することができます。
- **PRINT SETTINGS**: カレントのプリンターに対して印刷設定を設定します。**WP PRINT** がこの後に呼び出された場合、プリンター設定が PRINT SETTINGS ダイアログで変更されていれば、その設定を使用します (ただし余白ページ設定だけは常に4D Write Proドキュメントの設定を使用します)
- **OPEN PRINTING JOB** と **CLOSE PRINTING JOB**: これらのコマンドの間に **WP PRINT** を呼び出すことによって、一つ以上の4D Write Proドキュメントを一件の印刷ジョブに挿入することができます。

PRINT SELECTION ( {aTable}{;}{\* | >} )

引数	型	説明
aTable	テーブル	→ セレクションを印刷するテーブル, または 省略した場合、デフォルトテーブル
*   >	演算子	→ *: 印刷ダイアログを省略, または >: 印刷設定の再初期化をしない

## 説明

**PRINT SELECTION** は、aTableのカレントセレクションを印刷します。レコードは、カレントプロセスのテーブルのカレント出力フォームを使用して印刷されます。**PRINT SELECTION**は、デザインモードのプリント...メニューと同じ動作を実行します。セレクションが空の場合、**PRINT SELECTION**は何も行いません。

デフォルトで、**PRINT SELECTION** は印刷の前にプリンターダイアログボックス (32-bit 版) あるいはプリントジョブダイアログボックス (64-bit 版) を表示します。ユーザが印刷ダイアログボックスでキャンセルを行った場合、コマンドはキャンセルされ、印刷を行いません。

オプション引数の \* または > を使用して、ダイアログボックスの表示を取り消すことができます:

- \* 引数は、現在の印刷設定 (デフォルトの設定、もしくは **PAGE SETUP** や **SET PRINT OPTION** コマンドで定義した設定) に従い、印刷処理を行います。
- > 引数は上記に加え、現在の印刷設定を再初期化することなく印刷を行います。この設定は、以前に設定した内容を継続し、(例えばループ中で) 連続した **PRINT SELECTION** コマンドを使用する時に役立ちます。この引数の使用例は**PRINT RECORD** コマンドの説明を参照してください。

印刷中には、デザインモードのプロパティリストウィンドウで有効にされたフォームおよびオブジェクトのイベントと、実際に発生しているイベントに応じて、出力フォームのフォームメソッドとオブジェクトメソッドが実行されます:

- On Headerイベントはヘッダを印刷する直前に生成されます。
- On Printing Detailイベントはレコードを印刷する直前に生成されます。
- On Printing Breakイベントはブレイクエリアを印刷する直前に生成されます。
- On Printing Footerイベントはフッタを印刷する直前に生成されます。

**PRINT SELECTION** が最初のヘッダーを印刷しているかどうかは、On Headerイベントで **Before selection** を判定することによって調べることができます。またOn Printing Footerイベントで **End selection** を判定することによって、最後のフッターかどうかをチェックすることができます。これら関数の詳細は、それぞれのコマンドや **Form event**、**Level** の説明を参照してください。

**PRINT SELECTION** を使用し、小計やブレイク付きでセレクションを印刷するには、まずそのセレクションをソートしなければなりません。次に、レポートの各ブレイクエリアに、小計を変数に代入するオブジェクトメソッドを持つ変数を配置します。変数に値を代入する、**Sum** や **Average** のような統計関数と算術関数を使用することもできます。詳細は**Subtotal**、**BREAK LEVEL**、**ACCUMULATE** コマンドの説明を参照してください。

**警告:** **PRINT SELECTION** のコンテキストで **PAGE BREAK** コマンドを使用してはいけません。**PAGE BREAK** は **Print form** のコンテキストで使用します。

**PRINT SELECTION** の呼び出し後、プリントが正常に終了するとシステム変数OKに1がセットされます。プリントが中断された場合には、システム変数OKには0がセットされます (例えばユーザが印刷ダイアログボックスでキャンセルをクリックした場合)。

**4D Server:** このコマンドは、ストアードプロシージャのフレームワークにおいて4D Server上で実行することができます。この状況では、次の制約があります:

- サーバマシン上ではダイアログボックスを一切表示しないでください (特定の必要性がある場合を除く)。そのためにはこのコマンドを \* または > 引数付きで呼び出さなければなりません。
- プリンター関連の問題が発生しても (用紙切れ、プリンター接続切断等)、エラーメッセージは生成されません。

## 例題

以下の例は、最初に[People]テーブルのすべてのレコードを選択します。次に**DISPLAY SELECTION**コマンドを使用してすべてのレコードを表示し、ユーザがプリントするレコードを選択します。最後に**USE SET**コマンドにより、選択されたレコードを**PRINT SELECTION**で印刷します:

```
ALL RECORDS ([People]) ` 全レコード選択  
DISPLAY SELECTION ([People];*) ` レコード表示  
USE SET ("UserSet") ` ユーザが指定したレコードのみを使用  
PRINT SELECTION ([People]) ` 印刷実行
```

### 原理

---

印刷時可変オプションは、以下のオブジェクトにおいて利用可能です：

- ピクチャー型のフィールドと変数
- テキスト型のフィールドと変数
- 4D Write Proエリア(4D Write Proリファレンスマニュアルの[4D Write Pro エリアを使用する](#) の章にオプションの詳細な説明あり)

このオプションは、プロパティリスト内の"印刷"テーマ内にあります：



**OBJECT SET PRINT VARIABLE FRAME**と**OBJECT GET PRINT VARIABLE FRAME**コマンドの使用時にも利用可能です。

**注:** サブフォームにおいても似たようなオプションが存在します。これについてのより詳細な情報に付いては、[リストサブフォーム](#)の章内にある"印刷"を参照してください。

このプロパティは、レコードの中身に応じてサイズが変化するオブジェクトの印刷モードを管理します。これらのオブジェクト固定長フレームまたは可変長フレームでの印刷を設定することができます。固定長フレームオブジェクトは、フォーム上でオブジェクト作成するように、オブジェクトのサイズの制限内で印刷をします。可変長フレームオブジェクトはオブジェクトの中身を全て印刷するために、印刷時に展開します。

可変サイズとして印刷されるオブジェクト幅(オブジェクトプロパティによって定義)はこのオプションによって影響はされないという点に注意してください。オブジェクトの中身に応じて、高さのみが変化します。

フォーム内において複数の可変長フレームを隣同士に配置することはできません。非可変長フレームオブジェクトであれば、可変サイズで印刷されるオブジェクトのどちら側でも配置することができます(可変長フレームオブジェクトは最低でも横のオブジェクトより一行分長く、全てのオブジェクトは上部に配置されるからです)。この条件が遵守されない場合、もう一つのフィールドは可変長フレームオブジェクトの全ての水平方向の部分に繰り返しあらわれます。

出力フォームのコンテキストにおいては、詳細エリアには可変長サイズオブジェクトのみを配置することができます。

**注:** **Print object**と**Print form**コマンドはこのオプションをサポートしません。

### ピクチャー

---

ピクチャーは表示フォーマットが許容するのであれば、固定長フレーム、可変長フレームどちらでも印刷することができます。可変長フレームでの印刷が可能なのは以下の表示フォーマットのみです：

- トランケート(中央合わせ)
- バックグラウンド
- トランケート(中央合わせしない)

これらのピクチャーフォーマットについての詳細な情報については、[表示フォーマット](#)を参照してください。

- **印刷時可変**オプションがチェックされている場合、ピクチャーはそのサイズを考慮にいれた高さで印刷されます。ピクチャーフレームはピクチャー全体を表示するために、必要に応じて印刷時に拡大されます。
- このオプションをチェックしなかった場合、ピクチャーは(フォーム内で設定された)固定の高さで印刷されます。

### テキスト

---

- **印刷時可変**オプションがチェックされている場合、テキストはそのサイズを考慮にいれた高さで印刷されます。テキストフィールドは内包するテキストが全て印刷されるように印刷時に自動的に拡張されます。
- このオプションがチェックされていない場合、テキストは(フォームで設定された)固定長で印刷されます。



## Print object

Print object ( { \* ; } object { ; posX { ; posY { ; width { ; height } } } } ) -> 戻り値

引数	型	説明
*	演算子	⇒ 指定時objectはオブジェクト名 (文字列) 省略時 objectは変数
object	フォームオブジェクト	⇒ オブジェクト名 (* 指定時) または 変数 (* 省略時)
posX	倍長整数	⇒ オブジェクトの横位置
posY	倍長整数	⇒ オブジェクトの縦位置
width	倍長整数	⇒ オブジェクトの幅 (ピクセル)
height	倍長整数	⇒ オブジェクトの高さ (ピクセル)
戻り値	ブール	⇒ True = オブジェクトが完全に印刷された; そうでなければFalse

### 説明

**Print object** コマンドを使用して *object* と \* 引数で指定したフォームオブジェクトを、*posX* と *posY* の位置に、任意のサイズで印刷できます。

**Print object** コマンドを呼び出す前に、印刷するテーブルまたはオブジェクトを含むプロジェクトフォームを新しい **FORM LOAD** で指定しなければなりません。

オプションの \* 引数を渡すと、*object* 引数にはオブジェクト名 (文字列) を渡します。\* 引数を渡さない場合、*object* には変数を指定します。この場合、文字列ではなく変数参照 (オブジェクトタイプのみ) を渡します。

*posX* と *posY* 引数はオブジェクトを印刷する開始位置を指定します。これらの値はピクセル単位で指定します。これらの引数を省略すると、オブジェクトはフォーム上の位置に基づいて印刷されます。

*width* と *height* 引数はフォームオブジェクトの幅と高さを指定します。**Print object** コマンドは可変長のオブジェクトを管理しません。**OBJECT GET BEST SIZE** コマンドでオブジェクトのサイズを管理しなければなりません。また **OBJECT GET BEST SIZE** コマンドでテキストを含むオブジェクトのもっとも適切なサイズを知ることができます。同様に、**Print object** はページブレークを自動では生成しません。必要に応じて開発者が管理しなければなりません。

4Dコマンドを使用してオブジェクトプロパティ (色やサイズなど) をオンザフライで変更できます。

オブジェクトが完全に印刷されるとコマンドはTrueを返します。そうでないばあい、言い換えればフレームワーク内のオブジェクトに割り当てられたデータをすべて印刷できなかった場合、コマンドはFalseを返します。特にリストボックスのすべての行を印刷できなかった場合、コマンドはFalseを返します。この場合**Print object** コマンドを、それがTrueを返すまで繰り返し呼び出します。特別なメカニズムが自動で使用され、オブジェクトの内容が呼び出しごとに自動でスクロールされます。

#### 注:

- 4Dの現在のバージョンでは、リストボックスタイプのオブジェクトのみがこのメカニズムを持っています (他のオブジェクトではコマンドは常に True を返します)。4Dの将来のバージョンでこの機能は他の可変長オブジェクトに拡張されます。
- **LISTBOX GET PRINT INFORMATION** コマンドを使用して、処理中に印刷状況を知ることができます。

**Print object** コマンドは事前に**OPEN PRINTING JOB** で開かれた印刷ジョブのコンテキストでのみ使用できます。このコンテキストの外で呼び出された場合、コマンドはなにも行いません。同じ印刷ジョブ内で**Print object** コマンドを複数回呼び出すことができます。

**注:** 階層リスト、サブフォーム、およびWebエリアを印刷することはできません。

### 例題 1

フォーム上の10個のオブジェクトを印刷する例:

```
PRINT SETTINGS
If (OK=1)
  OPEN PRINTING JOB
  If (OK=1)
    FORM LOAD ("PrintForm")
```

```
x:=100
y:=50
GET PRINTABLE AREA (hpaper;wpaper)
For ($i;1;10)
    OBJECT GET BEST SIZE (*;"Obj"+String($i);bestwidth;bestheight)
    $send:=Print object (*;"Obj"+String($i))
    y:=y+bestheight+15
    If (y>hpaper)
        PAGE BREAK (>)
        y:=50
    End if
End for
End if
CLOSE PRINTING JOB
End if
```

## 例題 2

---

リストボックスを完全に印刷する例:

```
Repeat
    $send:=Print object (*;"mylistbox")
Until ($send)
```

## 4D Write Pro ドキュメントに含める式の制限

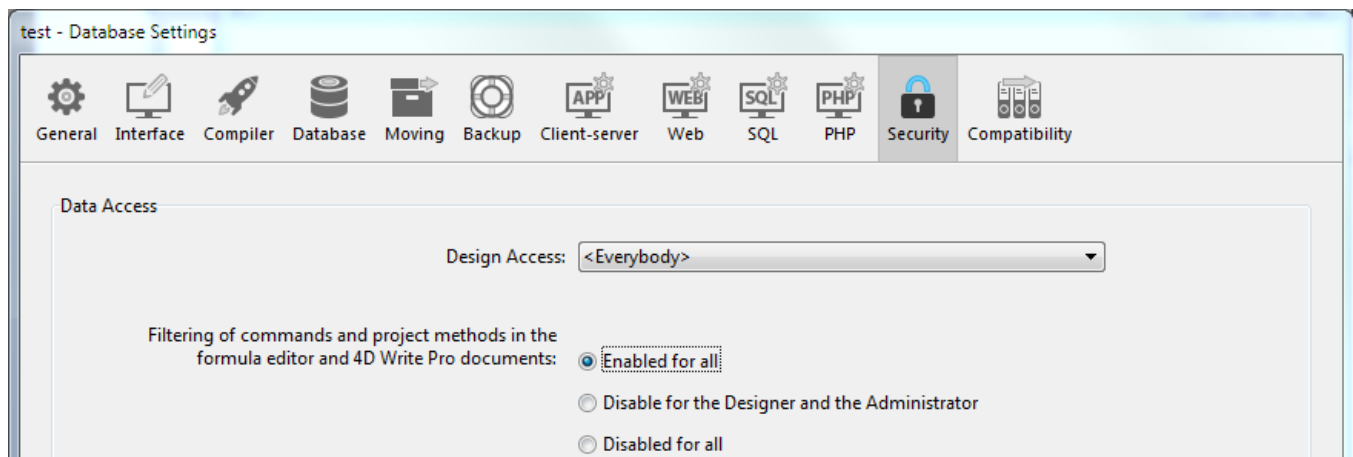
### 4D Write Pro ドキュメントに使用できる式の制限

4D Write Pro ドキュメントには、変数やフィールドに限らず、フォーミュラやプロジェクトメソッド、4D コマンドなどの動的な 4D 式への参照を含めることができます。ドキュメントの表示および印刷の際に、これらの参照は評価されます。セキュリティ上の理由により、適切でない式が使用されてしまったり、データベースに想定外の変更を加えるなどの影響を与えてしまったりしないよう、動的な式の評価は開発者によって制限する必要があります。例えば、**DELETE SELECTION** のようなコマンドや "DeleteOrders" といったメソッドの実行は制限したほうがよいでしょう。

**注:** 式を挿入するには、**ST INSERT EXPRESSION** コマンドを使うか、HTML エディターで 4D Write Pro ドキュメントを編集します。

### 4D のフィルタリング機能をサポート

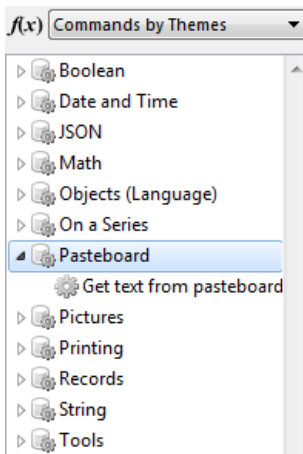
4D v16 より、4D Write Pro ドキュメントに挿入される 4D コマンドやプロジェクトメソッドの評価は、データベースレベルで設定される通常の 4D のフィルタリング機能によって管理されます。このオプションは "データベース設定" ダイアログの**セキュリティ**ページで提供されています:



デフォルトでは、すべてのユーザーに対して制限が課されるため、明示的に許可されたコマンドやメソッド以外を使用すると、評価エラーが発生します。この制限を部分的に緩和することも (Designer と Administrator は制限しない)、誰も制限しないことも可能です。このオプションは 4D データベース全体に対して有効で、ユーザーによるフォーミュラの評価をすべて管理します。このオプションについての詳細は、[デザインリファレンスマニュアル](#)を参照ください。

制限が有効になっている場合:

- ユーザーはフォーミュラエディターの右端にあるコマンドリストに表示されているコマンドしか使用できません:



- ユーザーはプロジェクトメソッドを実行できません。4D Write ドキュメントで使用するには、**SET ALLOWED METHODS** コマンドを使って明示的に許可する必要があります。

4D Write Pro ドキュメント内で式の評価中に許可されていない 4D コマンドやメソッドが見つかったら、値が "# # Error #"



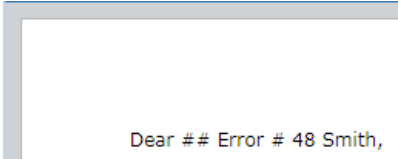
48" に置き換わります。

## 例題

4D Write Pro ドキュメントに次の式を挿入します:

```
ST INSERT EXPRESSION (*;"WriteProArea";"Gender")
```

セキュリティの設定で制限がかかっていると、**Gender** メソッドは評価されません:

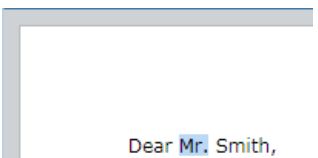


Dear ## Error # 48 Smith,

その場合には次のようなコードを実行します:

```
ARRAY TEXT (aTallow;1)  
aTallow{1}:="Gender"  
SET ALLOWED METHODS (aTallow)
```

すると **Gender** メソッドが評価され、値が返されます:



Dear Mr. Smith,

## 互換性

---

以前のリリースでは、4D Write Pro ドキュメントのフィルタリング機能は有効になっていませんでした。4D Write Pro ドキュメントで 4D メソッドを参照しているデータベースを 4D v16 以上に変換した場合、これらは評価されなくなり、"## Error # 48" メッセージが表示されます。このような場合には、**SET ALLOWED METHODS** コマンドを使用して、当該メソッドを許可リストに追加します。

## 📄 4D Write ドキュメントの読み込み

新しい4D Write Proオブジェクトの主要な機能の一つとして、既存の4D Writeドキュメントの読み込みと変換機能が挙げられます。これによって4D Write プラグインに依存しているアプリケーションを移行させることができます。

### 互換性に関する注意:

- サポートされるのは、4D Write ドキュメントのうち最後の世代 ("4D Write v7") のドキュメントに限られます。
- 4D Write ドキュメントから 4D Write Pro エリアへのコピー・ペーストは現時点ではまだサポートされていません。4D Writeドキュメントの読み込みは 4D Write Pro ランゲージコマンドの使用によってのみ可能です。

### 4D Write ドキュメントを読み込むには

4D Write Proオブジェクトに 4D Writeドキュメントを読み込む方法は二つあります:

- ディスク上に保存されている 4D Write ファイルに関しては、**WP Import document** コマンドを使用してください。
- BLOBフィールドに格納されている 4D Write エリアに関しては、**WP New** コマンドを使用してください。

より詳細な情報に関しては、それぞれのコマンドの詳細を参照してください。

### 4D Write のプロパティで復元されるもの

4D Write プラグインから4D Write Proへの移行を簡単にするために、4D Write の機能のうち、出来る限り多くのものが4D Write Proオブジェクトでサポートされようとしています。

以下の段落では、4D Write プラグインのプロパティのうち、**WP Import document** または **WP New** コマンドを使用して4D Write Proエリアへと読み込んだときに復元されるプロパティについてまとめています。

しかしながら一部の機能において、バグとはみなされない小さな差異が生じることがあります。これらは例えば、4D Write Proで使用されるデフォルトの行頭の記号や、下線のタイプの小さな変化などによるものです。

### ドキュメント情報

4D Write プラグイン	4D Write Pro
作成日時	利用可能
修正日時	利用可能
ロック	利用不可(読み込み専用オブジェクトプロパティを使用して下さい)
タイトル	利用可能
件名	利用可能(標準テキストのみ)
作者	利用可能
会社名	利用可能
注記	利用可能

### ドキュメントの表示の引数

4D Writeプラグイン	4D Write Pro
ページモード	読み込まれません(コンテキストメニューの、ドキュメント/ページモードを使用して下さい)
ルーラー	利用不可
枠	利用不可
ヘッダー	利用不可
フッター	利用不可
最初のページのヘッダー	利用不可
最初のページのフッター	利用不可
画像	利用不可
縦スクロールバー	読み込まれません(オブジェクトプロパティの縦スクロールバーを使用して下さい)
横スクロールバー	読み込まれません(オブジェクトプロパティの横スクロールバーを使用して下さい)
非表示文字	利用不可
参照	利用不可( <a href="#">ST SET OPTIONS</a> を参照して下さい)
カラム分割	利用不可
縦スプリッター	利用不可
横スプリッター	利用不可
Wysiwyg	利用不可
ズーム	読み込まれません(コンテキストメニューの、ドキュメント/ズームを使用して下さい)

## ドキュメントの引数

4D Write プラグイン	4D Write Pro
単位	利用不可
言語	利用不可
カラム数	利用不可
行間隔	利用不可
ウィンドウ&オーファン	利用不可
デフォルトのタブ	利用可能
先頭のタブ	利用不可
URLカラー	利用不可
アクセス済みのURLカラー	利用不可

## ドキュメントのページ付の引数

## 4D Writeプラグイン

ページ幅	利用可能
ページの高さ	利用可能
最初のページ番号	利用可能(v16以降)
最初のページのヘッダーとフッターを別にする	利用可能(v16以降)
左右のページでヘッダーとフッターが異なる	利用可能(v16以降)
ページバインディング	利用可能(v16以降)
反対側のページ	利用可能(v16以降)
ページマージン	利用可能
ヘッダーの上マージン	利用可能(v16以降)
ヘッダーの下マージン	利用可能(v16以降)
フッターの上マージン	利用可能(v16以降)
フッターの下マージン	利用可能(v16以降)
最初のページの上マージン	利用可能(v16以降)
最初のページの下マージン	利用可能(v16以降)
最初のページのヘッダーの上マージン	利用可能(v16以降)
最初のページのヘッダーの下マージン	利用可能(v16以降)
最初のページのフッターの上マージン	利用可能(v16以降)
最初のページのフッターの下マージン	利用可能(v16以降)
最初のページを右側にする	利用可能(v16以降)

## 4D Write Pro

## ドキュメントの印刷引数

### 4D Write プラグイン 4D Write Pro

用紙の種類	利用不可
横向きに印刷	利用不可
幅	利用不可
高さ	利用不可
ユーザー設定マージン	利用不可
倍率	利用不可
X 解像度	利用不可
Y 解像度	利用不可

## 画像

### 互換性に関する注意:

- 4D Write Pro はページ内画像の絶対位置に関してはまだサポートしていません。インライン画像のみがサポートされ、読み込まれます。
- Windows用64-bit版では、Mac OSのPICTフォーマットの画像を含んでいる4D Writeドキュメントの読み込みはサポートされていません。この画像タイプが含まれたドキュメントを読み込むには、あらかじめそれらの画像を異なるフォーマットへと変換しておくか、または32-bit版の4Dを使用する必要があります。また、PICTフォーマットは廃止予定であり、今後使用されるべきではない点にも注意してください ([PICT フォーマットのピクチャー](#) を参照してください)。

4D Write プラグイン	4D Write Pro
X (左)	(& position :absolute) (ページ内の画像のみ)
Y (上)	(& position :absolute) (ページ内の画像のみ)
幅	利用可能
高さ	利用可能
ページ番号	利用不可
背景画像	利用不可
最初のページには含めない	利用不可
ビューポートモード(縮小して画像を入れる)	利用可能
式である	利用不可
サイズを保持	利用不可

## 文字プロパティ

4D Write プラグイン	4D Write Pro (span プロパティ)
イタリック	利用可能
ボールド	利用可能
取り消し線	利用可能
下線	利用可能
シャドウ	利用可能
指数(上付きまたは下付き)	利用可能
大文字(英大文字または小型英大文字)	利用可能
フォントファミリー	利用可能
フォントサイズ	利用可能
テキストカラー	利用可能
テキスト背景カラー	利用可能
下線カラー	利用可能
取り消し線カラー	利用可能
シャドウカラー	利用可能
ユーザープロパティ	利用不可
スペルチェック(シンタックス&グラマー onまたはoff)	利用不可
表示	利用不可
スタイルシート	読み込まれません(スタイルは読み込まれますが、スタイルシートは利用不可です)

## 段落プロパティ

4D Write プラグイン	4D Write Pro
均等位置	利用可能
Interline	利用可能
箇条書き	利用可能
左マージン	利用可能
右マージン	利用可能
テキストインデント	利用可能
境界線スタイル	利用可能
境界線カラー	利用可能
境界線の背景カラー	利用可能
左境界線	利用可能
右境界線	利用可能
上境界線&上の内側の境界線	利用可能
下の境界線&下の内側の境界線	利用可能
境界線間隔	利用可能
スタイルシート	利用可能
タブ	利用可能

## ハイパーリンク

4D Write プラグイン	4D Write Pro
URL リンク	利用可能
4D メソッドリンク	利用不可
ドキュメントを開くリンク	利用不可

## 4D 式

4D Write プラグイン	4D Write Pro
4D 式	利用可能
日付と時間	利用可能
HTML 式	利用不可
RTF 式	利用不可

## テキストデータ

4D Write プラグイン	4D Write Pro
メインテキストデータ	利用可能
ヘッダーテキストデータ	利用不可
フッターテキストデータ	利用不可

## WP Import document

WP Import document ( filePath ) -> 戻り値

引数	型	説明
filePath	文字	→ 4D Write ドキュメント(.4w7または.4wt)へのパス
戻り値	Object	↩ 4D Write Proオブジェクト

### 説明

**WP Import document** コマンドは既存の4D Write Proドキュメントまたは4D Write ドキュメント(.4w7 または .4wt)を新規の4D Write Proオブジェクトへと変換します。

*filePath* 引数には、ディスク上に保存されたドキュメントへのパスを渡します。以下のタイプのドキュメントがサポートされています:

- 以前の4D Write ドキュメント(.4w7 or .4wt)。4D Write Proオブジェクト内で現在サポートされている4D Writeの昨日の詳細な一覧に着いては、[4D Write ドキュメントの読み込み](#) の章を参照して下さい。
- 4D Write Pro(.4wp)フォーマットドキュメント。4D Write Proドキュメントフォーマットについてのより詳細な情報に関しては、[.4wp ドキュメントフォーマット](#) を参照して下さい。

パスは完全なパスを渡す必要があります。ドキュメントがストラクチャーファイルと同等の階層の保存されている場合に限り、ドキュメント名のみを渡す事もできます。

実行後、コマンドはこの変換の結果を格納した4D Write Proオブジェクトを返します。

**注:** 4D BLOBフィールドに保存されたドキュメントを読み込みたい場合には、**WP New** コマンドの使用も検討して下さい。

*filePath* 引数に渡したパスが無効な場合、またはファイルが見つからない場合、または指定したファイルがサポートされていないフォーマットだった場合、エラーが返されます。

### 例題

```
C_OBJECT (WPDoc)
WPDoc:=WP Import document("C:\\documents\\4DWriteDocs\\Letter.4w7")
```

ST SET OPTIONS ( { \* ; } object ; option ; value { ; option2 ; value2 ; ... ; optionN ; valueN } )

引数	型	説明
*	演算子	⇒ 指定時: object はオブジェクト名(文字列) 省略時: object は変数またはフィールド
object	フォームオブジェクト	⇒ オブジェクト名( * 指定時)、または変数やフィールド( * 省略時)
option	倍長整数	⇒ 設定したいオプション
value	倍長整数	⇒ オプションの新しい値

## 説明

**ST SET OPTIONS**コマンドは、*object* 引数で指定されたスタイル付テキストフィールドまたはテキスト変数に関する様々なオプションを変更することができます。

任意の \* 演算子を渡した場合、*object* 引数でオブジェクト名を文字列で指定します。省略時には *object* 引数でフィールドまたは変数を指定します。

修正したいオプションを指定する値を *option* 引数に、新しく設定したい値を *value* に渡します。

*option* 引数は、"**Multistyle Text**"テーマ内にある以下の定数をサポートしています：

定数	型	値	コメント
ST Expressions display mode	倍長整数	1	<i>value</i> 引数にはST ValuesまたはST Referencesのどちらかが入ります

*value* 引数には、以下の定数のどれか一つをわたす事ができます：

定数	型	値	コメント
ST References	倍長整数	1	式のソースの文字列を表示します。
ST Values	倍長整数	0	4D式の、計算された値を表示します。

値の表示:

Current time:	14:39:10
Field contents:	Bravo

ソースの表示:

Current time:	String(Current time)
Field contents:	[Table_1]Comment






























## 例題

以下のコードは、エリアの表示モードを切り替えます：

```
ST GET OPTIONS (*;"StyledText_t";ST Expressions display mode;$exprValue)
If ($exprValue=1)
  ST SET OPTIONS (*;"StyledText_t";ST Expressions display mode;ST Values)
Else
  ST SET OPTIONS (*;"StyledText_t";ST Expressions display mode;ST References)
End if
```



# 4D Write Proランゲージ

-  4D Write Pro オブジェクトについて
-  オブジェクト(フォーム)テーマのコマンドの使用
-  スタイル付テキストテーマのコマンドの使用
-  ドキュメントのコンテンツにプログラムでアクセスする
-  WP CREATE BOOKMARK New 16.0
-  WP DELETE BOOKMARK New 16.0
-  WP EXPORT DOCUMENT
-  WP EXPORT VARIABLE
-  WP GET ATTRIBUTES
-  WP Get bookmark range New 16.0
-  WP GET BOOKMARKS New 16.0
-  WP Get page count New 16.0
-  WP Get paragraphs
-  WP Get pictures
-  WP Get range
-  WP Get selection
-  WP Import document
-  WP INSERT BREAK New 16.0
-  WP INSERT DOCUMENT New 16.0
-  WP INSERT PICTURE New 16.0
-  WP Is font style supported
-  WP New Updated 16.0
-  WP PRINT Updated 16.0
-  WP RESET ATTRIBUTES
-  WP SELECT
-  WP SET ATTRIBUTES
-  WP USE PAGE SETUP
-  4D Write Pro
-  4D Write Pro属性

## 4D Write Pro オブジェクトについて

### 4D Write Pro オブジェクト

---

4Dアプリケーションにおいて、4D Write Proドキュメントは`object` 型の変数を使用して管理されます(詳細な情報に関しては、[C\\_OBJECT](#) コマンドを参照して下さい)。

4D Write Pro オブジェクトを一度メモリで参照した後は、以下の様な事をすることができます:

- "Write Pro エリア"というフォームオブジェクトを使用して、4Dフォームを通して表示・編集ができます。その場合、フォームエリアはオブジェクト変数と同じ名前である必要があります([4D Write Pro エリアを定義する](#)を参照して下さい)。
- 複数のテーマのコマンドによって、プログラムで管理することができます(以下参照)。
- **WP EXPORT DOCUMENT** コマンドを使用して、ディスク上にHTMLドキュメントとして書き出す事ができます。

### 4D Write Pro オブジェクトを扱う事ができるコマンド

---

4D Write Pro オブジェクトは、複数のテーマの4Dコマンドを使用して管理することができます:

- **4D Write Pro** テーマの4D Write Pro専用のコマンド。これらのコマンドはこの章内にて詳細な説明があります。
- **オブジェクト(フォーム)** コマンド。フォーマット機能・オブジェクトプロパティ機能を管理します。詳細な情報に関しては、[オブジェクト\(フォーム\)テーマのコマンドの使用](#) の章を参照して下さい。
- **スタイル付きテキスト** コマンド。コンテンツ機能を管理します。詳細な情報に関しては、[スタイル付テキストテーマのコマンドの使用](#) の章を参照して下さい。

## 📄 オブジェクト(フォーム)テーマのコマンドの使用

以下のコマンドは4D Write Pro フォームオブジェクトをサポートします:

コマンド	補足
OBJECT DUPLICATE	
OBJECT Get auto spellcheck/OBJECT SET AUTO SPELLCHECK	
OBJECT Get border style/OBJECT SET BORDER STYLE	
OBJECT Get context menu/OBJECT SET CONTEXT MENU	
OBJECT GET COORDINATES/OBJECT SET COORDINATES	
OBJECT Get data source/OBJECT SET DATA SOURCE	
OBJECT GET DRAG AND DROP OPTIONS/OBJECT SET DRAG AND DROP OPTIONS	
OBJECT Get enabled/OBJECT SET ENABLED	
OBJECT Get enterable/OBJECT SET ENTERABLE	
OBJECT GET EVENTS/OBJECT SET EVENTS	
OBJECT Get focus rectangle invisible/OBJECT SET FOCUS RECTANGLE INVISIBLE	
OBJECT Get font/OBJECT SET FONT	カレントセクションがあればそこに適用
OBJECT Get font size/OBJECT SET FONT SIZE	カレントセクションがあればそこに適用
OBJECT Get font style/OBJECT SET FONT STYLE	カレントセクションがあればそこに適用
OBJECT Get horizontal alignment/OBJECT SET HORIZONTAL ALIGNMENT	カレントセクションがあればそこに適用。4D Write Proエリアの <code>wk.justify</code> 定数をサポート
OBJECT GET RESIZING OPTIONS/OBJECT SET RESIZING OPTIONS	
OBJECT SET COLOR	カレントセクションがあればそこに適用
OBJECT GET RGB COLORS/OBJECT SET RGB COLORS	カレントセクションがあればそこに適用
OBJECT Get type	
OBJECT Get vertical alignment/OBJECT SET VERTICAL ALIGNMENT	段落の垂直方向の行揃え:段落の高さが段落のテキストの高さより高い場合にのみ効力を持ちます。
OBJECT Get visible/OBJECT SET VISIBLE	
OBJECT Is styled text	trueを返します
OBJECT MOVE	
OBJECT GET SCROLL POSITION/OBJECT SET SCROLL POSITION	
OBJECT GET SUBFORM CONTAINER SIZE	
OBJECT Get name	
OBJECT Get pointer	

上記にないOBJECT コマンドはどれも4D Write Pro エリアに対しては使用できません。

## 例題

4D Write Pro エリアの垂直方向の行揃えを中央に設定したい場合を考えます:

```
Case of
: (Form event=On_Clicked)
  OBJECT SET HORIZONTAL ALIGNMENT (*;"myObject";Align_center)
  hAlignLeft:=0 //次回表示のためにプロパティを保存
  hAlignRight:=0
End case
```

## 📄 スタイル付テキストテーマのコマンドの使用

以下のコマンドは4D Write Pro オブジェクトをサポートします:

コマンド	補足
ST COMPUTE EXPRESSIONS	
ST FREEZE EXPRESSIONS	
ST GET ATTRIBUTES / ST SET ATTRIBUTES	
ST Get content type	画像コンテンツの型に新しいタイプ(6)が追加されました
ST Get expression / ST INSERT EXPRESSION	4D v16 より、4D Write Pro ドキュメントの式にはデフォルトで制限がかけられており、明示的に許可する必要があります。詳細については、 <a href="#">4D Write Pro ドキュメントに含める式の制限</a> を参照ください。
ST GET OPTIONS / ST SET OPTIONS	
ST Get plain text / ST SET PLAIN TEXT	
ST Get text / ST SET TEXT	
ST GET URL / ST INSERT URL	

### 例題

4D Write Pro エリアの選択範囲を変数の中身で置き換えたい場合を考えます:

```
C_TEXT(fullName)

Case of
: (Form event=On_Clicked)
    ST INSERT EXPRESSION(myArea;"fullName";ST_Start_highlight;ST_End_highlight)
End case
```

### 式の挿入

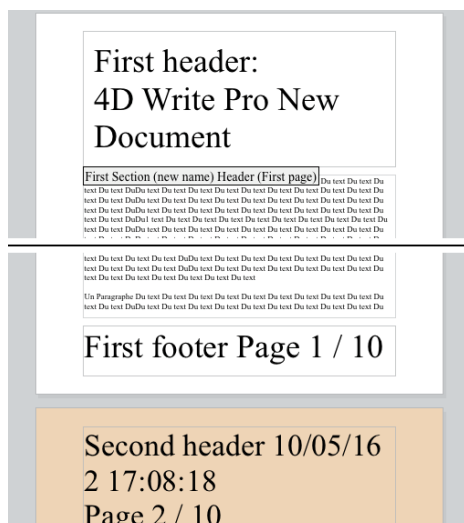
**ST INSERT EXPRESSION** コマンドを使って、ドキュメントやページの属性に関連づいた式を挿入することができます。

式のシンタックス	有効範囲	型	説明
\$wp_title	ドキュメント全域	テキスト	<u>wk title</u> 属性が定義するタイトル
\$wp_author	ドキュメント全域	テキスト	<u>wk author</u> 属性が定義する著者名
\$wp_subject	ドキュメント全域	テキスト	<u>wk subject</u> 属性が定義する題名
\$wp_company	ドキュメント全域	テキスト	<u>wk company</u> 属性が定義する会社
\$wp_notes	ドキュメント全域	テキスト	<u>wk notes</u> 属性が定義するコメント
\$wp_dateCreation	ドキュメント全域	日付	<u>wk date creation</u> 属性が定義する作成日
\$wp_dateModified	ドキュメント全域	日付	<u>wk date modified</u> 属性が定義する最終更新日
\$wp_pageNumber	ヘッダー & フッター (それ以外はエラー)	倍長 整数	ドキュメントの先頭 (デフォルト) あるいはセクションの先頭からのページ数
\$wp_pageCount	ヘッダー & フッター (それ以外はエラー)	倍調 整数	ページ総数

式を挿入するには、まずカーソルが任意のエリア (ヘッダー、フッター、ドキュメント本文) にあることを確認し、**ST INSERT EXPRESSION** コマンドをコールします。例えば、選択したフッターエリアにページ番号を挿入する場合:

```
ST INSERT EXPRESSION (*;"4DWPArea";"$wp_pageNumber")
```

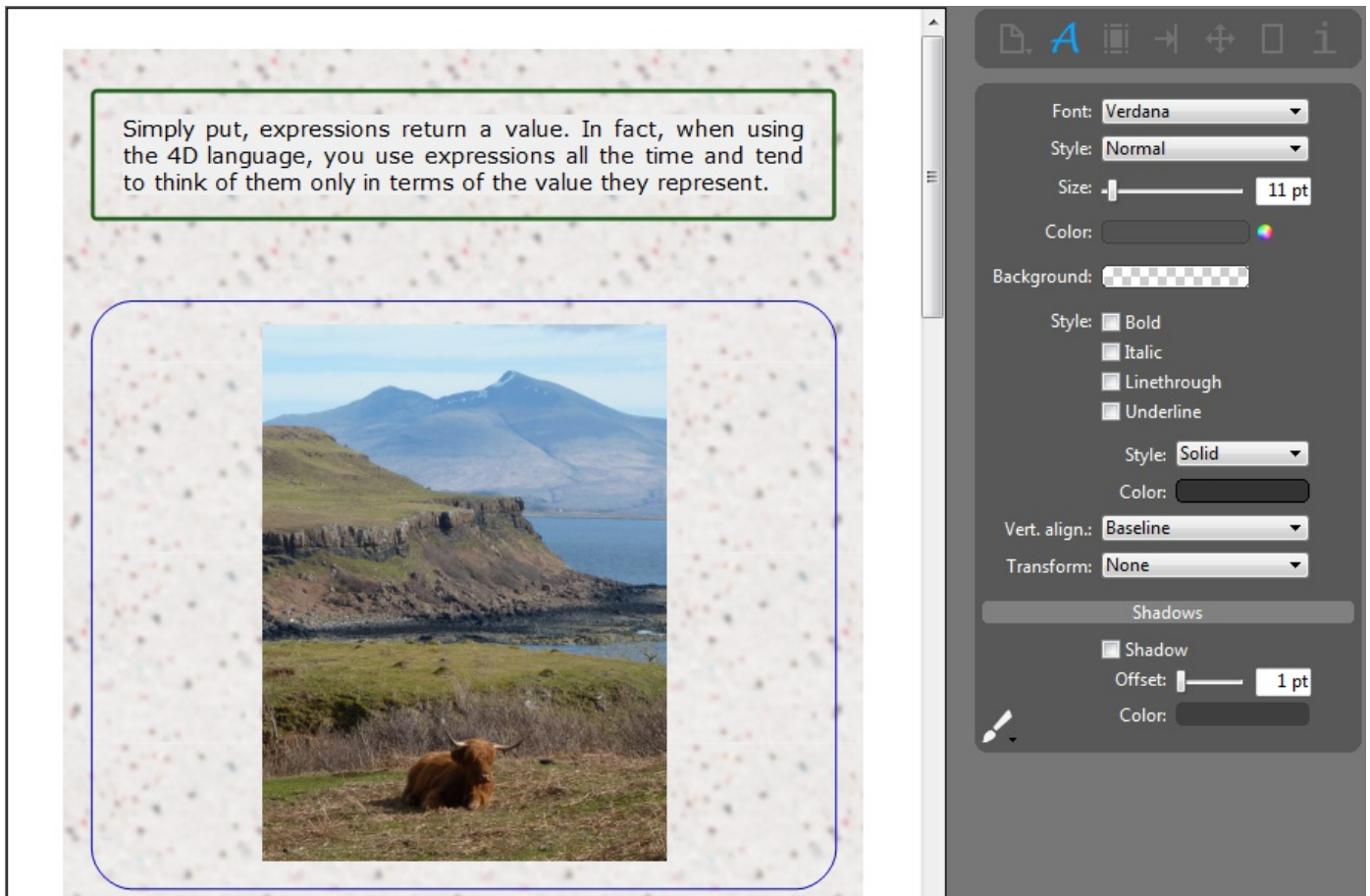
例えば、下のようなドキュメントデザインを定義することができます:



## ドキュメントのコンテンツにプログラムでアクセスする

4D Write Proではドキュメント内でテキストと画像の属性を変更するためのコマンドをフルセット提供しています。これらの機能のおかげで、4Dデベロッパーは4D Write Proドキュメントに対して(ボタン、メニュー、チェックボックス等を使用した)独自のインターフェースをデザインすることができます。コマンドはドキュメント全体、または特定の部分(レンジ)に対して適用可能で、どちらもユーザー選択あるいはカスタムの値に基づきます。利用可能なプロパティには、ドキュメント単位、マージン、パディング、背景、段落設定、カラー、フォント、フォントスタイル、そして画像プロパティなどが含まれます。

例えば、**4D Write Pro エリア**ライブラリオブジェクトはこれらの機能を存分に活用し、洗練された4D Write Proインターフェースを提供しています。



## セレクションレンジコマンド

いくつかのコマンドは、ドキュメント内の選択範囲を管理するために特化しています。選択されたテキストは(不通過の)フォーマットタグを含み得ることから、4D Write Proはレンジというものを使用します。レンジは4D Write ドキュメントの一部を表すオブジェクトです。

- **WP Get range**( *wpArea ; startRange ; endRange* ) -> *rangeObj*: 引数として渡した境界に対応する新しい範囲を返します。
- **WP Get selection**( {\*;} *wpArea* ) -> *rangeObj*: カレントのユーザーセレクションに対応した新しいレンジを返します。
- **WP Get pictures**( *rangeObj* ) -> *rangeObj*: ピクチャーのみを含んだ新しいレンジを返します。
- **WP Get paragraphs**( *rangeObj* ) -> *rangeObj*: 段落のみを含んだ新しいレンジを返します。
- **WP SELECT**( {\*;} *wpArea* {; *rangeObj*} {; *startRange ; endRange*} ): レンジに対応したテキストを選択します。

## ブックマークコマンド

4D Write Proでは**ブックマーク**と呼ばれる、ドキュメントの一部に対して動的な参照を作成し、使用することができます。ブックマークとは、4D Write Proドキュメント内の特定のレンジに関連づけられた名前付き参照です。

ブックマークは動的です。これはつまり、ブックマークに関連付いた文章をユーザーが移動または削除した場合には、紐付いているレンジも自動的に更新され、ブックマークはドキュメント内の同じ内容を引き続き参照し続けるという事です。例えば:

- ドキュメント内の20ページ目にある"Hello world"を参照する、"MyBM"というブックマークを作成したとします。
- 次に、ドキュメントの最初に50ページ挿入したとします。
- "MyBM"ブックマークを使用する事で、今度はドキュメント内の70ページ目にある、同じ"Hello world"というテキストに自動的にアクセスすることができます。

ドキュメント内には無制限のブックマークを含める事ができます。複数のブックマークが同じレンジを参照することもできますし、ブックマークのレンジは重複することもできます。しかしながら、それぞれブックマーク名はドキュメント内において固有でなければなりません。ブックマークは、**MissingRef** あるいは **WP New** コマンドを使用した場合にはインポートされません(保存先のドキュメントのブックマークは上書きされません)。

一度作成すると、ブックマークはドキュメント内に保存されています。ブックマークはドキュメントとともに保存され、複数のコマンドによって管理することができます。また、テンプレートドキュメントの一部を参照するのも使用できます。これらの部分はデータベース内のドキュメントと自動的に組み合わせることもでき、その結果請求書やカタログといった動的な出力ドキュメントを生成することもできます。

複数のコマンドを使用して、ブックマークを作成、削除、使用することができます:

- **WP CREATE BOOKMARK** はレンジから新しいブックマークを作成します。
- はドキュメント内で定義されている全てのブックマークを取得します。
- **WP Get bookmark range** は既存のブックマークからレンジを取得します。
- **WP DELETE BOOKMARK** はブックマークを削除します。

## 属性管理コマンド

---

以下のコマンドはドキュメントの属性を取得あるいは設定します:

- **WP SET ATTRIBUTES**( *rangeObj* | *wpDoc* ; *attribName* ; *attribValue* { ; *attribName2* ; *attribValue2* ; ... ; *attribNameN* ; *attribValueN* } ): ドキュメントまたはレンジ内の一つ以上の属性/値のペアを設定します。
- **WP GET ATTRIBUTES**( *rangeObj* | *wpDoc* ; *attribName* ; *attribValue* { ; *attribName2* ; *attribValue2* ; ... ; *attribNameN* ; *attribValueN* } ): ドキュメントまたはレンジ内の属性のカレントの値を取得します。
- **WP RESET ATTRIBUTES**( *rangeObj* ; *attribName* { ; *attribName2* ; ... ; *attribNameN* } ): ドキュメントまたはレンジ内の属性の値を再設定します。

属性は**4D Write Pro属性**の章に詳細な説明があります。

## フォント管理コマンド

---

このコマンドはレンジのスタイルサポートに関する情報を取得します:

- **WP Is font style supported**( *rangeObj* ; *wpFontStyle* ) -> *true* または *false*: レンジが特定のスタイルをサポートするかどうかを調べます(インターフェースのデザインに有用です)。



WP CREATE BOOKMARK ( rangeObj ; bkName )

引数	型		説明
rangeObj	Object	⇒	4D Write Proレンジ
bkName	文字	⇒	作成するブックマーク名

## 説明

**WP CREATE BOOKMARK**コマンドは、*rangeObj* 引数で指定された*bkName*引数で指定された名前の4D Write Proレンジを、レンジの親ドキュメント内に作成します。

ブックマークとはレンジに対する命名された参照で、例えばテンプレートを作成する目的などでドキュメントの特定の部分にアクセスし再利用する事ができるようになります。詳細な情報については、の章を参照して下さい。

*bkName* 引数には新しいブックマークの名前を渡して下さい。ブックマーク名はHTML/CSS名に準拠している必要があります(スペース文字などの無効な文字は自動的に削除されます)。ブックマーク名にはスペース文字を含める事はできますが、名前の先頭と最後につくスペースは削除されます。例えば、" my bookmark "という名前は、保存した際には"my bookmark"となります。ブックマーク名はドキュメント内において固有でなければなりません。同じ名前のブックマークが既にドキュメント内に存在していた場合、それは上書きされます。

ブックマークは同一ドキュメント内において無制限に作成することができます。同一のレンジを使用した複数のブックマークを作成することもできます。作成したあとは、ブックマークは自動的に親ドキュメントに保存され、そのドキュメント自身と共に保存されます。

## 例題 1

ドキュメント内で現在選択されているテキストを参照する新しいブックマークを作成したい場合を考えます。以下のように書くことができます:

```
C_OBJECT ($range)
$range:=WP Get selection (*;"WPDocument")
WP CREATE BOOKMARK ($range;"my_bookmark")
```

## 例題 2

既存のブックマークの名前を変更したい場合を考えます。そのためには、同じレンジを使用した新しいブックマークを作成し、古い方を削除する必要があります:

```
C_TEXT ($bookmarkOldName)
C_TEXT ($bookmarkNewName)
C_LONGINT ($p)
C_OBJECT ($wpRange)

$bookmarkOldName:="MyBookmark"
$bookmarkNewName:="MyNewBookmark"

ARRAY TEXT ($_bookmarks;0)
WP GET BOOKMARKS (WParea;$_bookmarks)

$p:=Find in array ($_bookmarks;$bookmarkOldName)
If ($p>0)
    $wpRange:=WP Get bookmark range (WParea;$bookmarkOldName)
    WP DELETE BOOKMARK (WParea;$bookmarkOldName)
    WP CREATE BOOKMARK ($wpRange;$bookmarkNewName)
End if
```



## WP DELETE BOOKMARK

WP DELETE BOOKMARK ( wpDoc ; bkName )

引数	型		説明
wpDoc	Object	→	4D Write Proドキュメント
bkName	文字	→	削除するブックマークの名前

### 説明

**WP DELETE BOOKMARK** コマンドは *wpDoc* 引数で指定した4D Write Proドキュメント内の、*bkName* 引数で指定した名前のブックマークを削除します。

*bkName* 引数のブックマークがの *wpDoc* ドキュメント内に存在しない場合、コマンドは何もしません。

### 例題

既存のブックマークの名前を変更したい場合を考えます。そのためには、同じレンジを使用した新しいブックマークを作成し、古い方を削除する必要があります:

```
C_TEXT ($bookmarkOldName)
C_TEXT ($bookmarkNewName)
C_LONGINT ($p)
C_OBJECT ($wpRange)

$bookmarkOldName:="MyBookmark"
$bookmarkNewName:="MyNewBookmark"

ARRAY TEXT ($_bookmarks;0)
WP GET BOOKMARKS (WParea;$_bookmarks)

$p:=Find in array ($_bookmarks;$bookmarkOldName)
If ($p>0)
    $wpRange:=WP Get bookmark range (WParea;$bookmarkOldName)
    WP DELETE BOOKMARK (WParea;$bookmarkOldName)
    WP CREATE BOOKMARK ($wpRange;$bookmarkNewName)
End if
```

WP EXPORT VARIABLE ( wpDoc ; destination ; format {; option} )

引数	型	説明
wpDoc	Object	→ 4D Write Pro変数
destination	テキスト変数, BLOB変数	← 書き出したコンテンツを受け取る変数
format	倍長整数	→ 変数出力フォーマット
option	倍長整数, 文字	→ 書き出しオプション

## 説明

**WP EXPORT VARIABLE** コマンドはwpDoc 変数内の4D Write Proオブジェクトをformat 引数で指定したフォーマットでdestination 引数で指定した4D変数へと書き出します。

wpDoc 引数には、書き出したい4D Write Proオブジェクトを渡します。

destination 引数には、書き出された4D Write Pro オブジェクトを受け取る変数を渡します。必要な変数の型は、format 引数で指定した書き出しフォーマットに応じて変化します:

- ネイティブな.4wpフォーマットを渡した場合、必要な変数はBLOB型になります。
- HTMLフォーマットを渡した場合、必要な変数はテキスト型になります。

format 引数には、**4D Write Pro** テーマから、使用する書き出しフォーマットを設定する定数を渡します。それぞれのフォーマットは特定の用法と関連しています。サポートされているフォーマットは以下の通りです:

定数	型	値	コメント
wk 4wp	倍 長 整 数	4	4D Write Proドキュメントは、ネイティブなアーカイブフォーマット(zipになったHTMLと、別のフォルダに保存された画像)にて保存されています。4D特有のタグも含まれ、4D式は計算はされていません。このフォーマットは、特に4D Write Proドキュメントをディスク上に損失なく保存してアーカイブする事に適しています。
wk mime html	倍 長 整 数	1	4D Write Proドキュメントは標準のMIME HTMLとして保存され、htmlドキュメントと画像はMIME パーツとして埋め込まれます(base64でエンコードされます)。式は計算され4D特有のタグは除去されます。このフォーマットはSMTP_QuickSend コマンドを使用してHTML Eメールを送信するのに特に適しています。
wk web page html 4D	倍 長 整 数	3	4D Write ProドキュメントはHTMLとして保存され、4D独自のタグを含みます。それぞれの式はノンブレークのスペースとして挿入されます。このフォーマットはロスレスであるため、テキストフィールドへの保存目的に適しています。

## 注:

- "4D独自のタグ" とは4Dネームスペースと4D CSSスタイルを含めた4D XHTMLのことです。
- 式は、書き出しをする前であれば**ST FREEZE EXPRESSIONS** コマンドを使用することでいつでも値を確定させることができます。
- 4D Write Pro ドキュメントフォーマットについてのより詳細な情報については、**.4wp ドキュメントフォーマット** を参照して下さい。

options 引数には、書き出しの詳細を設定するオプションを渡します。詳細は以下の通りです:

- 倍長整数 の値を渡してHTMLコードのスタイルを定義します。以下の定数を使用することができます。

定数	型	値	コメント
wk html debug	倍長整数	1	フォーマット済みのHTMLコード("整形済みフォーマット")。デバッグが容易。
wk normal	倍長整数	0	標準のHTML コード

- HTML デバッグオプションなし(デフォルト):

```
<html xmlns="http://www.w3.org/1999/xhtml"> <head> <title>New 4D Write Pro Document</title> <style
type="text/css">body { background-color:#FFFFFF }ul, ol { margin:0;padding:0 }p,li { white-space:pre-
wrap;margin:0pt;padding:0pt;font-family:'Times New Roman' }p.Normal,li.Normal { text-align:left }p._p1,li._p1
{ font-family:'Arial';font-size:18pt;color:#1D1D1D }img_img1 { width:51pt;height:51pt }</style> </head>
<body> <p class="Normal _p1"> <span>23/12/14</span>,</p> <p class="Normal _p1">Dear <span
style="color:#BE0E00">Shaun Stoltz</span><span style="color:#800062"> <span style="color:#000000">from
```

- HTML デバッグオプションあり:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>New 4D Write Pro Document</title>
<style type="text/css">
body { background-color:#FFFFFF }
ul, ol { margin:0;padding:0 }
```

- または文字列を渡す事もできます。以下のプロパティがサポートされています(ただし `wp mime html` フォーマットが使用されている場合に限り):
  - CID ホストドメイン名: 生成されたCIDのURLに追加されるホストドメイン(区切り文字として '@' を含む)。例えば, "gmail.com" を渡した場合、CIDユニークIDが123のときには '123@gmail.com' が挿入されます。省略された場合、デフォルトで、CIDユニークIDのみが使用されます(ほとんどのメールサーバーで使用可能です)。

## 例題

スタイル付テキストと4D参照と画像を含んだEメールを送信したい場合を考えます。この場合、MIME HTMLフォーマットで書き出された4D Write Proエリアを、4D Internet Commandを使用して送信することができます:

```
C_LONGINT($smtpid_t;$err_t;$smtpOption_t;$smtpPort_t)
C_TEXT($str;$emailBody_t;$smtpHost_t;$emailTo_t;$emailFrom_t;$smtpPass_t)

//エリアを適切なフォーマットで書き出す
WP_EXPORT VARIABLE(myWPArea;$str;wk_mime_html)
$emailTo_t:="johnsmith@4d.com"
$emailFrom_t:="testWritePro@gmail.com"
$emailBody_t:=$str
$smtpHost_t:="smtp.gmail.com"
$smtpOption_t:=9
$smtpPort_t:=465
$smtpPass_t:="QRN_on_bretzelburg"

$err_t:=SMTP_QuickSend($smtpHost_t;$emailFrom_t;$emailTo_t;$emailTitle_t;\
$emailBody_t;$smtpOption_t;$smtpPort_t;$smtpUser_t;$smtpPass_t)

If(($err_t=0))
ALERT("Email sent to "+emailTo_t)
Else
ALERT("Error in parameters, please try again.")
End if
```

WP GET ATTRIBUTES ( rangeObj | wpDoc ; attribName ; attribValue { ; attribName2 ; attribValue2 ; ... ; attribNameN ; attribValueN } )

引数	型	説明
rangeObj   wpDoc	Object	→ 4D Write Proレンジまたはドキュメント
attribName	文字	→ 取得する属性の名前
attribValue	文字, 実数, ブール, 配列	← テキストのレンジの属性のカレント値

## 説明

**WP GET ATTRIBUTES**コマンドは、4D Write Proレンジまたはドキュメント内のあらゆる属性の値を返します。このコマンドを使用すると4D Write Proのどのような内部属性(文字、段落、ドキュメント、画像など)にもアクセスする事ができます。

第一引数には、4D Write Proレンジオブジェクト(*rangeObj*)または4D Write ドキュメント参照(*wpDoc*)のいずれかを渡します。レンジオブジェクト(*rangeObj*)は他のコマンドによって作成される4D Write Proドキュメントの一部です。以下の表はコマンドのターゲットオブジェクトと属性による**WP GET ATTRIBUTES**コマンドのスコープをまとめたものです:

引数	共通の属性 ("verticalAlign"を除く)	ドキュメント のみの属性	段落のみの属性	文字のみの属性 (と"verticalAlign")
<b>WP Get paragraphs</b> の <i>rangeObj</i>	段落	ドキュメント	段落	段落
<b>WP Get pictures</b> の <i>rangeObj</i>	画像	ドキュメント	-	-
<b>WP Get range</b> または <b>WP Get selection</b> の <i>rangeObj</i>	レンジの段落	ドキュメント	レンジの段落(関係する段落)	レンジの文字
<i>wpDoc</i>	ドキュメント	ドキュメント	ドキュメント全体の段落	ドキュメント全体の文字

共通の属性("margin", "padding"など)は、ドキュメント、段落、画像に共通する属性ということです。

*attribName*引数に渡す属性とそれに対応する値の包括的な一覧については、**4D Write Pro属性**の章を参照して下さい。

引数として渡したレンジまたはドキュメントの、同じ属性内に異なる値があった場合、コマンドは以下のように対応します:

- 数値に対しては、wk mixedを返します。
- 配列の場合、空の配列を返します(*attribValue*が配列として定義されていた場合にはタブストップ、カラー)。ただしシャドウのオフセットに関しては、配列の値は常に2つのエントリーを含むので、wk shadow offsetを返します。これらのエントリーは水平方向あるいは垂直方向のオフセット(あるいはその両方)がミックスされていた場合には個別にwk mixedに設定することもできます。
- 文字列の値に対しては、空の文字列を返します。
- ピクチャーの値に対しては、空のピクチャーを返します。

## 例題

選択されたエリアの背景色を取得したい場合を考えます:

```
$range:=WP Get selection(*;"WParea")
WP GET ATTRIBUTES($range;wk_background_color;$bcol)
```

## WP Get bookmark range

WP Get bookmark range ( wpDoc ; bkName ) -> 戻り値

引数	型		説明
wpDoc	Object	→	4D Write Proドキュメント
bkName	テキスト	→	レンジを取得したいブックマーク名
戻り値	Object	↩	ブックマークのレンジ

### 説明

---

**WP Get bookmark range** コマンドは、*wpDoc*引数で指定した4D Write Proドキュメント内の、*bkName*引数で指定されたブックマークのレンジを格納したテキストレンジオブジェクト(*rangeObj*)を返します。

*bkName* のブックマークが*wpDoc* のドキュメント内に存在しない場合、空の*rangeObj*オブジェクトが返されます。

### 例題

---

ドキュメント内の"MyBookmark"ブックマークのレンジを表示したい場合を考えます:

```
C_OBJECT($wpRange)
$wpRange:=WP Get bookmark range (WParea;"MyBookmark")
WP SELECT (WParea;$wpRange)
```

## WP GET BOOKMARKS

WP GET BOOKMARKS ( wpDoc ; arrBKNames )

引数	型		説明
wpDoc	Object	→	4D Write Proドキュメント
arrBKNames	テキスト配列	←	ブックマーク名の配列

### 説明

---

**WP GET BOOKMARKS** コマンドは *wpDoc* 引数で指定した4D Write Proドキュメント内で定義されている全てのブックマーク名を含む配列を返します。

コマンド実行後、*arrBKNames* 変数にはドキュメント内の全てのブックマーク名が格納されます。配列内ではブックマーク名はドキュメント内での位置によって並べ替えられています。

同じ位置から始まる複数のブックマークが存在する場合、それらはアルファベット順に並べられます。

### 例題

---

ドキュメント内で定義されているブックマークの数を知りたい場合を考えます：

```
ARRAY TEXT ($_bookmarks;0)
WP GET BOOKMARKS (WParea;$_bookmarks)
ALERT ("The document contains "+Size of array($_bookmarks)+" bookmarks.")
```



## ⚙️ WP Get page count

WP Get page count ( wpDoc ) -> 戻り値

引数	型		説明
wpDoc	Object	→	4D Write Pro ドキュメント
戻り値	倍長整数	↩	ドキュメントのページ数

### 説明

---

**WP Get page count** コマンドは *wpDoc* に指定した 4D WritePro ドキュメントの総ページ数を返します。

### 例題

---

Items テーブルのカレントセレクション内で、"Manual" フィールドに格納されている 4D Write Pro ドキュメントの総ページ数を確認します:

```
C_LONGINT($pageCount)
C_LONGINT($totalCount)
FIRST RECORD([Items])
While(Not(End selection([Items])))
    $pageCount:=WP Get page count([Items]Manual)
    $totalCount:=$totalCount+$pageCount
NEXT RECORD([Items])
End while
ALERT("マニュアルの総ページ数: "+String($totalCount))
```

## ⚙️ WP Get paragraphs

WP Get paragraphs ( rangeObj ) -> 戻り値

引数	型		説明
rangeObj	Object	→	段落を取得するレンジ
戻り値	Object	↩	段落のみを格納したレンジ

### 説明

---

新しい**WP Get paragraphs**コマンドは、*rangeObj*引数内のレンジ内に含まれる段落のみだけを内包した特定のレンジオブジェクトを返します。返されたレンジオブジェクトの段落は**WP GET ATTRIBUTES**や**WP SET ATTRIBUTES**コマンドなどで使用する事ができ、それによって段落属性のみを管理する事ができます。

*rangeObj*引数には、有効な4D Write Proの標準のレンジオブジェクトを渡して下さい。*rangeObj*は4D Write Proドキュメントの一部であり、**WP Get selection**や**WP Get range**コマンド等を使用して作成する事ができます。

### 例題

---

段落に対してのみpaddingを設定したい場合を考えます:

```
$oParagraphs := WP Get paragraphs ($oSelection)
WP SET ATTRIBUTES ($oParagraphs; wk_padding; 20)
```

## WP Get pictures

WP Get pictures ( rangeObj ) -> 戻り値

引数	型		説明
rangeObj	Object	→	ピクチャーを取得するレンジ
戻り値	Object	↩	ピクチャーのみを含んだレンジオブジェクト

### 説明

新しい**WP Get pictures**コマンドは、*rangeObj*引数に渡したオブジェクト内に含まれるピクチャーのみを格納した特定のレンジオブジェクトを返します。返された画像レンジオブジェクトは、**WP GET ATTRIBUTES**と**WP SET ATTRIBUTES**によってピクチャーの属性のみを管理するのに使用します。

*rangeObj*引数には、有効な4D Write Pro 標準レンジオブジェクトを渡して下さい。*rangeObj* オブジェクトは4D Write Pro ドキュメントの一部で、**WP Get selection**や**WP Get range**コマンドを使用して作成する事が可能です。

### 例題

ピクチャーの境界線カラーのみを変えたい場合を考えます:

```
$oPicts:=WP Get pictures($oSelection)
WP SET ATTRIBUTES($oPicts;wk border color;"blue")
```

## WP Get range

WP Get range ( wpArea ; startRange ; endRange ) -> 戻り値

引数	型	説明
wpArea	Object	→ 4D Write Proオブジェクト変数またはフィールド
startRange	倍長整数	→ エリア内でのレンジの開始のオフセット
endRange	倍長整数	→ エリア内でのレンジの終わりのオフセット
戻り値	Object	↻ レンジオブジェクト

### 説明

**WP Get range** コマンドは *wpArea* 引数の 4D Write Pro エリア内から、*startRange* と *endRange* の間に含まれるセレクションを新しいテキストレンジオブジェクト (*rangeObj*) として返します。

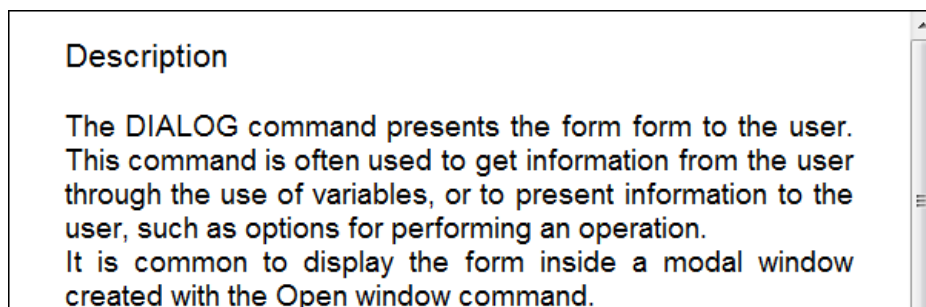
*wpArea* 引数には 4D Write Pro オブジェクト変数または変数を渡します。 *wpArea* 引数に有効な 4D Write Pro エリアが渡されなかった場合、空の *rangeObj* が返されます。

*startRange* と *endRange* 引数には、ドキュメント内で選択したい文字の最初の位置と最後の位置に対応する値を渡します。 *startRange* 引数に *wk start text* 定数を渡す事でドキュメントの最初を、 *endRange* 引数に *wk end text* 定数を渡す事でドキュメントの最後を指定する事ができます。 4D Write Pro ドキュメントは表示されているテキストだけではなく範囲内に含まれるフォーマットタグもあるという点に注意して下さい。

コマンドは新しい *rangeObj* を返します。 *rangeObj* は 4D Write Pro テキストレンジオブジェクトであり、これを使用してテキストセレクションの属性を管理する事ができます(その際には新しい **WP GET ATTRIBUTES** と **WP SET ATTRIBUTES** コマンドを使用します)。 これには三つの読み込みのみ属性 (*wk range start*, *wk range end* and *wk range owner*) があり、レンジ自身を定義するのに使用されます。

### 例題

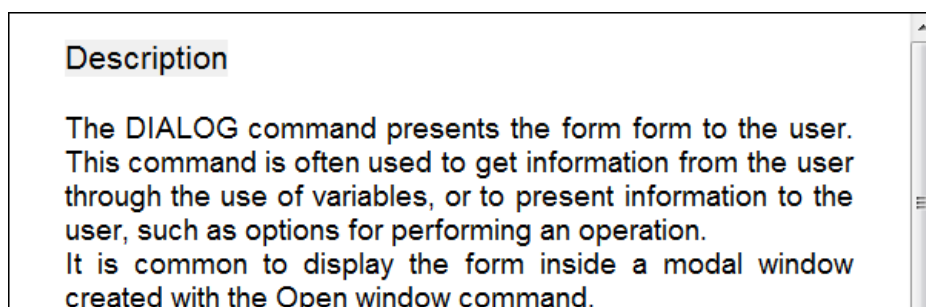
4D Write Pro フィールドの、最初から数えて 12 文字のレンジを選択したい場合を考えます。 フィールドはフォームオブジェクト内に表示されているとします:



以下のコードを実行した場合:

```
$range2:=WP Get range([SAMPLE]WP;wk_start_text;12)
WP SELECT(*;"WParea";$range2)
```

...結果は以下の画像のようになります:



WP Get selection ( { \* ; } wpArea ) -> 戻り値

引数	型	説明
*	演算子	→ 指定時、wpAreaはオブジェクト名(文字列)。省略時はwpAreaはオブジェクトフィールドあるいは変数。
wpArea	文字, Object	→ フォームオブジェクト名(*指定時)、または4D Write Proオブジェクト変数またはフィールド(*省略時)
戻り値	Object	↻ レンジオブジェクト

### 説明

**WP Get selection** コマンドは *wpArea* 引数で指定した4D Write Proエリア内で現在選択されているテキストに基づいた新しいテキストレンジオブジェクト(*rangeObj*)を返します。

任意の \* 引数を渡した場合、*wpArea*がフォームオブジェクト名(文字列)である事を指示します。この引数を渡さなかった場合、*wpArea*が4D Write Proオブジェクト変数あるいはフィールドであることを指示します。*wpArea*引数に有効な4D Write Proエリアが渡されなかった場合、空の*rangeObj*が返されます。

このコマンドは新しい*rangeObj*を返します。*rangeObj*は4D Write Proテキストレンジオブジェクトであり、これを使用してテキストセレクション属性を管理する事ができます(その際新しい**WP GET ATTRIBUTES**と**WP SET ATTRIBUTES**を使用します)。これには三つのプライベートな読み込みのみ属性(*wk range start*, *wk range end* and *wk range owner*)があり、レンジ自身を定義するのに使用されます。

### 例題

4D Write Proエリアから選択されたテキストを取得したい場合を考えます:

```
$range:=WP Get selection (*;"WParea")
```

WP INSERT BREAK ( rangeObj ; breakType ; mode {; rangeUpdate} )

引数	型		説明
rangeObj	Object	→	4D Write Proレンジオブジェクト
breakType	倍長整数	→	挿入するブレイクの型
mode	倍長整数	→	挿入モード
rangeUpdate	倍長整数	→	レンジ更新モード

## 説明

**WP INSERT BREAK** コマンドは *mode* 引数や *rangeUpdate* 引数で指定された形で、*rangeObj* 引数で指定されたレンジ内に *breakType* 引数で指定された型の新しいブレイクを挿入します。

*rangeObj* 引数には、有効な4D Write Pro標準レンジオブジェクトを渡します。*rangeObj* は4D Write Proドキュメントの一部で、**WP Get selection**、**WP Get bookmark range** あるいは **WP Get range** コマンドなどを使用して作成する事ができます。

*breakType* 引数には、**4D Write Pro**テーマからの以下の定数のどれか一つを渡し、挿入するブレイクのタイプを定義します:

定数	型	値	コメント
wk line break	倍長整数	0	(同じ段落内での)改行
wk page break	倍長整数	2	改ページ: 新しいページを定義します。
wk section break	倍長整数	1	セクションブレイク: 新しいセクションを定義します。

*mode* 引数には、対象となる *rangeObj* レンジ内で使用されるブレイクの挿入モードを指定する定数を渡します:

定数	型	値	コメント
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する
wk prepend	倍長整数	1	コンテンツをレンジの始めに挿入する
wk replace	倍長整数	0	レンジのコンテンツを置き換える

任意の *rangeUpdate* 引数内には、以下の定数のいずれか一つを渡し、挿入したコンテンツが挿入後のレンジに含まれるかどうかを指定します:

定数	型	値	コメント
wk exclude from range	倍長整数	1	更新されたレンジには、以前のコンテンツのみを含める
wk include in range	倍長整数	0	更新されたレンジに挿入コンテンツを含める

*rangeUpdate* 引数を渡さなかった場合、デフォルトで、挿入されたコンテンツは挿入後のレンジに含まれます。

## 例題

請求書を作成する際、最後のページを除いてページブレイク(改ページ)を挿入したい場合を考えます:

```

$nbInvoices:=Records in selection([INVOICE])
For($j;1;$nbInvoices)
    ... //請求書処理
    If($j#$nbInvoices) //最後のページを除いて改ページを挿入
        WP INSERT BREAK($buildRange;wk page break;wk append;wk exclude from range)
    End if
End for
    
```

WP INSERT DOCUMENT ( rangeObj ; wpDoc ; mode {; rangeUpdate} )

引数	型		説明
rangeObj	Object	→	4D Write Pro レンジオブジェクト
wpDoc	Object	→	4D Write Pro ドキュメント
mode	倍長整数	→	挿入モード
rangeUpdate	倍長整数	→	レンジ更新モード

## 説明

**WP INSERT DOCUMENT** コマンドは *mode* や *rangeUpdate* で指定した方法で、*rangeObj* に指定されたレンジに *wpDoc* のドキュメントを挿入します。

*rangeObj* には有効な 4D Write Pro 標準のレンジオブジェクトを渡します。レンジオブジェクトは 4D Write Pro ドキュメントの一部で、次のコマンドを使用して作成することができます: **WP Get selection**, **WP Get bookmark range**, **WP Get range**。

挿入する *wpDoc* ドキュメントは **WP New** または **WP Import document** で作成された 4D Write Pro ドキュメントオブジェクトを受け渡すことができます。挿入されるのは本文の要素のみです (挿入先のセクションおよびブックマークは変更されません)。挿入は実質的にはコピー処理されるため、*wpDoc* は再利用することができます。

*mode* パラメーターには、*rangeObj* レンジにドキュメントを挿入する際の挿入モードを指定するために、**4D Write Pro** テーマから一つ以上の定数を受け渡します:

定数	型	値	コメント
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する
wk prepend	倍長整数	1	コンテンツをレンジの始めに挿入する
wk replace	倍長整数	0	レンジのコンテンツを置き換える

上に紹介した定数を、次の挿入オプションと組み合わせることができます:

定数	型	値	コメント
wk freeze expressions	倍長整数	64	挿入の瞬間に式を評価します。
wk inherit style from paragraph	倍長整数	32	挿入されたコンテンツは、段落のデフォルト文字スタイルを継承します。
wk keep paragraph styles	倍長整数	128	移行先の段落スタイルを保持します。

任意の *rangeUpdate* パラメーターには、次の定数のいずれか一つを渡して、挿入コンテンツを処理後のレンジに含めるかどうかを指定します:

定数	型	値	コメント
wk exclude from range	倍長整数	1	更新されたレンジには、以前のコンテンツのみを含める
wk include in range	倍長整数	0	更新されたレンジに挿入コンテンツを含める

*rangeUpdate* パラメーターを省略した場合 (デフォルト)、挿入コンテンツは処理後のレンジに含まれます。

## 例題 1

ドキュメントの内容を、他のドキュメント内で選択されたテキストで置き換える場合を考えます:

```
$tempRange:=WP Get selection (WPTemplate) //WPTemplateドキュメント内のユーザーセレクションを取得
$doctoCopy:=WP New ($tempRange) //WPTemplateに基づいて新しいドキュメントを作成
WP INSERT DOCUMENT (WPDoc;$doctoCopy;wk_replace) //新しいドキュメントの中身でWPDocの中身を置き換え
```

## 例題 2

---

あらかじめフォーマットされ、それぞれがブックマークとして保存された複数のパーツで構成された、テンプレートドキュメントを定義しました。このテンプレートから任意のブックマークを新規ドキュメントとして抽出し、作成中のドキュメントに挿入することができます。

```
ARRAY TEXT($_BookmarkNames;0)
WP GET BOOKMARKS ([TEMPLATES]WP;$_BookmarkNames) // テンプレートからブックマークを取得します
$targetRange:=WP New //空のドキュメントを作成 (これが最終的なドキュメントになります)

$P:=Find in array($_BookmarkNames;"Main_Header") // 使用したいブックマークを名称で探します
If ($P>0)
    $Range:=WP Get bookmark range (WParea;$_BookmarkNames{$P}) // ブックマークからレンジを取得します
    $RangeDoc:=WP New ($Range) // レンジから新規ドキュメントを作成します
    WP INSERT DOCUMENT ($targetRange;$RangeDoc;wk_append+wk_freeze_expressions) // wk_append =
$RangeDocドキュメントは $targetRange の先頭に挿入されます
End if
```



WP INSERT PICTURE ( rangeObj ; picture ; mode {; rangeUpdate} )

引数	型	説明
rangeObj	Object	→ 4D Write Proレンジ
picture	ピクチャー, 文字	→ ピクチャーフィールドまたは変数、あるいはディスク上のピクチャーファイルへのパス
mode	倍長整数	→ 挿入モード
rangeUpdate	倍長整数	→ レンジ更新モード

## 説明

**WP INSERT PICTURE** コマンドは、指定した挿入モード *mode* に従って、*picture* のピクチャーを *rangeObj* で指定したレンジに挿入します。ピクチャーは文字として に挿入されます。

*rangeObj* には有効な 4D Write Pro 標準レンジオブジェクトを受け渡します。*rangeObj* オブジェクトは 4D Write Pro ドキュメントの一部で、**WP Get selection**、**WP Get bookmark range** や **WP Get range** コマンドを使用して作成することが可能です。

*picture* には次のものを渡すことができます:

- 4D ピクチャーフィールドまたは変数
- ディスク上のピクチャーファイルへのパス (システムシンタックス) を格納した文字列  
文字列を受け渡す場合には、フルパス名またはデータベースのストラクチャーファイルを起点とした相対パス名を指定することができます。データベースのストラクチャーファイルと同じ場所にピクチャーが保存されている場合には、ファイル名だけを渡すこともできます。

このピクチャーのフォーマットは4Dがネイティブにサポートされるものであればいずれの形式でも構いません (**ピクチャ** 参照)。**PICTURE CODEC LIST** コマンドを使用すると、使用可能なフォーマットのリストを取得できます。ピクチャーファイルに複数のフォーマット (コーデック) が含まれていると、4D Write Pro は表示用にひとつ、そして (異なる場合には) 印刷用にひとつ、"最適な" フォーマットを自動的に選択して保存します。

*mode* パラメーターには、ピクチャーの挿入モードを指定する定数を受け渡します:

定数	型	値	コメント
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する
wk prepend	倍長整数	1	コンテンツをレンジの始めに挿入する
wk replace	倍長整数	0	レンジのコンテンツを置き換える

任意の *rangeUpdate* パラメーターには、挿入したピクチャーをレンジに含めるか否かを指定する定数を受け渡すことができます:

定数	型	値	コメント
wk exclude from range	倍長整数	1	更新されたレンジには、以前のコンテンツのみを含める
wk include in range	倍長整数	0	更新されたレンジに挿入コンテンツを含める

*rangeUpdate* パラメーターを省略した場合には、挿入したピクチャーはレンジに含められます。

## 例題

この例題では、ユーザーが選択したピクチャーがレンジオブジェクトに挿入できなかった場合に警告します:

```
C_OBJECT($wpRange)
$wpRange:=WP Get selection([EXAMPLES]wpDoc)

C_BOOLEAN($fail)
$fail:=False

// ユーザーに挿入するピクチャーを選択してもらいます
$imgRef:=Open document("")
```

```
// ユーザーがキャンセルしなかった場合
If (OK=1)
// サポートされている形式のピクチャーファイルの場合
If (Is picture file (document))
// ユーザーが選択したピクチャーを挿入します
WP INSERT PICTURE ($wpRange; document; wk_replace)
Else
    $fail:=True
End if
Else
    $fail:=True
End if
// 挿入に失敗した場合には、ユーザーに警告します
If ($fail)
    ALERT ("Picture insertion failed")
End if
```

## WP Is font style supported

WP Is font style supported ( rangeObj ; wpFontStyle ) -> 戻り値

引数	型	説明
rangeObj	Object	→ 解析するレンジオブジェクト
wpFontStyle	倍長整数	→ フォントスタイル定数:wk font bold, wk font italic, wk text underline style, wk text linethrough style
戻り値	ブール	→ レンジの一部がwpFontStyleをサポートする場合にはTrue、それ以外にはFalse

### 説明

---

**WP Is font style supported**コマンドは引数で指定したレンジオブジェクトのどこかがwpFontStyleスタイルをサポートする場合、Trueを返します。

rangeObj引数には有効な4D Write proレンジオブジェクトを渡して下さい。rangeObjは4D Write Proドキュメントの一部であり、**WP Get selection**や**WP Get range**コマンド、あるいは**WP Get paragraphs / WP Get pictures**を使用して作成することができます。

wpFontStyle引数には、"4D Write Pro"定数テーマ内にある以下のスタイル定数のうちいずれか一つを渡します:

定数	型	値	コメント
wk font bold	文字列	fontBold	<p>テキストの太さを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● <code>wk true</code> を指定すると、選択された文字を太字フォントスタイルへと設定します。<b>WP GET ATTRIBUTES</b>コマンドを使用するとき、選択された文字のうち少なくとも1文字が太字フォントスタイルをサポートする場合、<code>wk true</code>が返されます。</li> <li>● <code>wk false</code> (デフォルト)を指定すると、選択された文字から太字フォントスタイルを除去します。<b>WP GET ATTRIBUTES</b>コマンドを使用するとき、選択された文字がどれも太字フォントスタイルをサポートしない場合、<code>wk false</code>が返されます。</li> </ul> <p>テキストの、イタリックスタイルを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● <code>wk true</code> は選択された文字をイタリック体あるいは斜体スタイルに設定します。<b>WP GET ATTRIBUTES</b>コマンドは選択された文字のうち少なくとも一つ以上の文字がイタリック体あるいは斜体フォントスタイルをサポートする場合、<code>wk true</code>が返されます。</li> <li>● <code>wk false</code> (デフォルト)は選択された文字のうちイタリック体あるいは斜体フォントスタイルがある場合にはそれらを除去します。コマンドは選択された文字のうちイタリック体あるいは斜体フォントスタイルをサポートする文字がない場合、<code>wk false</code>が返されます。</li> </ul> <p>テキストの打ち消し線のスタイルを(あれば)指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● <code>wk none</code> (デフォルト): 打ち消し線エフェクトはありません。</li> <li>● <code>wk solid</code>: 選択されたテキストに対して実線を描画します。</li> <li>● <code>wk dotted</code>: 選択されたテキストに対して点線を描画します。</li> <li>● <code>wk dashed</code>: 選択されたテキストに対して破線を描画します。</li> <li>● <code>wk double</code>: 選択されたテキストに対して二重線を描画します。</li> <li>● <code>wk semi transparent</code>: 選択されたテキストに対してグレーの線を描画します。他のスタイルと組み合わせる事が可能です。</li> <li>● <code>wk word</code>: 単語にのみ線を描画します(空白のスペースは描画しません)。他のスタイルと組み合わせる事が可能です。</li> </ul> <p>テキストの下線のスタイルを(あれば)指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● <code>wk none</code> (デフォルト): 下線エフェクトはありません。</li> <li>● <code>wk solid</code>: 実線の下線を描画します。</li> <li>● <code>wk dotted</code>: 点線の下線を描画します。</li> <li>● <code>wk dashed</code>: 破線の下線を描画します。</li> <li>● <code>wk double</code>: 二重線の下線を描画します。</li> <li>● <code>wk semi transparent</code>: グレーの下線を描画します。他の線スタイルと組み合わせる事が可能です。</li> <li>● <code>wk word</code>: 単語にのみ下線を描画します(空白のスペースは描画しません)。他の線スタイルと組み合わせる事が可能です。</li> </ul>
wk font italic	文字列	fontItalic	
wk text linethrough style	文字列	textLinethroughStyle	
wk text underline style	文字列	textUnderlineStyle	

一般的にこのコマンドは、選択されたテキストに基づいたスタイルオプションを提供するポップアップメニューなどのカスタムのインターフェースオブジェクトをデベロッパーが実装する目的で提供されています。

WP New {{ source }} -> 戻り値

引数	型	説明
source	文字, BLOB, Object	→ 4D HTMLソースまたは4D Write Blob
戻り値	Object	↻ 4D Write Pro オブジェクト

## 説明

**WP New** コマンドは4D Write Pro オブジェクトを作成し、返します。

*source* 引数を省略した場合、コマンドはデフォルトで空の4D Write Proオブジェクトを返します。

また *source* 引数を使用した場合、新しい4D Write Pro オブジェクトは*source* 引数の中身をコンテンツとして返されます。渡せる内容は以下の通りです:

- 文字列 の引数: この場合、4D HTMLソースを渡します。つまり、`wk web page html 4D` オプションを使用した**WP EXPORT VARIABLE** で書き出されたテキストです。このテキストは参照(4Dタグと式)と埋め込まれた画像を含むことができます。
- *blob* 引数: この場合、いかのどちらかを渡す事ができます:
  - BLOBに保存された4D Write Pro(.4wp)フォーマットドキュメント。4D Write Proドキュメントフォーマットの詳細については、[.4wp ドキュメントフォーマット](#) を参照して下さい。
  - BLOBに読み込まれている以前の4D Writeエリア(.4w7 または .4wt を含んだBlobがサポートされます)。4D Write Proオブジェクト内で現在サポートされている4D Writeの機能の詳細な一覧については、[4D Write ドキュメントの読み込み](#) の章を参照して下さい。

ディスク上に保存されている4D Write ドキュメント(.4w7 or .4wt) を読み込みたい場合、**WP Import document** コマンドの使用も検討してみてください。

- *object* 引数: この場合には、4D Write Pro レンジオブジェクトを渡します。**WP New** は指定したレンジから新規ドキュメントを作成して、これを返します。指定レンジがドキュメントの全レンジでない場合、最初のセクションだけがエクスポートされ、ブックマークなどが存在する場合には、これらは無視されます。

返されるオブジェクトは、例えば **MissingRef** コマンドに受け渡し可能な、完全なドキュメントです。

## 例題 1

空の4D Write Proオブジェクトを作成したい場合を考えます:

```
myWPObject:=WP New
```

## 例題 2

簡単な4D式の参照を含んだ4D Write Proオブジェクトを作成したい場合を考えます:

```
C_TEXT(myText)
myText:="Today is "
ST INSERT EXPRESSION(myText;"string(current date;System date long)";ST_End_text)
myWPA:=WP New(myText)
```

## 例題 3

以前作成したテンプレートを使用して4D Write Proエリアを初期化したい場合を考えます:

```
//既存のエリアからテンプレートを書き出し
C_TEXT(wpTemplate)
```

```
WP_EXPORT VARIABLE (myWPArea;wpTemplate;wk_web_page_html_4D)
```

```
// 新規エリアに対してテンプレートを使用
```

```
C_OBJECT (myNewWPA)
```

```
myNewWPA:=WP New (wpTemplate)
```

## 例題 4

4Dフィールドに保存されている4D Write ドキュメントを新しい4D Write Proエリア内に読み込みたい場合を考えます:

```
C_OBJECT (wpArea)
```

```
wpArea=WP New ([Templates]Reference_)
```

## 例題 5

あらかじめフォーマットされ、それぞれがブックマークとして保存された複数のパーツで構成された、テンプレートドキュメントを定義しました。このテンプレートから任意のブックマークを新規ドキュメントとして抽出し、作成中のドキュメントに挿入することができます。

```
ARRAY TEXT ($_BookmarkNames;0)
```

```
WP_GET BOOKMARKS ([TEMPLATES]WP;$_BookmarkNames) // テンプレートからブックマークを取得します
```

```
$targetRange:=WP New //空のドキュメントを作成 (これが最終的なドキュメントになります)
```

```
$p:=Find in array ($_BookmarkNames;"Main_Header") // 使用したいブックマークを名称で探します
```

```
If ($p>0)
```

```
    $Range:=WP Get bookmark range (WPArea;$_BookmarkNames{$p}) // ブックマークからレンジを取得します
```

```
    $RangeDoc:=WP New ($Range) // レンジから新規ドキュメントを作成します
```

```
    WP_INSERT_DOCUMENT ($targetRange;$RangeDoc;wk_append+wk_freeze_expressions) // wk_append =
```

```
$RangeDocドキュメントは $targetRange の先頭に挿入されます
```

```
End if
```

WP PRINT ( wpDoc {; printLayout} )

引数	型	説明
wpDoc	Object	→ 4D Write Pro のドキュメント名
printLayout	倍長整数	→ 4D Write Pro ドキュメントの印刷レイアウト: 0 (デフォルト)=4D Write Pro レイアウト、1=HTML WYSIWYG

## 説明

**WP PRINT** コマンドは、*wpDoc* で指定した 4D Write Proドキュメントの印刷ジョブをローンチし、64-bit版 4Dで **OPEN PRINTING JOB** と **CLOSE PRINTING JOB** の間に呼び出された場合には、ドキュメントをカレントの印刷ジョブに追加します (後述参照)。**WP PRINT** は通常の **PRINT SETTINGS** あるいは **SET PRINT OPTION** コマンドで定義された印刷設定を使用しますが、ページの余白に関しては常に 4D Write Pro ドキュメントの設定を使用します。また、**WP PRINT** はカレントのページ設定オプション (ページサイズやページの向きなど) を使用しますが、**WP USE PAGE SETUP** が先に呼び出されていた場合には、指定ドキュメントにおけるこれらの設定を使用します。

任意の *printLayout* パラメーターを使って、印刷出力に HTML WYSIWYG ビューを適用することができます。"4D Write Pro" テーマの次の定数を受け渡すことができます:

定数	型	値	コメント
wk 4D Write Pro layout	倍長整数	0	標準の4D Write Proレイアウト。いくつかの特定のスタイル属性を含める事ができます。
wk html wysiwyg	倍長整数	1	このレイアウトでは、4D Write Proの高度な属性のうち、全てのブラウザで互換性があるもの以外は除去されます。

*printLayout* を省略した場合のデフォルトは (0) です。

**注:** **WP PRINT** を使用して印刷した場合、エリアのビュープロパティ (**ビュープロパティの設定** 参照) の設定にかかわらず、4D Write Pro ドキュメントは常にページビューモードのとおり印刷されます。

## 32-bit版に関する注記

**WP PRINT** コマンドは 32-bit版の 4D でもサポートされていますが、32-bit版では **OPEN PRINTING JOB** によって開始された 4D のプリントジョブ内で呼び出すことはできません。32-bit版の 4D でプリントジョブ内でコールした場合にはエラーが生成されます。

## 例題

変数の値に応じて、4D Write Pro ドキュメントを標準レイアウト、あるいは HTML wysiwyg レイアウトで印刷します:

```
// HTML wysiwyg レイアウトまたは4D Write Pro レイアウトで印刷します
If(rb_htmlwysiwyg=1)
    WP PRINT(writeProDoc;wk_html_wysiwyg)
Else
    WP PRINT(writeProDoc;wk_4D_Write_Pro_layout)
End if
```

## WP RESET ATTRIBUTES

WP RESET ATTRIBUTES ( rangeObj ; attribName { ; attribName2 ; ... ; attribNameN } )

引数	型		説明
rangeObj	Object	→	4D Write Proレンジ
attribName	文字	→	除去したい属性の名前

### 説明

**WP RESET ATTRIBUTES** コマンドは *rangeObj* 4D Write Pro レンジオブジェクト内の一つ以上の属性の値をリセットします。このコマンドは4D Write Proのどのような内部(文字、段落、ドキュメント、または画像)の属性でも除去することができます。

*rangeObj* オブジェクトは別のコマンドで作成することができる4D Write Proドキュメントの一部です。以下の表は、ターゲットとなるオブジェクトと属性に応じた**WP RESET ATTRIBUTES**のスコープを表しています:

引数	共通の属性 ("verticalAlign"を 除く)	ドキュメント のみの属性	段落のみの属性	文字のみの属性 (と"verticalAlign")
<b>WP Get paragraphs</b> で取得した <i>rangeObj</i>	段落	ドキュメント	段落	段落
<b>WP Get pictures</b> で取得した <i>rangeObj</i>	ピクチャー	ドキュメント	-	-
<b>WP Get range</b> または <b>WP Get selection</b> で取得した <i>rangeObj</i>	レンジの段落	ドキュメント	レンジの段落(レンジとかぶる段落)	レンジの文字
<i>wpDoc</i>	ドキュメント	ドキュメント	ドキュメント全体の段落	ドキュメント全体の文字

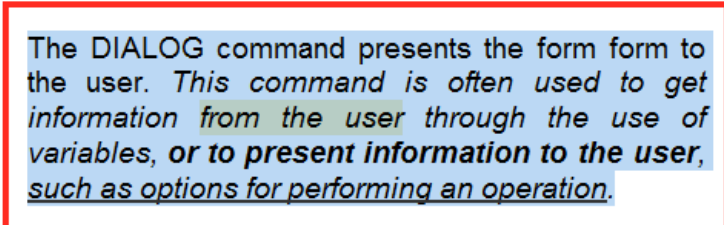
**WP RESET ATTRIBUTES** コマンドを使用して属性の値が除去されたとき、値はデフォルト値にリセットされます。デフォルト値については**4D Write Pro属性**の章に記されています。

*rangeObj*のレンジ内にリセットしたい属性がなかった場合、コマンドは何もしません。

### 例題

以下のセレクションから複数の属性を除去したい場合を考えます:

Description



The DIALOG command presents the form form to the user. *This command is often used to get information from the user through the use of variables, or to present information to the user, such as options for performing an operation.*

It is common to display the form inside a modal window created with the Open window command.

以下のコードを実行します:

```
$range:=WP Get selection (*;"WParea")
WP RESET ATTRIBUTES ($range;wk padding)
WP RESET ATTRIBUTES ($range;wk background color)
WP RESET ATTRIBUTES ($range;wk text underline style)
WP RESET ATTRIBUTES ($range;wk margin)
WP RESET ATTRIBUTES ($range;wk border style)
```



その結果ドキュメントは以下ようになります:

## Description

The DIALOG command presents the form form to the user. *This command is often used to get information from the user through the use of variables, or to **present information to the user**, such as options for performing an operation.*

It is common to display the form inside a modal window created with the Open window command.

WP SELECT ( { \* ; } wpArea { ; rangeObj } { ; startRange ; endRange } )

引数	型	説明
*	演算子	→ 指定時、wpAreaはフォームオブジェクト名(文字列)。省略時、wpAreaはオブジェクトフィールドまたは変数。
wpArea	文字、Object	→ フォームオブジェクト名(*指定時)または4D Write Proオブジェクト変数またはフィールド(*省略時)
rangeObj	Object	→ セレクションを作成するのに使用するレンジオブジェクト
startRange	倍長整数	→ テキストレンジの開始のオフセット
endRange	倍長整数	→ テキストレンジの終わりのオフセット

## 説明

**WP SELECT** コマンドは *wpArea* 引数で指定した4D Write Proエリアから、*rangeObj* または *startRange* と *endRange* で定義された新しい範囲に基づく新しいテキストのセレクションを作成します。

任意の \* 引数を渡した場合、*wpArea* 引数がフォームオブジェクト名(文字列)であることを指示します。この引数を省略した場合、*wpArea* 引数が4D Write Proオブジェクト変数あるいはフィールドであることを指示します。*wpArea* 引数に有効な4D Write Proエリアが渡されなかった場合、このコマンドは何もしません。

選択のレンジを定義するためには、既存のレンジオブジェクトを *rangeObj* 引数に渡すか、範囲を指定する *startRange* / *endRange* のペアを渡します:

- **第一シンタックス:** `WP SELECT({* ;} wpArea ; rangeObj)`  
*rangeObj* は4D Write Proドキュメントの一部です。これは **WP Get range**、**WP Get selection**、**WP Get paragraphs**、あるいは **WP Get pictures** 等の別のコマンドを使用して作成することができます。
- **第二シンタックス:** `WP SELECT({* ;} wpArea ; startRange ; endRange)`  
 この場合、*startRange* と *endRange* 引数にはドキュメントで選択したい最初の文字と最後の文字の位置に対応する値を渡します。*startRange* 引数に `wk start text` 定数を渡す事でドキュメントの最初を定義する事ができ、また *endRange* 引数に `wk end text` 定数を渡す事でドキュメントの最後を定義することができます。4D Write Proドキュメントは表示されているテキストだけではなく範囲内に含まれるフォーマットタグもあるという点に注意して下さい。

## 例題

以下のコードは:

```
$range2:=WP Get range ([SAMPLE]WP;wk start text;12)
WP SELECT (*;"WParea";$range2)
```

... このコードと同じ意味になります:

```
WP SELECT (*;"WParea";wk start text;12)
```

## WP SET ATTRIBUTES

```
WP SET ATTRIBUTES ( rangeObj | wpDoc ; attribName ; attribValue { ; attribName2 ; attribValue2 ; ... ; attribNameN ; attribValueN } )
```

引数	型	説明
rangeObj   wpDoc	Object	→ 4D Write Pro レンジまたはドキュメント
attribName	文字	→ 設定する属性名
attribValue	文字, 実数, ブール	→ 新しい属性の値

### 説明

新しい**WP SET ATTRIBUTES**コマンドを使用すると4D Write Proレンジあるいはドキュメントのあらゆる属性の値を設定することができます。このコマンドはあらゆる4D Write Pro内部(文字、段落、ドキュメント、またはピクチャー)の属性にアクセスすることができます。

第一引数には、4D Write Proレンジオブジェクト(*rangeObj*)または4D Writeドキュメント参照(*wpDoc*)を渡します。*rangeObj*は別のコマンドによって作成することのできる4D Write Proドキュメントの一部です。以下の表は以下の表は、ターゲットとなるオブジェクトと属性に応じた**WP SET ATTRIBUTES**のスコープを表しています:

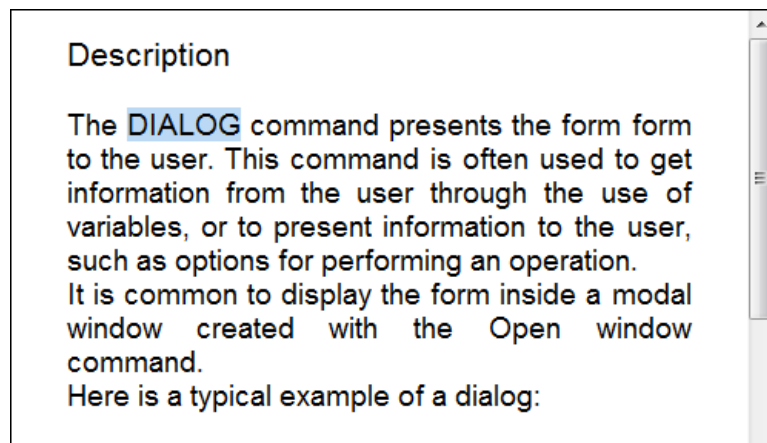
引数	共通の属性 ("verticalAlign"を 除く)	ドキュメン トのみの属 性	段落のみの属性	文字のみ (と"verticalAlign")の 属性
<b>WP Get paragraphs</b> で取得し た <i>rangeObj</i>	段落	ドキュメン ト	段落	段落
<b>WP Get pictures</b> で取得し た <i>rangeObj</i>	ピクチャー	ドキュメン ト	-	-
<b>WP Get range</b> または <b>WP Get selection</b> で取得した <i>rangeObj</i>	レンジの段落	ドキュメン ト	レンジの段落(レン ジとかぶる段落)	レンジの文字
<i>wpDoc</i>	ドキュメント	ドキュメン ト	ドキュメント全体 の段落	ドキュメント全体の文 字

共通の属性("margin"や"padding"など)は、ドキュメント、段落、ピクチャーに共通する属性という意味です。

*attribName*引数に渡す属性の包括的な一覧と、それに対応する値については、**4D Write Pro属性**の章を参照して下さい。

### 例題 1

この4D Write Proエリアにて、以下のように単語を選択した場合を考えます:



以下のコードを実行した場合:

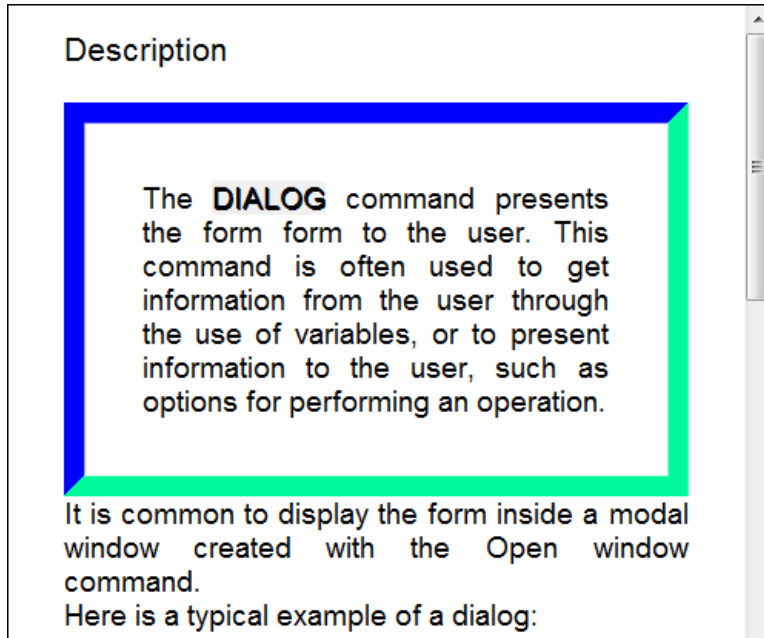
```
$range:=WP Get selection (*;"WParea") //選択されたレンジを取得  
  
// 選択されたテキストに対して影のオフセットをpt単位で設定
```

```

WP SET ATTRIBUTES ($range;wk text shadow offset;1)
//段落のパディングを設定
WP SET ATTRIBUTES ($range;wk padding;1)
//10 ptの境界線を定義
WP SET ATTRIBUTES ($range;wk border style;wk solid)
WP SET ATTRIBUTES ($range;wk border width;10)
//境界線カラーを設定
WP SET ATTRIBUTES ($range;wk border color;"blue")
WP SET ATTRIBUTES ($range;wk border color bottom;"#00FA9A") //ミディアムグリーン
WP SET ATTRIBUTES ($range;wk border color right;"#00FA9A")

```

以下の様な結果が得られます:



## 例題 2

以下の例では、`wk inside`と`wk outside`定数の使用を説明します:

```

$wpRange:=WP Get selection(writeProdoc)
WP SET ATTRIBUTES ($wpRange;wk border style+wk inside;wk dotted)
WP SET ATTRIBUTES ($wpRange;wk border style+wk outside;wk solid)
WP SET ATTRIBUTES ($wpRange;wk border color+wk outside;"#00FA9A")

```

ドキュメント全体が選択されていた場合、結果は以下のようになります:

```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus.
Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,
dolor.
.....
Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper
congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim
est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper.
.....
Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim.
Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras
vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit
odio eu enim.
.....
Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis
in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris
ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non
diam sodales hendrerit.

```

## WP USE PAGE SETUP

WP USE PAGE SETUP ( wpDoc )

引数	型		説明
wpDoc	Object	→	4D Write Proドキュメント名

### 説明

---

**WP USE PAGE SETUP** コマンドはカレントのプリンターページ設定を、4D Write Proドキュメント属性のページサイズとページの向きに変更します。このコマンドは、カレントのプリンターページ設定を4D Write Proドキュメントページ設定と同期させるために、**WP PRINT**の直前に呼び出される必要があります。

その他の設定は **PRINT SETTINGS** 4Dコマンドによって定義されます。カレントの印刷設定は 4Dセッション全体に対して設定されます。

### 例題

---

ドキュメントを印刷する前に、そのサイズと向きをドキュメント内に保存されている値に設定したい場合を考えます：

```
WP USE PAGE SETUP (writeProDoc)
```



定数	型	値	コメント
wk 4D Write Pro layout	倍長整数	0	標準の4D Write Proレイアウト。いくつかの特定のスタイル属性を含める事ができます。
wk 4wp	倍長整数	4	4D Write Proドキュメントは、ネイティブなアーカイブフォーマット(zipになったHTMLと、別のフォルダに保存された画像)にて保存されています。4D特有のタグも含まれ、4D式は計算はされていません。このフォーマットは、特に4D Write Proドキュメントをディスク上に損失なく保存してアーカイブする事に適しています。
wk append	倍長整数	2	コンテンツをレンジの終わりに挿入する
wk armenian	倍長整数	19	伝統的なアルメニア数字スタイルを使用( <a href="#">wk list style type</a> 用の値)
wk author	文字列	author	ドキュメントの著者名を指定(文字列)
wk auto	倍長整数	0	プロパティの値(定数)で、適用されると要素の中身またはコンテキストに応じて自動的に適応されます。
wk background clip	文字列	backgroundClip	<p>背景の塗りエリアを指定します。取り得る値:</p> <ul style="list-style-type: none"> <li><a href="#">wk border box</a> (デフォルト): 背景は境界線の外側の端まで塗られます。</li> <li><a href="#">wk content box</a>: 背景はコンテンツボックスの内部まで塗られます。</li> <li><a href="#">wk padding box</a>: 背景はパディングの橋野外側まで塗られます(もしくはあった場合には境界線の内側まで)。</li> </ul> <p>要素の背景色を指定します。取り得る値:</p> <ul style="list-style-type: none"> <li>CSS カラー("#010101" または "#FFFFFF" または "red")</li> <li>4D カラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列</li> </ul> <p>ドキュメントのデフォルトは"#FFFFFF" と <a href="#">wk transparent</a>で、段落と画像に対しては"transparent"がデフォルトです。</p> <p>背景として使用する画像を指定します。設定可能な値は以下の通りです:</p> <ul style="list-style-type: none"> <li>画像URL(文字列)。絶対パスまたはストラクチャーファイルあらの相対パスが可能です。</li> <li>ピクチャー変数またはフィールド</li> </ul> <p>(<b>WP GET ATTRIBUTES</b>によって)返された値: URI(ネットワークURLまたはデータURI)。ネットワークURLで参照されない画像については、最初のURLとは同じでないかもしれませんが(ネットワークURLのみが保持されます)。ローカルファイルのURLについては、画像ストリームそのものがドキュメント内に保存されるので、返されたURLはデータURIとbas64でエンコードされた画像ストリームです。</p> <p>背景画像どのような位置におかれるかを指定します。取りうる値は以下の通りです:</p>
wk background color	文字列	backgroundColor	
wk background image	文字列	backgroundImage	

- |                            |             |                  |                                                                                                                                                                                                                                                                             |
|----------------------------|-------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wk<br>background<br>origin | 文<br>字<br>列 | backgroundOrigin | <ul style="list-style-type: none"> <li>● <a href="#">wk padding box</a> (デフォルト): 背景画像はパディング(または内側の境界線の端)からスタートします。</li> <li>● <a href="#">wk border box</a>: 背景画像は境界線(外側の境界線の端)の四角形からスタートします。</li> <li>● <a href="#">wk content box</a>: 背景画像はコンテンツ四角形からスタートします。</li> </ul> |
|----------------------------|-------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

背景画像の水平方向のスタート位置を指定します。取り得る値は以下の通りです:

- |                                |             |                     |                                                                                                                                                                                                                                                        |
|--------------------------------|-------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wk<br>background<br>position h | 文<br>字<br>列 | backgroundPositionH | <ul style="list-style-type: none"> <li>● <a href="#">wk left</a> (デフォルト): 背景画像は水平方向に対して要素の左側からスタートします。</li> <li>● <a href="#">wk center</a>: 背景画像は水平方向に対して要素の中央からスタートします。</li> <li>● <a href="#">wk right</a>: 背景画像は水平方向に対して要素の右側からスタートします。</li> </ul> |
|--------------------------------|-------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

背景画像の垂直方向のスタート位置を指定します。取り得る値は以下の通りです:

- |                                |             |                     |                                                                                                                                                                                                                                                        |
|--------------------------------|-------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wk<br>background<br>position v | 文<br>字<br>列 | backgroundPositionV | <ul style="list-style-type: none"> <li>● <a href="#">wk top</a> (デフォルト): 背景画像は垂直方向に対して要素の上側からスタートします。</li> <li>● <a href="#">wk middle</a>: 背景画像は垂直方向に対して要素の中央からスタートします。</li> <li>● <a href="#">wk bottom</a>: 背景画像は垂直方向に対して要素の下側からスタートします。</li> </ul> |
|--------------------------------|-------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

背景画像をどのように繰り返すか(あるいは繰り返さないか)を指定します。取り得る値は以下の通りです:

- |                            |             |                  |                                                                                                                                                                                                                                                                                                    |
|----------------------------|-------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wk<br>background<br>repeat | 文<br>字<br>列 | backgroundRepeat | <ul style="list-style-type: none"> <li>● <a href="#">wk repeat</a> (デフォルト): 背景画像は垂直方向にも水平方向にも繰り返されます。</li> <li>● <a href="#">wk no repeat</a>: 背景画像は繰り返されません。</li> <li>● <a href="#">wk repeat x</a>: 背景画像は水平方向にのみ繰り返されます。</li> <li>● <a href="#">wk repeat y</a>: 背景画像は垂直方向にのみ繰り返されます。</li> </ul> |
|----------------------------|-------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

背景画像の水平方向のサイズを指定します。取り得る値は以下の通りです:

- |                            |             |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------|-------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wk<br>background<br>size h | 文<br>字<br>列 | backgroundSizeH | <ul style="list-style-type: none"> <li>● <a href="#">wk auto</a> (デフォルト): 背景画像はその幅を保ちます。</li> <li>● <a href="#">wk contain</a>: 背景画像のアスペクト比を保ったまま、画像全体が表示される範囲で最大のサイズへと拡大/縮小します。このオプションは他のサイズ属性の値も変更します。</li> <li>● <a href="#">wk cover</a>: 背景画像のアスペクト比を保ったまま、背景全体を画像でカバーできる大きさまで拡大/縮小します。このとき背景画像の一部は見切れる可能性があります。このオプションは他のサイズ属性の値も変更します。</li> <li>● 定義されたサイズ: 背景画像の水平方向のサイズは実数値あるいは文字列値を使用して表現されます: <ul style="list-style-type: none"> <li>○ 実数: サイズは<a href="#">wk layout unit</a>で指定します。</li> <li>○ 文字列: 連結された値と単位のCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。<br/>最小値: 0pt、最大値: 1000pt。相対値(パーセンテージ、%)はサポートされています。</li> </ul> </li> </ul> |
|----------------------------|-------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

背景画像の垂直方向のサイズを指定します。取り得る値は以下の通りです:

- |    |             |  |                                                                                                                                                                                                                                                                                                                          |
|----|-------------|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| wk | 文<br>字<br>列 |  | <ul style="list-style-type: none"> <li>● <a href="#">wk auto</a> (デフォルト): 背景画像はその高さを保ちます。</li> <li>● <a href="#">wk contain</a>: 背景画像のアスペクト比を保ったまま、画像全体が表示される範囲で最大のサイズへと拡大/縮小します。このオプションは他のサイズ属性の値も変更します。</li> <li>● <a href="#">wk cover</a>: 背景画像のアスペクト比を保ったまま、背景全体を画像でカバーできる大きさまで拡大/縮小します。このとき背景画像の一部は見切れる</li> </ul> |
|----|-------------|--|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



wk background size v	文字列	backgroundSizeV	<p>る可能性があります。このオプションは他のサイズ属性の値も変更します。</p> <ul style="list-style-type: none"> <li>定義されたサイズ: 背景画像の垂直方向のサイズを実数値あるいは文字列値を使用して表現します: <ul style="list-style-type: none"> <li>実数: サイズはwk layout unitで指定します。</li> <li>文字列: 連結された値と単位のCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。</li> </ul> </li> </ul> <p>最小値: 0pt、最大値: 1000pt。相対値(パーセンテージ、%)はサポートされています。</p>
wk bar	倍長整数	4	タブ位置に垂直のバーを挿入します(wk tab stop types用の値)
wk baseline	倍長整数	4	要素のベースラインを親要素のベースラインと合わせる(wk vertical align用の値)
wk border box	倍長整数	0	背景は境界線ボックスに合わせて見切れます(wk background clip用の値)
wk border color	文字列	borderColor	<p>境界線4つともカラーを設定します。取りうる値は以下の通りです:</p> <ul style="list-style-type: none"> <li>CSSカラー("#010101"または"#FFFFFF"または"red")。</li> <li>4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul> <p>デフォルト値は"#000000"です(文字列値の場合)。複数のカラーがある場合、<b>WP GET ATTRIBUTES</b>は空の文字列を返します。</p> <p>下の境界線のカラーを設定します。取りうる値は以下の通りです:</p>
wk border color bottom	文字列	borderColorBottom	<ul style="list-style-type: none"> <li>CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。</li> <li>4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul> <p>左の境界線のカラーを設定します。取りうる値は以下の通りです:</p>
wk border color left	文字列	borderColorLeft	<ul style="list-style-type: none"> <li>CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。</li> <li>4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul> <p>右の境界線のカラーを設定します。取りうる値は以下の通りです:</p>
wk border color right	文字列	borderColorRight	<ul style="list-style-type: none"> <li>CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。</li> <li>4Dカラーの倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列</li> </ul>

## 配列

上の境界線のカラーを設定します。取りうる値は以下の通りです:

wk border  
color top

文字列  
borderColorTop

- CSSカラー("#010101"または"#FFFFFF"または"red")。デフォルトは"#000000"です。
- 4Dカラーの倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納している倍長整数配列

角の丸い境界線を指定します。取り得る値は以下の通りです:

wk border  
radius

文字列  
borderRadius

- wk none (デフォルト): 境界線は丸い角を持ちません。
- 整数値または文字列値を使用して表現された半径:
  - 整数値: wk layout unitでの半径
  - 文字列値: 値と単位が連結されたCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチを意味します。

境界線の4辺全てを指定します。取り得る値は以下の通りです:

wk border  
style

文字列  
borderStyle

- wk none (デフォルト): 境界線なし
- wk hidden: wk noneと同様(境界線の競合解決時を除く)
- wk solid: 標準の境界線
- wk dotted: ドットの境界線
- wk dashed: 破線の境界線
- wk double: 二重境界線
- wk groove: 3D くぼみ境界線(実際の効果は境界線カラーによります)
- wk ridge: 3D 隆起境界線(実際の効果は境界線カラーによります)
- wk inset: 3D インセット境界線(実際の効果は境界線カラーによります)

下部の境界線を指定します。取り得る値は以下の通りです:

wk border  
style  
bottom

文字列  
borderStyleBottom

- wk none (デフォルト): 下部境界線なし
- wk hidden: wk noneと同様(境界線の競合解決時を除く)
- wk solid: 標準の下部境界線
- wk dotted: ドットの下部境界線
- wk dashed: 破線の下部境界線
- wk double: 二重下部境界線
- wk groove: 3D くぼみ下部境界線(実際の効果は下部境界線カラーによります)
- wk ridge: 3D 隆起下部境界線(実際の効果は下部境界線カラーによります)
- wk inset: 3D インセット下部境界線(実際の効果は下部境界線カラーによります)

左の境界線を指定します。取り得る値は以下の通りです:

wk border  
style left

文字列  
borderStyleLeft

- wk none (デフォルト): 左境界線なし
- wk hidden: wk noneと同様(境界線の競合解決時を除く)
- wk solid: 標準の左境界線
- wk dotted: ドットの左境界線
- wk dashed: 破線の左境界線
- wk double: 二重左境界線
- wk groove: 3D くぼみ左境界線(実際の効果は左境界線カラーによります)
- wk ridge: 3D 隆起左境界線(実際の効果は左境界線カラーによります)
- wk inset: 3D インセット左境界線(実際の効果は左境界線カラーによります)

右の境界線を指定します。取り得る値は以下の通りです:

- [wk none](#) (デフォルト): 右境界線なし
- [wk hidden](#): [wk none](#)と同様(境界線の競合解決時を除く)
- [wk solid](#): 標準の右境界線
- [wk dotted](#): ドットの右境界線
- [wk dashed](#): 破線の右境界線
- [wk double](#): 二重右境界線
- [wk groove](#): 3D くぼみ右境界線(実際の効果は右境界線カラーによります)
- [wk ridge](#): 3D 隆起右境界線(実際の効果は右境界線カラーによります)
- [wk inset](#): 3D インセット右境界線(実際の効果は右境界線カラーによります)

上部境界線を指定します。取り得る値は以下の通りです:

- [wk none](#) (デフォルト): 上部境界線なし
- [wk hidden](#): [wk none](#)と同様(境界線の競合解決時を除く)
- [wk solid](#): 標準の上部境界線
- [wk dotted](#): ドットの上部境界線
- [wk dashed](#): 破線の上部境界線
- [wk double](#): 二重上部境界線
- [wk groove](#): 3D くぼみ上部境界線(実際の効果は上部境界線カラーによります)
- [wk ridge](#): 3D 隆起上部境界線(実際の効果は上部境界線カラーによります)
- [wk inset](#): 3D インセット上部境界線(実際の効果は上部境界線カラーによります)

四辺全ての境界線の幅を指定します。境界線幅を指定する前に境界線スタイルを指定する必要があります。取り得る値は以下の通りです:

- 整数値あるいは文字列値を使用して表現された幅:
  - 整数値: [wk layout unit](#)での幅
  - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
- デフォルト値: 2pt

下部境界線の幅を指定します。取り得る値は以下の通りです:

- 整数値あるいは文字列値を使用して表現された幅:
  - 整数値: [wk layout unit](#)での幅
  - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
- デフォルト値: 2pt

左境界線の幅を指定します。取り得る値は以下の通りです:

- 整数値あるいは文字列値を使用して表現された幅:
  - 整数値: [wk layout unit](#)での幅
  - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
- デフォルト値: 2pt

右境界線の幅を指定します。取り得る値は以下の通りです:

- 整数値あるいは文字列値を使用して表現された幅:
  - 整数値: [wk layout unit](#)での幅
  - 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。
- デフォルト値: 2pt

上部境界線の幅を指定します。取り得る値は以下の通りです:

wk border width top	文字列 borderWidthTop	<ul style="list-style-type: none"> <li>整数値あるいは文字列値を使用して表現された幅: <ul style="list-style-type: none"> <li>整数値: <a href="#">wk layout unit</a>での幅</li> <li>文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12 points、1.5cmは1.5センチを意味します。</li> </ul> </li> <li>デフォルト値: 2pt</li> </ul>
wk bottom	倍長整数 1	背景画像の位置を設定( <a href="#">wk background position v</a> の値の場合)、あるいは要素の下部を行での最低の要素に合わせる( <a href="#">wk vertical align</a> の値の場合)
wk capitalize	倍長整数 1	それぞれの単語の最初の文字を大文字に変換します( <a href="#">wk text transform</a> 用の値)
wk center	倍長整数 2	テキストまたは画像を中央に配置します( <a href="#">wk background position h</a> 、 <a href="#">wk text align</a> 、または <a href="#">wk tab stop types</a> 用の値)
wk circle	倍長整数 11	円形の絵文字が使用されます( <a href="#">wk list style type</a> 用の値)
wk cjk ideographic	倍長整数 24	標準の表意文字の数字が使用されます( <a href="#">wk list style type</a> 用の値)
wk club	倍長整数 27	クラブ型の絵文字が使用されます( <a href="#">wk list style type</a> 用の値)
wk company	文字列 company	ドキュメントに関連づけられる会社を指定します(文字列)
wk contain	倍長整数 -1	画像の幅と高さがコンテンツエリアの内側に合致する最大のサイズまで画像を拡大/縮小します( <a href="#">wk background size h</a> と <a href="#">wk background size v</a> 用の値)
wk content box	倍長整数 2	背景はコンテンツボックスのサイズで見切れる( <a href="#">wk background clip</a> 用の値)、または背景画像はコンテンツの左上端からスタートする( <a href="#">wk background origin</a> 用の値)
wk cover	倍長整数 -2	背景画像を、背景エリアがその画像で完全に隠れる最小のサイズまで拡大/縮小します( <a href="#">wk background size h</a> または <a href="#">wk background size v</a> 用の値)
wk custom	倍長整数 29	カスタムのマーカースを使用します( <a href="#">wk list style type</a> 用の値)
wk dashed	倍長整数 3	破線を使用( <a href="#">wk text linethrough style</a> または <a href="#">wk text underline style</a> 用の値)
wk date	文	ドキュメントの作成日(日付)を返します。この値は読み込みのみで、設定する事

wk date creation	文字列	dateCreation	ドキュメントの作成日(日付)を返します。この値は読み込みのみであり、設定する事はできません。
wk date modified	文字列	dateModified	ドキュメントの最終更新日(日付)を返します。この値は読み込みのみであり、設定する事はできません。
wk decimal	倍長整数	3	小数点の桁揃え( <a href="#">wk tab stop types</a> 用の値)または使用される数字( <a href="#">wk list style type</a> 用の値)
wk decimal greek	倍長整数	28	ギリシャ数字が使用されます( <a href="#">wk list style type</a> 用の値)
wk decimal leading zero	倍長整数	13	小数は先頭にゼロがつけられます( <a href="#">wk list style type</a> 用の値)
wk default	倍長整数	-1	プロパティ(定数)のデフォルト値が使用されます。
wk diamond	倍長整数	26	ダイヤ型の絵文字が使用されます( <a href="#">wk list style type</a> 用の値)
wk direction	文字列	direction	段落のテキストの方向を指定します。取り得る値は以下の通りです: <ul style="list-style-type: none"> <li>• <a href="#">wk left to right</a> (デフォルト)</li> <li>• <a href="#">wk right to left</a></li> </ul>
wk disc	倍長整数	10	黒丸のマーカが使用されます( <a href="#">wk list style type</a> 用の値)
wk dotted	倍長整数	2	破線あるいは境界線が使用されています( <a href="#">wk border style</a> 、 <a href="#">wk text linethrough style</a> と <a href="#">wk text underline style</a> 用の値)
wk double	倍長整数	4	二重線あるいは境界線が使用されています( <a href="#">wk border style</a> 、 <a href="#">wk text linethrough style</a> と <a href="#">wk text underline style</a> 用の値)
wk dpi	文字列	dpi	内部でのピクセル<->ポイント変換に使用されるDPI(整数)。常に96です(読み込みのみ)
wk end text	倍長整数	0	ドキュメントの終わりをテキストレンジの終わりに設定します。
wk exclude from range	倍長整数	1	更新されたレンジには、以前のコンテンツのみを含める

wk false	調整数	0	
wk font	文字列	font	<p><b>FONT STYLE LIST</b> コマンドによって返される、完全なフォント名とスタイル名を指定します。無効なフォント名を設定した場合、コマンドは何もしません。デフォルト値:"Times New Roman"</p> <p>テキストの太さを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <u>wk true</u> を指定すると、選択された文字を太字フォントスタイルへと設定します。<b>WP GET ATTRIBUTES</b> コマンドを使用するとき、選択された文字のうち少なくとも1文字が太字フォントスタイルをサポートする場合、<u>wk true</u>が返されます。</li> <li>• <u>wk false</u> (デフォルト)を指定すると、選択された文字から太字フォントスタイルを除去します。<b>WP GET ATTRIBUTES</b> コマンドを使用するとき、選択された文字がどれも太字フォントスタイルをサポートしない場合、<u>wk false</u>が返されます。</li> </ul>
wk font bold	文字列	fontBold	
wk font family	文字列	fontFamily	<p><u>wk font</u> で定義されたフォントファミリー名を指定します。デフォルトの値は"Times New Roman"です。</p> <p>選択された文字に異なるフォントファミリープロパティが含まれる場合、<b>WP GET ATTRIBUTES</b> コマンドは空の文字列を返します。</p> <p>テキストの、イタリックスタイルを指定します(利用可能なフォントスタイルによります)。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <u>wk true</u> は選択された文字をイタリック体あるいは斜体スタイルに設定します。<b>WP GET ATTRIBUTES</b> コマンドは選択された文字のうち少なくとも一つ以上の文字がイタリック体あるいは斜体フォントスタイルをサポートする場合、<u>wk true</u>が返されます。</li> <li>• <u>wk false</u> (デフォルト)は選択された文字のうちイタリック体あるいは斜体フォントスタイルがある場合にはそれらを除去します。コマンドは選択された文字のうちイタリック体あるいは斜体フォントスタイルをサポートする文字がない場合、<u>wk false</u>が返されます。</li> </ul>
wk font italic	文字列	fontItalic	
wk font size	文字列	fontSize	<p>テキストのフォントサイズを指定します。取り得る値は以下の通りです(単位はポイントのみ):</p> <ul style="list-style-type: none"> <li>• 実際の値(デフォルト = 12)</li> <li>• 値と単位が連結されたCSS文字列。例: 12pt (12 ポイントを意味する)</li> </ul>
wk freeze expressions	倍長調整数	64	挿入の瞬間に式を評価します。
wk georgian	倍長調整数	20	伝統的なグルジア数字が使用されます( <u>wk list style type</u> 用の値)
wk groove	倍長調整数	6	3Dくぼみ境界線が使用されます( <u>wk border style</u> 用の値)
wk hebrew	倍長調整数	21	伝統的なヘブライ数字が使用されます( <u>wk list style type</u> 用の値)
			要素の高さを設定します。高さのプロパティには、パディング、境界線、マー

ジンは含まれません。このプロパティは要素のパディング、境界線、マージンの内側のエリアの高さを設定します。取り得る値は以下の通りです:

- `wk auto` (デフォルト): 高さは要素の中身に基づいて決められます。
- 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズです:
  - 実数: `wk layout unit`で設定された単位でのサイズ
  - 文字列: 値と単位が結合されたCSS文字列。例: 12ptは12ポイントを意味し、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10000pt

`wk height`属性は(定義されていれば)`wk min height`によって上書きされます。

**注:** 現行の実装では、`wk height`定数はピクチャーに対してのみ使用可能です。

`wk height` 文字列 height

`wk hidden` 倍長整数 5

境界線は使用されません。なし(境界線なし)と同じですが、境界線上での対立があった場合にはこちらが優先されます(`wk border style`用の値)

`wk hiragana` 倍長整数 22

ひらがなの数字が使用されます(`wk list style type`用の値)

`wk hollow square` 倍長整数 25

空の四角の絵文字が使用されます(`wk list style type`用の値)

`wk html debug` 倍長整数 1

フォーマット済みのHTMLコード("整形済みフォーマット")。デバッグが容易。

`wk html wysiwyg` 倍長整数 1

このレイアウトでは、4D Write Proの高度な属性のうち、全てのブラウザで互換性があるもの以外は除去されます。

画像を指定します。取り得る値は以下の通りです:

- 画像URL(文字列)。絶対パス、あるいはストラクチャーからの相対パスを指定できます。
- ピクチャー変数またはフィールド

`wk image` 文字列 image

返される値(**WP GET ATTRIBUTES**): URI(ネットワークURLまたはデータURI)。ネットワークURLで参照されていない画像については最初のURLと同一ではないかもしれませんが(ネットワークURLのみが保存されます)。ローカルファイルのURLに対しては、画像ストリーム自身がドキュメントに保存されるので、返されるURLはデータURIとbase64にエンコードされた画像ストリームです。

`wk image alternative text` 文字列 imageAltText

画像が表示できない際の、画像の代替テキストを指定します。

`wk include in range` 倍長整数 0

更新されたレンジに挿入コンテンツを含める

`wk inherit style from paragraph` 倍長整数 32

挿入されたコンテンツは、段落のデフォルト文字スタイルを継承します。

paragraph	数		
wk inset	倍長整数	8	3Dインセット境界線が使用されます( <a href="#">wk border style</a> 用の値)
wk inside	文字列	Inside	選択されたエリアに複数の段落が含まれる場合、属性は、対応して被っている段落のプロパティに対してのみ適用することを指定します(範囲の外側は含まれません)。境界線、パディング、マージン属性に対してのみ適用可能で、指定された属性に追加されなければなりません。 <b>WP SET ATTRIBUTES</b> コマンドの例2を参照して下さい。
wk justify	倍長整数	5	4D Write Proエリアに対してのみ適用可能です。
wk katakana	倍長整数	23	カタカナの数字が使用されます( <a href="#">wk list style type</a> 用の値)
wk keep paragraph styles	倍長整数	128	移行先の段落スタイルを保持します。
wk layout unit	文字列	userUnit	<p>値が整数あるいは実数として設定・取得された際の一部の属性の次元の単位を指定します。取り得る値は以下の通り:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk unit cm</a> (デフォルト): センチメートル</li> <li>• <a href="#">wk unit pt</a>: ポイント</li> <li>• <a href="#">wk unit px</a>: ピクセル</li> <li>• <a href="#">wk unit percent</a> (<a href="#">wk line height</a>、<a href="#">wk background size h</a> / <a href="#">wk background size v</a>に対してのみ使用可能)</li> <li>• <a href="#">wk unit mm</a>: ミリメートル</li> <li>• <a href="#">wk unit inch</a>: インチ</li> </ul>
wk left	倍長整数	0	テキストまたはタブを左側に揃える( <a href="#">wk text align</a> または <a href="#">wk tab stop types</a> 用の値)か、背景画像の開始位置を設定します( <a href="#">wk background position h</a> 用の値)
wk left to right	倍長整数	0	左から右へのテキストの読み書き方向が使用されます( <a href="#">wk direction</a> 用の値)
wk line break	倍長整数	0	(同じ段落内での)改行
wk line height	文字列	lineHeight	<p>行間スペースを指定します。取り得る値は以下の通り:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk normal</a> (デフォルト): テキストサイズに基づいた値を使用</li> <li>• 整数値あるいは文字列値で表現された高さを使用: <ul style="list-style-type: none"> <li>◦ 整数値: <a href="#">wk layout unit</a> の単位での高さ</li> <li>◦ 文字列値: 値と単位が結合されたCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチメートルを意味します。相対値(パーセンテージ)はサポートされています。</li> </ul> </li> </ul>
	倍長整数		



wk list auto	整数	2147483647	自動リストスタイル値を保存・適用します。
wk list font	文字列	listFont	<p><b>FONT STYLE LIST</b>コマンドで返される、完全なフォント名を指定してリスト項目マーカを表示するようにします(しかし段落のテキストは表示しません)。システムがフォント名を認識しない場合、その代理を使用します。無効なフォント名を設定した場合、コマンドは何もしません。デフォルト値: "Times"</p> <p><u>wk list font</u>によって定義されたフォントファミリー名を使用してリスト項目マーカを表示します(しかし段落のテキストは表示しません)。デフォルト値は"Times New Roman"です。</p> <p>並べられたリストの始まる値を設定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <u>wk auto</u> (デフォルト): 始まる値は、前のリスト項目があればそれに応じます。</li> <li>• 整数値: 始まる値</li> </ul>
wk list font family	文字列	listFontFamily	<p><u>wk list font</u>によって定義されたフォントファミリー名を使用してリスト項目マーカを表示します(しかし段落のテキストは表示しません)。デフォルト値は"Times New Roman"です。</p> <p>並べられたリストの始まる値を設定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <u>wk auto</u> (デフォルト): 始まる値は、前のリスト項目があればそれに応じます。</li> <li>• 整数値: 始まる値</li> </ul>
wk list start number	文字列	listStartNumber	<ul style="list-style-type: none"> <li>• <u>wk auto</u> (デフォルト): 始まる値は、前のリスト項目があればそれに応じます。</li> <li>• 整数値: 始まる値</li> </ul>
wk list string format LTR	文字列	listStringFormatLtr	<p>左から右への段落方向二対してのリスト項目マーカ文字列フォーマット。定義されると、リストのデフォルトのリスト項目マーカ文字列を上書きします。</p> <ul style="list-style-type: none"> <li>• 順番に並んでいないリスト: 文字列がリスト項目マーカとして使用されます(通常は単一文字による文字列、例: "-")</li> <li>• 順番に並んでいるリスト: "#"文字を含んだ文字列。"#"は計算された数字や文字のプレースホルダーになります。デフォルトは"#."なので、例えばカレントのリスト項目番号が15でリストスタイルタイプが小数だった場合、リスト項目マーカ文字列は"15."になります。</li> </ul>
wk list string format RTL	文字列	listStringFormatRtl	<p>右から左への段落方向に対してのリスト項目マーカ文字列フォーマット。定義されると、リストのデフォルトのリスト項目マーカ文字列を上書きします。</p> <ul style="list-style-type: none"> <li>• 順番に並んでいないリスト: 文字列がリスト項目マーカとして使用されます(通常は単一文字による文字列、例: "-")</li> <li>• 順番に並んでいるリスト: "#"文字を含んだ文字列。"#"は計算された数字や文字のプレースホルダーになります。デフォルトは"#."なので、例えばカレントのリスト項目番号が15でリストスタイルタイプが小数だった場合、リスト項目マーカ文字列は"15."になります。</li> </ul>
wk list style image	文字列	listStyleImage	<p>順番に並んでいないリストにおいて、画像をリスト項目マーカとして指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <u>wk none</u> (デフォルト): リスト項目マーカは画像によっては定義されていません。</li> <li>• ローカルファイルが像URL(文字列)。絶対パスあるいはデータベースのソースディレクトリからの相対パスが指定可能です。</li> </ul> <p>返される値(<b>WP GET ATTRIBUTES</b>): URI(ネットワークURLまたはデータURI)。ネットワークURLで参照されていない画像については最初のURLと同一ではないかもしれません(ネットワークURLのみが保存されます)。ローカルファイルのURLに対しては、画像ストリーム自身がドキュメントに保存されるので、返されるURLはデータURIとbase64にエンコードされた画像ストリームです。</p>
wk list style image height	文字列	listStyleImageHeight	<p>リスト項目マーカとして使用される画像の高さを設定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <u>wk auto</u> (デフォルト): 高さは画像のサイズに基づいて決められます。</li> <li>• 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>◦ 実数: <u>wk layout unit</u>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大</li> </ul> </li> </ul>

値: 1000pt。

順番付けされた、あるいは順番付けされていないリスト項目マーカのタイプを指定します。取り得る値は以下の通りです:

- [wk disc](#) (デフォルト)
- [wk circle](#)
- [wk square](#)
- [wk decimal](#): 1 2 3
- [wk decimal leading zero](#): 01 02 03
- [wk lower latin](#): a b c
- [wk lower roman](#): i ii iii iv
- [wk upper latin](#): A B C
- [wk upper roman](#): I II III IV
- [wk lower greek](#): α、β、γ、等
- [wk armenian](#)
- [wk georgian](#)
- [wk hebrew](#)
- [wk hiragana](#)
- [wk katakana](#)
- [wk cjk ideographic](#)
- [wk hollow square](#)
- [wk diamond](#)
- [wk club](#)
- [wk decimal greek](#)
- [wk custom](#): 順番付けされていないリストで "-" をデフォルトのリスト項目マーカとして使用します。これは標準のリスト項目マーカを変更する事なく [wk list string format LTR](#) または [wk list string format RTL](#) のリスト項目マーカをカスタマイズするのに便利なスタイルです。
- [wk none](#)

wk list style type 文字列 listStyleType

wk lower greek 倍長整数 18 小文字の伝統的なギリシャ文字が使用されます([wk list style type](#)用の値)

wk lower latin 倍長整数 14 小文字のASCII文字が使用されます([wk list style type](#)用の値)

wk lower roman 倍長整数 15 小文字のローマ数字が使用されます([wk list style type](#)用の値)

wk lowercase 倍長整数 2 全ての文字を小文字に変換します([wk text transform](#)用の値)

要素の全てのマージンのサイズを指定します。取り得る値は以下の通りです:

wk margin 文字列 margin

- 整数値あるいは文字列値を使用して表現されたサイズ:
  - 整数: [wk layout unit](#)の単位でのサイズ
  - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイント、1.5cmは1.5センチメートルを意味します。
- [wk none](#) (デフォルト): 特にマージンを指定しません。

要素の下部のマージンのサイズを指定します。取り得る値は以下の通りです:

wk margin bottom	文字列	marginBottom	<ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul> <p>要素の左のマージンのサイズを指定します。取り得る値は以下の通りです:</p>
wk margin left	文字列	marginLeft	<ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul> <p>要素の右のマージンのサイズを指定します。取り得る値は以下の通りです:</p>
wk margin right	文字列	marginRight	<ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul> <p>要素の上部のマージンのサイズを指定します。取り得る値は以下の通りです:</p>
wk margin top	文字列	marginTop	<ul style="list-style-type: none"> <li>● 整数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 整数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>● <a href="#">wk none</a> (デフォルト): 特にマージンを指定しません。</li> </ul>
wk middle	倍長整数	2	<p>背景画像の位置を設定(<a href="#">wk background position</a> v用の値)、あるいは要素を親要素の中央に配置します(<a href="#">wk vertical align</a>用の値)。</p>
wk mime html	倍長整数	1	<p>4D Write Proドキュメントは標準のMIME HTMLとして保存され、htmlドキュメントと画像はMIMEパーツとして埋め込まれます(base64でエンコードされます)。式は計算され4D特有のタグは除去されます。このフォーマットは<b>SMTP_QuickSend</b> コマンドを使用してHTML Eメールを送信するのに特に適しています。</p> <p>要素の最小限の高さを設定します。これは<a href="#">wk height</a>プロパティの値が<a href="#">wk min height</a>プロパティより小さくなることを防ぎます。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>● <a href="#">wk auto</a> (デフォルト): 最小限の高さは要素の中身に基づいて決められます。</li> <li>● 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ: <ul style="list-style-type: none"> <li>○ 実数: <a href="#">wk layout unit</a>の単位でのサイズ</li> <li>○ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 1000pt。</li> </ul> </li> </ul> <p><a href="#">wk min height</a>の値は<a href="#">wk height</a>属性を上書きします。</p> <p><b>注:</b> 現行の実装では、これはピクチャーに対してのみ使用可能です。</p> <p>要素の最小限の幅を設定します。これは<a href="#">wk width</a>プロパティの値が<a href="#">wk min width</a>プロパティより小さくなることを防ぎます。取り得る値は以下の通りです:</p>
wk min height	文字列	minHeight	

- wk auto (デフォルト): 最小限の幅は要素の中身に基づいて決められます。
- 定義されたサイズ: 実数値あるいは文字列値を使用して表現されたサイズ:
  - 実数: wk layout unitの単位でのサイズ
  - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 1000pt。

wk min widthの値はwk width属性を上書きします。

**注:** 現行の実装では、これはピクチャーに対してのみ使用可能です。

wk min width

文字列  
minWidth

wk mixed

倍長整数  
-2147483648

レンジまたはドキュメント内のある属性に対して異なる値がある場合に返されま

す。  
段落に新たな行を追加する際のスタイルシートを指定します。取り得る値は以下の通りです:

wk new line style sheet

文字列  
newLineStyleSheet

- 既存のスタイルシート名
- wk none (デフォルト)

wk no repeat

倍長整数  
3

背景画像は繰り返されません(wk background repeat用の値)

wk none

倍長整数  
0

wk normal

倍長整数  
0

標準のHTML コード

wk notes

文字列  
notes

ドキュメントについてのコメントを指定します(文字列)

wk outset

倍長整数  
9

3Dアウトセット境界線が使用されます(wk border style用の値)

wk outside

文字列  
Outside

選択されたエリアが複数の段落を含むとき、ある属性が対応する段落の外側のプロパティにのみ適用されることを指定します(内側には適用されません)。これは境界線、パディング、そしてマージン属性に対してのみ指定可能で、特定の属性に対して追加される必要があります。**WP SET ATTRIBUTES**コマンドの例2を参照して下さい。

要素の全てのサイドのパディングを指定します。取り得る値は以下の通りです:

wk padding

文字列  
padding

- 整数値あるいは文字列値を使用して表現されるサイズ:
  - 整数: wk layout unitの単位でのサイズ
  - 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。
- wk none (デフォルト): 特定のpaddingを指定しません。

要素の下部のpaddingを指定します。取り得る値は以下の通りです:

wk padding bottom	文字列	paddingBottom	<ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk padding box	倍長整数	1	<p>背景はpaddingボックスのサイズに切り取られます(<code>wk background clip</code>用の値)、あるいは、背景画像はpaddingの左上端の際から開始されます(<code>wk background origin</code>用の値)。</p> <p>要素の左側のpaddingを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul> <p>要素の右側のpaddingを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul> <p>要素の上部のpaddingを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk padding left	文字列	paddingLeft	<ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul> <p>要素の右側のpaddingを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul> <p>要素の上部のpaddingを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk padding right	文字列	paddingRight	<ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul> <p>要素の上部のpaddingを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk padding top	文字列	paddingTop	<ul style="list-style-type: none"> <li>• 整数値あるいは文字列値を使用して表現されるサイズ: <ul style="list-style-type: none"> <li>◦ 整数: <code>wk layout unit</code>の単位でのサイズ</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例:12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。</li> </ul> </li> <li>• <code>wk none</code> (デフォルト): 特定のpaddingを指定しません。</li> </ul>
wk page break	倍長整数	2	改ページ: 新しいページを定義します。
wk prepend	倍長整数	1	コンテンツをレンジの始めに挿入する
wk range end	文字列	rangeEnd	(読み込み専用のレンジ属性)
wk range owner	文字列	rangeOwner	(読み込み専用のレンジ属性)
wk range start	文字列	rangeStart	(読み込み専用のレンジ属性)
wk range type	文字列	rangeType	(Read-only range attribute) Type of 4D Write Pro range. Can be 0: default range (Default), 1: paragraph range, 2: image range
wk repeat	倍長整数	0	背景画像は垂直方向にも水平方向にも繰り返されます( <code>wk background repeat</code> 用の値)

wk repeat x	数 倍 長 整 数	1	背景画像は水平方向にのみ繰り返されます( <a href="#">wk background repeat</a> 用の値)
wk repeat y	数 倍 長 整 数	2	背景画像は垂直方向にのみ繰り返されます( <a href="#">wk background repeat</a> 用の値)
wk replace	数 倍 長 整 数	0	レンジのコンテンツを置き換える
wk ridge	数 倍 長 整 数	7	3D 隆起境界線が使用されます( <a href="#">wk border style</a> 用の値)
wk right	数 倍 長 整 数	1	テキストまたはタブを右揃えにする( <a href="#">wk text align</a> または <a href="#">wk tab stop types</a> 用の値)、あるいは背景画像の開始位置を設定します( <a href="#">wk background position h</a> 用の値)
wk right to left	数 倍 長 整 数	1	右から左への方向が使用されます( <a href="#">wk direction</a> 用の値)
wk section break	数 倍 長 整 数	1	セクションブレイク:新しいセクションを定義します。
wk semi transparent	数 倍 長 整 数	5	半透明の線が使用されます( <a href="#">wk text linethrough style</a> または <a href="#">wk text underline style</a> 用の値)
wk small uppercase	数 倍 長 整 数	4	全ての文字を小さな大文字(スモールキャップス)に変換します( <a href="#">wk text transform</a> 用の値)
wk solid	数 倍 長 整 数	1	実線、あるいは境界線が使用されます( <a href="#">wk border style</a> , <a href="#">wk text linethrough style</a> または <a href="#">wk text underline style</a> 用の値)
wk square	数 倍 長 整 数	12	四角のマーカースが使用されます( <a href="#">wk list style type</a> 用の値)
wk start text	数 倍 長 整 数	1	ドキュメントの最初をテキストレンジの初めに設定します。
wk style sheet	文 字 列	styleSheet	<p>選択された要素に対してカレントのスタイルシートを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk none</a> (デフォルト)</li> <li>• 既存のスタイルシート名</li> </ul>

wk subject	文字列	subject	ドキュメントの題名を指定します(文字列)
wk subscript	倍長整数	6	要素を下付きのテキストとして配置させます( <a href="#">wk vertical align</a> 用の値)
wk superscript	倍長整数	5	要素を上付きのテキストとして配置させます( <a href="#">wk vertical align</a> 用の値)
wk tab stop offsets	文字列	tabStopOffsets	<p>段落のタブストップを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>スカラー値(デフォルトは35.45pt): 段落全体のデフォルトのオフセットです。<b>WP GET ATTRIBUTES</b>コマンドは最後に使用されたオフセットを返します(最後の絶対オフセットより前のオフセットについては、デフォルトの相対オフセットになります)。</li> <li>タブ値の配列: 絶対値の順番付けされたリストで、左のマージンから始まります。最後の値によって定義されたタブオフセットは、段落内で入力される追加のタブ文字にて繰り返し使用されます。タブオフセットが段落の幅より大きい場合、テキストは次の行に移行し、最初のタブ値から開始されます。配列内の値が以前の値より小さい場合、それは無視されます。</li> </ul> <p><b>注:</b> 同じ呼び出しにおいて異なる属性に対して配列とスカラーを同時に使用することはできません。</p> <p>値はCSS文字列(デフォルト)または<a href="#">wk layout unit</a>での単位を使用する実数値で表現されます。最大値は1000ptです。</p> <p>段落でのタブストップタイプを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li><a href="#">wk left</a> (デフォルト): テキストはタブストップから右側に拡張されます。</li> <li><a href="#">wk right</a>: タブスペースが埋まるまで、テキストはタブストップから左側に拡張されます。</li> <li><a href="#">wk center</a>: テキストはタブストップにて中央に配置されます。</li> <li><a href="#">wk decimal</a>: 小数点より前のテキストは左に拡張され、小数点より後のテキストは右に拡張されます。</li> <li><a href="#">wk bar</a>: 指定された位置に垂直の線がおかれます。</li> <li>タブストップ値の配列(タブストップが配列を通して定義されていた場合)</li> </ul> <p>段落内でのテキストの行揃えを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li><a href="#">wk left</a> (デフォルト)</li> <li><a href="#">wk right</a></li> <li><a href="#">wk justify</a></li> <li><a href="#">wk center</a></li> </ul> <p>テキストのカラーを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>CSSカラー("#010101" や "#FFFFFF" または "red")。デフォルトは文字列の場合"#000000"です。</li> <li>4Dカラー倍長整数値(<b>OBJECT SET COLOR</b>コマンドを参照して下さい)</li> <li>R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列</li> </ul> <p>段落の最初の行のインデントを指定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>実数: <a href="#">wk layout unit</a>の単位でのサイズ。デフォルトは0です。</li> <li>文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値:</li> </ul>
wk tab stop types	文字列	tabStopTypes	
wk text align	文字列	textAlign	
wk text color	文字列	color	
wk text indent	文字列	textIndent	

10000pt

テキストの打ち消し線のカラーを指定します。取り得る値は以下の通りです:

wk text  
linethrough  
color 文  
字列 textLinethroughColor

- CSSカラー("#010101"や"#FFFFFF" または "red")
- 4Dカラー倍長整数値(**OBJECT SET COLOR**コマンドを参照してください)
- R、G、B、コンポーネントそれぞれの要素(0-255)を格納した倍長整数配列

デフォルトは、文字列なら"currentColor"、あるいは倍長整数なら wk default です。

テキストの打ち消し線のスタイルを(あれば)指定します。取り得る値は以下の通りです:

wk text  
linethrough  
style 文  
字列 textLinethroughStyle

- wk none (デフォルト): 打ち消し線エフェクトはありません。
- wk solid: 選択されたテキストに対して実線を描画します。
- wk dotted: 選択されたテキストに対して点線を描画します。
- wk dashed: 選択されたテキストに対して破線を描画します。
- wk double: 選択されたテキストに対して二重線を描画します。
- wk semi transparent: 選択されたテキストに対してグレーの線を描画します。他のスタイルと組み合わせる事が可能です。
- wk word: 単語にのみ線を描画します(空白のスペースは描画しません)。他のスタイルと組み合わせる事が可能です。

選択されたテキストのシャドウカラーを指定します。取り得る値は以下の通りです:

wk text  
shadow  
color 文  
字列 textShadowColor

- CSSカラー("#010101" や "#FFFFFF" または "red")
- 4Dカラー倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列
- wk transparent (デフォルト)

wk text  
shadow  
offset 文  
字列 textShadowOffset

シャドウエフェクトのオフセットを指定します。取り得る値:

- ポイント単位で指定されたサイズ。デフォルト値: 1pt

テキスト内での大文字と小文字を指定します。取り得る値は以下の通り:

wk text  
transform 文  
字列 textTransform

- wk capitalize: 最初の文字は大文字に設定されます。
- wk lowercase: 文字は小文字に設定されます。
- wk uppercase: 文字は大文字に設定されます。
- wk small uppercase: 文字はスモールキャップスに設定されます。
- wk none (デフォルト): 変換なし

テキストの下線のカラーを指定します。取り得る値:

wk text  
underline  
color 文  
字列 textUnderlineColor

- CSS カラー("#010101" や "#FFFFFF" または "red")
- 4Dカラー倍長整数値(**OBJECT SET COLOR**コマンドを参照して下さい)
- R、G、Bコンポーネントそれぞれの要素(0-255)を格納した倍長整数配列

デフォルトは、文字列なら"currentColor"、あるいは倍長整数なら wk default です。

テキストの下線のスタイルを(あれば)指定します。取り得る値は以下の通りです:

wk text 文

- wk none (デフォルト): 下線エフェクトはありません。
- wk solid: 実線の下線を描画します。
- wk dotted: 点線の下線を描画します。



underline style	文字列	textUnderlineStyle	<ul style="list-style-type: none"> <li>• <u>wk dashed</u>: 破線の下線を描画します。</li> <li>• <u>wk double</u>: 二重線の下線を描画します。</li> <li>• <u>wk semi transparent</u>: グレーの下線を描画します。他の線スタイルと組み合わせる事が可能です。</li> <li>• <u>wk word</u>: 単語にのみ下線を描画します(空白のスペースは描画しません)。他の線スタイルと組み合わせる事が可能です。</li> </ul>
wk title	文字列	title	ドキュメントのタイトル(文字列)を指定します。デフォルトは"New 4D Write Pro Document"です。
wk top	倍長整数	0	背景画像の位置を指定する( <u>wk background position v</u> 用の値)、あるいは要素を行上の最も高い要素の上に揃えます( <u>wk vertical align</u> 用の値)。
wk transparent	倍長整数	-1	透明色を指定します( <u>wk background color</u> あるいは <u>wk text shadow color</u> 用の値)
wk true	倍長整数	1	
wk unit cm	文字列	cm	使用される単位はセンチメートルです( <u>wk layout unit</u> 用の値)
wk unit inch	文字列	in	使用される単位はインチです( <u>wk layout unit</u> 用の値)
wk unit mm	文字列	mm	使用される単位はミリメートルです( <u>wk layout unit</u> 用の値)
wk unit percent	文字列	%	使用される単位はパーセンテージです( <u>wk layout unit</u> 用の値)
wk unit pt	文字列	pt	使用される単位はポイントです( <u>wk layout unit</u> 用の値)
wk unit px	文字列	px	使用される単位はピクセルです( <u>wk layout unit</u> 用の値)
wk upper latin	倍長整数	16	大文字のASCII文字が使用されます( <u>wk list style type</u> 用の値)
wk upper roman	倍長整数	17	大文字のローマ数字が使用されます( <u>wk list style type</u> 用の値)
wk uppercase	倍長整数	3	全ての文字を大文字に変換します( <u>wk text transform</u> 用の値)
wk value unit not	倍長	100000	<b>WP GET ATTRIBUTES</b> コマンドに対して、カレントの値の単位がパーセン

unit not percentage	整数	-100000	ページではないのに文字列値を渡した場合に返されます(不正な結果です)。
wk value unit percentage	倍長整数	-100001	<b>WP GET ATTRIBUTES</b> コマンドに対して、カレントの値の単位がパーセンテージなのに数値を渡した場合に返されます(不正な結果です)。
wk version	文字列	version	<p>ドキュメントの内部4DWPバージョンを返します(実数)。この数値は<b>WP GET ATTRIBUTES</b>を使用してのみ読み込み可能です。設定することはできません。</p> <p>要素の垂直揃えを設定します。文字、段落、ピクチャーに対して使用可能です。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk baseline</a> (デフォルト): 要素のベースラインを親要素のベースラインと揃えます。</li> <li>• <a href="#">wk top</a>: 要素の上を、行上で最も高い要素の上と揃えます。</li> <li>• <a href="#">wk bottom</a>: 要素の下を、行上で最も低い要素の下と揃えます。</li> <li>• <a href="#">wk middle</a>: 要素は親要素の中央に配置されます。</li> <li>• <a href="#">wk superscript</a>: 要素を上付き文字として配置します。</li> <li>• <a href="#">wk subscript</a>: 要素を下付き文字として配置します。</li> </ul> <p>文字においては、<a href="#">wk top</a>と<a href="#">wk bottom</a>は<a href="#">wk baseline</a>と同じ効果を持ちます。段落においては、<a href="#">wk baseline</a>、<a href="#">wk superscript</a>と<a href="#">wk subscript</a> は<a href="#">wk top</a>と同じ効果を持ちます。</p>
wk web page complete	倍長整数	2	.htm または .html 拡張子。このドキュメントは標準HTMLとして保存され、そのリソースは別に保存されます。4Dタグは除去され、式は値が計算されます。このフォーマットは特に4D Write Pro ドキュメントWeb ブラウザで表示したい場合に特に適しています。
wk web page html 4D	倍長整数	3	4D Write ProドキュメントはHTMLとして保存され、4D独自のタグを含みます。それぞれの式はノンブレイクのスペースとして挿入されます。このフォーマットはロスレスであるため、テキストフィールドへの保存目的に適しています。
wk width	文字列	width	<p>要素の幅を設定します。取り得る値は以下の通りです:</p> <ul style="list-style-type: none"> <li>• <a href="#">wk auto</a> (デフォルト): 幅は、要素の中身に基づいて設定されます。</li> <li>• 定義されたサイズ: サイズは実数値あるいは文字列値を使用して表現されたサイズになります: <ul style="list-style-type: none"> <li>◦ 実数: <a href="#">wk layout unit</a>の単位でのサイズ。</li> <li>◦ 文字列: 値と単位が連結されたCSS文字列。例: 12ptは12ポイントを、1.5cmは1.5センチメートルを意味します。最小値: 0pt、最大値: 10000pt</li> </ul> </li> </ul> <p><a href="#">wk width</a> 属性は、<a href="#">wk min width</a>が定義されていればそれによって上書きされます。</p> <p><b>注:</b> 現行の実装では、<a href="#">wk width</a> はピクチャーに対してのみ使用可能です。</p>
wk word	倍長整数	6	単語のみに下線を引きます(空白スペースには引きません)( <a href="#">wk text linethrough style</a> あるいは <a href="#">wk text underline style</a> 用の値)

## 4D Write Proリファレンス - 新着

16.0

15



-  WP DELETE BOOKMARK New 16.0
-  WP Get bookmark range New 16.0
-  WP GET BOOKMARKS New 16.0
-  WP Get page count New 16.0
-  WP INSERT BREAK New 16.0
-  WP INSERT DOCUMENT New 16.0
-  WP INSERT PICTURE New 16.0
-  WP New Updated 16.0
-  WP PRINT Updated 16.0
-  4D Write Pro エリアを定義する Updated 16.0
-  4D Write Pro エリアを使用する Updated 16.0
-  4D Write Pro ドキュメントに含める式の制限 New 16.0
-  4D Write ドキュメントの読み込み Updated 16.0
-  スタイル付テキストテーマのコマンドの使用 Updated 16.0
-  ドキュメントのコンテンツにプログラムでアクセスする Updated 16.0