Classes

ActiveClass : STRING;	6 ～	アクティブなクラスの名前を返します。
ClassList(index : LONGINT): STRING;	1 ～	指定した索引番号をもつクラスの名前を返します。 例えば、索引番号を4とすると4番目のクラスの名前を返します。
ClassNum:LONGINT;	1 ～	アクティブなドキュメント内のクラスの数を返します。
DelClass(className : STRING);	1 ～	アクティブなドキュメントから指定した名前のクラスを削除します。 削除するクラスにある図形は「一般」クラスに割り当てられます。
FillColorByClass;	8 ～	デフォルトの面の色として、アクティブなクラスの面の色を使います。
FPatByClass;	8 ～	デフォルトの面の模様として、アクティブなクラスの面の模様を使います。
GetClassArrow(className : STRING; VAR style : INTEGER; VAR size : REAL; VAR angle : INTEGER);	10 ～ 12.5	クラスのマーカー設定を返します（この関数は廃止されています。 代わりにGetClassBeginningMarkerまたはGetClassEndMarkerを使用してください。）。
GetClassBeginningMarker(name : STRING; VAR style : LONGINT; VAR angle : INTEGER; VAR size, width : REAL; VAR thicknessBasis : INTEGER; VAR thickness : REAL): BOOLEAN;	2008 ～	指定したクラスの始点マーカーのすべての設定値を返します。
GetClassEndMarker(name : STRING; VAR style : LONGINT; VAR angle : INTEGER; VAR size, width : REAL; VAR thicknessBasis : INTEGER; VAR thickness : REAL): BOOLEAN;	2008 ～	指定したクラスの終点マーカーのすべての設定値を返します。 正常終了するとTRUEが返されます。
GetClassOptions : INTEGER;	8.5 ～	アクティブなドキュメントの、他のクラスの表示方法を値で返します。
GetClFillBack(className : STRING; VAR colorRV, colorGV, colorBV : LONGINT);	8 ～	クラスに設定されている面の地色の成分を返します。 色は赤、緑、青の成分として返されます。 値の範囲は0から65535までです。
GetClFillFore(className : STRING; VAR colorRV, colorGV, colorBV : LONGINT);	8 ～	クラスに設定されている面の色成分を返します。 色は赤、緑、青の成分として返されます。 値の範囲は0から65535までです。
GetClFPat(className : STRING): LONGINT;	8 ～	クラスに設定されている面の模様を返します。 ビットマップは0から71までの正の値で示されています。 ハッチングは負の値で示されています。
GetClLS(className : STRING): INTEGER;	8 ～	クラスに設定されている線の種類を返します。

Classes

GetClLW(className : STRING): INTEGER;	8 ～	クラスに設定されている線の太さを返します。
GetClOpacity(className : STRING): INTEGER;	2008 ～	指定したクラスの不透明度を返します。
GetClPenBack(className : STRING; VAR colorRV, colorGV, colorBV : LONGINT);	8 ～	クラスに設定されている線の地色の成分を返します。 色は赤、緑、青の成分として返されます。 値の範囲は0から65535までです。
GetClPenFore(className : STRING; VAR colorRV, colorGV, colorBV : LONGINT);	8 ～	クラスに設定されている線の色成分を返します。色は赤、緑、青の成分として返されます。値の範囲は0から65535までです。
GetClUseGraphic(className : STRING): BOOLEAN;	8 ～	クラスに設定されているグラフィック属性が有効か無効かを返します。
GetClVectorFill(className : STRING; VAR hatchName : STRING): BOOLEAN;	9 ～	クラスに設定されているハッチングの名前を返します。そのクラスがハッチングを利用している場合はTRUE、利用していない場合はFALSEを返します。
GetCVis(className : STRING): INTEGER;	1 ～	クラスの表示状態を返します。(0:表示／-1:非表示／2:グレイ表示)
GrayClass(className : STRING);	8 ～	指定したクラスをグレイ表示にします。
HideClass(className : STRING);	1 ～	指定したクラスを隠します。
LSByClass;	8 ～	デフォルトの線の種類として、アクティブなクラスの線の種類を使います。
LWByClass;	8 ～	デフォルトの線の太さとして、アクティブなクラスの線の太さを使います。
MarkerByClass;	8 ～	デフォルトのマーカーの種類として、アクティブなクラスのマーカーの種類を使います。
NameClass(className : STRING);	1 ～	新しいクラスを指定した名前で作成し、アクティブにします。指定した名前のクラスが存在する場合は、そのクラスがアクティブになります。
OpacityByClass;	2008 ～	アクティブクラスの不透明度にファイルのデフォルト設定を使用します。
PenColorByClass;	8 ～	デフォルトの線の色として、アクティブなクラスの線の色を使います。
RenameClass(className, newName : STRING);	8.5 ～	クラスの名前を変更します。変更されたクラスに割り当てられていたオブジェクトがすべて更新されます。
SetClassArrow(className : STRING; style : INTEGER; size : REAL; angle : INTEGER);	10 ～ 12.5	クラスのマーカーを設定します。(この関数は廃止されています。代わりに、SetClassBeginningMarkerまたはSetClassEndMarkerを使用してください。)

Classes

GetClassBeginningMarker(name : STRING; style : LONGINT; angle : INTEGER; size, width : REAL; thicknessBasis : INTEGER; thickness : REAL): BOOLEAN;	2008 ～	指定したクラスの始点マーカーのすべての設定値を設定します。 正常終了するとTRUEが返されます。
GetClassEndMarker(name : STRING; style : LONGINT; angle : INTEGER; size, width : REAL; thicknessBasis : INTEGER; thickness : REAL): BOOLEAN;	2008 ～	指定したクラスの終点マーカーのすべての設定値を設定します。 正常終了するとTRUEが返されます。
SetClassOptions(classOpts : INTEGER);	8.5 ～	他のクラスの表示状態を設定します。
SetClFillBack(className : STRING; colorR, colorG, colorB : LONGINT);	8 ～	クラスの面の地色を設定します。 色は赤、緑、青の成分で指定します。 値の範囲は0から65535までです。
SetClFillFore(className : STRING; colorR, colorG, colorB : LONGINT);	8 ～	クラスの面の色を設定します。 色は赤、緑、青の成分で指定します。 値の範囲は0から65535までです。
SetClFPat(className : STRING; fillpattern : LONGINT);	8 ～	クラスの面の模様を設定します。 ビットマップは0から71までの正の値で示されています。 ハッチングは負の値で示されています。
SetClLS(className : STRING; LS : INTEGER);	8 ～	クラスの線の種類を設定します。
SetClLW(className : STRING; LW : INTEGER);	8 ～	クラスの線の太さを設定します。
SetClOpacity(className : STRING; opacity : INTEGER);	2008 ～	指定したクラスの不透明度を設定します。
SetClPenBack(className : STRING; colorR, colorG, colorB : LONGINT);	8 ～	クラスの線の地色を設定します。 色は赤、緑、青の成分で指定します。 値の範囲は0から65535までです。
SetClPenFore(className : STRING; colorR, colorG, colorB : LONGINT);	8 ～	クラスの線の色を設定します。 色は赤、緑、青の成分で指定します。 値の範囲は0から65535までです。

Classes

SetClUseGraphic(className : STRING; use : BOOLEAN);	8 ～	クラスのグラフィック属性を設定します。
SetClUseTexture(className : STRING; use : BOOLEAN);	8 ～	クラスのテクスチャ使用の有無を設定します。
SetClVectorFill(className, hatchName : STRING): BOOLEAN;	9 ～	クラスのハッチングを設定します。
ShowClass(className : STRING);	1 ～	指定したクラスを表示します。

Command		
Absolute;	1 ～	座標指定を絶対指定にします。
AcquireExportPDFSettingsAndLocation(inbSeparateDocuments : BOOLEAN): BOOLEAN;	12.5 ～	「PDF取り出し」の設定とPDFファイル名（TRUE=フォルダ名）を尋ねます。 PDFの一括取り出しに対応するための操作です。
AngleVar;	1 ～	変数を極座標として角度指定に使用することを宣言します。
CallTool(toolID : INTEGER);	4 ～	指定した番号のツールを選択します。 ツールが使われた後は、この手続きが実行される前に選択されていたツールを選択しま
ClosePDFDocument;	12.5 ～	OpenPDFDocumentで開始されたPDFドキュメントの作成プロセスが完了します。 これは、PDFの一括取り出しに対応するための操作です。
DoMenuText(menuItem : STRING);	4 ～	この関数はVectorworks上では使えません。
DoMenuTextByName(subMenu : STRING; index : INTEGER);	5 ～	Vectorworksのメニューコマンドを実行します。 実行するメニューがチャンクの一部の場合はチャンクメニュー番号パラメータで指定しま す。
ExportPDFPages(savedViewNameStr : STRING): INTEGER;	12.5 ～	現在のドキュメントがPDFとして取り出されます。 この関数を使用する前にOpenPDFDocumentを使用する必要があります。 これは、PDFの一括取り出しに対応するための操作です。
Move(moveDX, moveDY : REAL);	1 ～	現在のペン位置を相対的に移動します。
MoveTo(pX, pY : REAL);	1 ～	ペン位置を絶対座標の位置に移動します。
NoAngleVar;	1 ～	AngleVarの宣言を解除します。
OpenPDFDocument(inFilenameStr : STRING): BOOLEAN;	12.5 ～	PDFドキュメントへの取り出しを開始します。 この関数を使用する前にAcquireExportPDFSettingsAndLocationを使用する必要があり ます。これは、PDFの一括取り出しに対応するための操作です。
PenLoc(VAR pX, pY : REAL);	1 ～	現在のペンの座標位置を返します。
PopAttrs;	4 ～	記憶されている属性を現在の設定にします。
PrintUsingPrintDialog : INTEGER;	10.5 ～	アクティブなドキュメントをプリントします。プリントダイアログが表示されます。 用紙設定が正しくない場合は、プリントダイアログの前に用紙設定ダイアログが表示され
PrintWithoutUsingPrintDialog : INTEGER;	10.5 ～	アクティブなドキュメントをプリントします。 プリントダイアログ、用紙設定ダイアログのどちらも表示しません。
PushAttrs;	4 ～	現在の属性の設定を記憶します。
Relative;	1 ～	座標指定を相対指定にします。
Run(p : PROCEDURE);	1 ～	VectorScript プログラムを実行します。
SetTool(theTool : INTEGER);	1 ～	ツールパレットからツールを選択します。 ツールが使われた後も、そのツールを選択したままにします。

Criteria

Angle(c : CRITERIA): REAL;	1 ～	線分や円弧の角度を返します。検索条件に合致した図形が複数ある場合は合計を返します。
Area(c : CRITERIA): REAL;	1 ～ 12.0.1	検索条件に合致した図形の面積を返します。 この関数は古いものです。CriteriaAreaを使用してください。
BotBound(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の下辺のY座標を返します。 検索条件に合致した図形が複数ある場合は、最後に合致した図形の値を返します。
ComponentArea(c : CRITERIA; index : INTEGER): REAL;	2011 ～	指定した構成要素の片面の面積を返します。 3D 図形の開口を考慮 (差し引き) します。
ComponentVolume(c : CRITERIA; index : INTEGER): REAL;	2011 ～	指定した構成要素の3D 容積を返します。 3D 図形の開口を考慮 (差し引き) します。
Count(c : CRITERIA): LONGINT;	1 ～	検索条件に合致した図形の数を返します。
CriteriaArea(c : CRITERIA): REAL;	12 ～	検索条件に合致した図形の面積を返します。 値は現在の面積の単位になります。
CriteriaSurfaceArea(c : CRITERIA): REAL;	12 ～	検索条件に合致したソリッドの表面積を返します。 値は現在の面積の単位になります。
CriteriaVolume(c : CRITERIA): REAL;	12 ～	検索条件に合致した図形の体積を返します。 値は現在の体積の単位になります。
DSelectObj(c : CRITERIA);	1 ～	検索条件に合致したすべての図形の選択を解除します。
EditProperties(c : CRITERIA);	12.5 ～	指定した検索条件に一致するすべての図形の「プロパティ」ダイアログを表示します。
Eval(h : HANDLE; c : CRITERIA): REAL;	1 ～	指定した図形の検索条件に合致したデータを実数で返します。 検索条件がレコードの場合、図形に連結されている/いないを返します。 レコードフィールドの場合、値を実数で返します。
EvalStr(h : HANDLE; c : CRITERIA): STRING;	1 ～	指定した図形の検索条件に合致したデータを文字列で返します。 検索条件にレコードを使用した場合、指定したレコードが図形に連結されている/いないを返します。 レコードフィールドを使用した場合、値を文字列で返します。
ForEachObject(callback : PROCEDURE; c : CRITERIA);	1 ～	検索条件に合致した図形を、指定した手続き型サブルーチンで処理します。 手続き型サブルーチンには、検索条件に合致した図形のハンドルを渡すための、ハンドル型のパラメータがなければなりません。

Criteria

Height(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の高さを返します。 検索条件に合致した図形が複数ある場合は合計を返します。
Hide(c : CRITERIA);	1 ～	検索条件に合致した図形を隠します。
IsFlipped(c : CRITERIA): REAL;	8 ～	反転されている図形 (壁の中のシンボルなど) で検索条件に合致したものの数を返します。
LeftBound(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の左辺のX座標を返します。 検索条件に合致した図形が複数ある場合は、最後に合致した図形の値を返します。
Length(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の長さを返します。 検索条件に合致した図形が複数ある場合は合計を返します。
ObjectType(c : CRITERIA): INTEGER;	1 ～	検索条件に合致した図形の種類を返します。 検索条件に合致した図形が複数ある場合は、最後に合致した図形の値を返します。
Perim(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の周長を返します。 検索条件に合致した図形が複数ある場合は合計を返します。
RightBound(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の右辺のX座標を返します。 検索条件に合致した図形が複数ある場合は、最後に合致した図形の値を返します。
RoofArea_Heated(c : CRITERIA): REAL;	2009 ～	検索条件に合致した屋根、屋根面の勾配に沿った暖房領域の面積を返します。 暖房領域は軒の出を含まない領域です。
RoofArea_HeatedProj(c : CRITERIA): REAL;	2009 ～	検索条件に合致した屋根、屋根面の暖房領域の水平投影面積を返します。 暖房領域は軒の出を含まない領域です。
RoofArea_Total(c : CRITERIA): REAL;	2009 ～	検索条件に合致した屋根、屋根面の勾配に沿った総面積を返します。
RoofArea_TotalProj(c : CRITERIA): REAL;	2009 ～	検索条件に合致した屋根、屋根面の水平投影面積を返します。
SelectObj(c : CRITERIA);	1 ～	検索条件に合致したすべての図形を選択します。
Show(c : CRITERIA);	1 ～	隠されているまたは、グレイ表示されている図形で検索条件に合致したものを表示します。
SlabThickness(c : CRITERIA): REAL;	12 ～	検索条件に合致したスラブ (屋根面、床、柱) の厚さを返します。
SurfaceArea(c : CRITERIA): REAL;	8 ～ 12.0.1	検索条件に合致したソリッドの表面積を返します。この関数は古いものです。代わりにCriteriaSurfaceAreaを使用してください。

Criteria

TopBound(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の上辺のY座標を返します。 検索条件に合致した図形が複数ある場合は合計を返します。
Volume(c : CRITERIA): REAL;	8 ～ 12.0.1	検索条件に合致したソリッドの体積を返します。 この関数は古いものです。代わりにCriteriaVolumeを使用してください。
WallArea_Gross(c : CRITERIA): REAL;	12 ～	検索条件に合致した壁の2D表面積 (グロス) を返します。
WallArea_Net(c : CRITERIA): REAL;	12 ～	検索条件に合致した壁の2D表面積 (グロス) からドアと窓の面積を除いた値を返します。
WallAverageHeight(c : CRITERIA): REAL;	12 ～	検索条件に合致した壁の頂点、開始位置、終了位置の高さを含めて計算した平均の高さを返します。
WallThickness(c : CRITERIA): REAL;	12 ～	検索条件に合致した壁の厚さを返します。
Width(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の幅を返します。 検索条件に合致した図形が複数ある場合は合計を返します。
XCenter(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の中心のX座標を返します。 検索条件に合致した図形が複数ある場合は合計を返します。
XCoordinate(c : CRITERIA): REAL;	2011 ～	ユーザ原点に基づく図形の X 座標を返します。
YCenter(c : CRITERIA): REAL;	1 ～	検索条件に合致した図形の中心のY座標を返します。 検索条件に合致した図形が複数ある場合は合計を返します。
YCoordinate(c : CRITERIA): REAL;	2011 ～	ユーザ原点に基づく図形の Y 座標を返します。
ZCenter(c : CRITERIA): REAL;	2009 ～	検索条件に合致した図形の中心のZ座標を返します。 検索条件に合致した図形が複数ある場合は合計を返します。
ZCoordinate(c : CRITERIA): REAL;	2011 ～	ユーザ原点に基づく図形の Z 座標を返します。

Database / Record

DelRecord(h : HANDLE; name : STRING);	7 ～	ハンドルで指定した図形に連結されているレコードを削除します。
Field(h : HANDLE; s1, s2, s3 : STRING);	1 ～	この関数はVectorworks上では使えません。
GetFldName(h : HANDLE; index : INTEGER): STRING;	1 ～	レコードフィールドの名前を返します。
GetFldType(h : HANDLE; t : INTEGER): INTEGER;	1 ～	レコードフィールドの形式を返します。
GetParametricRecord(h : HANDLE): HANDLE;	2011 ～	参照された図形に添付されているパラメトリックレコードのハンドルを返します。 パラメトリックレコードはパラメトリック図形のパラメータを保持する隠しレコードです。 パラメトリック図形だけがパラメトリックレコードを持ちます。
GetRecord(h : HANDLE; cnt : INTEGER): HANDLE;	1 ～	ハンドルで指定した図形に連結されているレコードのハンドルを返します。
GetRField(h : HANDLE; record, field : STRING): DYNARRAY[] of CHAR;	8.5 ～	ハンドルで指定した図形に連結されたレコードフィールドの値を文字列で返します。
LinkText(h : HANDLE; rec, fld : STRING);	1 ～	ハンドルで指定した文字図形にレコードとレコードフィールドを連結させます。 この機能は「文字をレコードに連結」メニューに相当し、シンボル内の文字の制御に使います。
NewField(recName, fieldName : STRING; fieldValue : DYNARRAY[] of CHAR; fType, fFlag : INTEGER);	1 ～	レコードに新しいレコードフィールドを追加します。 指定した名前のレコードが存在しなければ、新しいレコードを作成します。
NumFields(h : HANDLE): INTEGER;	1 ～	レコードフィールドの数を返します。
NumRecords(h : HANDLE): INTEGER;	1 ～	ハンドルで指定した図形に連結されているレコードの数を返します。
Record(h : HANDLE; s : STRING);	1 ～	ハンドルで指定した図形にレコードを連結し直します。
SetRecord(h : HANDLE; record : STRING);	1 ～	ハンドルで指定した図形にレコードを連結します。

Database / Record

SetRField(h : HANDLE; record, field : STRING; value : DYNARRAY[] of CHAR);	1 ～	ハンドルで指定した図形に連結しているレコードフィールドに値を設定します。

Dialogs - Classic

AddButton(buttonStr : STRING; itemID, buttonType, x1, y1, x2, y2 : INTEGER);	1 ～ 2009	カスタムダイアログにボタンを作成します。 ボタンの種類(1:プッシュボタン／2:チェックボックス／3:ラジオボタン)
AddChoiceItem(choiceTitle : STRING; itemID, choiceType, x1, y1, x2, y2 : INTEGER);	7 ～ 2009	カスタムダイアログににリストアイテムを作成します。 リストアイテムの種類(1:ポップアップ／2:リストボックス(複数選択可)／3:リストボックス(複数選択不可))
AddField(fieldStr : STRING; itemID, fieldType, x1, y1, x2, y2 : INTEGER);	1 ～ 2009	カスタムダイアログに編集可能／不可能なテキストフィールドを作成します。 フィールドの種類(1:編集不可能なフィールド／2:編集可能なフィールド)
AddGroupBox(s : STRING; item, left, top, right, bottom : INTEGER);	8 ～ 2009	カスタムダイアログにグループボックスを作成します。
AddHelpItem(itemID, x1, y1, x2, y2 : INTEGER);	8.5 ～ 2009	カスタムダイアログにヘルプフィールドを作成します。
BeginDialog(dialogID, dialogType, x1, y1, x2, y2 : INTEGER);	1 ～ 2009	カスタムダイアログの定義を開始します。 カスタムダイアログの構成はBeginDialogとEndDialogの間に記述します。
ClrDialog;	1 ～ 2009	一番上に表示されているダイアログを消去します。
DelChoice(item, whichChoice : INTEGER);	8 ～ 2009	リストアイテムにある項目を削除します。
DialogEvent(VAR item : INTEGER);	1 ～ 2009	ユーザーが動作を起こすまで待機し、クリックされたアイテムの番号を返します。
DrawDialog;	1 ～ 2009	一番上に表示されているダイアログを再描画します。
EndDialog;	1 ～ 2009	カスタムダイアログの定義を終了します。
GetDialog(dialogID : INTEGER);	1 ～ 2009	指定した番号のダイアログを表示します。
SetTitle(theTitle : STRING);	6.0.2 ～ 2009	カスタムダイアログのタイトルを設定します。

Dialogs - Handler

GetChoiceStr(item, whichChoice : INTEGER; VAR s : STRING);	8 ～ 2009	リストアイテムにある項目の名前を返します。
GetField(fieldID : INTEGER): DYNARRAY[] of CHAR;	1 ～ 2009	テキストフィールドに入力された文字列を返します。
GetSelChoice(item, atChoice : INTEGER; VAR choiceNumber : INTEGER; VAR choiceString : STRING);	8 ～ 2009	リストアイテムで選択されている項目の番号と名前を返します。 atChoiceは現在使用していません。0を入力しておいてください。
InsertChoice(item, before : INTEGER; choice : STRING);	8 ～ 2009	リストアイテムに新しい項目を追加します。
ItemSel(fieldID : INTEGER): BOOLEAN;	1 ～ 2009	チェックボックスまたはラジオボタンの選択状態を返します。
NumChoices(item : INTEGER): INTEGER;	8 ～ 2009	リストアイテムにある項目の数を返します。
SelChoice(item, choice : INTEGER; select : BOOLEAN);	8 ～ 2009	リストアイテムにある項目の選択状態を設定します。
SelField(fieldID : INTEGER);	1 ～ 2009	フィールドの文字列をアクティブにして強調表示します。
SetField(fieldID : INTEGER; str : DYNARRAY[] of CHAR);	1 ～ 2009	テキストフィールドに表示する文字列を設定します。 編集可能なテキストフィールド／編集不可能なテキストの両方で利用できます。
SetHelpString(itemID : INTEGER; helpString : STRING);	8.5 ～ 2009	ヘルプフィールドに表示する文字列を設定します。
SetItem(fieldID : INTEGER; select : BOOLEAN);	1 ～ 2009	チェックボックスまたはラジオボタンの選択状態を設定します。
SetItemEnable(item : INTEGER; enable : BOOLEAN);	8 ～ 2009	アイテムの編集可／不可を設定します。 編集不可に設定した場合、アイテムはグレイ表示され、ユーザは選択できません。
SetTextEditable(item : INTEGER; editable : BOOLEAN);	11.5 ～ 2009	テキストフィールドの状態を設定します。

Dialogs - Modern

AddChoice(dialogID, componentID : LONGINT; choiceText : STRING; itemIndex : INTEGER);	2010 ～	プルダウンメニューやリストアイテムに項目を追加します。
AddListBoxTabStop(dialogID, itemID : LONGINT; tabStop : INTEGER);	10 ～	リストボックスにタブストップを設定します。
AdjustComponentPixelPos(nDialogID, nComponentID : LONGINT; nHorizontalPixels, nVerticalPixels : INTEGER): BOOLEAN;	2008 ～	指定したレイアウトダイアログのコンポーネントのピクセル幅とピクセル高さを調整します。
AlignItemEdge(dialogID, itemID, whichEdge : LONGINT; alignID, alignMode : INTEGER);	9 ～	ダイアログ上の指定したアイテムの位置を揃えます。
ClearGradientSliderSegments(dialogID, componentID : LONGINT);	10 ～	グラデーションスライダーからすべての変化位置を消去します。
CreateCenteredStaticText(dialogID, controlID : LONGINT; text : STRING; widthInCharacters : INTEGER);	12.0.1 ～	CreateStaticTextのように、編集不可能なフィールドを作成します。 文字列はダイアログのフィールドの中心に作成されます。
CreateCheckBox(dialogID, itemID : LONGINT; text : STRING);	9 ～	チェックボックスを作成します。
CreateCheckBoxGroupBox(dialogID, itemID : LONGINT; name : STRING; hasFrame : BOOLEAN);	10.5 ～	チェックボックスのグループボックスを作成します。 チェックボックスのグループボックスにはラベルをつけることができます。 hasFrameがTRUEならば、枠線が表示されます。
CreateClassPullDownMenu(nDialogID, nComponentID : LONGINT; nWidthInChars : INTEGER);	2008 ～	クラスのプルダウンメニューを作成します。
CreateColorPopup(dialogID, itemID, widthInCharacters : LONGINT);	12 ～	256色のカラーパレットを表示する色ポップアップを作成します。
CreateControl(dialogID, itemID, controlKind : LONGINT; name : STRING; data : LONGINT);	9 ～	イメージ、カラーパレット、スライダーなどのアイテムを作成します。
CreateCustomControl(dialogID, componentID : LONGINT; iWidth, iHeight : INTEGER);	2008 ～	レイアウトコントロールをVectorScriptに作成し、外部ダイアログハンドラの GS_OverrideControlと組み合わせて使用します。
CreateDesignLayerPullDownMenu(nDialogID, nComponentID : LONGINT; nWidthInChars : INTEGER);	2008 ～	デザインレイヤのプルダウンメニューを作成します。
CreateEditInteger(dialogID, itemID, defaultValue, widthInCharacters : LONGINT);	9 ～	整数型の編集フィールドを作成します。

Dialogs - Modern

CreateEditReal(dialogID, itemID, editRealType : LONGINT; defaultValue : REAL; widthInCharacters : LONGINT);	9 ～	実数型の編集フィールドを作成します。
CreateEditText(dialogID, itemID : LONGINT; defaultText : STRING; widthInCharacters : LONGINT);	9 ～	文字列型の編集フィールドを作成します。
CreateEditTextBox(dialogID, itemID : LONGINT; defaultText : STRING; widthInCharacters, heightInLines : LONGINT);	10 ～	スクロールすることで複数行を編集可能なフィールドを作成します。
CreateEnhancedPullDownMenu(dialogID, componentID : LONGINT; iWidthInCharacters : INTEGER; bShowIconInMainWindow : BOOLEAN);	12.5 ～	拡張プルダウンメニューを作成します。
CreateGradient(name : STRING): HANDLE;	10 ～	グラデーションを新規に作成します。
CreateGroupBox(dialogID, itemID : LONGINT; text : STRING; hasFrame : BOOLEAN);	9 ～	グループボックスを作成します。 ボックスの幅はボックス内の最も長いアイテムの長さによって決定され、高さはボックス内のアイテムの高さの合計によって決定されます。
CreateIconPushButton(nDialogID, nComponentID : LONGINT; nIconID, nWidthInChars : INTEGER);	12.5 ～	アイコンプッシュボタンを作成し、アイコンIDと幅（文字数）を指定します。
CreateImageControl(dialogID, componentID : LONGINT; iWidthPixels, iHeightPixels : INTEGER; hImage : HANDLE);	12 ～	イメージを作成します。
CreateLayout(dialogTitle : STRING; hasHelp : BOOLEAN; defaultButtonName, cancelButtonName : STRING): LONGINT;	9 ～	ダイアログのレイアウトを作成し、ダイアログ番号を返します。
CreateLineAttributePopup(dialogID, itemID : LONGINT);	12 ～	線種と線の太さの選択肢を表示するポップアップを作成します。
CreateLineStylePopup(dialogID, itemID : LONGINT);	12 ～	線種の選択肢を表示するポップアップを作成します。
CreateLineWeightPopup(dialogID, itemID : LONGINT);	12 ～	線の太さの選択肢を表示するポップアップを作成します。
CreateListBox(dialogID, itemID, widthInCharacters, heightInCharacters : LONGINT);	9 ～	リストボックスを作成します。

Dialogs - Modern

CreateListBoxN(dialogID, itemID, widthInCharacters, heightInCharacters : LONGINT; isMultipleSelect : BOOLEAN);	10.5 ～	リストボックスを作成します。 isMultipleSelectパラメータをTRUEにすると、リストボックス内の項目を複数選択することができます。
CreateMarkerPopup(dialogID, componentID : LONGINT);	10.5 ～	マーカーポップアップを作成します。
CreatePatternPopup(dialogID, itemID : LONGINT);	12 ～	模様ポップアップを作成します。
CreatePullDownMenu(dialogID, itemID, widthInCharacters : LONGINT);	9 ～	プルダウンメニューを作成します。
CreatePullDownMenuGroupBox(liDialogID, liComponentID : LONGINT; iPullDownWidth : INTEGER; strLabel : STRING; bHasFrame : BOOLEAN);	12.5 ～	プルダウンメニューのグループボックスを作成します。
CreatePushButton(dialogID, itemID : LONGINT; text : STRING);	9 ～	プッシュボタンを作成します。
CreateRadioButton(dialogID, itemID : LONGINT; text : STRING);	9 ～	ラジオボタンを作成します。
CreateRadioButtonGroupBox(dialogID, itemID : LONGINT; name : STRING; hasFrame : BOOLEAN);	10.5 ～	ラジオボタンのグループボックスを作成します。 ラジオボタンのグループボックスにはラベルをつけることができます。 hasFrameがTRUEならば、枠線が表示されます。
CreateResizableLayout(dialogTitle : STRING; hasHelp : BOOLEAN; defaultButtonName, cancelButtonName : STRING; widthResizable, heightResizable : BOOLEAN): LONGINT;	12 ～	サイズ変更が可能なダイアログのレイアウトを作成し、ダイアログ番号を返します。
CreateRightStaticText(dialogID, itemID : LONGINT; text : STRING; widthInCharacters : INTEGER);	12.0.1 ～	CreateStaticTextのように、編集不可能なフィールドを作成します。 文字列はダイアログのフィールドの右に作成されます。
CreateSeparator(dialogID, componentID : LONGINT; iLength : INTEGER);	12.5 ～	セパレータを作成します。
CreateSheetLayerPullDownMenu(nDialogID, nComponentID : LONGINT; nWidthInChars : INTEGER);	2008 ～	シートレイヤのプルダウンメニューを作成します。
CreateStandardIconControl(dialogID, iconControlID : LONGINT; iconNumber : INTEGER);	11.5 ～	アプリケーションアイコンや警告アイコンで表示される標準アイコンアイテム作成します。

Dialogs - Modern

CreateStaticText(dialogID, itemID : LONGINT; text : STRING; widthInCharacters : LONGINT);	9 ～	編集不可能なフィールドを作成します。文字列の長さで自動的に幅を変更するには、幅のパラメータを-1にします。
CreateSwapControl(dialogID, swapControlID : LONGINT);	11.5 ～	スワップペインコントロールを作成します。 複数の重なりあったコントロールのグループを、同時に表示される1つのコントロールのグループとして扱えます。
CreateSwapPane(dialogID, swapControlID, newGroupID : LONGINT);	11.5 ～	スワップペインコントロールの中にスワップペインを作成します。 スワップコントロール内では、一度に1つのスワップペインだけが表示されます。
CreateSymbolDisplayControl(dialogID, itemID : LONGINT; symbolName : STRING; height, width, margin, renderMode, view : INTEGER);	12 ～	シンボル表示コントロールを作成します。
CreateTabControl(dialogID, itemID : LONGINT);	10.5 ～	タブペインコントロールを作成します。 タブペインコントロールは、情報を複数のタブペインに表示し、タブボタンでタブペイン間の切り替えができるようにします。
CreateTabPane(dialogID, itemID, groupID : LONGINT);	10.5 ～	タブペインコントロールの中にタブペインを作成します。
CreateThreeStateCheckBox(dialogID, componentID : LONGINT; strName : STRING);	12.5 ～	3状態チェックボックスを作成します。
CreateTreeControl(nDialogID, nComponentID : LONGINT; nWidthInChars, nHeightInChars : INTEGER);	2008 ～	ツリーアイテムを作成します。
DeleteAllItems(dialogID, itemID : LONGINT);	10 ～	リストボックスからすべての行を削除します。
DeregisterDialogFromTimerEvents(dialogID : LONGINT);	2010 ～	タイマーイベントレジストリからダイアログを削除します。
DeselectEditText(dialogID, controlID : LONGINT);	12.0.1 ～	全ての文字列型の編集フィールドの選択状態を解除します。
DisplaySwapPane(dialogID, swapControlID, groupNumber : LONGINT);	11.5 ～	スワップコントロール内に指定したスワップペインを表示します。
DisplayTabPane(dialogID, tabControlID, groupNumber : LONGINT);	12 ～	タブコントロール内に指定したタブパネルを表示します。
EnableItem(dialogID, componentID : LONGINT; enableState : BOOLEAN);	2010 ～	アイテムの編集可／不可を設定します。 編集不可に設定した場合、アイテムはグレイ表示され、ユーザは選択できません。
EnableLBDropOnIndices(dialogID, componentID : LONGINT; iStartIndex, iEndIndex : INTEGER; bEnable : BOOLEAN): BOOLEAN;	12.5 ～	指定した番号内でのドラッグ＆ドロップ操作を有効または無効にします。
EnableTextEdit(dialogID, componentID : LONGINT; editableState : BOOLEAN);	2010 ～	テキストフィールドの編集可／不可を設定します。

Dialogs - Modern

ExpandTreeControlItem(nDialogID, nComponentID : LONGINT; nItemID : INTEGER; bExpand : BOOLEAN);	12.5 ～	指定したツリーアイテムを開くまたは折りたたみます。
GetActiveEditItem(dialogID : LONGINT): LONGINT;	12.0.1 ～	アクティブな編集フィールドのIDを返します。 アクティブな編集フィールドがなければ、-1を返します。
GetActivePane(dialogID, tabControlID : LONGINT): LONGINT;	12 ～	タブまたはスワップコントロールに現在表示されているタブまたはスワップパネルを返します。
GetBooleanItem(dialogID, componentID : LONGINT; VAR outState : BOOLEAN);	2010 ～	チェックボックスまたはラジオボタンの選択状態を返します。
GetChoiceCount(dialogID, componentID : LONGINT; VAR outCount : INTEGER);	2010 ～	プルダウンメニューやリストアイテムの項目数を返します。
GetChoiceText(dialogID, componentID : LONGINT; itemIndex : INTEGER; VAR itemText : STRING);	2010 ～	インデックスを使用して、プルダウンメニューやリストアイテムのテキストを返します。
GetColorButton(dialogID, itemID : LONGINT; VAR red, green, blue : LONGINT);	10 ～	カラーボタンの色を返します。
GetColorChoice(dialogID, itemID : LONGINT; VAR colorIndex : INTEGER);	12 ～	選択されている色ポップアップを返します。
GetComponentRect(nDialogID, nComponentID : LONGINT; VAR nLeft, nTop, nRight, nBottom : INTEGER): BOOLEAN;	2008 ～	指定したレイアウトダイアログのコンポーネントの四角形の領域の座標を返します。
GetComponentTextWidth(nDialogID, nComponentID : LONGINT; VAR nWidthInLMUnits : INTEGER): BOOLEAN;	2008 ～	指定したレイアウトダイアログの単位でテキストフィールドの幅を返します。
GetControlData(dialogID, itemID : LONGINT; VAR data : LONGINT);	8.5 ～	指定したアイテムのデータを返します。
GetEditInteger(dialogID, itemID : LONGINT; VAR value : LONGINT): BOOLEAN;	9 ～	整数型の編集フィールドに入力されている整数値を返します。
GetEditReal(dialogID, itemID, editRealType : LONGINT; VAR value : REAL): BOOLEAN;	9 ～	実数型の編集フィールドに入力されている実数値を返します。

Dialogs - Modern

GetGradientSliderData(dialogID, componentID : LONGINT; segmentIndex : INTEGER; VAR spotPosition, midpointPosition : REAL; VAR red, green, blue : LONGINT);	10 ～	グラデーションスライダーにある変化位置の変化点、変化の中心点の位置と色を返します。
GetGradientSliderSelectedMarker(dialogID, componentID : LONGINT; VAR segmentIndex, markerType : INTEGER);	10 ～	グラデーションスライダーの選択されている点を返します。
GetIconPushButtonState(nDialogID, nComponentID : LONGINT; VAR bPressed : BOOLEAN): BOOLEAN;	2008 ～	指定したアイコンプッシュボタンの状態（押された、押されていないなど）を設定します。
GetImagePopupObject(dialogID, componentID : LONGINT; itemIndex : INTEGER): STRING;	10 ～	イメージポップアップメニューのメニュー項目番号から図形の名前を返します。
GetImagePopupObjectItemIndex(dialogID, componentID : LONGINT; objectName : STRING): INTEGER;	10 ～	イメージポップアップメニュー内の図形の名前からメニュー項目番号を返します。
GetImagePopupSelectedItem(dialogID, componentID : LONGINT): INTEGER;	10 ～	イメージポップアップメニューの選択されているメニュー項目番号を返します。
GetItemText(dialogID, componentID : LONGINT; VAR text : STRING);	2010 ～	テキストフィールドに入力された文字列を返します。（255文字以下）
GetLayoutDialogPosition(dialogID : LONGINT; VAR left, top, right, bottom : INTEGER): BOOLEAN;	11 ～	ダイアログウインドウの位置を検索し、ピクセルで返します。
GetLayoutDialogSize(dialogID : LONGINT; VAR width, height : INTEGER);	12 ～	レイアウトダイアログのサイズをピクセル単位で取得します。
GetLBHeaderTextWidth(className : STRING; allowForSortIcon : BOOLEAN): INTEGER;	2009 ～	リストブラウザの列ヘッダーで文字列が短縮されずに表示される幅をピクセルで返します。InsertLBColumnで幅を適切に設定する際に利用します。
GetLineAttributeData(dialogID, itemID : LONGINT; VAR lineStyle, lineWeight : INTEGER);	12 ～	線種と線の太さポップアップで現在選択されている項目を返します。
GetLineStyleChoice(dialogID, itemID : LONGINT; VAR lineStyle : INTEGER);	12 ～	線種ポップアップで現在選択されている項目を返します。

Dialogs - Modern

GetLineWeightChoice(dialogID, itemID : LONGINT; VAR lineWeight : INTEGER);	12 ～	線の太さポップアップで現在選択されている項目を返します。
GetMarkerChoice(dialogID, itemID : LONGINT; VAR index, style, angle : INTEGER; VAR size : REAL);	12 ～ 12.5	選択されているマーカースタイルを返します。(VW2008ではこの関数は廃止されています。GetMarkerValueを使用してください。)
GetMarkerPopupSelectedItem(dialogID, componentID : LONGINT; VAR style, angle, size : INTEGER): INTEGER;	10.5 ～	選択されているマーカースタイルを番号 (1から始まる) で返します。カスタムが選択されている場合は8を返します。
GetMarkerValue(dialogID, itemID : LONGINT; VAR style, angle : INTEGER; VAR length, width : REAL; VAR basis : INTEGER; VAR thickness : REAL);	2008 ～	マーカーポップアップの値を返します (VW2008より前のバージョンのマーカーポップアップ手続きの代わりになります。)。
GetMultilineText(dialogID, componentID : LONGINT; VAR text : DYNARRAY[] of CHAR);	2010 ～	テキストフィールドに入力された文字列を返します。
GetNumGradientSliderSegments(dialogID, componentID : LONGINT): INTEGER;	10 ～	グラデーションスライダーの変化位置の数を返します。
GetNumImagePopupItems(dialogID, componentID : LONGINT): INTEGER;	10 ～	イメージポップアップメニュー内のメニュー項目数を返します。
GetPatternData(dialogID, itemID : LONGINT; VAR patternIndex, foreColor, backColor : INTEGER);	12 ～	模様ポップアップで現在選択されている項目を返します。
GetPopUpChoiceIndex(dialogID : LONGINT; componentID : LONGINT; itemText : STRING; VAR itemIndex : INTEGER);	2011 ～	ポップアップの中から文字列を検索し一致するインデックス (0基準) を取得します。 存在しない場合、-1を返します。
GetSelectedChoiceIndex(dialogID, componentID : LONGINT; startIndex : INTEGER; VAR outSelectedIndex : INTEGER);	2010 ～	選択されている項目の番号を返します。
GetSelectedChoiceInfo(dialogID, componentID : LONGINT; startIndex : INTEGER; VAR outSelectedIndex : INTEGER; VAR outSelectedChoiceText : STRING);	2010 ～	選択されている項目の番号と名前を返します。

Dialogs - Modern

GetSelectionRange(dialogID, controlID : LONGINT; VAR startPos, endPos : INTEGER);	12.0.1 ～	指定したコントロール内の選択範囲の始点と終点を返します。
GetThreeStateCheckBoxState(dialogID, componentID : LONGINT; VAR iState : INTEGER);	12.5 ～	3状態チェックボックスの状態を返します。
GetTreeControlItemText(nDialogID : LONGINT; nComponentID : LONGINT; nItemID : INTEGER; VAR itemText : STRING): BOOLEAN;	2011 ～	ツリーコントロールから指定した項目のテキストを返します。
GetTreeControlSelectedItemText(nDialogID : LONGINT; nComponentID : LONGINT; VAR itemText : STRING): BOOLEAN;	2011 ～	ツリーコントロールから選択した項目のテキストを返します。
GetTreeControlItemData(nDialogID, nComponentID : LONGINT; nItemID : INTEGER; VAR nUserData : LONGINT);	12.5 ～	指定したツリーアイテムのユーザデータを返します。
GetTreeControlSelectedItem(nDialogID, nComponentID : LONGINT; VAR nItemID : INTEGER): BOOLEAN;	2008 ～	選択されているツリーアイテムの項目番号を返します。
InsertEnhancedPullDownMenuItem(dialogID, componentID : LONGINT; strName : STRING; iIconID : INTEGER): INTEGER;	12.5 ～	指定した拡張プルダウンメニュ に項目を挿入します。
InsertGradientSliderSegment(dialogID, componentID : LONGINT; spotPosition : REAL; red, green, blue : LONGINT): INTEGER;	10 ～	グラデーションスライダーに新しい変化位置を挿入し、指定した値でそのデータを初期化します。
InsertImagePopupObjectItem(dialogID, componentID : LONGINT; objectName : STRING): INTEGER;	10 ～	図形をイメージポップアップメニューに挿入します。
InsertImagePopupResource(dialogID, componentID, listID, index : LONGINT): LONGINT;	12 ～	リソースファイルから指定された項目を、イメージポップアップメニューに挿入します。挿入された項目のイメージポップアップのインデックスを返します。

Dialogs - Modern

InsertImagePopupSeparator(liDialogID, liComponentID : LONGINT; strLabel : STRING): INTEGER;	12 ～	イメージポップアップメニューリストの末尾に指定したラベルがある状態で、セパレータを挿入します。
InsertProposedClassOrLayerItem(nDialogID, nComponentID : LONGINT; strLabel : STRING; nIconIndex : INTEGER): BOOLEAN;	2008 ～	クラス項目またはレイヤ項目を、クラス、デザインレイヤ、またはシートレイヤのプルダウンメニューの任意の場所に挿入します。
InsertTreeControlItem(nDialogID, nComponentID : LONGINT; strItemLabel : STRING; nParentID, nAfterID : INTEGER): INTEGER;	2008 ～	ツリーアイテムに項目を挿入します。
IsItemEnabled(nDialogID, nComponentID : LONGINT): BOOLEAN;	12.5 ～	指定したアイテムの有効／無効を返します。
IsItemVisible(nDialogID, nComponentID : LONGINT): BOOLEAN;	12.5 ～	指定したアイテムの表示／非表示を返します。
NotifyPullDownClicked(nDialogID, nComponentID : LONGINT);	2008 ～	プルダウンメニューがクリックされたことを返します。動的メニューを生成することができます。
RefreshItem(liDialogID, liComponentID : LONGINT);	12.5 ～	指定した項目を更新します。
RegisterDialogForTimerEvents(dialogID, timerDelayInMilliseconds : LONGINT);	2010 ～	ダイアログを登録して一定時間ごとにイベントを受け取れるようにします。
RemoveAllImagePopupItems(dialogID, componentID : LONGINT);	10 ～	イメージポップアップメニューからすべてのメニュー項目を削除します。
RemoveChoice(dialogID, componentID : LONGINT; itemIndex : INTEGER);	2010 ～	プルダウンメニューやリストアイテムにある項目を削除します。
RemoveEnhancedPullDownMenuItemRange(dialogID, componentID : LONGINT; iStartItemIndexToRemove, iEndItemIndexToRemove : INTEGER);	12.5 ～	拡張プルダウンメニューの指定した範囲の項目を削除します。
RemoveGradientSliderSegment(dialogID, componentID : LONGINT; segmentIndex : INTEGER);	10 ～	グラデーションスライダーから変化位置を削除します。
RemoveImagePopupItem(dialogID, componentID : LONGINT; itemIndex : INTEGER);	10 ～	イメージポップアップメニューからメニュー項目を削除します。
RemoveListBoxTabStop(dialogID, itemID : LONGINT);	10 ～	リストボックスから最後のタブストップを削除します。

Dialogs - Modern

RemoveTreeControlItem(nDialogID, nComponentID : LONGINT; nItemID : INTEGER): BOOLEAN;	2008 ～	ツリーアイテムから項目を削除します。
RunLayoutDialog(dialogID : LONGINT; callback : PROCEDURE): LONGINT;	9 ～	指定したダイアログ番号とサブルーチンの名前で、ダイアログイベントループを実行します。
SelectChoice(dialogID, componentID : LONGINT; itemIndex : INTEGER; selectState : BOOLEAN);	2010 ～	プルダウンメニューやリストアイテムにある項目の選択状態を設定します。
SelectEditText(dialogID, componentID : LONGINT);	2010 ～	テキストフィールドを選択状態にします。
SelectTreeControlItem(nDialogID, nComponentID : LONGINT; nItemID : INTEGER);	12.5 ～	指定したツリーアイテムの項目を選択します。
SetBelowItem(dialogID, srcItemID, belowtItemID : LONGINT; indent, lineSpacing : INTEGER);	9 ～	基準とするアイテムの下に、アイテムを配置します。 挿入点からX軸、Y軸方向のオフセットを設定できます。
SetBooleanItem(dialogID, componentID : LONGINT; setState : BOOLEAN);	2010 ～	チェックボックスまたはラジオボタンの選択状態を設定します。
SetColorButton(dialogID, itemID, red, green, blue : LONGINT);	10 ～	カラーボタンの色を設定します。 黒ならばすべての色の成分を0、白ならばすべての色の成分を65535にします。
SetColorChoice(dialogID, itemID : LONGINT; colorIndex : INTEGER);	12 ～	色ポップアップの色を設定します。
SetComponentIndeterminate(nDialogID, nComponentID : LONGINT; bIndeterminateState : BOOLEAN): BOOLEAN;	2008 ～	指定された属性 (線種、 太さ、色など) を、3番目の未定の状態に設定します。
SetComponentSize(nDialogID, nComponentID : LONGINT; nWidthPixels, nHeightPixels : INTEGER): BOOLEAN;	2008 ～	指定したレイアウトダイアログのコンポーネントの幅と高さを設定します。
SetControlData(dialogID, itemID, data : LONGINT);	9 ～	指定したアイテムのデータを設定します。
SetControlText(DlogID, ItemID : INTEGER; newtext : STRING);	10 ～	ラジオボタン、チェックボックス、ボタンに表示する文字列を設定します。
SetEdgeBinding(dialogID, itemID : LONGINT; boundToLeft, boundToRight, boundToTop, boundToBottom : BOOLEAN);	12 ～	レイアウトダイアログに対してコントロールの縁を固定 (TRUE) します。

Dialogs - Modern

SetEditInteger(dialogID, itemID, value : LONGINT);	9 ～	整数型の編集フィールドに整数値を設定します。
SetEditReal(dialogID, itemID, editRealType : LONGINT; value : REAL);	9 ～	実数型の編集フィールドに実数値を設定します。
SetFirstGroupItem(dialogID, groupID, firstItemID : LONGINT);	9 ～	グループボックス内の最初のアイテムを設定します。
SetFirstLayoutItem(dialogID, firstItemID : LONGINT);	9 ～	レイアウト内の最初のアイテムを設定します。
SetFocusOnItem(liDialogID, liComponentID : LONGINT);	12.5 ～	指定した項目にキーボード入力フォーカスを設定します。
SetGradientSliderData(dialogID, componentID : LONGINT; VAR segmentIndex : INTEGER; spotPosition, midpointPosition : REAL; red, green, blue : LONGINT);	10 ～	グラデーションスライダーの変化位置の変化点の位置、変化の中心点の位置や色を設定します。
SetGradientSliderSelectedMarker(dialogID, componentID : LONGINT; segmentIndex, markerType : INTEGER);	10 ～	グラデーションスライダーの点を選択します。
SetHelpText(dialogID, componentID : LONGINT; helpText : STRING);	2010 ～	ヘルプフィールドに表示する文字列を設定します。
SetIconPushButtonState(nDialogID, nComponentID : LONGINT; bPressed : BOOLEAN): BOOLEAN;	2008 ～	指定したアイコンプッシュボタンの状態（押された、押されていないなど）を設定します。
SetImageControlHandle(dialogID, componentID : LONGINT; hImage : HANDLE);	12 ～	指定したイメージのイメージ定義ノードハンドルを設定します。
SetImageControlPath(nDialogID, nComponentID : LONGINT; strPath : STRING): BOOLEAN;	2008 ～	指定したイメージのイメージパスを設定します。 CreateImageControlと共に使用します。
SetImagePopupSelectedItem(dialogID, componentID : LONGINT; itemIndex : INTEGER);	10 ～	イメージポップアップメニューを設定します。
SetItemClickable(dialogID, componentID : LONGINT; clickable : BOOLEAN);	2010 ～	指定されたコントロールがクリックイベントを受け取れるようにします。 現在は静的文字列と画像が対応しています。
SetItemText(dialogID, componentID : LONGINT; text : DYNARRAY[] of CHAR);	2010 ～	テキストフィールドに表示する文字列を設定します。

Dialogs - Modern

SetItemToolTipText(nDialogID, nComponentID : LONGINT; strToolTip, strSubToolTip : STRING; nIndex, nSubIndex : INTEGER);	2008 ～	リストブラウザ、編集フィールド、プルダウンメニューの、ツールチップとして表示される文字列を設定します。
SetLayoutDialogPosition(dialogID : LONGINT; left, top : INTEGER): BOOLEAN;	11 ～	指定した位置にダイアログウインドウを移動します。
SetLayoutDialogSize(dialogID : LONGINT; width, height : INTEGER);	12 ～	レイアウトダイアログのサイズをピクセル単位で設定します。
SetLayoutOption(dialogID : LONGINT; option : INTEGER; value : LONGINT): BOOLEAN;	12 ～	レイアウトダイアログのオプションを設定します。
SetLineAttributeData(dialogID, itemID : LONGINT; lineStyle, lineWeight : INTEGER);	12 ～	線種と線の太さポップアップを設定します。 線種と線の太さの両方を指定できます。
SetLineStyleChoice(dialogID, itemID : LONGINT; lineStyle : INTEGER);	12 ～	線種ポップアップを設定します。
SetLineWeightChoice(dialogID, itemID : LONGINT; lineWeight : INTEGER);	12 ～	線の太さポップアップを設定します。
SetMarkerChoice(dialogID, itemID : LONGINT; index, style, angle : INTEGER; size : REAL);	12 ～ 12.5	マーカーポップアップを設定します（VW2008ではこの関数は廃止されています。代わりにSetMarkerValueを使用してください）。
SetMarkerValue(dialogID, itemID : LONGINT; style, angle : INTEGER; length, width : REAL; basis : INTEGER; thickness : REAL);	2008 ～	マーカーポップアップの値を設定します。 （VW2008より前のバージョンのマーカーポップアップ手続きの代わりになります。）
SetPatternData(dialogID, itemID : LONGINT; patternIndex, foreColor, backColor : INTEGER);	12 ～	模様ポップアップを設定します。
SetProportionalBinding(dialogID, itemID : LONGINT; leftProportional, rightProportional, topProportional, bottomProportional : BOOLEAN);	12 ～	レイアウトダイアログに対してコントロールの位置が比例距離を保つ設定をします。
SetRightItem(dialogID, srcItemID, rightItemID : LONGINT; indent, lineSpacing : INTEGER);	9 ～	基準とするアイテムの右に、アイテムを配置します。挿入点からX軸、Y軸方向のオフセットを設定できます。

Dialogs - Modern

SetSelectionRange(dialogID, controlID : LONGINT; startPos, endPos : INTEGER);	12.0.1 ～	指定したコントロール内の選択範囲の始点と終点を設定します。
SetSliderLiveUpdate(dialogID, componentID : LONGINT; liveUpdate : BOOLEAN);	2010 ～	スライダーアイテムのイベントをライブアップデートにするかを設定します。
SetStaticTextColor(dialogID, componentID : LONGINT; red, green, blue : INTEGER);	2010 ～	編集不可能なフィールドに色を指定します。
SetStaticTextStyle(dialogID, componentID : LONGINT; style : INTEGER);	2010 ～	編集不可能なフィールドに文字スタイルを指定します。
SetThreeStateCheckBoxState(dialogID, componentID : LONGINT; iState : INTEGER);	12.5 ～	3状態チェックボックスの状態を設定します。
SetTreeControlItemData(nDialogID, nComponentID : LONGINT; nItemID : INTEGER; nUserData : LONGINT);	12.5 ～	指定したツリーアイテムのユーザデータを設定します。
SetVSResourceFile(fileName : STRING): BOOLEAN;	9 ～	リソースファイルを開きます。 ファイルの名前は、拡張子を含まない名前で指定します。
ShowEditTileDialog(tileHandle : HANDLE);	2011 ～	タイル編集ダイアログを表示します。 タイルの形状と設定のどちらを編集するか選択することができます。
ShowEditTileSettingsDialog(VAR tileHandle : HANDLE);	2011 ～	タイル設定編集ダイアログを表示します。
ShowEnhancedPullDownMenuGroupIcon(liDialogID, liComponentID : LONGINT; bShowGroupIcon : BOOLEAN);	12.5 ～	指定した拡張プルダウンメニューにグループアイコンを表示するかどうかを設定します。
ShowItem(dialogID : LONGINT; item : INTEGER; show : BOOLEAN);	11.5 ～	ダイアログアイテムの表示／非表示を設定します。
ShowNewTileDialog : HANDLE;	2011 ～	新しいタイルダイアログを表示します。
UpdateSymbolDisplayControl(dialogID, itemID : LONGINT; symbolName : STRING; renderMode, view : INTEGER);	12 ～	シンボル表示コントロールをアップデートします。
VerifyLayout(dialogID : LONGINT): BOOLEAN;	9 ～	作成したレイアウトの整合性をチェックします。

Dialogs - Modern - Browser

AddLBImage(dialogID, componentID : LONGINT; resourceType, resourceID : INTEGER): INTEGER;	11 ～	イメージリソースをイメージリストに追加します。
AreLBColumnLinesEnabled(dialogID, componentID : LONGINT): BOOLEAN;	11 ～	列の枠線が描かれている／いないを返します。
AreLBRadioColumnLinesEnabled(dialogID, componentID : LONGINT; columnIndex : INTEGER): BOOLEAN;	11 ～	列の枠線がラジオボタンの間に描かれている／いないを返します。
CreateLB(dialogID, componentID : LONGINT; widthInCharacters, heightInCharacters : INTEGER);	11 ～	リストブラウザを作成します。
DeleteAllLBItems(dialogID, componentID : LONGINT): BOOLEAN;	12 ～	リストブラウザからすべての項目を削除します。
DeleteLBColumn(dialogID, componentID : LONGINT; columnIndex : INTEGER): BOOLEAN;	11 ～	指定したリストブラウザから列を削除します。
DeleteLBItem(dialogID, componentID : LONGINT; itemIndex : INTEGER): BOOLEAN;	11 ～	指定したリストブラウザからアイテムを削除します。
EnableLB(dialogID, componentID : LONGINT; enable : BOOLEAN): BOOLEAN;	12 ～	リストブラウザを使用する／しないを設定します。
EnableLBClickAllDataChange(dialogID, componentID : LONGINT; enable : BOOLEAN): BOOLEAN;	12 ～	クリックする際にaltキーまたはoptionキーを押していれば、ラジオボタンや複数選択可能な列にあるすべてのデータ項目を一回のクリックで変更する／しないを設定します。
EnableLBColumnLines(dialogID, componentID : LONGINT; enableColumnLines : BOOLEAN);	11 ～	列の枠線を作成する／しないを設定します。
EnableLBColumnTracking(dialogID, componentID : LONGINT; columnIndex : INTEGER; enableColumnTracking : BOOLEAN);	11 ～	列を編成する／しないを設定します。
EnableLBDragAndDrop(dialogID, componentID : LONGINT; enable : BOOLEAN): BOOLEAN;	12 ～	ドラッグ&ドロップ機能を使用する／しないを設定します。SetLBDragDropColumnを使って、ドラッグ＆ドロップ列を設定します。

Dialogs - Modern - Browser

EnableLBRadioColumnLines(dialogID, componentID : LONGINT; columnIndex : INTEGER; enableRadioColumnLines : BOOLEAN);	11 ～	ラジオボタンを列の線に配置する／しないを設定します。
EnableLBSingleLineSelection(dialogID, componentID : LONGINT; enable : BOOLEAN): BOOLEAN;	12 ～	1行のみ選択可能に設定します。
EnableLBSorting(dialogID, componentID : LONGINT; enableSorting : BOOLEAN);	11 ～	ソートする／しないを設定します。
EnableLBUpdates(liDialogID, liComponentID : LONGINT; bEnableUpdates : BOOLEAN);	12.5 ～	リストブラウザを更新する／しないを設定します。
EnsureLBItemIsVisible(dialogID, componentID : LONGINT; index : INTEGER): BOOLEAN;	12 ～	リストブラウザで、指定した行のデータが表示されるようにします。
FindLBColumnDataItem(dialogID, componentID : LONGINT; columnIndex : INTEGER; itemString : STRING; VAR columnDataItemIndex : INTEGER): BOOLEAN;	11 ～	文字列を指定して列に含まれているデータアイテムを検索します。
FindLBColumnItem(dialogID, componentID : LONGINT; columnIndex : INTEGER; itemString : STRING; VAR itemIndex : INTEGER): BOOLEAN;	11 ～	文字列を指定して列に含まれているアイテムを検索します。
GetLBColumnDataItemInfo(dialogID, componentID : LONGINT; columnIndex, columnDataItemIndex : INTEGER; VAR itemString : STRING; VAR imageOn, imageOff : INTEGER; VAR itemData : LONGINT): BOOLEAN;	11 ～	指定した列のデータを返します。
GetLBColumnHeaderJust(dialogID, componentID : LONGINT; columnIndex : INTEGER; VAR justification : INTEGER): BOOLEAN;	12 ～	指定した列ヘッダーの位置揃えを返します。

Dialogs - Modern - Browser

GetLBColumnHeaderToolTip(dialogID, componentID : LONGINT; columnIndex : INTEGER; VAR toolTipPrimaryText, toolTipSubText : STRING): BOOLEAN;	12 ～	リストブラウザの列ヘッダーでツールチップとして表示される文字列を返します。
GetLBColumnOwnerDrawnType(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR ownerDrawnType : INTEGER): BOOLEAN;	12 ～	リストブラウザの列の描画タイプを返します。
GetLBColumnSortState(dialogID, componentID : LONGINT; columnIndex : INTEGER): INTEGER;	12 ～	列のソート状態を返します。
GetLBColumnWidth(dialogID, componentID : LONGINT; columnIndex : INTEGER; VAR width : INTEGER): BOOLEAN;	2009 ～	指定したリストブラウザにある指定した列の幅を返します。
GetLBControlType(dialogID, componentID : LONGINT; columnIndex : INTEGER): INTEGER;	11 ～	列の制御タイプを返します。
GetLBEditDisplayType(dialogID, componentID : LONGINT; columnIndex : INTEGER): INTEGER;	11 ～	指定した列にあるリストアイテムの表示形式を返します。
GetLBEventInfo(dialogID, componentID : LONGINT; VAR eventType, rowIndex, columIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザで最後に実行されたイベントの情報を返します。
GetLBItemDashStyle(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR styleIndex, lineWeight : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの破線の種類を返します。
GetLBItemData(nDialogID, nComponentID : LONGINT; nItemIndex, nSubItemIndex : INTEGER; VAR nUserData : LONGINT);	12.5 ～	リストブラウザアイテムのユーザデータを返します。
GetLBItemDisplayType(dialogID, componentID : LONGINT; columnIndex : INTEGER): INTEGER;	11 ～	指定した列にあるリストアイテムの表示形式を返します。

Dialogs - Modern - Browser

GetLBItemFillBackColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの面の地色を返します。
GetLBItemFillForeColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの面の色を返します。
GetLBItemGradientOrImageRefNumber(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR refNumber : LONGINT): BOOLEAN;	12 ～	リストブラウザアイテムのグラデーションまたはイメージを返します。
GetLBItemInfo(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR itemString : STRING; VAR imageIndex : INTEGER): BOOLEAN;	11 ～	指定したアイテムのデータを返します。
GetLBItemPatternIndex(dialogID, componentID : LONGINT; itemIndex, the column index : INTEGER; VAR outPatIndex : INTEGER): BOOLEAN;	2010 ～	リストブラウザアイテムの模様番号を返します。
GetLBItemPenBackColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの線の地色を返します。
GetLBItemPenForeColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの線の色を返します。
GetLBItemTextColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの文字の色を返します。

Dialogs - Modern - Browser

GetLBItemTextJust(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR justification : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの文字の位置揃えを返します。
GetLBItemTextStyle(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR textStyle : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの文字のスタイルを返します。
GetLBMultImageIndexes(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; VAR imageIndex0, imageIndex1, imageIndex2 : INTEGER): BOOLEAN;	12 ～	リストブラウザの複数イメージ表示にあるイメージの番号を返します。
GetLBSortColumn(dialogID, componentID : LONGINT): INTEGER;	12 ～	リストブラウザでソートされている列の番号を返します。
GetNumLBColumnDataItems(dialogID, componentID : LONGINT; columnIndex : INTEGER): INTEGER;	11 ～	列のデータアイテムの数を返します。
GetNumLBColumns(dialogID, componentID : LONGINT): INTEGER;	11 ～	指定したリストブラウザにある列の数を返します。
GetNumLBItems(dialogID, componentID : LONGINT): INTEGER;	11 ～	指定したリストブラウザにあるアイテムの数を返します。
GetNumSelectedLBItems(dialogID, componentID : LONGINT): INTEGER;	12 ～	リストブラウザで選択されているアイテムの数を返します。
InsertLBColumn(dialogID, componentID : LONGINT; columnIndex : INTEGER; headerString : STRING; width : INTEGER): INTEGER;	11 ～	指定したリストブラウザに列を挿入します。 作成された列の番号を返します。
InsertLBColumnDataItem(dialogID, componentID : LONGINT; columnIndex : INTEGER; itemString : STRING; imageOn, imageOff : INTEGER; itemData : LONGINT): INTEGER;	11 ～	指定したデータを列に挿入します。作成されたアイテムの番号を返します。

Dialogs - Modern - Browser

InsertLBItem(dialogID, componentID : LONGINT; itemIndex : INTEGER; itemString : STRING): INTEGER;	11 ～	指定したリストブラウザにアイテムを挿入します。 作成されたアイテムの番号を返します。
IsLBColumnTrackingEnabled(dialogID, componentID : LONGINT; columnIndex : INTEGER): BOOLEAN;	11 ～	指定した列で編成の有効／無効を返します。
IsLBItemSelected(dialogID, componentID : LONGINT; itemIndex : INTEGER): BOOLEAN;	11 ～	指定したアイテムが現在選択されている／いないを返します。
IsLBSortingEnabled(dialogID, componentID : LONGINT): BOOLEAN;	11 ～	並び替え（ソート）の有効／無効を返します。
RefreshLB(dialogID, componentID : LONGINT): BOOLEAN;	12 ～	リストブラウザの内容を更新します。
RemoveAllLBColumnDataItems(dialogID, componentID : LONGINT; columnIndex : INTEGER);	11 ～	すべての列のデータアイテムを削除します。
RemoveLBColumnDataItem(dialogID, componentID : LONGINT; columnIndex, columnDataItemIndex : INTEGER): BOOLEAN;	11 ～	指定した列のデータアイテムを削除します。
SetFocusOnLB(dialogID, componentID : LONGINT): BOOLEAN;	12 ～	リストブラウザにキーボードまたは入力フォーカスを設定します。
SetLBColumnHeaderJust(dialogID, componentID : LONGINT; columnIndex, justification : INTEGER): BOOLEAN;	12 ～	指定した列ヘッダーの位置揃えを設定します。
SetLBColumnHeaderToolTip(dialogID, componentID : LONGINT; columnIndex : INTEGER; toolTipPrimaryText, toolTipSubText : STRING): BOOLEAN;	12 ～	リストブラウザの列ヘッダーでツールチップとして表示される文字列を設定します。
SetLBColumnImage(nDialogID, nComponentID : LONGINT; nColumnIndex, nImageIndex : INTEGER): BOOLEAN;	2008 ～	リストブラウザの列ヘッダーでテキストの代わりにアイコンを描画します。 AddLBImageとともに使用します。

Dialogs - Modern - Browser

SetLBColumnOwnerDrawnType(dialogID, componentID : LONGINT; itemIndex, subItemIndex, ownerDrawnType : INTEGER): BOOLEAN;	12 ～	リストブラウザの列の描画タイプを設定します。
SetLBColumnWidth(dialogID, componentID : LONGINT; fromColumn, toColumn, width : INTEGER): BOOLEAN;	11 ～	指定したリストブラウザの指定した範囲の列の幅を設定します。
SetLBControlType(dialogID, componentID : LONGINT; columnIndex, controlType : INTEGER): BOOLEAN;	11 ～	列に列の種類を設定します。
SetLBDragDropColumn(dialogID, componentID : LONGINT; columnIndex : INTEGER): BOOLEAN;	12 ～	ドラッグ&ドロップ列を設定します。 ドラッグ&ドロップ列は「コントロールタイプ6:数字」でなければなりません。
SetLBEditDisplayType(dialogID, componentID : LONGINT; columnIndex, displayType : INTEGER): BOOLEAN;	11 ～	指定した列のリストアイテムの表示形式を設定します。
SetLBItemDashStyle(dialogID, componentID : LONGINT; itemIndex, subItemIndex, styleIndex, lineWeight : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの破線の種類を設定します。
SetLBItemData(nDialogID, nComponentID : LONGINT; nItemIndex, nSubItemIndex : INTEGER; nUserData : LONGINT);	12.5 ～	リストブラウザアイテムのユーザデータを設定します。
SetLBItemDisplayType(dialogID, componentID : LONGINT; columnIndex, displayType : INTEGER): BOOLEAN;	11 ～	指定した列のリストアイテムの表示形式を設定します。
SetLBItemFillBackColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex, redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの面の地色を設定します。
SetLBItemFillForeColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex, redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの面の色を設定します。

Dialogs - Modern - Browser

SetLBItemGradientOrImageRefNumber(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; refNumber : LONGINT): BOOLEAN;	12 ～	リストブラウザアイテムのグラデーションまたはイメージを設定します。
SetLBItemInfo(dialogID, componentID : LONGINT; itemIndex, subItemIndex : INTEGER; itemString : STRING; imageIndex : INTEGER): BOOLEAN;	11 ～	アイテムにデータを設定します。
SetLBItemPatternIndex(dialogID, componentID : LONGINT; itemIndex, subItemIndex, patIndex : INTEGER): BOOLEAN;	2010 ～	リストブラウザアイテムの模様番号を設定します。
SetLBItemPenBackColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex, redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの線の地色を設定します。
SetLBItemPenForeColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex, redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの線の色を設定します。
SetLBItemTextColor(dialogID, componentID : LONGINT; itemIndex, subItemIndex, redIndex, greenIndex, blueIndex : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの文字の色を設定します。
SetLBItemTextJust(dialogID, componentID : LONGINT; itemIndex, subItemIndex, justification : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの文字の位置揃えを設定します。
SetLBItemTextStyle(dialogID, componentID : LONGINT; itemIndex, subItemIndex, textStyle : INTEGER): BOOLEAN;	12 ～	リストブラウザアイテムの文字のスタイルを設定します。
SetLBItemUsingColumnDataItem(dialogID, componentID : LONGINT; itemIndex, subItemIndex, columnDataItemIndex : INTEGER): BOOLEAN;	11 ～	指定した列のデータアイテムにリストアイテムデータを設定します。

Dialogs - Modern - Browser

SetLBMultImageIndexes(dialogID, componentID : LONGINT; itemIndex, subItemIndex, imageIndex0, imageIndex1, imageIndex2 : INTEGER): BOOLEAN;	12 ～	リストブラウザの複数イメージ表示にあるイメージの番号を設定します。
SetLBSelection(dialogID, componentID : LONGINT; firstItemIndex, lastItemIndex : INTEGER; select : BOOLEAN): BOOLEAN;	11 ～	指定した範囲のアイテムを選択します。
SetLBSortColumn(dialogID, componentID : LONGINT; columnIndex : INTEGER; isAscending : BOOLEAN);	12 ～	リストブラウザにある特定の列をソート列として設定します。

Dialogs - Predefined

AlertCritical(text, advice : STRING);	12 ～	ストップアラートダイアログを表示します。
AlertInform(text, advice : STRING; minorAlert : BOOLEAN);	12 ～	ノートアラートダイアログを表示します。
AlertInformDontShowAgain(text, advice : STRING; minorAlert : BOOLEAN; arrOptions : ARRAY);	2010 ～	二度と表示しないオプション付きのアラートダイアログ表示します。
AlertQuestion(question, advice : STRING; defaultButton : INTEGER; OKOverrideText, CancelOverrideText, customButtonAText, customButtonBText : STRING): INTEGER;	12 ～	注意アラートダイアログを表示します。
AlertQuestionDontShowAgain(question, advice : STRING; defaultButton : INTEGER; OKOverrideText, CancelOverrideText, customButtonAText, customButtonBText : STRING; arrOptions : ARRAY): INTEGER;	2010 ～	二度と表示しないオプション付きの注意アラートダイアログ表示します
AlrtDialog(message : STRING);	1 ～	警告ダイアログを表示します。
AngDialog(request, default : STRING): REAL;	1 ～	角度を入力するためのダイアログを表示し、入力された値を返します。
AngDialog3D(displayStr, xStr, yStr, zStr : STRING; VAR xAngleResult, yAngleResult, zAngleResult : REAL);	1 ～	3次元の角度を入力するためのダイアログを表示し、入力された値を返します。
DidCancel : BOOLEAN;	1 ～	直前に表示されたダイアログで、「キャンセル」ボタンが押された場合はTRUEを返しま
DistDialog(request, default : STRING): REAL;	1 ～	距離を入力するためのダイアログを表示し、入力された値を返します。
FormatTextDialog(VAR fontName : STRING; VAR style : INTEGER; VAR size : REAL; VAR spacing : INTEGER; VAR leading : REAL; VAR hAlignment, vAlignment : INTEGER; disableMask : INTEGER);	9 ～	文字設定ダイアログを表示し、その内容を返します。
IntDialog(request, default : STRING): INTEGER;	1 ～	整数を入力するためのダイアログを表示し、入力された値を返します。

Dialogs - Predefined

NonUndoableActionOK:BOOLEAN;	8 ～	取り消しができないことをダイアログでユーザーに知らせます。 ユーザが「OK」ボタンをクリックした場合はTRUEを、「キャンセル」ボタンをクリックした場合はFALSEを返します。
PtDialog(request, defaultX, defaultY : STRING; VAR x, y : REAL);	1 ～	座標を入力するためのダイアログを表示し、入力された値を返します。
PtDialog3D(displayStr, xStr, yStr, zStr : STRING; VAR xPt, yPt, zPt : REAL);	1 ～	3次元の座標を入力するためのダイアログを表示し、入力された値を返します。
RealDialog(request, default : STRING): REAL;	1 ～	実数を入力するためのダイアログを表示し、入力された値を返します。
StrDialog(request, default : STRING): STRING;	1 ～	文字列を入力するためのダイアログを表示し、入力された値を返します。
YNDialog(s : STRING): BOOLEAN;	1 ～	「はい」か「いいえ」を選択するダイアログを表示し、どちらのボタンが押されたかを返します。ユーザーが「はい」ボタンをクリックした場合はTRUEを、「いいえ」ボタンをクリックした場合はFALSEを返します。

Dimensions

AngularDim(startPtX, startPtY, endPtX, endPtY, vert1X, vert1Y : REAL; textOffsetDistance : REAL; arrow, textFlag : INTEGER; posAngle : REAL);	1 ～	角度の寸法線を作成します。
AssociateLinearDimension(h : HANDLE; selectedObjectsMode : BOOLEAN);	12.5 ～	寸法の終点がオブジェクトと一致した場合、寸法線をオブジェクトと関連づけます。 selectedObjectsModeがTRUEならば、選択されているオブジェクトだけがチェックされます。
CircularDim(startPtX, startPtY, endPtX, endPtY, box1X, box1Y, box2X, box2Y : REAL; textOffsetDistance : REAL; dimType, arrow, textFlag : INTEGER; shoulder : REAL);	1 ～	半径、直径の寸法線を作成します。
CreateChainDimension(h1, h2 : HANDLE): HANDLE;	12.5 ～	通過する2つの寸法またはチェーンが、単体の部分寸法オブジェクトになるための要件に合致すれば、新しい部分寸法オブジェクトを作成して返します。
DimArcText;	1 ～	直前に作成した円弧から寸法を作成します。
DimText;	1 ～	直前に作成した直線から寸法を作成します。
DoubleFixedTolerance(showVal, boxText : BOOLEAN; leader, trailer, topStr, botStr : STRING);	4 ～	作成直後の寸法線に対して、上下別の公差寸法（文字）を設定します。
DoubleTolerance(showVal, boxText : BOOLEAN; leader, trailer : STRING; topDistance, botDistance : REAL);	4 ～	作成直後の寸法線に対して、上下別の公差寸法表示を設定します。
GetDimText(h : HANDLE): STRING;	1 ～	ハンドルで指定した寸法線の値を文字列で返します。
HasDim(h : HANDLE): BOOLEAN;	1 ～	ハンドルで指定した図形が寸法文字を含んでいればTRUEを、含んでいなければFALSEを返します。
LimitTolerance(showVal, boxText : BOOLEAN; leader, trailer : STRING; lowDistance, hiDistance : REAL);	4 ～	作成直後の寸法線に対して、許容限界公差寸法を設定します。
LinearDim(startPtX, startPtY, endPtX, endPtY : REAL; offsetDistance : REAL; dimType, arrow, textFlag : INTEGER; textOffset : REAL);	1 ～	横、縦、斜めの寸法線を作成します。
SetDimText(h : HANDLE; leaderTrailer : STRING);	4 ～	ハンドルで指定した寸法線の値を、指定した文字列に置き換えます。

Dimensions

SingleTolerance(showVal, boxText : BOOLEAN; leader, trailer : STRING; limDistance : REAL);	4 ～	作成直後の寸法線に対して、上下等の公差寸法表示を設定します。

Document Attributes

AddTileGeometryObject(tileHandle : HANDLE; objectHandle :HANDLE):BOOLEAN;	2011 ～	指定したタイルリソースに指定した図形を追加します。
CreateImageFromPaint(paint : HANDLE; imageName : STRING): HANDLE;	10 ～	ペイントノードからイメージを作成します。
CreateTile(tileName : STRING): HANDLE;	2011 ～	タイルを新規に作成します。
FFillBack(VAR red, green, blue : LONGINT);	1 ～	現在設定されている面の地色の成分を返します。 値の範囲は0から65535までです。
FFillColorByClass : BOOLEAN;	8 ～	現在設定されている面の色に、クラス属性を使っている場合はTRUEを返します。
FFillFore(VAR red, green, blue : LONGINT);	1 ～	現在設定されている面の色の成分を返します。 値の範囲は0から65535までです。
FFillPat : LONGINT;	1 ～	現在設定されている模様番号を返します。
FFPatByClass : BOOLEAN;	8 ～	現在設定されている面の模様に、クラス属性を使っている場合はTRUEを返します。
FillBack(colorR, colorG, colorB : LONGINT);	1 ～	現在設定されている面の地色を変更します。 値の範囲は0から65535までです。
FillFore(colorR, colorG, colorB : LONGINT);	1 ～	現在設定されている面の色を変更します。 値の範囲は0から65535までです。
FillPat(patNumber : LONGINT);	1 ～	アクティブな面の模様を設定します。 この手続きの後に作成された図形は指定した面の模様が使われます。
FLSByClass : BOOLEAN;	8 ～	現在設定されている線の種類に、クラス属性を使っている場合はTRUEを返します。
FLWByClass : BOOLEAN;	8 ～	現在設定されている線の太さに、クラス属性を使っている場合はTRUEを返します。
FMarker(VAR style : INTEGER; VAR size : REAL; VAR ang : INTEGER);	6 ～ 12.5	現在設定されているマーカースタイル、長さ(インチ)、角度(度数)を返します。 (この関数は使用できないので、GetDefaultBeginningMarkerまたは GetDefaultEndMarkerを使用してください。)
FMarkerByClass:BOOLEAN;	8 ～	現在設定されているマーカーの種類に、クラス属性を使っている場合はTRUEを返しま
FPenBack(VAR red, green, blue : LONGINT);	1 ～	現在設定されている線の地色の成分を返します。 値の範囲は0から65535までです。
FPenColorByClass : BOOLEAN;	8 ～	現在設定されている線の色に、クラス属性を使っている場合はTRUEを返します。
FPenFore(VAR red, green, blue : LONGINT);	1 ～	現在設定されている線の色成分を返します。 値の範囲は0から65535までです。
FPenPat : INTEGER;	1 ～	現在設定されている線の模様を返します。
FPenSize : INTEGER;	1 ～	現在設定されている線の太さを返します。
GetDashStyle(swt : BOOLEAN; numPairs : INTEGER; pair1DX, pair1DY, pair2DX, pair2DY, pair3DX, pair3DY, pair4DX, pair4DY, pair5DX, pair5DY : REAL): INTEGER;	5 ～	現在設定されている破線の種類を返します。

Document Attributes

GetDashStyleIndex(swt : BOOLEAN; numPairs : INTEGER; pair1DX, pair1DY, pair2DX, pair2DY, pair3DX, pair3DY, pair4DX, pair4DY, pair5DX, pair5DY : REAL): INTEGER;	2010 ～	現在設定されている破線の種類を返します。
GetDefaultBeginningMarker(VAR style : LONGINT; VAR angle : INTEGER; VAR size, width : REAL; VAR thicknessBasis : INTEGER; VAR thickness : REAL; VAR visibility : BOOLEAN): BOOLEAN;	2008 ～	ファイルのデフォルト始点マーカーのすべての設定値を返します。
GetDefaultEndMarker(VAR style : LONGINT; VAR angle : INTEGER; VAR size, width : REAL; VAR thicknessBasis : INTEGER; VAR thickness : REAL; VAR visibility : BOOLEAN): BOOLEAN;	2008 ～	ファイルのデフォルト終点マーカーの設定値を返し、正常終了するとTRUEを返します。
GetDefaultOpacity(VAR opacity : INTEGER);	2008 ～	デフォルトの不透明度に返します。
GetDocumentDefaultSketchStyle : STRING;	11.5 ～	デフォルトのスケッチスタイルを返します。
GetGradientData(gradient : HANDLE; segmentIndex : INTEGER; VAR spotPosition, midpointPosition : REAL; VAR red, green, blue : LONGINT);	10 ～	変化位置の変化点、変化の中心点の位置と色を返します。
GetGradientMidpointPosition(gradient : HANDLE; segmentIndex : INTEGER; VAR position : REAL);	10 ～	変化位置の変化の中心点の位置を返します。
GetGradientSpotColor(gradient : HANDLE; segmentIndex : INTEGER; VAR red, green, blue : LONGINT);	10 ～	変化位置の変化点の色を返します。
GetGradientSpotPosition(gradient : HANDLE; segmentIndex : INTEGER; VAR position : REAL);	10 ～	変化位置の変化点の位置を返します。
GetNumGradientSegments(gradient : HANDLE): INTEGER;	10 ～	グラデーションの変化位置の数を返します。

Document Attributes

GetTileBackgroundColor(tileHandle : HANDLE; VAR red : LONGINT; VAR green : LONGINT; VAR blue : LONGINT);	2011 ～	指定したタイルリソースの背景色を取得します。
GetTileGeometryGroup(tileHandle : HANDLE): HANDLE;	2011 ～	タイルリソースの形状グループを取得する。
GetTileGroupParent(groupHandle : HANDLE): HANDLE;	2011 ～	指定したタイルグループを内包するコンテナのハンドルを取得する。
GetTileOffsetPoint(tileHandle : HANDLE; VAR offsetPoint :POINT);	2011 ～	指定したタイルリソースのオフセットポイントを取得します。
GetTileRepetitionPoint(tileHandle : HANDLE; VA_ repetitionPoint : POINT);	2011 ～	指定したタイルリソースの繰り返しポイントを取得します。
InsertGradientSegment(gradient : HANDLE; spotPosition, midpointPosition : REAL; red, green, blue : LONGINT): INTEGER;	10 ～	グラデーションに新しい変化位置を挿入し、そのデータを指定した値で初期化します。
IsTileGroupContainedObject(objectHandle : HANDLE)BOOLEAN;	2011 ～	指定した図形がタイルグループに含まれている/いないを返します。
IsUserColor(ColorIDX : INTEGER; VAR ColorName : DYNARRAY[] of CHAR): BOOLEAN;	2008 ～	色がユーザ色の場合は、TRUEを返します。
Marker(style : INTEGER; size : REAL; ang : INTEGER);	6 ～ 12.5	現在設定されているマーカースタイル、長さ（インチ）、角度（度数）を変更します。 （この関数は廃止されています:代わりに、SetDefaultBeginningMarkerまたは SetDefaultEndMarkerを使用してください。）
NumColors : INTEGER;	2008 ～	現在のドキュメントで最後に使用した線の色を返します。
NumDashStyles : INTEGER;	4 ～	現在設定されている破線の数を返します。
Opacity(opacity : INTEGER);	2008 ～	アクティブな不透明度を設定します。
PenBack(colorR, colorG, colorB : LONGINT);	1 ～	現在設定されている線の地色を変更します。 値の範囲は0から65535までです。
PenFore(colorR, colorG, colorB : LONGINT);	1 ～	現在設定されている線の色を変更します。 値の範囲は0から65535までです。
PenPat(patNumber : INTEGER);	1 ～	アクティブな線の模様／種類を設定します。
PenSize(lw : INTEGER);	1 ～	現在設定されている線の太さを変更します。

Document Attributes

RemoveGradientSegment(gradient : HANDLE; segmentIndex : INTEGER);	10 ～	グラデーションから変化位置を削除します。
SetDashStyle(swt : BOOLEAN; numPairs : INTEGER; pair1DX, pair1DY, pair2DX, pair2DY, pair3DX, pair3DY, pair4DX, pair4DY, pair5DX, pair5DY : REAL);	4 ～	指定された線の長さと間隔をもとに破線の種類を追加します。
SetDashStyleN(name : STRING; swt : BOOLEAN; numPairs : INTEGER; pair1DX, pair1DY, pair2DX, pair2DY, pair3DX, pair3DY, pair4DX, pair4DY, pair5DX, pair5DY : REAL);	2010 ～	指定された線の長さと間隔をもとに破線の種類を追加します。
SetDefaultBeginningMarker(style : LONGINT; angle : INTEGER; size, width : REAL; thicknessBasis : INTEGER; thickness : REAL; visibility : BOOLEAN): BOOLEAN;	2008 ～	ファイルのデフォルト始点マーカーのすべての設定値を設定します。
SetDefaultEndMarker(style : LONGINT; angle : INTEGER; size, width : REAL; thicknessBasis : INTEGER; thickness : REAL; visibility : BOOLEAN): BOOLEAN;	2008 ～	ファイルのデフォルト終点マーカーのすべての設定値を設定します。 正常終了するとTRUEが返されます。
SetDefaultOpacity(opacity : INTEGER);	2008 ～	デフォルトの不透明度を設定します。
SetDefaultOpacityByClass;	2008 ～	現在のクラスの不透明度を、デフォルトの不透明度に設定します。
SetDocumentDefaultSketchStyle(sketchName : STRING): BOOLEAN;	11.5 ～	デフォルトのスケッチスタイルを設定します。
SetGradientData(gradient : HANDLE; VAR segmentIndex : INTEGER; spotPosition, midpointPosition : REAL; red, green, blue : LONGINT);	10 ～	変化位置の変化点、変化の中心点の位置や色を設定します。
SetGradientMidpointPosition(gradient : HANDLE; segmentIndex : INTEGER; position : REAL);	10 ～	変化位置の変化の中心点の位置を設定します。

Document Attributes

SetGradientSpotColor(gradient : HANDLE; segmentIndex : INTEGER; red, green, blue : LONGINT);	10 ～	変化位置の変化点を設定します。
SetGradientSpotPosition(gradient : HANDLE; VAR segmentIndex : INTEGER; position : REAL);	10 ～	変化位置の変化点の位置を設定します。
SetTileBackgroundColor(tileHandle : HANDLE; backgroundColorR : LONGINT; backgroundColorG : LONGINT; backgroundColorB : LONGINT);	2011 ～	指定したタイルリソースの背景色を設定します。
SetTileOffsetPoint(tileHandle : HANDLE; offsetPoint : POINT);	2011 ～	指定したタイルリソースのオフセットポイントを設定します。
SetTileRepetitionPoint(tileHandle : HANDLE; repetitionPoint : POINT);	2011 ～	指定したタイルリソースの繰り返しポイントを設定します。
SheetList(sheetIndex : INTEGER): STRING;	10 ～	番号で指定した登録画面の名前を返します。
SheetNum : INTEGER;	10 ～	ドキュメントに含まれる登録画面の数を返します。
ShowCreateImageDialog : HANDLE;	10 ～	ダイアログを表示して、イメージ属性を作成するためのファイルをユーザーに選択させま
ShowGradientEditorDialog(VAR gradient : HANDLE);	10 ～	グラデーションの編集ダイアログを表示します。

Document List Handling

AddResourceToList(listID : LONGINT; resource : HANDLE): LONGINT;	12 ～	指定されたリソースを指定されたリソースリストに加えます。 リソースのインデックスを返します。
BuildResourceList(type, folderIndex : INTEGER; subFolderName : STRING; VAR numItems : LONGINT): LONGINT;	12 ～	カレントファイルと、指定されたフォルダがあるならば、指定されたタイプの全てのリソースのリストを作成します。 リストIDを返します。
DeleteResourceFromList(listID, index : LONGINT);	12 ～	指定したリソースリストから指定した番号のオブジェクトを削除します。
FActLayer : HANDLE;	1 ～	アクティブなレイヤ上にある最上位の図形のハンドルを返します。 図形が存在しない場合はNILを返します。
FIn3D(objectHd : HANDLE): HANDLE;	1 ～	ハンドルで指定した3次元図形の中で最上位の図形のハンドルを返します。
FInFolder(sfHd : HANDLE): HANDLE;	1 ～	ハンドルで指定したフォルダの中で最上位のシンボルのハンドルを返します。 図形が存在しない場合はNILを返します。
FInGroup(ObjectHd : HANDLE): HANDLE;	1 ～	ハンドルで指定したグループの中で最上位の図形のハンドルを返します。
FInLayer(h : HANDLE): HANDLE;	1 ～	ハンドルで指定したレイヤ上にある最上位の図形のハンドルを返します。 図形が存在しない場合はNILを返します。
FInSymDef(sdHd : HANDLE): HANDLE;	1 ～	ハンドルで指定した登録シンボルの中で最上位の図形のハンドルを返します。
FObject:HANDLE;	1 ～	ドキュメント内にある最上位の図形のハンドルを返します。 図形が存在しない場合はNILを返します。
FSActLayer:HANDLE;	1 ～	アクティブなレイヤ上で選択されている最上位の図形のハンドルを返します。 図形が存在しない場合はNILを返します。
FSObject(h : HANDLE): HANDLE;	1 ～	ハンドルで指定したレイヤ上で選択されている最上位の図形のハンドルを返します。 図形が存在しない場合はNILを返します。
FSymDef : HANDLE;	1 ～	シンボルライブラリに登録されている最上位のシンボルのハンドルを返します。 図形が存在しない場合はNILを返します。
GetActualNameFromResourceList(listID, index : LONGINT): STRING;	12 ～	指定リソースリストの特定アイテムの実際の名前を返します。 このコールにより、別のファイルから同名でリソースに追加されたファイル名が削除されます。
GetNameFromResourceList(listID, index : LONGINT): STRING;	12 ～	指定リソースリストの特定アイテムの表示名を返します。 指定リストには、リソースに追加された同名のファイル名が含まれています。

Document List Handling

GetResourceFromList(listID, index : LONGINT): HANDLE;	12 ～	リソースが現在のドキュメントにあれば、現在リソースリストから特定のアイテムを返します。 そうでなければ、0を返します。
ImportResourceToCurrentFile(listID, index : LONGINT): HANDLE;	12 ～	指定リソースリストから特定のアイテムを現在のファイルに取り込みます(特定のアイテムが現在のファイルになかった場合)。それは、リソースを返します。
LActLayer : HANDLE;	1 ～	アクティブなレイヤ上の最下位の図形のハンドルを返します。
LNewObj : HANDLE;	1 ～	直前に作成された図形のハンドルを返します。
Lobject : HANDLE;	1 ～	ドキュメント内にある最下位の図形のハンドルを返します。
LSActLayer : HANDLE;	1 ～	アクティブなレイヤ上で選択されている最下位の図形のハンドルを返します。 図形が存在しない場合はNILを返します。
NextDObj(h : HANDLE): HANDLE;	1 ～	ハンドルで指定した図形の次にある図形のうち、選択されていないもののハンドルを返します。図形が存在しない場合はNILを返します。
NextLayer(h : HANDLE): HANDLE;	1 ～	ハンドルで指定したレイヤの次にあるレイヤのハンドルを返します。 図形が存在しない場合はNILを返します。
NextObj(h : HANDLE): HANDLE;	1 ～	ハンドルで指定した図形の次にある図形のハンドルを返します。 図形が存在しない場合はNILを返します。
NextSObj(h : HANDLE): HANDLE;	1 ～	ハンドルで指定した図形の次にある図形のうち、選択されている図形のハンドルを返します。図形が存在しない場合はNILを返します。
NextSymDef(symHd : HANDLE): HANDLE;	1 ～	ハンドルで指定した登録シンボルの次にある登録シンボルのハンドルを返します。 図形が存在しない場合はNILを返します。
PrevDObj(h : HANDLE): HANDLE;	1 ～	ハンドルで指定した図形の前にある図形のうち、選択されていない図形のハンドルを返します。 図形が存在しない場合はNILを返します。
PrevLayer(h : HANDLE): HANDLE;	1 ～	ハンドルで指定したレイヤの前にあるレイヤのハンドルを返します。
PrevObj(h : HANDLE): HANDLE;	1 ～	ハンドルで指定した図形の前にある図形のハンドルを返します。 図形が存在しない場合はNILを返します。
PrevSObj(h : HANDLE): HANDLE;	1 ～	ハンドルで指定した図形の前にある図形のうち、選択されている図形のハンドルを返します。
PrevSymDef(symHd : HANDLE): HANDLE;	1 ～	ハンドルで指定した登録シンボルの前にある登録シンボルのハンドルを返します。
ResourceListSize(listID : LONGINT): LONGINT;	12 ～	指定したリソースリストのアイテム数を返します。

Document List Handling

SetParent(obj, container : HANDLE): BOOLEAN;	10 ～	現在のコンテナから図形を削除し、指定したコンテナへ置きます。

Document Settings

DeleteAllDLComponents : BOOLEAN;	12.5 ～	「ダブルラインの設定」の全構成要素を削除します。
DeleteDLComponent(index : INTEGER): BOOLEAN;	12.5 ～	「ダブルラインの設定」のindex番目の構成要素を削除します。
DoubLines(doubleLineDistance : REAL);	1 ～ 12.0.1	ダブルラインの間隔を設定します。
DrwSize(rows, columns : INTEGER);	1 ～	印刷領域の大きさを設定します。各用紙の大きさは、プリンタの種類に応じて「ファイル」メニューの「用紙設定...」で設定します。
GetCurrentPlanarRefID : LONGINT;	2011 ～	現在の平面参照番号を返します。 これはどんな平面であってもよいです ： ワーキング平面、スクリーン平面、コンテナのグランドプレーン、任意の平面
GetDefaultTextSize : REAL;	8 ～	デフォルトの文字の大きさをポイントで返します。
GetDLComponentClass(index : INTEGER; VAR componentClass : LONGINT): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素のクラスを返します。
GetDLComponentFill(index : INTEGER; VAR fill : LONGINT): BOOLEAN;	12.5 ～	「ダブルラインの設定」の、番号で指定した構成要素の面の模様番号を取得します。
GetDLComponentFillColors(index : INTEGER; VAR fillForeColor, fillBackColor : INTEGER): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の面の色と地色を返します。
GetDLComponentName(index : INTEGER): STRING;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の名前を返します。
GetDLComponentPenColors(index : INTEGER; VAR leftPenForeColor, leftPenBackColor, rightPenForeColor, rightPenBackColor : INTEGER): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の左右の線の色と地色を返します。
GetDLComponentPenStyles(index : INTEGER; VAR penStyleLeft, penStyleRight : INTEGER): BOOLEAN;	12.5 ～	「ダブルラインの設定」の、番号で指定した構成要素の左右の線の種類を取得します。
GetDLComponentPenWeights(index : INTEGER; VAR penWeightLeft, penWeightRight : INTEGER): BOOLEAN;	12.5 ～	「ダブルラインの設定」の、番号で指定した構成要素の左右の線の太さを取得します。
GetDLComponentUseFillClassAttr(index : INTEGER; VAR useClassAttr : BOOLEAN): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の面の模様に、クラス属性を使っている場合はTRUEを返します。

Document Settings

GetDLComponentUsePenClassAttr(index : INTEGER; VAR leftPenUseClassAttr, rightPenUseClassAttr : BOOLEAN): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の左右の線種に、クラス属性を使っている場合はTRUEを返します。
GetDLComponentWidth(index : INTEGER; VAR width : REAL): BOOLEAN;	12.5 ～	「ダブルラインの設定」のインデックスで、構成要素の幅を取得します。
GetDLControlOffset : REAL;	12.5 ～	「ダブルラインの設定」の、オフセットの値を返します。
GetDLOptions : INTEGER;	12.5 ～	「ダブルラインの設定」オプションを設定します。(0:線を作る／1:面を作る／2:線と面を作
GetDLSeparation : REAL;	12.5 ～	「ダブルラインの設定」の幅を取得します。
GetDrawingSizeRect(VAR p1X, p1Y, p2X, p2Y : REAL);	8 ～	用紙の大きさを、左上と右下の座標で返します。
GetFName : STRING;	1 ～	アクティブなドキュメントの名前を返します。
GetNumberOfDLComponents(VAR numComponents : INTEGER): BOOLEAN;	12.5 ～	「ダブルラインの設定」の構成要素の数を取得します。
GetOrigin(VAR x, y : REAL);	1 ～	現在の原点位置を座標で返します。
GetPref(prefIndex : INTEGER): BOOLEAN;	6 ～	環境設定の指定した項目の選択状態をBOOLEAN型の値で返します。
GetPrefInt(prefIndex : INTEGER): INTEGER;	8 ～	環境設定の指定した項目に設定されているINTEGER型の値を返します。
GetPrefLongInt(prefIndex : INTEGER): LONGINT;	9 ～	環境設定の指定した項目に設定されているLONGINT型の値を返します。
GetPrefReal(prefIndex : INTEGER): REAL;	9 ～	環境設定の指定した項目に設定されているREAL型の値を返します。
GetPrefRGB(prefIndex : INTEGER; VAR colorRV, colorGV, colorBV : LONGINT);	2009 ～	指定した項目の色成分を返します。
GetPrefString(prefIndex : INTEGER): STRING;	9 ～	環境設定の指定した項目に設定されているSTRING型の値を返します。
GetWallPrefStyle : STRING;	12 ～	アクティブなドキュメントの壁スタイル名を返します。
GridLines(gridDistance : REAL);	1 ～	グリッドの幅を設定します。

Document Settings

InsertNewDLComponent(beforeIndex : INTEGER; widthDistance : REAL; fill : LONGINT; penWeightLeft, penWeightRight, penStyleLeft, penStyleRight : INTEGER): BOOLEAN;	12.5 ～	「ダブルラインの設定」の、番号で指定した構成要素の前に新しい構成要素を挿入します。
PenGrid(gridDistance : REAL);	1 ～	スナップグリッドの幅を設定します。
SetConstrain(str : STRING);	1 ～	スナップパレットのON／OFFを設定します。
SetDimStd(whichStandard : INTEGER);	4 ～	デフォルトの寸法線の種類を変更します。
SetDLComponentClass(index : INTEGER; componentClass : LONGINT): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素のクラスを設定します。
SetDLComponentFill(index : INTEGER; fill : LONGINT): BOOLEAN;	12.5 ～	「ダブルラインの設定」の、番号で指定した構成要素の面を模様番号で設定します。
SetDLComponentFillColors(index, fillForeColor, fillBackColor : INTEGER): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の面の色と地色を設定します。
SetDLComponentName(index : INTEGER; componentName : STRING): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の名前を設定します。
SetDLComponentPenColors(index, leftPenForeColor, leftPenBackColor, rightPenForeColor, rightPenBackColor : INTEGER): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の左右の線の色と地色を設定します。
SetDLComponentPenStyles(index, penStyleLeft, penStyleRight : INTEGER): BOOLEAN;	12.5 ～	「ダブルラインの設定」の、番号で指定した構成要素の左右の線の種類を設定します。
SetDLComponentPenWeights(index, penWeightLeft, penWeightRight : INTEGER): BOOLEAN;	12.5 ～	「ダブルラインの設定」の、番号で指定した構成要素の左右の線の太さを設定します。
SetDLComponentUseFillClassAttr(index : INTEGER; useClassAttr : BOOLEAN): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の面の模様に、クラス属性を設定します。

Document Settings

SetDLComponentUsePenClassAttr(index : INTEGER; leftPenUseClassAttr, rightPenUseClassAttr : BOOLEAN): BOOLEAN;	2008 ～	「ダブルラインの設定」の、番号で指定した構成要素の左右の線種に、クラス属性を設定します。
SetDLComponentWidth(index : INTEGER; widthDistance : REAL): BOOLEAN;	12.5 ～	「ダブルラインの設定」のindex番目の構成要素の厚みを設定します。
SetDLControlOffset(controlOffsetDistance : REAL);	12.5 ～	「ダブルラインの設定」の、オフセットの値を設定します。
SetDLOptions(options : INTEGER);	12.5 ～	「ダブルラインの設定」オプションを設定します。(0:線を作る／1:面を作る／2:線と面を作る)
SetDLSeparation(separationDistance : REAL);	12.5 ～	「ダブルラインの設定」の幅を設定します。
SetOrigin(x, y : REAL);	1 ～	ファイルの原点位置を移動します。
SetOriginAbsolute(xValue, yValue : REAL);	8 ～	原点の位置を絶対座標で設定します。
SetPref(index : INTEGER; status : BOOLEAN);	6 ～	環境設定の指定した項目の選択状態をBOOLEAN型の値で設定します。
SetPrefInt(index, value : INTEGER);	8 ～	環境設定の指定した項目をINTEGER型の値で設定します。
SetPrefLongInt(index : INTEGER; value : LONGINT);	9 ～	環境設定の指定した項目をLONGINT型の値で設定します。
SetPrefReal(index : INTEGER; value : REAL);	9 ～	環境設定の指定した項目をREAL型の値で設定します。
SetPrefRGB(prefIndex : INTEGER; colorRV, colorGV, colorBV : LONGINT);	2009 ～	指定した項目の色成分を設定します。
SetPrefString(index : INTEGER; value : STRING);	9 ～	環境設定の指定した項目をSTRING型の値で設定します。
SetPrimaryDim(h : HANDLE; showValue, boxText : BOOLEAN; leader, trailer : STRING; precision : LONGINT);	7 ～	ハンドルで指定した基準寸法線を設定します。
SetSecondaryDim(h : HANDLE; showValue, boxText : BOOLEAN; leader, trailer : STRING; precision : LONGINT);	7 ～	ハンドルで指定した補助寸法線を設定します。

Document Settings

SetUnits(fraction, display : LONGINT; format : INTEGER; upi : REAL; name, squareName : STRING);	1 ～	アクティブなドキュメントの単位を設定します。
SetWallPrefStyle(sysName : STRING): BOOLEAN;	12 ～	アクティブなドキュメントの壁スタイル名を設定します。

File I/O

Append(fileName : STRING);	1 ～	指定したパスを持つファイルを追加モードで開きます。ファイル内の既存データは上書きされません。
Close(fileName : STRING);	1 ～	指定したパスを持つファイルを閉じます。
ConvertHSF2PosixPath(HSFPath : DYNARRAY[] of CHAR; VAR outPosixPath : DYNARRAY[] of CHAR): BOOLEAN;	2010 ～	ファイルパスの「:」 を 「/」に変更します。Macintoshのみ有効
ConvertPosix2HSFPath(PosixPath : DYNARRAY[] of CHAR; VAR outHSFPath : DYNARRAY[] of CHAR): BOOLEAN;	2010 ～	ファイルパスの「/」 を 「:」に変更します。Macintoshのみ有効
EOF(fileName : STRING): BOOLEAN;	1 ～	開いているファイルの読み込み位置が、終端であればTRUEを返します。
EOLN(fileName : STRING): BOOLEAN;	1 ～	開いているファイルの読み込み位置が、行の終端であればTRUEを返します。
FindFileInPluginFolder(filename : STRING; VAR path : STRING): BOOLEAN;	12 ～	すべてのプラグインフォルダからファイル名を検索します。ファイルが見つかるとTRUEを返し、見つからなければFALSEを返します。見つかった場合、結果はパスパラメータに返されます。
GetFile(VAR fileName : STRING);	1 ～	ファイルを開くダイアログを表示し、ユーザーが選択したファイルのパスを返します。
GetFileInfo(filename : STRING; VAR fullReadPath, fullWritePath : STRING; VAR readFileExists, writeFileExists, locked, hasReadPermission, hasWritePermission, hasFolderPermission :	12 ～	この関数は、ファイルの属性を返します。
GetFolderPath(whichPath : INTEGER): STRING;	8 ～	指定した種類のフォルダへのパスを文字列で返します。
GetLastFileErr : INTEGER;	8.5 ～	ファイルの入出力作業でエラーが発生した場合に、エラーコードを返します。
Open(fileName : STRING);	1 ～	指定したファイルのパスを使って、ファイルを開きます。
PutFile(commentStr, defaultStr : STRING; VAR fileName : STRING);	1 ～	ファイル保存ダイアログを表示し、ユーザーが指定したファイルのパスを返します。
Read(VAR z1,z2,...,zN : ANY);	1 ～	開いているファイルからデータを読み込みます。
ReadLn(VAR z1,z2,...,zN : ANY);	1 ～	開いているファイルから1行分のデータを読み込みます。この手続きを実行後、次に読み込むべき位置を示すマーカーは、次の行の先頭になっています。
Rewrite(fileName : STRING);	1 ～	指定したファイルのパスに新しいファイルを作成します。指定したファイルのパスにファイルがすでに存在する場合は、ファイルの内容を消去して開きます。

File I/O

Space(n : INTEGER);	1 ～	指定した数のスペースを、現在書き込み中のファイルに出力します。
StdRead(VAR z1,z2,...,zN : ANY);	8 ～	開いているファイルからデータを読み込みます。
StdReadLn(VAR z1,z2,...,zN : ANY);	8 ～	開いているファイルから1行分のデータを読み込みます。この手続きを実行後、次に読み込むべき位置を示すマーカーは次行の先頭になります。ReadLnと異なり、すべてのキャラクタ(タブやスペースなどの区切り記号も含む)を読み込めます。
Tab(n : INTEGER);	1 ～	指定した数のタブを、現在書き込み中のファイルに出力します。
UseDefaultFileErrorHandling(enable : BOOLEAN);	8.5 ～	入出力エラーダイアログの表示／非表示を設定します。
Write(z1,z2,...,zN : ANY);	1 ～	開いているファイルにデータを書き込みます。
WriteLn(z1,z2,...,zN : ANY);	1 ～	開いているファイルにデータを書き込み、改行します。
WriteLnMac(z1,z2,...,zN : ANY);	9 ～	開いているファイルにデータを書き込み、改行します。ファイルはどのプラットフォームでもVectorScriptとして読み戻すことができます。
WriteMac(z1,z2,...,zN : ANY);	9 ～	開いているファイルにデータを書き込みます。ファイルはどのプラットフォームでもVectorScriptとして読み戻すことができます。

General Edit

Backward;	1 ～	選択されている図形の前後関係を、ひとつ後ろに移動させます。
FlipHor;	1 ～	選択されている図形を水平反転させます。
FlipVer;	1 ～	選択されている図形を垂直反転させます。
Forward;	1 ～	選択されている図形の前後関係を、ひとつ前に移動させます。
IsObjectFlipped(h : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した3次元図形（回転体、柱状体、多段柱状体、シンボル、ソリッド、プラグインオブジェクト）が反転されている場合はTRUEを返します。
LckObjs;	1 ～	アクティブなレイヤ上で選択されている図形をロックします。コピー、複製を行うことはできますが、その他の変更はできません。
MirrorXY3D;	5.0.2 ～	選択されている3次元図形をXY座標を軸に反転します。
MoveBack;	1 ～	選択されている図形の前後関係を、最後に移動させます。
MoveFront;	1 ～	選択されている図形の前後関係を、最前に移動させます。
ResetBBox(h : HANDLE);	1 ～	ハンドルで指定した図形の領域を再計算します。
ResetOrientation3D;	5.0.2 ～	3次元表示のパラメータをリセットします。
Rotate(rotationAngle : REAL);	1 ～	選択されている図形を、中心を基点として回転させます。
Rotate3D(xAngle, yAngle, zAngle : REAL);	1 ～	直前に作成された3次元図形を、指定した角度で回転させます。
RotatePoint(pX, pY : REAL; rotationAngle : REAL);	6 ～	選択されている図形を、指定した座標を基点として回転させます。
Scale(scaleXR, scaleYR : REAL);	1 ～	アクティブなレイヤ上の選択されている図形を拡大／縮小します。
UnLckObjs;	1 ～	アクティブなレイヤ上で選択されている図形のロックを解除します。

Graphic Calculation

Distance(x1, y1, x2, y2 : REAL): REAL;	1 ～	指定した2点間の距離を返します。
EllipseEllipseIntersect(upperLeft1, lowerRight1, upperLeft2, lowerRight2 : POINT; VAR int1, int2, int3, int4 : POINT): INTEGER;	10 ～	指定した2つの楕円の交点数と座標を計算します。
EqualPt(p1X, p1Y, p2X, p2Y : REAL): BOOLEAN;	1 ～	指定した2つの座標が同じならばTRUEを返します。
EqualRect(rectAp1X, rectAp1Y, rectAp2X, rectAp2Y, rectBp1X, rectBp1Y, rectBp2X, rectBp2Y : REAL): BOOLEAN;	1 ～	指定した2つの四角形が同じならばTRUEを返します。
HCenter(h : HANDLE; VAR pX, pY : REAL);	1 ～	ハンドルで指定した図形の中心の座標を返します。ほとんどの図形の場合、バウンダリボックスの中心を返します。円、円弧、円弧壁の場合、図形の中心を返します。
LineEllipseIntersect(a1, a2, upperRight, lowerLeft : POINT; VAR int1 : POINT; VAR legal1 : BOOLEAN; VAR int2 : POINT; VAR legal2 : BOOLEAN);	10 ～	線分と楕円の交点を返します。
LineLineIntersection(l1start, l1end, l2start, l2end : POINT; VAR parallel, intOnLines : BOOLEAN; VAR sectpt : POINT);	10 ～	2つの線分の交点座標を返します。
PtInPoly(pX, pY : REAL; h : HANDLE): BOOLEAN;	1 ～	ハンドルで指定した多角形／曲線の内側に、指定した座標が入っていればTRUEを返します。
PtInRect(pointX, pointY, rect1X, rect1Y, rect2X, rect2Y : REAL): BOOLEAN;	1 ～	指定した四角形の座標の内側に、指定した座標が入っていればTRUEを返します。
Split2DObjectByLine(objectHd : HANDLE; p1X, p1Y, p2X, p2Y : REAL; VAR listHds : HANDLE);	12 ～	ハンドルで指定した図形を2点で定義される直線で分割します。分割された図形はlistHdsハンドルでアクセスします。
SrndArea(pX, pY : REAL): REAL;	1 ～	選択された多角形の中の、指定した測定点のある部分の面積を返します。
UnionRect(p1X, p1Y, p2X, p2Y, p3X, p3Y, p4X, p4Y : REAL; VAR p5X, p5Y, p6X, p6Y : REAL);	1 ～	指定した2つの四角形が重なった部分の四角形の座標を返します。

Hatches / Vector Fills

AddVectorFillLayer(xStart, yStart, xRepeat, yRepeat, xOffset, yOffset, dashFactor : REAL; lineWeight, colorIndex : INTEGER);	7.0.1 ～	ハッチングに線を追加します。
BeginVectorFillN(VAR vectorFillName : STRING; pageSpace, rotateInWall : BOOLEAN; colorIndex : INTEGER);	8 ～	新しいハッチングを作成します。パラメータで与えたハッチングの名前がすでに存在した場合、名前を変更して返します。
CreateStaticHatch(inHatchName : STRING; pX, pY : REAL; rotationAngle : REAL): HANDLE;	10.1 ～	選択している図形に名前で指定したハッチングを、挿入する位置と角度を指定して貼り付けます。ハッチングメニューとほとんど同じ機能をもっています。
CreateStaticHatchFromObject(inObj : HANDLE; inHatchName : STRING; pX, pY : REAL; rotationAngle : REAL): HANDLE;	10.5 ～	ハンドルで指定した図形に指定した名前のハッチングを、挿入する位置と角度を指定して貼り付けます。
DelVectorFill(vectorFillName : STRING);	7.0.1 ～	指定したハッチングを削除します。
EndVectorFill;	7.0.1 ～	ハッチングの作成を終了します。
GetVectorFill(theObj : HANDLE; VAR hatchName : STRING): BOOLEAN;	7.0.1 ～	ハンドルで指定した図形のハッチングの名前を返します。
GetVectorFillDefault(VAR vectorFillName : STRING): BOOLEAN;	7.0.1 ～	デフォルトのハッチングの名前を返します。
NumVectorFills : LONGINT;	7.0.1 ～	アクティブなドキュメント内のハッチングの数を返します。;
SetVectorFill(theObj : HANDLE; hatchName : STRING): BOOLEAN;	7.0.1 ～	ハンドルで指定した図形にハッチングを設定します。
SetVectorFillDefault(VAR vectorFillName : STRING): BOOLEAN;	7.0.1 ～	デフォルトのハッチングを変更します。
VectorFillList(index : LONGINT): STRING;	7.0.1 ～	指定した索引番号をもつハッチングの名前を返します。

Hatches / Vector Fills

ActLayer : HANDLE;	1 ～	アクティブなレイヤのハンドルを返します。
CopyMode(mode : INTEGER);	1 ～	アクティブなレイヤの表示モードを設定します。表示モード(1:ペイント／2:オーバーレイ／3:インバート／4:イレーズ／5:反ペイント／6:反オーバーレイ／7:反インバート／8:反
CreateLayer(layerName : STRING; layerType : INTEGER): HANDLE;	10.5 ～	指定した種類のレイヤを作成します。レイヤの種類(1:デザインレイヤ／2:シートレイヤ)
DisplayLayerScaleDialog;	12 ～	縮尺ダイアログを表示します。
Flayer : HANDLE;	1 ～	最初のレイヤのハンドルを返します。
GetLayer(h : HANDLE): HANDLE;	1 ～	ハンドルで指定した図形が存在するレイヤのハンドルを返します。
GetLayerByName(layerName : STRING): HANDLE;	8.5 ～	指定したレイヤのハンドルを返します。
GetLayerElevation(h : HANDLE; VAR baseElev, thickness : REAL);	10 ～	ハンドルで指定したレイヤの高さ(Z)と厚み(ΔZ)を返します。
GetLayerOptions : INTEGER;	8.5 ～	アクティブなドキュメントの、他のレイヤの表示方法を値で返します。
GetLayerRenderMode(theLayer : HANDLE): INTEGER;	10 ～	ハンドルで指定したレイヤのレンダリングモードを返します。
GetLName(h : HANDLE): STRING;	1 ～	ハンドルで指定したレイヤの名前を返します。
GetLScale(h : HANDLE): REAL;	1 ～	ハンドルで指定したレイヤの縮尺を返します。
GetLVis(h : HANDLE): INTEGER;	1 ～	ハンドルで指定したレイヤの表示状態を返します。レイヤの表示状態(0:表示／1:グレイ表示／2:非表示)
GetSheetLayerUserOrigin(layerHandle : HANDLE; VAR xOrigin, yOrigin : REAL): BOOLEAN;	10.5 ～	指定したシートレイヤの原点を返します。
GetZVals(VAR zVal, deltaZVal : REAL);	5 ～	アクティブなレイヤの高さ(Z)と厚み(ΔZ)を返します。
GrayLayer;	1 ～	アクティブなレイヤをグレイ表示にします。
HideLayer;	1 ～	アクティブなレイヤを隠します。
IsLayerReferenced(layer : HANDLE; VAR pathname : STRING): BOOLEAN;	10 ～	ハンドルで指定したレイヤがファイル共有で参照されているレイヤの場合は、参照元ドキュメントへのパスを返します。
Layer(name : STRING);	1 ～	新しいレイヤを指定した名前で作成します。指定した名前のレイヤがすでに存在する場合、そのレイヤをアクティブにします。

Layers

LayerRef(layerName : STRING);	4 ～	指定したレイヤとアクティブなレイヤをリンクさせます。
LFillBack(colorR, colorG, colorB : LONGINT);	1 ～	アクティブなレイヤの面の地色を設定します。値の範囲は0から65535までです。
LFillFore(colorR, colorG, colorB : LONGINT);	1 ～	アクティブなレイヤの面の色を設定します。値の範囲は0から65535までです。
Llayer : HANDLE;	1 ～	最下位のレイヤのハンドルを返します。
LPenBack(colorR, colorG, colorB : LONGINT);	1 ～	アクティブなレイヤの線の地色を設定します。値の範囲は0から65535までです。
LPenFore(colorR, colorG, colorB : LONGINT);	1 ～	アクティブなレイヤの線の色を設定します。値の範囲は0から65535までです。
NumLayers : INTEGER;	1 ～	アクティブなドキュメント内のレイヤの数を返します。
NumObj(h : HANDLE): LONGINT;	1 ～	ハンドルで指定したレイヤ上の図形の数を返します。
SetLayerElevation(h : HANDLE; baseElev, thickness : REAL);	10 ～	ハンドルで指定したレイヤの高さ(Z)と厚み(ΔZ)を設定します。
SetLayerOptions(layerOpts : INTEGER);	8.5 ～	他のレイヤの表示状態を設定します。
SetLayerRenderMode(theLayer : HANDLE; newRenderMode : INTEGER; immediate, doProgress : BOOLEAN);	10 ～	ハンドルで指定したレイヤのレンダリングモードを設定します。
SetLayerTransparency(transparency : REAL);	12 ～	アクティブデザインレイヤの不透明度を設定します。
SetLScale(h : HANDLE; scale : REAL);	6 ～	ハンドルで指定したレイヤの縮尺を変更します。
SetScale(actualSize : REAL);	1 ～	アクティブなレイヤの縮尺を設定します。
SetSheetLayerUserOrigin(layerHandle : HANDLE; xOrigin, yOrigin : REAL): BOOLEAN;	10.5 ～	指定したシートレイヤの原点を設定します。
SetZVals(zDistance, deltaZDistance : REAL);	5 ～	アクティブなレイヤの高さ(Z)と厚み(ΔZ)を設定します。
ShowLayer;	1 ～	アクティブなレイヤを表示します。
Abs(v : REAL): REAL;	1 ～	指定した数値(REAL、LONGINT、INTEGER)の絶対値を返します。

Math - General

ArcCos(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）のアークコサインの値を返します。
ArcSin(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）のアークサインの値を返します。
ArcTan(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）のアークタンジェントの値を返します。
Cos(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）のコサインの値を返します。
Deg2Rad(degreeValue : REAL): REAL;	1 ～	指定したデグリー値（度数）をラジアン値に変換して返します。
Exp(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）の指数を返します。
Ln(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）の自然対数を返します。
Rad2Deg(radianValue : REAL): REAL;	1 ～	指定したラジアン値をデグリー値（度数）に変換して返します。
Random : REAL;	8 ～	0.0から1.0までの乱数を実数で返します。
Round(v : REAL): LONGINT;	1 ～	指定した実数を四捨五入して、LONGINT型の整数に変換して返します。
Sin(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）のサインの値を返します。
Sqr(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）の2乗の値を返します。REAL型の数値を与えると、REAL型で返します。LONGINT型の数値を与えると、LONGINT型で返します。INTEGER型の数値を与えると、INTEGER型またはLONGINT型で返します。
Sqrt(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）の平方根を返します。
Tan(v : REAL): REAL;	1 ～	指定した数値（REAL、LONGINT、INTEGER）のタンジェントの値を返します。
Trunc(v : REAL): LONGINT;	1 ～	指定した実数の小数点以下を切り捨て、LONGINT型の整数に変換して返します。

Math - Vectors

Ang2Vec(angleR, Length : REAL): VECTOR;	1 ～	指定した角度と距離を2次元のベクトルに変換して返します。
AngBVec(v1, v2 : VECTOR): REAL;	1 ～	2つのベクトルがなす角 (0?180度) を返します。
Comp(v1, v2 : VECTOR; VAR v3, v4 : VECTOR);	1 ～	v3にはv1のv2方向成分、v4にはv1のv2に直角方向の成分を返します。
CrossProduct(v1, v2 : VECTOR): VECTOR;	8.5 ～	指定した2つのベクトルの外積を返します。
DotProduct(v1, v2 : VECTOR): REAL;	8 ～	指定した2つのベクトルの内積を返します。
Norm(Vec : VECTOR): REAL;	1 ～	指定したベクトルの長さを返します。
Perp(Vec : VECTOR): VECTOR;	1 ～	指定したベクトルの垂直ベクトルを返します。
UnitVec(Vect : VECTOR): VECTOR;	1 ～	指定したベクトルの標準単位を返します。
Vec2Ang(Vect : VECTOR): REAL;	1 ～	指定したベクトルの角度を返します。

Object Attributes

GetClass(h : HANDLE): STRING;	1 ～	ハンドルで指定した図形のクラスの名前を返します。
GetEntityMatrix(objectHandle : HANDLE; VAR offsetX, offsetY, offsetZ : REAL; VAR rotationXAngle, rotationYAngle, rotationZAngle : REAL): BOOLEAN;	2011 ～	プレイナー図形の平面マトリックスを取得します。
GetFillBack(h : HANDLE; VAR red, green, blue : LONGINT);	6 ～	ハンドルで指定した図形の面の地色の成分を返します。値の範囲は0から65535までです。
GetFillFore(h : HANDLE; VAR red, green, blue : LONGINT);	6 ～	ハンドルで指定した図形の面の色成分を返します。値の範囲は0から65535までです。
GetFillIAxisEndPoint(objectHandle : HANDLE; VAR xIAxisEndPoint, yIAxisEndPoint : REAL);	10 ～	I軸の終点座標を返します。
GetFillJAxisEndPoint(objectHandle : HANDLE; VAR xJAxisEndPoint, yJAxisEndPoint : REAL);	10 ～	J軸の終点座標を返します。
GetFillOriginPoint(objectHandle : HANDLE; VAR xOriginPoint, yOriginPoint : REAL);	10 ～	原点座標を返します。
GetFillPoints(objectHandle : HANDLE; VAR xOriginPoint, yOriginPoint, xIAxisEndPoint, yIAxisEndPoint, xJAxisEndPoint, yJAxisEndPoint : REAL);	10 ～	原点やI、J軸の終点座標を返します。
GetFPat(h : HANDLE): LONGINT;	1 ～	ハンドルで指定した図形の面の模様番号を返します。
GetLS(h : HANDLE): INTEGER;	1 ～	ハンドルで指定した図形の線の種類を返します。
GetLW(h : HANDLE): INTEGER;	1 ～	ハンドルで指定した図形の線の太さを返します。
GetMarker(h : HANDLE; VAR start, end : BOOLEAN; VAR style : INTEGER; VAR size : REAL);	10 ～ 12.5	ハンドルで指定した図形のマーカー情報を返します。GetArrowHeadsの代わりの関数です。サイズは、マーカーの1インチの1/16384 です。

Object Attributes

GetObjArrow(obj : HANDLE; VAR style : INTEGER; VAR size : REAL; VAR angle : INTEGER; VAR start, end : BOOLEAN);	10 ～ 12.5	図形のマーカースタイルを返します(この関数は廃止されています。代わりにGetObjBeginningMarkerまたはGetObjEndMarkerを使用してください。)。
GetObjBeginningMarker(object : HANDLE; VAR style : LONGINT; VAR angle : INTEGER; VAR size, width : REAL; VAR thicknessBasis : INTEGER; VAR thickness : REAL; VAR visibility : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した図形の始点マーカーの全設定値を返します。
GetObjEndMarker(object : HANDLE; VAR style : LONGINT; VAR angle : INTEGER; VAR size, width : REAL; VAR thicknessBasis : INTEGER; VAR thickness : REAL; VAR visibility : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した図形の終点マーカーのすべての設定値を返します。
GetOpacity(h : HANDLE; VAR opacity : INTEGER);	2008 ～	ハンドルで指定した図形の不透明度を取得します。
GetOpacityByClass(h : HANDLE; VAR isByClass : BOOLEAN);	2008 ～	ハンドルで指定した図形がクラスの不透明度を使用しているかどうかを返します。
GetPenBack(h : HANDLE; VAR red, green, blue : LONGINT);	6 ～	ハンドルで指定した図形の線の地色の成分を返します。値の範囲は0から65535までです。
GetPenFore(h : HANDLE; VAR red, green, blue : LONGINT);	6 ～	ハンドルで指定した図形の線の色成分を返します。値の範囲は0から65535までです。
GetViewMatrix(objectHandle : HANDLE; VAR offsetX, offsetY, offsetZ : REAL; VAR rotationXAng, rotationYAng, rotationZAng : REAL): BOOLEAN;	10.5 ～	レイヤやビューポート図形の見る方向を返します。
IsFillColorByClass(h : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した図形の面の色が、クラス属性を使用していた場合はTRUEを返します。

Object Attributes

IsFPatByClass(h : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した図形の面の模様が、クラス属性を使用していた場合はTRUEを返します。
IsLSByClass(h : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した図形の線の種類が、クラス属性を使用していた場合はTRUEを返します。
IsLWByClass(h : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した図形の線の太さが、クラス属性を使用していた場合はTRUEを返します。
IsMarkerByClass(h : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した図形のマーカーの種類が、クラス属性を使用していた場合はTRUEを返します。
IsPenColorByClass(h : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した図形の線の色が、クラス属性を使用していた場合はTRUEを返します。
SetClass(h : HANDLE; class : STRING);	1 ～	ハンドルで指定した図形のクラスを変更します。
SetEntityMatrix(objectHandle :HANDLE; offsetX, offsetY, offsetZ : REAL; rotationXAngle, rotationYAngle, rotationZAngle : REAL): BOOLEAN;	2011 ～	プレイナー図形の平面ﾏﾄﾘｯｸｽを設定します。 すでに同じﾏﾄﾘｯｸｽを持つ平面が書類中にあれば、ｵﾌﾞｼﾞｪｸﾄはその平面に設定されます。 そうでなければ、書類に新しい平面が設定されます。
SetFillBack(h : HANDLE; colorR, colorG, colorB : LONGINT);	1 ～	ハンドルで指定した図形の面の地色を、指定した色に変更します。値の範囲は0から65535までです。
SetFillColorByClass(h : HANDLE);	8 ～	ハンドルで指定した図形の面の色に、現在のクラス属性を使います。
SetFillFore(h : HANDLE; colorR, colorG, colorB : LONGINT);	1 ～	ハンドルで指定した図形の面の色を、指定した色に変更します。値の範囲は0から65535までです。
SetFillIAxisEndPoint(objectHandle : HANDLE; xIAxisEndPoint, yIAxisEndPoint : REAL);	10 ～	I軸の終点座標を設定します。
SetFillJAxisEndPoint(objectHandle : HANDLE; xJAxisEndPoint, yJAxisEndPoint : REAL);	10 ～	J軸の終点座標を設定します。
SetFillOriginPoint(objectHandle : HANDLE; xOriginPoint, yOriginPoint : REAL);	10 ～	原点座標を設定します。
SetFPat(h : HANDLE; fillPattern : LONGINT);	1 ～	ハンドルで指定した図形の面の模様を、指定した模様番号に変更します。
SetFPatByClass(h : HANDLE);	8 ～	ハンドルで指定した図形の面の模様に、現在のクラス属性を使います。

Object Attributes

SetLS(h : HANDLE; ls : INTEGER);	1 ～	ハンドルで指定した図形の線の模様／種類を変更します。
SetLSByClass(h : HANDLE);	8 ～	ハンドルで指定した図形の線の種類に、現在のクラス属性を使います。
SetLW(h : HANDLE; lw : INTEGER);	1 ～	ハンドルで指定した図形の線の太さを変更します。
SetLWByClass(h : HANDLE);	8 ～	ハンドルで指定した図形の線の太さに、現在のクラス属性を使います。
SetMarker(h : HANDLE; start, end : BOOLEAN; style : INTEGER; size : REAL);	10 ～ 12.5	ハンドルで指定した図形にマーカーを設定します。（この関数は廃止されています。代わりに、SetObjBeginningMarkerまたはSetObjEndMarkerを使用してください。）
SetMarkerByClass(h : HANDLE);	8 ～	ハンドルで指定した図形のマーカーの種類に、現在のクラス属性を使います。
SetObjArrow(obj : HANDLE; style : INTEGER; size : REAL; angle : INTEGER; start, end : BOOLEAN);	10 ～ 12.5	ハンドルで指定した図形にマーカーを設定します。（この関数は廃止されています。代わりに、SetObjBeginningMarkerまたはSetObjEndMarkerを使用してください。）
SetObjBeginningMarker(object : HANDLE; style : LONGINT; angle : INTEGER; size, width : REAL; thicknessBasis : INTEGER; thickness : REAL; visibility : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した図形の始点マーカーのすべての設定値を設定します。
SetObjEndMarker(object : HANDLE; style : LONGINT; angle : INTEGER; size, width : REAL; thicknessBasis : INTEGER; thickness : REAL; visibility : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した図形の終点マーカーの全設定値を設定します。正常終了するとTRUEを返します。
SetOpacity(h : HANDLE; opacity : INTEGER);	2008 ～	ハンドルで指定した図形の不透明度を設定します。

Object Attributes

SetOpacityByClass(h : HANDLE);	2008 ～	ハンドルで指定した図形の不透明度に、現在のクラス属性を使います。
SetPenBack(h : HANDLE; colorR, colorG, colorB : LONGINT);	1 ～	ハンドルで指定した図形の線の地色を、指定した色に変更します。値の範囲は0から65535までです。
SetPenColorByClass(h : HANDLE);	8 ～	ハンドルで指定した図形の線の色に、現在のクラス属性を使います。
SetPenFore(h : HANDLE; colorR, colorG, colorB : LONGINT);	1 ～	ハンドルで指定した図形の線の色を、指定した色に変更します。値の範囲は0から65535までです。
SetViewMatrix(objectHandle : HANDLE; offsetX, offsetY, offsetZ : REAL; rotationXAng, rotationYAng, rotationZAng : REAL): BOOLEAN;	10.5 ～	レイヤやビューポート図形の見る方向を設定します。
UpdateThumbnailPreview(resourceHandle : HANDLE): BOOLEAN;	11 ～	Vectorworksのリソース（ハッチング、テクスチャ、シンボル、プラグインオブジェクトなど）のサムネイルプレビューを作成、または更新します。

Object Editing

BeginMultipleDuplicate;	2011 ～	この手続きは EndMultipleDuplicate と共に使用して、複製された図形の拘束を保持しま
CreateDuplicateObject(objectToDuplicate, containerHandle : HANDLE): HANDLE;	12.5 ～	指定した図形を複製し、新しく作成された図形をハンドルで指定したコンテナに挿入します。コンテナのハンドルがNILならば、新しく作成された図形をアクティブなコンテナに挿入します。
DeleteObjs;	1 ～	アクティブなレイヤ上の選択された図形を削除します。
DeleteSymbolDefinition(hSymDef : HANDLE; bCompletely : BOOLEAN);	2011 ～	ドキュメントから参照中のシンボル定義を削除します。 bCompletely が TRUE ならば、すべてのシンボルを削除します。 FALSE ならば、ローカルなシンボルに置き換えます。
DelObject(h : HANDLE);	1 ～	ハンドルで指定した図形、レイヤ、ワークシートを削除します。
Duplicate(offsetDX, offsetDY : REAL);	1 ～	選択されている図形を複製し、指定した位置に移動させます。「環境設定」で「ずれを伴う複製」にチェックが入っていないと、複製した図形は移動しません。
EndMultipleDuplicate;	2011 ～	この手続きを BeginMultipleDuplicate と共に使用して、複製された図形の拘束を保持し
HDuplicate(objectHandle : HANDLE; x, y : REAL): HANDLE;	10 ～	ハンドルで指定した図形を複製し、指定した距離で動かします。
HMove(h : HANDLE; xOffset, yOffset : REAL);	1 ～	ハンドルで指定した図形を移動します。
HMoveBackward(h : HANDLE; toBack : BOOLEAN);	8.5 ～	ハンドルで指定した図形を、前後関係の後ろへ移動します。
HMoveForward(h : HANDLE; toFront : BOOLEAN);	8.5 ～	ハンドルで指定した図形を、前後関係の前へ移動します。
HRotate(h : HANDLE; centerX, centerY : REAL; rotationAngle : REAL);	6 ～	ハンドルで指定した図形を、指定した座標を基点として回転させます。
Move3D(xDistance, yDistance, zDistance : REAL);	1 ～	直前に作成された3次元図形を、相対的な量で移動します。
Move3DObj(h : HANDLE; xDistance, yDistance, zDistance : REAL);	1 ～	ハンドルで指定した3次元図形を、相対的な量で移動します。
MoveObjs(moveDX, moveDY : REAL; allLayers, allObjects : BOOLEAN);	1 ～	選択されている図形を、相対的な量で移動します。
SetBBox(h : HANDLE; p1X, p1Y, p2X, p2Y : REAL);	1 ～	ハンドルで指定した図形の領域を、指定した座標で設定します。
SetHDef(oldH, newH : HANDLE);	6 ～	ワークシートやシンボル、リンクされたレイヤで参照されている図形を置き換えます。

Object Events

GetEvent : LONGINT;	2011 ～	
SetCntrlPtVis(inCustomObj : HANDLE; inContrlPtIndex : INTEGER; inIsVisible : BOOLEAN);	2011 ～	
SetObjPropCharVS(PropertyID : LONGINT; PropertyVal : CHAR): BOOLEAN;	2011 ～	
SetObjPropDoubleVS(PropertyID : LONGINT; PropertyVal : REAL): BOOLEAN;	2011 ～	
SetObjPropDoubleVS(PropertyID :LONGINT; PropertyVal :REAL): BOOLEAN;	2011 ～	
vsoAddParamWidget(widgetID : LONGINT; paramName, locName : STRING): BOOLEAN;	2011 ～	オブジェクト情報パレットに表示するパラメータの、ウィジェットを追加します。
vsoAppendParamWidget(parameterID : LONGINT; text : STRING; data : LONGINT): BOOLEAN;	2011 ～	オブジェクト情報パレットの、最後に表示するウィジェットを設定します。
vsoAppendWidget(widgetType, mappingID : LONGINT; text : STRING; data : LONGINT):BOOLEAN;	2011 ～	オブジェクト情報パレットの、最後に表示するウィジェットを設定します。
vsoGetEventInfo(VAR outObjEvent, outEventData : LONGINT);	2011 ～	オブジェクトイベントに関する、コンプリートなメッセージ情報を取得します。
vsoInsertAllParams : BOOLEAN;	2011 ～	プラグインオブジェクトの全てのパラメータを、オブジェクト情報パレットに表示するウィジェットとして、挿入します。
vsoInsertParamWidget(position, parameterID : LONGINT; text : STRING; data : LONGINT): BOOLEAN ;	2011 ～	オブジェクト情報パレットに表示するウィジェットを設定します。
vsoInsertWidget(paramID, widgetType, mappingID : LONGINT; text : STRING; data : LONGINT):BOOLEAN;	2011 ～	オブジェクト情報パレットに表示するウィジェットを設定します。

Object Events

vsoPrmName2WidgetID(recName : STRING; paramName : STRING; VAR outWidgetID : LONGINT): BOOLEAN;	2011 ～	
VSOSetEventResult(inEventResult : LONGINT);	2011 ～	
vsoSetObjToolName(eventData : LONGINT; toolName : STRING);	2011 ～	類似オブジェクト作成のために、ツール名を設定します。 kObjOnGetToolNameイベント内で使用します。
vsoStateAddCurrent(hObj : HANDLE; message : LONGINT): LONGINT;	2011 ～	
vsoStateClear(hObj : HANDLE);	2011 ～	
vsoStateGet(hObj : HANDLE; state : INTEGER): BOOLEAN;	2011 ～	
vsoStateGetExitGroup(hObj : HANDLE; VAR outGrpType : LONGINT): BOOLEAN;	2011 ～	
vsoStateGetLayrChng(hObj : HANDLE; VAR outOldScale, outNewScale : REAL; VAR outScaleText : BOOLEAN): BOOLEAN;	2011 ～	
vsoStateGetNameChng(hObj : HANDLE; VAR outOldName, outNewName : STRING): BOOLEAN;	2011 ～	
vsoStateGetObjChng(hObj : HANDLE; VAR outChangeID : LONGINT): BOOLEAN;	2011 ～	
vsoStateGetParamChng(hObj : HANDLE; VAR outWidgID : LONGINT; VAR outPrmIdx : INTEGER; VAR outOldVal : STRING):BOOLEAN;	2011 ～	

Object Events

vsoStateGetPos(hObj :HANDLE; VAR outX, outY, outZ :REAL; VAR outIs3D :BOOLEAN):BOOLEAN;	2011 ～	
vsoStateGetRot(hObj : HANDLE; VAR outDiffAng : REAL; VAR outIs3D : BOOLEAN): BOOLEAN;	2011 ～	
vsoWidgetGetEnable(widgetID : LONGINT): BOOLEAN;	2011 ～	
vsoWidgetGetRecParam(widgetID : LONGINT): STRING;	2011 ～	
vsoWidgetGetText(widgetID : LONGINT): STRING;	2011 ～	
vsoWidgetGetVisible(widgetID : LONGINT): BOOLEAN;	2011 ～	
vsoWidgetPopupAdd(widgetID : LONGINT; id : STRING; text : STRING);	2011 ～	
vsoWidgetPopupClear(widgetID : LONGINT);	2011 ～	
vsoWidgetPopupGet(widgetID, index : LONGINT; VAR outId, outText : STRING);	2011 ～	
vsoWidgetPopupGetCnt(widgetID : LONGINT): LONGINT;	2011 ～	
vsoWidgetPopupSet(widgetID, index : LONGINT; id, text : STRING);	2011 ～	
vsoWidgetSetEnable(widgetID : LONGINT; enabled : BOOLEAN);	2011 ～	
vsoWidgetSetIndentLevel(widgetID, indentLevel : LONGINT);	2011 ～	
vsoWidgetSetText(widgetID : LONGINT; text : STRING);	2011 ～	

Object Events

vsoWidgetSetVisible(widgetID : LONGINT; visible : BOOLEAN);	2011 ~	
vstAddButtonMode(inIconID : INTEGER);	2011 ~	
vstAddPDMenuItem(group : INTEGER; item : STRING);	2011 ~	
vstAddPDMenuMode(label : STRING);	2011 ~	
vstAddRadioMode(inInitialSetting, inNumButtons : INTEGER; inRsrcID_1, inRsrcID_2, inRsrcID_3 : INTEGER; inRsrcID_4, inRsrcID_5, inRsrcID_6 : INTEGER);	2011 ~	
vstCustomProcNNA(inEvent : LONGINT; VAR outEvtResult : LONGINT; inMode : LONGINT; inDiameter, inSpacing : REAL): BOOLEAN;	2011 ~	
vstDefault2DToolDraw;	2011 ~	
vstDefault3DToolDraw;	2011 ~	
vstDrawCoordArcN(ptLeftTop, ptRghtBot : POINT; startAngle, sweepAngle : REAL);	2011 ~	
vstDrawCoordEllipse(ptLeftTop, ptRghtBot : POINT);	2011 ~	
vstDrawCoordLine(pt1, pt2 : POINT);	2011 ~	
vstDrawCoordLine3D(pt1X, pt1Y, pt1Z : REAL; pt2X, pt2Y, pt2Z : REAL);	2011 ~	
vstDrawCoordLineN(pt1, pt2 : POINT; planeRefID : LONGINT);	2011 ~	
vstDrawCoordLineN3D(pt1, pt2 : POINT; planeRefID : LONGINT);	2011 ~	
vstDrawCoordRect(ptLeftTop, ptRghtBot : POINT);	2011 ~	
vstEnableMode(inModeNumber : INTEGER; inEnable : BOOLEAN);	2011 ~	
vstGetCurrPt2D(VAR, outY : REAL);	2011 ~	

Object Events

vstGetCurrPt3D(VAR outX, outY, outZ : REAL; result : BOOLEAN);	2011 ～	
vstGetDataLong(inDataID : LONGINT; VAR outData : LONGINT; VAR result : BOOLEAN);	2011 ～	
vstGetDataReal(inDataID : LONGINT; VAR outData : REAL; VAR result : BOOLEAN);	2011 ～	
vstGetDataString(inDataID : LONGINT; VAR outData : DYNARRAY[] of CHAR; VAR result : BOOLEAN);	2011 ～	
vstGetEventInfo(VAR outAction, outMessage1, outMessage2 : LONGINT);	2011 ～	VSツールイベントパラメータを取得します。
vstGetEventResult(VAR outGetVal : LONGINT);	2011 ～	
vstGetInitObject(message1 : LONGINT): HANDLE;	2011 ～	このツールがコピーされる図形のハンドルを返します。 「類似オブジェクト作成」の一部です。 kToolInitByObjectイベント内で使用します。
vstGetModeHelpBase(VAR outTextRsrcIDBase : INTEGER);	2011 ～	
vstGetModeValue(inModeGroup : LONGINT; VAR outValue : LONGINT);	2011 ～	
vstGetPickObject : HANDLE;	2011 ～	
vstGetPt2D(inPtIndex : LONGINT; VAR outX, outY : REAL; result : BOOLEAN);	2011 ～	
vstGetPt3D(inPtIndex : LONGINT; VAR outX, outY, outZ : REAL; result : BOOLEAN);	2011 ～	
vstGetRsrcFileID(VAR outFileID : INTEGER);	2011 ～	
vstGetString(inStrListID, inStrID : INTEGER; VAR outString : STRING);	2011 ～	ツールのリソースファイルに格納された文字列にアクセスします。
vstGetToolObject : HANDLE;	2011 ～	vstSetPtBehaviorは、ツールの完全なオブジェクトを作成することができます。 この関数は、そのオブジェクトを返します。
vstNameUndoEvent(inUndoEventName : STRING);	2011 ～	

Object Events

vstNumPts(VAR outNumPts : LONGINT);	2011 ～	
vstRestoreWPHybridTool(message1 : LONGINT);	2011 ～	ハイブリッドツールであるvstSetWPHybridToolの後で、作業平面を復元します。
vstSetCursorByView;	2011 ～	ビューに適したカーソルを設定します。
vstSetCustomProc(inRoutineName : STRING);	2011 ～	
vstSetDataLong(inDataID, inDataVal : LONGINT; VAR result : BOOLEAN);	2011 ～	
vstSetDataReal(inDataID : LONGINT; inDataVal : REAL; VAR result : BOOLEAN);	2011 ～	
vstSetDataString(inDataID : LONGINT; inDataVal : DYNARRAY[] of CHAR; VAR result : BOOLEAN);	2011 ～	
vstSetEventInfo(inAction, inMessage1, inMessage1 : LONGINT; inRsrcFileID : INTEGER);	2011 ～	VSツールイベントの戻り値を設定します。
vstSetEventResult(inSetVal : LONGINT);	2011 ～	
vstSetHelpString(inHelpStr : STRING);	2011 ～	
vstSetModeHelpBase(inTextRsrcIDBase : INTEGER);	2011 ～	
vstSetPDMenuSel(group : INTEGER; selectedItem : INTEGER);	2011 ～	
vstSetPtBehavior(inStatusType : LONGINT);	2011 ～	
vstSetRsrcFile(inFileName : STRING);	2011 ～	
vstSetWPHybridTool(message1 : LONGINT);	2011 ～	ハイブリッドツールに対して、レイヤに作業平面を設定します。

Object Info

ConsolidatePlanarObjects(obj1 : HANDLE; obj2 : HANDLE): BOOLEAN;	2011 ～	2番目のプレイナー図形の平面を変更して、最初の図形の平面上に配置します。 また、図形を移動して、平面の変更がその位置に影響を与えないようにします。
Get2DPt(obj : HANDLE; index : INTEGER; VAR locX, locY : REAL);	8.5 ～	ハンドルで指定した2次元図形の頂点座標を返します。
GetBBox(h : HANDLE; VAR p1X, p1Y, p2X, p2Y : REAL);	1 ～	ハンドルで指定した図形がおさまる最小の四角形の座標を返します。
GetObjectVariableBoolean(h : HANDLE; index : INTEGER): BOOLEAN;	9 ～	ハンドルで指定した図形の論理 (BOOLEAN) 設定値を返します。
GetObjectVariableHandle(h : HANDLE; index : INTEGER): HANDLE;	12 ～	ハンドルで指定した図形のハンドル (HANDLE) 設定値を返します。
GetObjectVariableInt(h : HANDLE; index : INTEGER): INTEGER;	9 ～	ハンドルで指定した図形の整数 (INTEGER) 設定値を返します。
GetObjectVariableLongInt(h : HANDLE; index : INTEGER): LONGINT;	9 ～	ハンドルで指定した図形の整数 (LONGINT) 設定値を返します。
GetObjectVariablePoint(h : HANDLE; index : INTEGER; VAR outPX, outPY, outPZ : REAL): BOOLEAN;	2011 ～	図形の設定値を返します。 2D、3Dの点を返す設定値で利用します。
GetObjectVariableReal(h : HANDLE; index : INTEGER): REAL;	9 ～	ハンドルで指定した図形の実数 (REAL) 設定値を返します。
GetObjectVariableString(h : HANDLE; index : INTEGER): STRING;	9 ～	ハンドルで指定した図形の文字列 (STRING) 設定値を返します。
GetParent(h : HANDLE): HANDLE;	8.5 ～	ハンドルで指定した図形を内包しているコンテナのハンドルを返します。 コンテナがない場合はレイヤのハンドルを返します。

Object Info

GetPlanarRef(h : HANDLE): LONGINT;	2011 ～	指定した図形の平面参照番号を取得します。
GetSymLoc(symHd : HANDLE; VAR pX, pY : REAL);	1 ～	ハンドルで指定したシンボルやプラグインオブジェクトの挿入点の座標を返します。
GetSymRot(symHd : HANDLE): REAL;	1 ～	ハンドルで指定したシンボルの回転角度を返します。
GetType(h : HANDLE): INTEGER;	1 ～ 2010	ハンドルで指定した図形の種類を番号で返します。
GetTypeN(h : HANDLE): INTEGER;	2011 ～	ハンドルで指定した図形の種類を番号で返します。
HAngle(h : HANDLE): REAL;	1 ～	ハンドルで指定した図形の角度を返します。
HArea(h : HANDLE): REAL;	1 ～ 12.0.1	ハンドルで指定した図形の面積を返します。この関数は古いものです。代わりにObjAreaを使用してください。
HHeight(h : HANDLE): REAL;	1 ～	ハンドルで指定した図形の高さを返します。高さとは、3次元図形の高さではなく、Y方向の高さです。
HLength(h : HANDLE): REAL;	1 ～	ハンドルで指定した図形の長さを返します。
HPerim(h : HANDLE): REAL;	1 ～	ハンドルで指定した図形の周囲の長さを返します。
HWidth(h : HANDLE): REAL;	1 ～	ハンドルで指定した図形の幅を返します。
ObjArea(h : HANDLE): REAL;	12.5 ～	ハンドルで指定した図形の面積を返します。 値は現在の面積の単位になります。
SetAngle(h : HANDLE; value : REAL);	2009 ～	ハンドルで指定した図形の角度を設定します。
SetHeight(h : HANDLE; value : REAL);	2009 ～	ハンドルで指定した図形の高さを設定します。

Object Info

SetObjectVariableBoolean(h : HANDLE; index : INTEGER; status : BOOLEAN);	9 ～	ハンドルで指定した図形の論理 (BOOLEAN) 値を設定します。
SetObjectVariableHandle(h : HANDLE; index : INTEGER; value : HANDLE);	12 ～	ハンドルで指定した図形のハンドル (HANDLE) 値を設定します。
SetObjectVariableInt(h : HANDLE; index, value : INTEGER);	9 ～	ハンドルで指定した図形の整数 (INTEGER) 値を設定します。
SetObjectVariableLongInt(h : HANDLE; index : INTEGER; value : LONGINT);	9 ～	ハンドルで指定した図形の整数 (LONGINT) 値を設定します。
SetObjectVariablePoint(h : HANDLE; index : INTEGER; pX, pY, pZ : REAL):BOOLEAN;	2011 ～	図形の設定値を設定します。 2D、3Dの点の設定値で利用します。
SetObjectVariableReal(h : HANDLE; index : INTEGER; value : REAL);	9 ～	ハンドルで指定した図形の実数 (REAL) 値を設定します。
SetObjectVariableString(h : HANDLE; index : INTEGER; value : STRING);	9 ～	ハンドルで指定した図形の文字列 (STRING) を設定します。
SetPlanarRef(h : HANDLE; refID : LONGINT);	2011 ～	指定した図形の平面参照番号を設定します。
SetPlanarRefIDToGround	2011 ～	指定した図形の平面参照番号を設定します。
SetWidth(h : HANDLE; value : REAL);	2009 ～	ハンドルで指定した図形の幅を設定します。

Object Names

DelName(name : STRING);	1 ～	図形の名前をドキュメントから削除します。 図形そのものは削除されません。
GetColorName(ColorIndex : INTEGER): STRING;	2011 ～	指定した色番号の名前を返します。
GetDashStyleName(DashStyleIndex : INTEGER): STRING;	2011 ～	指定した破線番号の名前を返します。
GetName(h : HANDLE): STRING;	1 ～	ハンドルで指定した図形の名前を返します。
GetObject(name : STRING): HANDLE;	1 ～	指定した名前が付いた図形のハンドルを返します。 図形が存在しない場合はNILを返します。
Index2Name(index : LONGINT): STRING;	8 ～	内部で使用している索引番号を文字列で返します。
Name2Index(name : STRING): LONGINT;	8 ～	内部で使用している文字列の索引番号を返します。
NameList(index : LONGINT): STRING;	1 ～	指定した索引番号の名前を返します。
NameNum:LONGINT;	1 ～	アクティブなドキュメント内の名前が付いている図形の数を返します。
NameObject(objName : STRING);	1 ～	直後に作成される図形やワークシートに名前を付けます。
SetColorName(ColorIndex : INTEGER; ColorName : STRING): BOOLEAN;	2011 ～	指定した色番号の名前を設定します。
SetDashStyleName(DashStyleIndex : INTEGER; DashStyleName : STRING): BOOLEAN;	2011 ～	指定した破線番号の名前を設定します。
SetName(h : HANDLE; name : STRING);	1 ～	ハンドルで指定した図形の名前を設定します。

Objects - 2D

AddHole(VAR objectToGetHole : HANDLE; holeTemplate : HANDLE): BOOLEAN;	10.1 ～	ハンドルで指定した図形で図形に穴を開けます。 実行後の図形属性は曲線になります。
AddSurface(s1, s2 : HANDLE): HANDLE;	8.5 ～	ハンドルで指定した2つの図形を貼り合わせて図形を作成します。 貼り合わせて作成された図形のハンドルを返します。
Arc(p1X, p1Y, p2X, p2Y : REAL; StartAngle, ArcAngle : REAL);	1 ～	指定した座標の四角形に内接する円弧を作成します。
ArcByCenter(x, y, radius : REAL; startAngl, sweepAngle : REAL);	10 ～	中心点、半径、始発角度、円弧角から円弧を作成します。
ClipSurface(s1, s2 : HANDLE);	8.5 ～	ハンドルで指定した図形で図形を切断します。
CombineIntoSurface(ptX, ptY : REAL): HANDLE;	10.1 ～	選択している図形から、指定した点を囲む曲線を作成します。
CreateRWBackground(imageResource : HANDLE): HANDLE;	2009 ～	ハンドルで指定したイメージリソースを使用して、背景テクスチャリソースを作成します。 幅と高さはデフォルト値 (ページサイズとの相対サイズ) に設定されます。
GetArc(h : HANDLE; VAR startAngleR, arcAngleR : REAL);	1 ～	ハンドルで指定した円弧の開始角度と円弧の角度を返します。
GetLocPt(h : HANDLE; VAR pX, pY : REAL);	1 ～	ハンドルで指定した基準点の座標を返します。
GetRRDiam(h : HANDLE; VAR xDiam, yDiam : REAL);	1 ～	ハンドルで指定した隅の丸い四角形の、隅の丸みの直径を返します。
GetSegPt1(h : HANDLE; VAR pX, pY : REAL);	1 ～	ハンドルで指定した直線の始点の座標を返します。
GetSegPt2(h : HANDLE; VAR pX, pY : REAL);	1 ～	ハンドルで指定した直線の終点の座標を返します。
IntersectSurface(s1, s2 : HANDLE): HANDLE;	8.5 ～	ハンドルで指定した図形で図形を抜き取ります。 ただし、グループ図形やシンボル図形にはこの機能は使えません。
Line(lineDX, lineDY : REAL);	1 ～	現在のペン位置から、相対的に移動した座標の位置へ線を描画します。
LineTo(pX, pY : REAL);	1 ～	現在のペン位置から、絶対座標の位置へ線を描画します。
Locus(pX, pY : REAL);	1 ～	指定した座標の位置に基準点を作成します。

Objects - 2D

MakePolygon(inSourceObject : HANDLE): HANDLE;	10.1 ～	ハンドルで指定した図形から多角形を作成します。
MakePolyline(inSourceObject : HANDLE): HANDLE;	10.1 ～	ハンドルで指定した図形から曲線を作成します。
ModelPt2DToScreenPt(VAR pX, pY : REAL);	2008 ～	平面の回転でワールド座標系からスクリーン座標にポイント変えます。
ModelVecToScreenVec(VAR pX, pY : REAL);	2008 ～	平面の回転でベクトルをワールド座標系からスクリーン座標に変えます。 回転のみを考慮に入れます。
Oval(p1X, p1Y, p2X, p2Y : REAL);	1 ～	指定した座標の四角形に内接する長円を作成します。
OvalN(orginX, orginY, directionX, directionY : REAL; width, height : REAL);	2008 ～	指定した座標の四角形に内接する長円を作成します。
Rect(p1X, p1Y, p2X, p2Y : REAL);	1 ～	指定した座標の四角形を作成します。
RectangleN(orginX, orginY, directionX, directionY : REAL; width, height : REAL);	2008 ～	指定した座標の四角形を作成します。
RRect(p1X, p1Y, p2X, p2Y : REAL; DiamDX, DiamDY : REAL);	1 ～	指定した座標の四角形を、指定した直径で隅を丸く作成します。
RRectangleN(orginX, orginY, directionX, directionY : REAL; width, height, xDiam, yDiam : REAL);	2008 ～	指定した座標と直径で隅の丸い四角形を作成します。
ScreenPtToModelPt2D(VAR pX, pY : REAL);	2008 ～	平面の回転で座標をスクリーン座標からモデル座標に変換します。
ScreenVecToModelVec(VAR pX, pY : REAL);	2008 ～	平面の回転でベクトルをスクリーン座標からモデル座標に変換します。 回転のみを考慮します。
SetArc(h : HANDLE; startAngle, arcAngle : REAL);	1 ～	ハンドルで指定した円弧の開始角度と円弧の角度を変更します。
SetSegPt1(h : HANDLE; pX, pY : REAL);	1 ～	ハンドルで指定した直線の始点の座標を、指定した座標に変更します。
SetSegPt2(h : HANDLE; pX, pY : REAL);	1 ～	ハンドルで指定した直線の終点の座標を、指定した座標に変更します。

Objects - 3D

Add3DPt(pX, pY, pZ : REAL);	1 ～	BeginPoly3DとEndPoly3Dの間で使います。
AddVertex3D(objectHd : HANDLE; pX, pY, pZ : REAL);	6 ～	ハンドルで指定した3D多角形に頂点を追加します。
BeginMesh;	1 ～	この手続きを実行した後、EndMeshが実行されるまでに作成された図形をもとにメッシュ図形を作成します。
BeginMXtrd(startDistance, endDistance : REAL);	1 ～	この手続きを実行した後、EndMXtrdが実行されるまでに作成された2次元図形をもとに多段柱状体を作成します。
BeginPoly3D;	1 ～	この手続きを実行した後、EndPoly3Dが実行されるまでに追加された頂点をもとに3D多角形を作成します。
BeginSweep(startAngle, arcAngle, incAngle : REAL; pitchDistance : REAL);	1 ～	この手続きが実行した後、EndSweepが実行されるまでに作成された2次元図形をもとに回転体を作成します。
BeginXtrd(startDistance, endDistance : REAL);	1 ～	この手続きを実行した後、EndXtrdが実行されるまでに作成された2次元図形を底面とする柱状体を作成します。
Centroid3D(object : HANDLE; VAR xCG, yCG, zCG : REAL): BOOLEAN;	10.1 ～	ハンドルで指定した3次元図形の質量中心の座標を返します。 計算できた場合は、TRUEを返します。
ConvertTo3DPolys(original : HANDLE): HANDLE;	10 ～	ハンドルで指定した図形を3D多角形に変換します。
CreateContourCurves(inSourceObject : HANDLE; delta : REAL; ptOnPlaneX, ptOnPlaneY, ptOnPlaneZ, normalX, normalY, normalZ : REAL	10.1 ～	ハンドルで指定したソリッド図形に、指定した間隔で等高線を作成します。
CreateExtrudeAlongPath(pathHandle, profileHandle : HANDLE): HANDLE;	9 ～	上面と側面の図形から3Dパス図形を作成します。 作成できた場合は、そのハンドルを返します。
CreateImageProp(propName : STRING; textureRef : LONGINT; height, width : REAL; enforceImageAspectRatio, crossedPlanes, createPlugin, autoRotate, createSymbol : BOOLEAN): HANDLE;	11.5 ～	オプションを指定して添景を作成します。
CreateTaperedExtrude(profileH : HANDLE; angle, height : REAL): HANDLE;	9 ～	錐状体を作成します。
EndMesh;	1 ～	BeginMeshを実行した後、この手続きが実行されるまでに作成された図形をもとにメッシュ図形を作成します。

Objects - 3D

EndMXtrd;	1 ～	BeginMXtrdを実行した後、この手続きが実行されるまでに作成された2次元図形をもとに多段柱状体を作成します。
EndPoly3D;	1 ～	BeginPoly3Dを実行した後、この手続きが実行されるまでに追加された頂点をもとに3D多角形を作成します。
EndSweep;	1 ～	BeginSweepを実行した後、この手続きが実行されるまでに作成された2次元図形をもとに回転体を作成します。
EndXtrd;	1 ～	BeginXtrdが実行された後、この手続きが実行されるまでに作成された2次元図形を底面とする柱状体を作成します。
Flip3DObj(h : HANDLE; horiz : BOOLEAN);	2010 ～	3D図形をX軸かY軸で反転させます。
Get3DCntr(h : HANDLE; VAR pX, pY : REAL; VAR zValue : REAL);	1 ～	ハンドルで指定した3次元図形の中心の座標を返します。
Get3DInfo(h : HANDLE; VAR height, width, depth : REAL);	1 ～	ハンドルで指定した3次元図形の高さ、幅、奥行きを返します。
Get3DOrientation(h : HANDLE; VAR xRot, yRot, zRot : REAL; VAR isMirroredXY : BOOLEAN): BOOLEAN;	8 ～	ハンドルで指定した3次元図形の回転角度を返します。
GetLocus3D(h : HANDLE; VAR pX, pY, pZ : REAL);	6 ～	ハンドルで指定した3D基準点の座標を返します。
GetPolyPt3D(objectHd : HANDLE; index : INTEGER; VAR pX, pY, pZ : REAL);	7 ～	ハンドルで指定した3D多角形の指定した頂点番号の座標を返します。
HExtrude(objectH : HANDLE; bottom, top : REAL): HANDLE;	10 ～	柱状体を作成します。
Locus3D(pX, pY, pZ : REAL);	6 ～	座標を指定して3D基準点を作成します。
MeshToGroup(meshObj : HANDLE): HANDLE;	10 ～	ハンドルで指定したメッシュ図形を3D多角形で構成されたグループ図形に変換します。
Moments3D(object : HANDLE; VAR lxx, lyy, lzz : REAL): BOOLEAN;	10.1 ～	図形の質量中心を通る軸での慣性モーメントを返します。
Poly3D(x1,y1,z1,...,xn,yn,zn : REAL);	1 ～	座標を指定して3D多角形を作成します。

Objects - 3D

Products3D(object : HANDLE; VAR lxy, lyz, lzx : REAL): BOOLEAN;	10.1 ～	図形の質量中心を通る平面での慣性乗数を返します。
Set3DInfo(h : HANDLE; heightDistance, widthDistance, depthDistance : REAL);	1 ～	ハンドルで指定した3次元図形の高さ、幅、奥行きを変更します。
Set3DRot(h : HANDLE; xAngle, yAngle, zAngle : REAL; xDistance, yDistance, zDistance : REAL);	1 ～	ハンドルで指定した3次元図形を、指定した回転軸を中心に回転させます。
SetPolyPt3D(objectHd : HANDLE; index : INTEGER; pX, pY : REAL; zValue : REAL);	7 ～	ハンドルで指定した3D多角形の指定した頂点番号の座標を変更します。
SetRot3D(h : HANDLE; xAngle, yAngle, zAngle : REAL; xDistance, yDistance, zDistance : REAL);	1 ～	ハンドルで指定した3次元図形を、指定した角度と軸で回転させます。

Objects - Architectural

BeginColumn(columnDistance : REAL);	7 ～	柱の高さを設定します。この手続きを実行してから、EndGroupが実行されるまでの間に作成された2次元図形を底面とする柱を作成します。
BeginFloor(thicknessDistance : REAL);	4 ～	床の厚みを設定します。この手続きを実行してから、EndObjectが実行されるまでの間に作成された2次元図形から床を作成します。
CreateSlab(profile : HANDLE): HANDLE;	2011 ～	スラブを作成します。
CreateSlabStyle(slabStyleName : STRING): HANDLE;	2011 ～	指定した名前の新しいスラブスタイルを作成します。 すでに名前が使われている場合は次に可能な名前となります。
ConvertToUnstyledSlab(slab:HANDLE);	2011 ～	スラブスタイルを「なし」に設定します。
DeleteAllComponents(object : HANDLE): BOOLEAN;	12.5 ～	ハンドルで指定した壁または壁スタイルの全構成要素を削除します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
DeleteComponent(object : HANDLE; componentIndex : INTEGER): BOOLEAN;	12 ～	壁の中心線を番号を指定して削除します。
GetComponentAutoBoundEdgeOffset(object : HANDLE; componentIndex : INTEGER; VAR autoBoundEdgeOffset : INTEGER): BOOLEAN;	2011 ～	図形の構成要素の自動設定された境界の辺のオフセットを取得します。
GetComponentClass(object : HANDLE; componentIndex : INTEGER; VAR componentClass : LONGINT): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素からクラスを取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentFill(object : HANDLE; componentIndex : INTEGER; VAR fill : LONGINT): BOOLEAN;	12 ～	壁の中心線の線の模様を返します。成功した場合はTRUEを返します。
GetComponentFillColors(object :HANDLE; componentIndex :INTEGER; VAR fillForeColor :INTEGER; VAR fillBackColor :INTEGER): BOOLEAN;	12.5 ～	ハンドルで指定した壁の、指定した番号の構成要素から面の色と地色を取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentFollowBottomWallPeaks(object : HANDLE; componentIndex : INTEGER; VAR followBottomWallPeaks : BOOLEAN): BOOLEAN;	2011 ～	図形の構成要素が壁の下端に端を合わせるかどうかのフラグを取得します。

Objects - Architectural

GetComponentFollowTopWallPeaks(object : HANDLE; componentIndex : INTEGER; VAR followTopWallPeaks : BOOLEAN): BOOLEAN;	**2011 ～**	図形の構成要素が壁の上端に端を合わせるかどうかのフラグを取得します。
GetComponentManualEdgeOffset(object : HANDLE; componentIndex : INTEGER; VAR manualEdgeOffset : REAL): BOOLEAN;	**2011 ～**	図形の構成要素の手動設定された境界の辺のオフセットを取得します。
GetComponentName(object : HANDLE; componentIndex : INTEGER): STRING;	**2008 ～**	ハンドルで指定した壁の、指定した番号の構成要素から名前を取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentNetArea(object : HANDLE; componentIndex : INTEGER): REAL;	**2011 ～**	図形の構成要素の正味の面積を取得します。
GetComponentNetVolume(object : HANDLE; componentIndex : INTEGER): REAL;	**2011 ～**	図形の構成要素の正味の体積を取得します。
GetComponentPenColors(object : HANDLE; componentIndex : INTEGER; VAR leftPenForeColor, leftPenBackColor : INTEGER; VAR rightPenForeColor, rightPenBackColor : INTEGER): BOOLEAN;	**2008 ～**	ハンドルで指定した壁の、指定した番号の構成要素から線の色を取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentPenStyles(object : HANDLE; componentIndex : INTEGER; VAR leftPenStyle, rightPenStyle : INTEGER): BOOLEAN;	**12 ～**	ハンドルで指定した壁で、指定した番号の構成要素から左右の線の種類を取得します。 ハンドルがNILならば、デフォルト壁スタイルの値を取得します。
GetComponentPenWeights(object :HANDLE; componentIndex : INTEGER; VAR leftPenWeight, rightPenWeight : INTEGER): BOOLEAN;	**12 ～**	ハンドルで指定した壁で、指定した番号の構成要素から左右の線の太さを取得します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
GetComponentTexture(object : HANDLE; componentIndex : INTEGER; VAR texture : LONGINT): BOOLEAN;	**2011 ～**	図形の構成要素のテクスチャを取得します。

Objects - Architectural

GetComponentUseFillClassAttr(object : HANDLE; componentIndex : INTEGER; VAR useFillClassAttributes : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素で面の模様で、クラス属性を使用しているかどうかを返します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentUsePenClassAttr(object : HANDLE; componentIndex : INTEGER; VAR useLeftPenClassAttributes, useRightPenClassAttributes : BOOLEAN	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素で左右の線種で、クラス属性を使用しているかどうかを返します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentWallBottomOffset(object : HANDLE; componentIndex : INTEGER; VAR offsetFromWallBottom : REAL): BOOLEAN;	2011 ～	図形の構成要素の壁の下端からのオフセットを取得します。
GetComponentWallTopOffset(object : HANDLE; componentIndex : INTEGER; VAR offsetFromWallTop : REAL): BOOLEAN;	2011 ～	図形の構成要素の壁の上端からのオフセットを取得します。
GetComponentWidth(object : HANDLE; componentIndex : INTEGER; VAR width : REAL): BOOLEAN;	12 ～	ハンドルで指定した壁で、指定した番号の構成要素から幅を取得します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
GetCoreWallComponent(object : HANDLE): INTEGER;	2010 ～	壁の構成要素の中のコア指定されている番号を返します。
GetDatumSlabComponent(object : HANDLE): INTEGER;	2011 ～	図形の構成要素のスラブ基準面を取得します。
GetNumberOfComponents(object : HANDLE; VAR numComponents : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁の構成要素の数を取得します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
GetSlabHeight(slab : HANDLE): REAL;	2011 ～	スラブの高さを取得します。
GetSlabPreferences : HANDLE;	2011 ～	スラブ設定を取得します。 これは構成要素の関数やスタイル選択に使えます。
GetSlabPreferencesStyle : LONGINT;	2011 ～	スラブ設定のスラブスタイルを取得します。
GetSlabStyle(slab : HANDLE): LONGINT;	2011 ～	スラブのスラブスタイルを取得します。

Objects - Architectural

GetWallPreferences : HANDLE;	2011 ～	壁設定を取得します。 これは構成要素の関数やスタイル選択に使えます。
InsertNewComponent(object : HANDLE; beforeComponentIndex : INTEGER; width : REAL; fill : LONGINT; leftPenWeight, rightPenWeight :INTEGER; leftPenStyle, rightPenStyle : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁で、指定した番号の構成要素の前に、新しい構成要素を追加します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
ModifySlab(slab, modifier : HANDLE; isClipObject : BOOLEAN; componentFlags : LONGINT): BOOLEAN;	2011 ～	スラブの貼り合わせ、切り欠きをします。
SetComponentAutoBoundEdgeOffset(object : HANDLE; componentIndex, autoBoundEdgeOffset : INTEGER): BOOLEAN;	2011 ～	図形の構成要素の自動設定された境界の辺のオフセットを設定します。
SetComponentClass(object : HANDLE; componentIndex : INTEGER; componentClass : LONGINT): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素のクラスを設定します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentFill(object : HANDLE; componentIndex : INTEGER; fill : LONGINT): BOOLEAN;	12 ～	ハンドルで指定した壁で、指定した番号の構成要素の面を模様番号で設定します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
SetComponentFillColors(object : HANDLE; componentIndex : INTEGER; fillForeColor, fillBackColor : INTEGER): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の面の色と地色を設定します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentFollowBottomWallPeaks(object : HANDLE; componentIndex : INTEGER; followBottomWallPeaks : BOOLEAN): BOOLEAN;	2011 ～	図形の構成要素が壁の下端に端を合わせるかどうかのフラグを設定します。
SetComponentFollowTopWallPeaks(object : HANDLE; componentIndex : INTEGER; followTopWallPeaks : BOOLEAN): BOOLEAN;	2011 ～	図形の構成要素が壁の上端に端を合わせるかどうかのフラグを設定します。

Objects - Architectural

SetComponentManualEdgeOffset(object : HANDLE; componentIndex : INTEGER; manualEdgeOffset : REAL): BOOLEAN;	2011 ～	図形の構成要素の手動設定された境界の辺のオフセットを設定します。
SetComponentName(object : HANDLE; componentIndex : INTEGER; componentName : STRING): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の名前を設定します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentPenColors(object :HANDLE; componentIndex :INTEGER; leftPenForeColor, leftPenBackColor :INTEGER; rightPenForeColor, rightPenBackColor : INTEGER): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の線の色を設定します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentPenStyles(object : HANDLE; componentIndex : INTEGER; leftPenStyle, rightPenStyle : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁で、指定した番号の構成要素の左右の線の種類を設定します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
SetComponentPenWeights(object :HANDLE; componentIndex : INTEGER; leftPenWeight, rightPenWeight : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁で、指定した番号の構成要素の左右の線の太さを設定します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
SetComponentTexture(object : HANDLE; componentIndex : INTEGER; texture : LONGINT): BOOLEAN;	2011 ～	図形の構成要素のテクスチャを設定します。
SetComponentUseFillClassAttr(object : HANDLE; componentIndex : INTEGER; useFillClassAttributes : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の面の模様に、クラス属性を設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentUsePenClassAttr(object : HANDLE; componentIndex : INTEGER; useLeftPenClassAttributes, useRightPenClassAttributes : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の構成要素の左右の線種に、クラス属性を設定します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。

Objects - Architectural

SetComponentWallBottomOffset(object : HANDLE; componentIndex : INTEGER; offsetFromWallBottom : REAL): BOOLEAN;	**2011 ～**	図形の構成要素の壁の下端からのオフセットを設定します。
SetComponentWallTopOffset(object : HANDLE; componentIndex : INTEGER; offsetFromWallTop : REAL): BOOLEAN;	**2011 ～**	図形の構成要素の壁の上端からのオフセットを設定します。
SetComponentWidth(object : HANDLE; componentIndex : INTEGER; width : REAL): BOOLEAN;	**12 ～**	ハンドルで指定した壁で、指定した番号の構成要素の幅を設定します。 ハンドルがNILならば、デフォルト壁スタイルに作用します。
SetCoreWallComponent(object : HANDLE; coreWallComponent : INTEGER);	**2010 ～**	壁、円弧壁、壁スタイル、壁設定のコアを設定する。
SetDatumSlabComponent(object : HANDLE; datumSlabComponent : INTEGER);	**2011 ～**	図形の構成要素のスラブ基準面を設定します。
SetSlabHeight(slab : HANDLE; height : REAL);	**2011 ～**	スラブの高さを設定します。
SetSlabPreferencesStyle(slabStyle : LONGINT);	**2011 ～**	スラブ設定のスラブスタイルを設定します。
SetSlabStyle(slab : HANDLE; slabStyle : LONGINT);	**2011 ～**	スラブのスラブスタイルを設定します。
SlabFromPoly(poly : HANDLE);	**2011 ～**	ハンドルで指定した曲線からスラブを作成します。新しいスラブには現在のスラブ設定が適用されます。

Objects - Custom

CreateCustomObject(objectName : STRING; pX, pY : REAL; rotationAngle : REAL): HANDLE;	8.5 ～	位置と角度を指定してプラグインオブジェクトを配置し、そのハンドルを返します。
CreateCustomObjectN(objectName : STRING; pX, pY : REAL; rotationAngle : REAL; showPref : BOOLEAN): HANDLE;	10 ～	位置と角度を指定してプラグインオブジェクトを作成し、そのハンドルを返します。 また、設定ダイアログの表示／非表示を設定できます。
CreateCustomObjectPath(objectName : STRING; path, profileGroup : HANDLE): HANDLE;	8.5 ～	パスを用いたカスタムオブジェクトを作成し、そのハンドルを返します。
EnableParameter(inPlugin : HANDLE; inParameterName : STRING; inSetEnabled : BOOLEAN);	10 ～	プラグインのパラメータに連動している編集可能なアイテムの動作を設定します。
GetCustomObjectChoice(objectName, parameterName : STRING; choiceIndex : INTEGER): STRING;	8 ～	プラグイン、パラメータの名前、メニュー項目番号からメニュー項目の名前を返します。 この関数はプラグインオブジェクトのコマンドの中でのみ使用できます。
GetCustomObjectColor(objectHand : HANDLE; inTagID : INTEGER; VAR outColorIndex : INTEGER): BOOLEAN;	2008 ～	SetCustomObjectColorで設定され「objectHand」に設定されている色を返します。 色はinTagIDに保持されます。
GetCustomObjectInfo(VAR objectName : STRING; VAR objectHand, recordHand, wallHand : HANDLE): BOOLEAN;	8 ～	プラグインオブジェクトの情報を返します。 この関数はプラグインオブジェクトのコマンドの中でのみ使用できます。
GetCustomObjectPath(objectHand : HANDLE): HANDLE;	8.5 ～	ハンドルで指定したプラグインオブジェクトのパス図形のハンドルを返します。
GetCustomObjectProfileGroup(objectHand : HANDLE): HANDLE;	8.5 ～	ハンドルで指定したプラグインオブジェクトの輪郭図形のハンドルを返します。
GetCustomObjectSelectionGroup(objectHand : HANDLE): HANDLE;	2009 ～	グループのハンドルを返します。このグループには、パラメータで指定したプラグインオブジェクトが選択された際に、強調表示する形状が含まれています。
GetCustomObjectWallHoleGroup(objectHand : HANDLE): HANDLE;	2009 ～	グループのハンドルを返します。このグループには、パラメータで指定したプラグインオブジェクトが壁に挿入された際に、壁に作られる開口部を定義するする形状が含まれています。

Objects - Custom

GetLocalizedPluginChoice(inPluginName, inParameterName : STRING; inChoiceIndex : INTEGER; VAR outChoice : STRING): BOOLEAN;	10 ～	ローカライズ後の名前を返します。プラグイン、パラメータの名前にはユニバーサルな名前を入れます。
GetLocalizedPluginName(inPluginName : STRING; VAR outName : STRING): BOOLEAN;	10 ～	プラグインの名前からローカライズ後の名前を返します。
GetLocalizedPluginParameter(inPluginName, inParameterName : STRING; VAR outParameter : STRING): BOOLEAN;	10 ～	プラグイン、パラメータの名前からローカライズ後の名前を返します。 プラグイン、パラメータの名前にはユニバーサルな名前を入れます。
GetPluginChoiceIndex(inPluginName, inParameterName, inChoiceName : STRING; VAR outIndex : INTEGER): BOOLEAN;	10 ～	inPluginNameand inParameterNameにより指定されたoutIndexとともにTRUEを返します。 それぞれのパラメータ名にはユニバーサルな名前を入れます。
GetPluginInfo(VAR pluginName : STRING; VAR recordHand : HANDLE): BOOLEAN;	8.5 ～	実行されているプラグインの名前とレコードのハンドルを返します。
GetPluginString(stringIndex : INTEGER): STRING;	10 ～	索引番号で指定された文字を返します。
HasPlugin(itemUniversalName : STRING; VAR PaletteName : STRING): BOOLEAN;	10 ～	ツールアイテムやメニューコマンドがアクティブな作業画面に存在している場合は、そのパレットやメニューの名前とTRUEを返します。
IsNewCustomObject(objectName : STRING): BOOLEAN;	8 ～	指定した名前のプラグインオブジェクトが最初に生成された場合はTRUEを返します。 この関数はプラグインオブジェクトのコマンドの中でのみ使用できます。
IsPluginFormat(theFormat : HANDLE): BOOLEAN;	11 ～	レコードフォーマットがプラグインオブジェクト、プラグインツール、プラグインメニューのパラメータに使われている場合は、TRUEを返します。
NumCustomObjectChoices(objectName, parameterName : STRING): INTEGER;	8 ～	プラグイン、パラメータの名前からメニュー項目の数を返します。 この関数はプラグインオブジェクトのコマンドの中でのみ使用できます。
SetCustomObjectColor(objectHand : HANDLE; inTagID, inColoIndex : INTEGER): BOOLEAN;	2008 ～	「objectHand」に色を保管/設定します。 GetCustomObjectColorを使ってこの色を使用できます。 色はinTagIDに保持されます。
SetCustomObjectPath(objectHand, path : HANDLE): BOOLEAN;	8.5 ～	ハンドルで指定したプラグインオブジェクトのパス図形を設定します。

Objects - Custom

SetCustomObjectProfileGroup(objectHand, profileGroupHand : HANDLE): BOOLEAN;	8.5 ～	ハンドルで指定したプラグインオブジェクトの輪郭図形のハンドルを設定します。
SetCustomObjectSelectionGroup(objectHand, selGroup : HANDLE): BOOLEAN;	2009 ～	ハンドルで指定したプラグインオブジェクトが選択された際に、強調表示する形状を設定します。
SetCustomObjectWallHoleGroup(objectHand, holeGroup : HANDLE): BOOLEAN;	2009 ～	ハンドルで指定したプラグインオブジェクトが壁に挿入された際に、壁に作られる開口部の形状を設定します。
SetParameterVisibility(inPlugin : HANDLE; inParameterName : STRING; inSetVisible : BOOLEAN);	10 ～	プラグインのパラメータに連動している編集可能なアイテムの表示／非表示を設定します。

Objects - Groups

AddVPAnnotationObject(viewportHandle, annotationHandle : HANDLE): BOOLEAN;	11 ～	指定したビューポートに指定した注釈を追加します。
BeginGroup;	1 ～	この手続きを実行した後、EndGroupが実行されるまでの間に作成された図形をグループ化します。
BeginGroupN(VAR groupHandle:HANDLE);	2011 ～	グループのハンドルを指定し、既存のグループに図形を作成して追加します。 nilに初期化されたハンドルを渡した場合は新しいグループが作成されます。
CreateVP(parentHandle : HANDLE): HANDLE;	11 ～	ビューポートを作成します。表示されているレイヤや、表示されているレイヤの中にあるグループのハンドルをコンテナのハンドルとして渡します。
EndGroup;	1 ～	BeginGroupを実行した後、この手続きを実行するまでの間に作成された図形をグループ化します。
GetVPClassVisibility(viewportHandle : HANDLE; className : STRING; VAR visibilityType : INTEGER): BOOLEAN;	11 ～	指定したビューポートの指定したクラスの表示形式を返します。
GetVPCropObject(viewportHandle : HANDLE): HANDLE;	11 ～	指定したビューポートの枠のハンドルを返します。
GetVPGroup(viewportHandle : HANDLE; groupType : INTEGER): HANDLE;	11 ～	指定したビューポートのグループの種類を返します。 グループの種類（1:枠／2:注釈／3:キャッシュ）
GetVPGroupParent(groupHandle : HANDLE): HANDLE;	11 ～	指定したビューポートグループのコンテナである、ビューポートのハンドルを返します。
GetVPLayerVisibility(viewportHandle, layerHandle : HANDLE; VAR visibilityType : INTEGER): BOOLEAN;	11 ～	指定したビューポートの指定したレイヤの表示形式を返します。
Group;	1 ～	アクティブなレイヤ上の選択されている図形をグループ化します。
GroupToMesh(groupObj : HANDLE): HANDLE;	10 ～	ハンドルで指定したグループ図形をメッシュ図形に変換します。
HUngroup(h : HANDLE);	8.5 ～	ハンドルで指定した図形をグループ解除します。
IsVPGroupContainedObject(objectHandle : HANDLE; groupType : INTEGER): BOOLEAN;	11 ～	指定した図形がビューポートグループに含まれている／いないを返します。

Objects - Groups

SetVPClassVisibility(viewportHandle : HANDLE; className : STRING; visibilityType : INTEGER): BOOLEAN;	11 ～	指定したビューポートの指定したクラスの表示形式を設定します。
SetVPCropObject(viewportHandle, cropHandle : HANDLE): BOOLEAN;	11 ～	指定したビューポートに指定した枠を設定します。 新しい図形が枠図形として有効であれば、置き換えられます。
SetVPLayerVisibility(viewportHandle, layerHandle : HANDLE; visibilityType : INTEGER): BOOLEAN;	11 ～	指定したビューポートの指定したレイヤの表示形式を設定します。
Ungroup;	1 ～	アクティブなレイヤ上の選択されている図形をグループ解除します。
UpdateVP(viewportHandle : HANDLE);	11 ～	指定したビューポートを更新します。 ワイヤーフレームレンダリング以外は再度レンダリングが行われます。
VPHasCropObject(viewportHandle : HANDLE): BOOLEAN;	11 ～	指定したビューポートに枠が含まれている／いないを返します。

Objects - Lights

ContainsLight(containerObject : HANDLE): BOOLEAN;	8 ～	ハンドルで指定したコンテナ（グループ、シンボル、レイヤ）内に光源が含まれている場合はTRUEを返します。
CreateLight(pXR, pYR, pZR : REAL; lightType : INTEGER; isOn, castShadow : BOOLEAN): HANDLE;	7 ～	座標を指定して光源を作成し、そのハンドルを返します。
GetBeamAngle(h : HANDLE; VAR beamAngleR : REAL);	7 ～	ハンドルで指定した光源（スポットライト）の光束の角度を返します。
GetLayerAmbientColor(layer : HANDLE; VAR red, green, blue : LONGINT);	8 ～	ハンドルで指定したレイヤの背景放射光の色成分を返します。 値の範囲は0から65535までです。
GetLayerAmbientInfo(layer : HANDLE; VAR isOn : BOOLEAN; VAR brightness : INTEGER);	8 ～	ハンドルで指定したレイヤの背景放射光の情報（スイッチ、明るさ）を返します。
GetLightColorRGB(light : HANDLE; VAR red, green, blue : LONGINT);	7.0.1 ～	ハンドルで指定した光源の放射光の色成分を返します。
GetLightDirection(h : HANDLE; VAR panAngleR, tiltAngleR : REAL);	7 ～	ハンドルで指定した光源の、パン角度（パノラマ上の開き）と傾きを返します。
GetLightFalloff(light : HANDLE; VAR distFalloff, angFalloff : INTEGER);	7.0.1 ～	ハンドルで指定した光源の、距離減衰と角度減衰を返します。 距離減衰と角度減衰（0:なし／1:標準／2:スムーズ／3:シャープ）
GetLightInfo(h : HANDLE; VAR lightType, brightness : INTEGER; VAR isOn, castShadow : BOOLEAN);	7 ～	ハンドルで指定した光源の情報を返します。
GetLightLocation(h : HANDLE; VAR pX, pY, pZ : REAL);	7 ～	ハンドルで指定した光源の位置を座標で返します。
GetSpreadAngle(h : HANDLE; VAR spreadAngleR : REAL);	7 ～	ハンドルで指定した光源（スポットライト）の拡散光の角度を返します。
SetBeamAngle(h : HANDLE; beamAngleR : REAL);	7 ～	ハンドルで指定した光源（スポットライト）の光束の角度を設定します。
SetLayerAmbientColor(layer : HANDLE; red, green, blue : LONGINT);	8 ～	ハンドルで指定したレイヤの背景放射光の色成分を設定します。値の範囲は0から65535までです。

Objects - Lights

SetLayerAmbientInfo(layer : HANDLE; isOn : BOOLEAN; brightness : INTEGER);	8 ～	ハンドルで指定したレイヤの背景放射光の情報（スイッチ、明るさ）を設定します。
SetLightColorRGB(light : HANDLE; red, green, blue : LONGINT);	7.0.1 ～	ハンドルで指定した光源が放射する光の色を設定します。
SetLightDirection(h : HANDLE; panAngleR, tiltAngleR : REAL);	7 ～	ハンドルで指定した光源のパン角度（パノラマ上の開き）と傾きを設定します。
SetLightFalloff(light : HANDLE; distFalloff, angFalloff : INTEGER);	7.0.1 ～	ハンドルで指定した光源の、距離減衰と角度減衰を設定します。 距離減衰と角度減衰（0:なし／1:標準／2:スムーズ／3:シャープ）
SetLightInfo(h : HANDLE; lightType, brightness : INTEGER; isOn, castShadow : BOOLEAN);	7 ～	ハンドルで指定した光源の情報を設定します。
SetLightLocation(h : HANDLE; pX, pY : REAL; zValue : REAL);	7 ～	ハンドルで指定した光源の位置を設定します。
SetSpreadAngle(h : HANDLE; spreadAngleR : REAL);	7 ～	ハンドルで指定した光源（スポットライト）の拡散光の角度を設定します。

Objects - NURBS

ConvertToNURBS(h : HANDLE; keepOrig : BOOLEAN): HANDLE;	10 ～	ハンドルで指定した図形をNURBS図形やNURBS図形のグループに変換します。
CreateInterpolatedSurface(surfaceHandle : HANDLE; numUPts, numVPts : LONGINT; uDegree, vDegree : INTEGER): HANDLE;	10 ～	度数と頂点数を指定して補間点によるNURBS曲面を作成します。
CreateLoftSurfaces(groupCurvesHd : HANDLE; bRule, bClose, bSolid : BOOLEAN): HANDLE;	10 ～	曲線の交点のグループに補間法を用いることでNURBS曲面を作成します。
CreateNurbsCurve(firstX, firstY, firstZ : REAL; byCtrlPts : BOOLEAN; degree : INTEGER): HANDLE;	9 ～	NURBS曲線を作成します。
CreateNurbsSurface(numUPts, numVPts : LONGINT; uDegree, vDegree : INTEGER): HANDLE;	9 ～	NURBS曲面を作成します。
CreateOffsetNurbsObjectHandle(h : HANDLE; offsetDistance : REAL): HANDLE;	10 ～	ハンドルで指定したNURBS図形オフセットしてNURBS図形を新規に作成し、そのハンドルを返します。
CreateSurfacefromCurvesNetwork : BOOLEAN;	10 ～	選択されている交差した曲線の網からNURBS曲面を作成します。
DrawNurbsObject(h : HANDLE);NURBS図形を描画します。	10 ～	
EvaluateNurbsSurfacePointAndNormal(surfaceHandle : HANDLE; u, v : REAL; VAR pointX, pointY, pointZ, normalX, normalY, normalZ : REAL): BOOLEAN;	10 ～	U方向／V方向の値を指定してNURBS曲面上の頂点を決定します。
ExtendNurbsCurve(curveHandle : HANDLE; distance : REAL; bStart, bLinear : BOOLEAN): HANDLE;	10 ～	始点から終点までの距離を指定してNURBS曲線を延長します。直線状または曲率に沿って延長することができます。
ExtendNurbsSurface(surfaceHandle : HANDLE; distance : REAL; bStart, bLinear, bUDir : BOOLEAN): HANDLE;	10 ～	U方向やV方向の始点から終点までの距離を指定してNURBS曲面を延長します。

Objects - NURBS

GetNurbsObjectDistanceFromPoint(h : HANDLE; pointX, pointY : REAL; VAR distance : REAL): BOOLEAN;	10 ～	NURBS図形と点の間の距離を返します。
GetParameterOnNurbsCurve(h : HANDLE; pointX, pointY, pointZ : REAL; VAR parameter : REAL; VAR index : LONGINT): BOOLEAN;	10 ～	NURBS曲線のハンドルと点から、点を投影して確定された点のパラメータを返します。 この関数は点が投影されたNURBS曲線の番号も返します。
GetPointAndParameterOnNurbsCurveAtGivenLength(inNurbCurve : HANDLE; inPercentOfLength : REAL; VAR pX, pY, pZ : REAL; VAR outParam : REAL; VAR outIndex : LONGINT): BOOLEAN;	10.1 ～	NURBS曲線上の指定した点の位置、パラメータの位置、番号を返します。
NurbsCurveEvalPt(objectHd : HANDLE; index : LONGINT; u : REAL; VAR pX, pY, pZ : REAL);	9.5 ～	ハンドルで指定したNURBS曲線上の結び目uの座標を返します。
NurbsCurveGetNumPieces(objectHd : HANDLE): INTEGER;	9 ～	ハンドルで指定したNURBS曲線の辺の数を返します。
NurbsCurveType(objectHd : HANDLE; index : LONGINT; VAR isByFit : BOOLEAN);	9 ～	ハンドルで指定したNURBS曲線の、指定した辺の種類を返します。
NurbsDegree(objectHd : HANDLE; index : INTEGER): INTEGER;	9 ～	ハンドルで指定したNURBS曲線／曲面の、指定した辺の角度を返します。
NurbsDelVertex(objectHd : HANDLE; index1, index2 : LONGINT);	9 ～	ハンドルで指定したNURBS曲線／曲面の、指定した頂点を消去します。
NurbsGetNumPts(objectHd : HANDLE; index : LONGINT): LONGINT;	9 ～	ハンドルで指定したNURBS曲線／曲面の頂点数を返します。
NurbsGetPt3D(objectHd : HANDLE; index1, index2 : LONGINT; VAR pX, pY, pZ : REAL);	9 ～	ハンドルで指定したNURBS曲線／曲面の、指定した頂点の座標を返します。

Objects - NURBS

NurbsGetWeight(objectHd : HANDLE; index1, index2 : LONGINT; VAR weight : REAL);	9 ～	ハンドルで指定したNURBS曲線／曲面の、指定した頂点の重みを返します。
NurbsKnot(objectHd : HANDLE; index1, index2 : LONGINT; VAR knot : REAL);	9 ～	ハンドルで指定したNURBS曲線／曲面の、指定した頂点の結び目を返します。
NurbsNumKnots(objectHd : HANDLE; index : LONGINT): LONGINT;	9 ～	ハンドルで指定したNURBS曲線／曲面の結び目の数を返します。
NurbsSetKnot(objectHd : HANDLE; index1, index2 : LONGINT; knot : REAL);	9 ～	ハンドルで指定したNURBS曲線／曲面の、指定した頂点の結び目を設定します。
NurbsSetPt3D(objectHd : HANDLE; index1, index2 : LONGINT; pX, pY, pZ : REAL);	9 ～	ハンドルで指定したNURBS曲線／曲面の、指定した頂点の座標を設定します。
NurbsSetWeight(objectHd : HANDLE; index1, index2 : LONGINT; weight : REAL);	9 ～	ハンドルで指定したNURBS曲線／曲面の、指定した頂点の重みを設定します。
NurbsSurfaceEvalPt(objectHd : HANDLE; u, v : REAL; VAR pX, pY, pZ : REAL);	9.5 ～	ハンドルで指定したNURBS曲面上の結び目u,vの座標を返します。
RevolveWithRail(profileH, railH, axisH : HANDLE): HANDLE;	10 ～	断面を指定した曲線に沿って回転させて、NURBS曲面またはNURBS曲面のグループを作成します。

Objects - Polys

AddPoint(pX, pY : REAL);	1 ～	頂点を追加します。BeginPolyとEndPolyの間で使用します。
ArcTo(pX, pY : REAL; radiusDistance : REAL);	4 ～	円弧の座標を頂点として追加します。 BeginPolyとEndPolyの間で使用します。
BeginPoly;	1 ～	多角形／曲線の作成を開始します。 BeginPolyとEndPolyの間にAddPointで頂点を指定します。
ClosePoly;	1 ～	以降に作成される多角形の始点と終点を自動的に閉じます。
CurveThrough(pX, pY : REAL);	4 ～	キュービックスプラインの座標を通過する部分を頂点として追加します。 BeginPolyとEndPolyの間で使用します。
CurveTo(pX, pY : REAL);	4 ～	ベジェ曲線の座標を頂点として追加します。 BeginPolyとEndPolyの間で使用します。
DelVertex(objectHd : HANDLE; vertexNum : INTEGER);	6 ～	ハンドルで指定した多角形／曲線から、指定した番号の頂点を削除します。
EndPoly;	1 ～	多角形／曲線の作成を終了します。
GetHole(inOutsidePolyline : HANDLE; inIndex : INTEGER; VAR outHole : HANDLE): BOOLEAN;	9 ～	ハンドルで指定した曲線の、指定した番号の穴図形のハンドルを返します。
GetNumHoles(inPolyline : HANDLE; VAR outNumHoles : INTEGER): BOOLEAN;	9 ～	ハンドルで指定した曲線の穴の数を返します。 その曲線に穴が空いていない場合はFALSEを返します。
GetPolylineVertex(obj : HANDLE; vertexNum : INTEGER; VAR pX, pY : REAL; VAR vertexType : INTEGER; VAR arcRadius : REAL);	8.5 ～	ハンドルで指定した多角形／曲線の頂点の値を返します。
GetPolyPt(objectHd : HANDLE; index : INTEGER; VAR pX, pY : REAL);	1 ～	ハンドルで指定した多角形／曲線の頂点の座標を返します。
GetVertexVisibility(h : HANDLE; vertnum : INTEGER): BOOLEAN;	10 ～	ハンドルで指定した多角形／曲線の頂点の表示／非表示を返します。
GetVertNum(PolyHd : HANDLE): INTEGER;	1 ～	ハンドルで指定した多角形／曲線の頂点の数を返します。

Objects - Polys

InsertVertex(objectHandle : HANDLE; x, y : REAL; beforeVertexNum, vertexType : INTEGER; arcRadius : REAL);	10 ～	新しい頂点を多角形／曲線に挿入します。頂点の種類が0でない場合は、図形を曲線に変換します。
OpenPoly;	1 ～	以降に作成される多角形の始点と終点を開きます。
Poly(x1,y1,x2,y2,...,xn,yn : REAL);	1 ～	指定した頂点を結び多角形を作成します。 頂点のパラメータはいくつでも指定できますが、偶数でなければいけません。
GetPolylineVertex(obj : HANDLE; vertexNum : INTEGER; pX, pY : REAL; vertexType : INTEGER; arcRadiusDistance : REAL; recalcBounds : BOOLEAN);	8.5 ～	ハンドルで指定した多角形／曲線の頂点の値を設定します。
SetPolyPt(objectHd : HANDLE; index : INTEGER; xR, yR : REAL);	1 ～	ハンドルで指定した多角形／曲線の頂点の座標を設定します。
SetVertexVisibility(h : HANDLE; vertnum : INTEGER; vis : BOOLEAN);	10 ～	ハンドルで指定した多角形／曲線の頂点の表示／非表示を設定します。
Smooth(smoothType : INTEGER);	1 ～	多角形／曲線のスムージングの種類を設定します。スムージングの種類(0:なし／1:ベジェ曲線／2:キュービックスプライン／3:円弧)

Objects - Roofs

AppendRoofEdge(theRoof : HANDLE; edgePtX, edgePtY : REAL; slopeAngle : REAL; projectionDistance, eaveHeightDistance : REAL);	8 ～	ハンドルで指定した屋根に、新しい縁を追加します。
BeginRoof(p1X, p1Y, p2X, p2Y, upslopeX, upslopeY : REAL; riseDistance, runDistance : REAL; miter : INTEGER; vertPart : REAL);	4 ～	この手続きを実行してから、EndGroupが実行されるまでに作成された2次元図形を平面とする屋根を作成します。
CreateBatDormer(roofObject : HANDLE): INTEGER;	8 ～	ハンドルで指定した屋根に、波形のドーマーを追加します。
CreateGableDormer(roofObject : HANDLE): INTEGER;	8 ～	ハンドルで指定した屋根に、切り妻のドーマーを追加します。
CreateHipDormer(roofObject : HANDLE): INTEGER;	8 ～	ハンドルで指定した屋根に、寄せ棟のドーマーを追加します。
CreateRoof(genGableWall : BOOLEAN; bearingInsetDistance, roofThickDistance : REAL; miterType : INTEGER; vertMiterDistance : REAL): HANDLE;	8 ～	新しい屋根を作成し、そのハンドルを返します。
CreateShedDormer(roofObject : HANDLE): INTEGER;	8 ～	ハンドルで指定した屋根に、片流れのドーマーを追加します。
CreateSkylight(roofObject : HANDLE): INTEGER;	8 ～	ハンドルで指定した屋根に、トップライトを追加します。
CreateTrapeziumDormer(roofObject : HANDLE): INTEGER;	8 ～	ハンドルで指定した屋根に、台形のドーマーを追加します。
GetBatAttributes(roofObject : HANDLE; dormerID : INTEGER; VAR useHeight : BOOLEAN; VAR heightDepth, bottomWidth, topWidth, baseHeight : REAL; VAR controlPoint, topSlope : REAL);	8 ～	ハンドルで指定した屋根の中の、波型のドーマーの情報を返します。

Objects - Roofs

GetDormerAttributes(roofObject : HANDLE; dormerID : INTEGER; VAR edgeIndex : INTEGER; VAR cornerOffset : REAL; VAR isPerpOffset : BOOLEAN; VAR perpOrHeightOffset : REAL; VAR symName : LONGINT; VAR centerSymbol : BOOLEAN; VAR symOffset : REAL);	8 ～	ﾄﾞｰﾏｰ属性
GetDormerThick(roofObject : HANDLE; VAR wallThick, roofThick : REAL);	8 ～	ハンドルで指定したドーマーの、壁と屋根の厚みを返します。
GetGableAttributes(roofObject : HANDLE; dormerID : INTEGER; VAR useHeight : BOOLEAN; VAR heightDepth, bottomWidth : REAL; VAR overhang, leftSlope, rightSlope : REAL);	8 ～	ハンドルで指定した屋根の中の、切り妻のドーマーの情報を返します。
GetHipAttributes(roofObject : HANDLE; dormerID : INTEGER; VAR useHeight : BOOLEAN; VAR heightDepth, bottomWidth, overhang : REAL; VAR leftSlope, rightSlope, frontSlope : REAL);	8 ～	ハンドルで指定した屋根の中の、寄せ棟のドーマーの情報を返します。
GetNumRoofElements(roofObject : HANDLE): INTEGER;	8 ～	ハンドルで指定した屋根の中の、ドーマーの数を返します。
GetRoofAttributes(theRoof : HANDLE; VAR genGableWall : BOOLEAN; VAR bearingInset, roofThick : REAL; VAR miterType : INTEGER; VAR vertMiter : REAL): BOOLEAN;	8 ～	ハンドルで指定した屋根の情報を返します。
GetRoofEdge(theRoof : HANDLE; index : INTEGER; VAR vertexPtX, vertexPtY : REAL; VAR slope, projection, eaveHeight : REAL): BOOLEAN;	8 ～	ハンドルで指定した屋根の中の、番号で指定した縁の情報を返します。

Objects - Roofs

GetRoofElementType(roofObject : HANDLE; dormerID : INTEGER; VAR edgeIndex : INTEGER; VAR isDormer : BOOLEAN; VAR dormerType : INTEGER);	9 ～	ハンドルで指定した屋根の情報を返します。
GetRoofFaceAttrib(roofFace : HANDLE; VAR roofRise, roofRun : REAL; VAR miterType, holeStyle : INTEGER; VAR vertPart, thickness : REAL);	9 ～	ハンドルで指定した屋根の属性を返します。
GetRoofFaceCoords(h : HANDLE; VAR axis1X, axis1Y, axis2X, axis2Y : REAL; VAR Zaxis : REAL; VAR upslopeX, upslopeY : REAL);	9 ～	ハンドルで指定した屋根の傾きを返します。
GetRoofVertices(roofObject : HANDLE): INTEGER;	8 ～	ハンドルで指定した屋根の中の、縁の数を返します。
GetShedAttributes(roofObject : HANDLE; dormerID : INTEGER; VAR useHeight : BOOLEAN; VAR heightDepth, bottomWidth, overhang, topSlope : REAL);	8 ～	ハンドルで指定した屋根の中の、片流れのドーマーの情報を返します。
GetSkylight(roofObject : HANDLE; skylightID : INTEGER; VAR edgeIndex : INTEGER; VAR cornerOffset, perpOffset : REAL; VAR symName : LONGINT);	8 ～	ハンドルで指定した屋根の中の、トップライトの情報を返します。
GetTrapeziumAttributes(roofObject : HANDLE; dormerID : INTEGER; VAR useHeight : BOOLEAN; VAR heightDepth, bottomWidth : REAL; VAR useTopWidth : BOOLEAN; VAR topWidth, leftSlope, rightSlope, topSlope : REAL);	8 ～	屋根の中の台形のドーマーの情報を返します。
RemoveRoofEdge(roofObject : HANDLE; index : INTEGER): BOOLEAN;	8 ～	ハンドルで指定した屋根の中の、番号で指定した縁を削除します。
RemoveRoofElement(roofObject : HANDLE; id : INTEGER);	8 ～	ハンドルで指定した屋根の中の、番号で指定したドーマーを削除します。

Objects - Roofs

SetBatAttributes(roofObject : HANDLE; dormerID : INTEGER; useHeight : BOOLEAN; heightDepthValueDistance : REAL; bottomWidthDistance, topWidthDistance : REAL; baseHeightDistance, controlPointDistance : REAL; topAngle : REAL);	8 ～	屋根の中の波形のドーマーを設定します。
GetDormerAttributes(roofObject : HANDLE; dormerID, edgeIndex : INTEGER; cornerOffsetDistance : REAL; isPerpOffset : BOOLEAN; perpOrHeightOffsetDistance : REAL; symName : LONGINT; centerSymbol : BOOLEAN; symOffsetDistance : REAL);	8 ～	ﾄﾞｰﾏｰ属性を設定します。
SetDormerThick(roofObject : HANDLE; wallThickDistance, roofThickDistance : REAL);	8 ～	ハンドルで指定したドーマーの、壁と屋根の厚みを設定します。
SetGableAttributes(roofObject : HANDLE; dormerID : INTEGER; useHeight : BOOLEAN; heightDepthDistance, bottomWidthDistance : REAL; overhangDistance : REAL; leftAngle, rightAngle : REAL);	8 ～	ハンドルで指定した屋根の中の、切り妻のドーマーを設定します。
SetHipAttributes(roofObject : HANDLE; dormerID : INTEGER; useHeight : BOOLEAN; heightDepthDistance, bottomWidthDistance : REAL; overhangDistance : REAL; leftAngle, rightAngle, frontAngle : REAL);	8 ～	ハンドルで指定した屋根の中の、寄せ棟のドーマーを設定します。
SetRoofAttributes(roofObject : HANDLE; genGableWall : BOOLEAN; bearingInsetDistance, roofThickDistance : REAL; miterType : INTEGER; vertMiterDistance : REAL);	8 ～	ハンドルで指定した屋根を設定します。

Objects - Roofs

SetRoofEdge(roofObject : HANDLE; index : INTEGER; vertexPtX, vertexPtY : REAL; edgeAngle : REAL; projectionDistance, eaveHeightDistance : REAL);	8 ～	ハンドルで指定した屋根の中の、縁を設定します。
SetShedAttributes(roofObject : HANDLE; dormerID : INTEGER; useHeight : BOOLEAN; heightDepthDistance, bottomWidthDistance : REAL; overhangDistance : REAL; topAngle : REAL);	8 ～	ハンドルで指定した屋根の中の、片流れのドーマーを設定します。
SetSkylight(roofObject : HANDLE; skylightID, edgeIndex : INTEGER; cornerOffsetDistance, perpOffsetDistance : REAL; symName : LONGINT);	8 ～	ハンドルで指定した屋根の中の、トップライトを設定します。
SetTrapeziumAttributes(roofObject : HANDLE; dormerID : INTEGER; useHeight : BOOLEAN; heightDpthDistance, bottomWidthDistance : REAL; useTopWidth : BOOLEAN; topWidthDistance : REAL; leftAngle, rightAngle, topAngle : REAL);	8 ～	台形のﾄﾞｰﾏｰ属性を設定します。

Objects - Solids

AddSolid(obj1, obj2 : HANDLE; VAR newSolid : HANDLE): INTEGER;	7 ～	ハンドルで指定した2つの3次元図形を噛み合わせます。 作成された図形のハンドルはnewSolidに返されます。
CalcSurfaceArea(solidObject : HANDLE): REAL;	8 ～ 12.0.1	ハンドルで指定した3次元図形（ソリッド）の表面積を返します。 この関数は古いものです。代わりにObjSurfaceAreaを使用してください。
CalcVolume(solidObject : HANDLE): REAL;	8 ～ 12.0.1	ハンドルで指定した3次元図形（ソリッド）の体積を返します。 この関数は古いものです。代わりにObjVolumeを使用してください。
CreateCone(centerX, centerY, centerZ, tipX, tipY, tipZ : REAL; radiusDistance : REAL): HANDLE;	8 ～	円錐を作成します。
CreateHemisphere(centerX, centerY, centerZ, topX, topY, topZ : REAL): HANDLE;	8 ～	半球を作成します。
CreateSphere(centerX, centerY, centerZ : REAL; radiusDistance : REAL): HANDLE;	8 ～	球を作成します。
IntersectSolid(obj1, obj2 : HANDLE; VAR newSolid : HANDLE): INTEGER;	7 ～	ハンドルで指定した2つの3次元図形の重なった部分を残します。 作成された図形のハンドルはnewSolidに返されます。
ObjSurfaceArea(solidObject : HANDLE): REAL;	12.5 ～	ハンドルで指定した3次元図形（ソリッド）の表面積を返します。 値は現在の体積の単位になります。
ObjVolume(solidObject : HANDLE): REAL;	12.5 ～	ハンドルで指定した3次元図形（ソリッド）の体積を返します。 値は現在の体積の単位になります。
SubtractSolid(obj1, obj2 : HANDLE; VAR newSolid : HANDLE): INTEGER;	7 ～	ハンドルで指定した2つの3次元図形を削り取ります。 作成された図形のハンドルはnewSolidに返されます。

Objects - Symbols

ActSymDef:HANDLE;	1 ～	現在ﾘｿｰｽﾌﾞﾗｳｻﾞ上で選択されている登録シンボルのハンドルを返します。
BeginFolder;	1 ～	この手続きを実行した後、EndFolderを実行するまでの間に作成されたシンボルをシンボルフォルダに入れます。 シンボルフォルダフォルダの名前は、この手続きの直前にNameObjectを実行して設定します。
BeginSym(symbolName : STRING);	1 ～	この手続きを実行した後、EndSymを実行するまでの間に作成された図形をシンボルとして登録します。 シンボルの名前は、この手続きの直前にNameObjectを実行して設定します。
CopySymbol(filePath, symbol : STRING): BOOLEAN;	1 ～	他のﾄﾞｷｭﾒﾝﾄから、ｱｸﾃｨﾌﾞなﾄﾞｷｭﾒﾝﾄにシンボルを取り込みます。取り込みが成功すると、TRUEを返します。
EndFolder;	1 ～	BeginFolderを実行した後、この手続きを実行するまでの間に作成されたシンボルをシンボルフォルダに入れます。
EndSym;	1 ～	BeginSymを実行した後、この手続きを実行するまでの間に作成された図形をシンボルとして登録します。
GetSDName(h : HANDLE): STRING;	1 ～	ハンドルで指定したシンボルの名前を返します。
GetSymbolOptionsN(name : STRING; VAR insertMode, breakMode : INTEGER; VAR className : STRING);	8.5 ～	シンボルの挿入位置と壁の処理、クラスの設定を返します。
GetSymbolType(objectHandle : HANDLE): INTEGER;	10 ～	シンボルの種類を返します。(0:2Dシンボル／1:3Dシンボル／2:ハイブリッドシンボル)
GetSymBrightMult(symbol : HANDLE): INTEGER;	8 ～	ハンドルで指定したシンボルの、光源の明るさの乗数をパーセンテージで返します。
GetSymLoc3D(objectHandle : HANDLE; VAR x, y, z : REAL);	10 ～	3次元空間でのシンボルやプラグインオブジェクトの位置を返します。
GetSymName(symHd : HANDLE): STRING;	1 ～	ハンドルで指定したシンボルの名前を返します。
InsertSymbolInFolder(targetFolder, symbolDef : HANDLE);	8.5 ～	ハンドルで指定したシンボルをシンボルフォルダに入れます。
SetActSymbol(name : STRING);	1 ～	名前で指定したシンボルをアクティブにします。
SetSymbolOptionsN(name : STRING; insertMode, breakMode : INTEGER; className : STRING);	8.5 ～	シンボルの挿入位置と壁の処理、デフォルトのクラスを設定します。
SetSymBrightMult(symbol : HANDLE; brightnessMultiplier : INTEGER);	8 ～	ハンドルで指定したシンボルの、光源の明るさの乗数をパーセンテージで設定します。

Objects - Symbols

Symbol(symbolName : STRING; pX, pY : REAL; rotationAngle : REAL);	1 ～	アクティブなレイヤ上の指定した座標に、シンボルを配置します。
SymbolToGroup(h : HANDLE; convertAction : INTEGER);	10 ～	オプションを指定してシンボルをグループに変換します。 (0:下の階層の図形はグループに変換しない／1:下の階層にあるシンボルとプラグインオブジェクトをグループに変換する／2:べての図形をグループに変換する)
SymDefNum : LONGINT;	1 ～	クティブなドキュメントに登録されているシンボルの数を返します。

Objects - Text

BeginText;	1 ～	文字図形の作成を開始します。BeginTextとEndTextの間に文字列を設定します。
CreateText(theText : DYNARRAY[] of CHAR);	8 ～	現在のペン位置に文字図形を作成します。
EndText;	1 ～	文字図形の作成を終了します。
GetFontID(fontName : STRING): INTEGER;	8 ～	フォントの番号を返します。
GetFontName(fontID : INTEGER): STRING;	8 ～	フォントの名前を返します。
GetText(objectHd : HANDLE): DYNARRAY[] of CHAR;	1 ～	ハンドルで指定した文字図形の文字列を返します。
GetTextFont(objectHd : HANDLE; Position : INTEGER): INTEGER;	6 ～	ハンドルで指定した文字図形の、指定した位置の文字のフォント番号を返します。 指定した位置に文字が存在しない場合は0が返ります。
GetTextJust(TextHd : HANDLE): INTEGER;	6 ～	ハンドルで指定した文字図形の位置揃えを返します。 位置揃え(1:標準／2:左寄せ／3:中央／4:右寄せ)
GetTextLeading(theText : HANDLE): REAL;	8 ～	ハンドルで指定した文字図形の行間隔を返します。
GetTextLength(TextHd : HANDLE): INTEGER;	8 ～	ハンドルで指定した文字図形の文字の数を返します。
GetTextOrientation(theText : HANDLE; VAR textOriginX, textOriginY : REAL; VAR textAng : REAL; VAR textIsMirrored : BOOLEAN);	8 ～	ハンドルで指定した文字図形の座標と角度を返します。
GetTextSize(TextHd : HANDLE; Position : INTEGER): REAL;	6 ～	ハンドルで指定した文字図形の、指定した位置の文字の大きさを返します。
GetTextSpace(theText : HANDLE): INTEGER;	8 ～	ハンドルで指定した文字図形の行間隔を返します。 行間(2:全角／3:1.5角／4:倍角)
GetTextStyle(TextHd : HANDLE; Position : INTEGER): INTEGER;	6 ～	ハンドルで指定した文字図形の、指定した位置の文字のスタイルを返します。 指定した位置に文字が存在しない場合は0が返ります。
GetTextVerticalAlign(TextHd : HANDLE): INTEGER;	8 ～	ハンドルで指定した文字図形の垂直方向の位置揃えを返します。

Objects - Text

GetTextWidth(theText : HANDLE): REAL;	8 ～	ハンドルで指定した文字図形の幅を返します。
GetTextWrap(theText : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した文字図形でラップテキストが有効の場合はTRUEを返します。
SetText(objectHd : HANDLE; text : DYNARRAY[] of CHAR);	1 ～	ハンドルで指定した文字図形の文字列を設定します。
SetTextAdorner(textBlock, textAdorner : HANDLE; pX, pY : REAL): Boolean;	2011 ～	この関数は、指定したテキストブロックとテキスト装飾の間の関係を作成します。 テキストブロックがVPにある場合、テキスト装飾も同様にVPに位置します。 同じテキスト図形に、複数のオブジェクトを装飾できます。
SetTextFont(objectHd : HANDLE; Start, Count, FontNum : INTEGER);	6 ～	ハンドルで指定した文字図形の指定した位置から、指定した長さまでのフォントを設定します。
SetTextJust(TextHd : HANDLE; JustFlag : INTEGER);	6 ～	ハンドルで指定した文字図形の位置揃えを設定します。 位置揃え(1:標準／2:左寄せ／3:中央／4:右寄せ) ---MC6～VW9.5 位置揃え(1:標準／2:中央／3:右寄せ) ----------VW10～
SetTextJustN(TextHd : HANDLE; JustFlag : INTEGER);	2011 ～	ハンドルで指定した文字図形の位置を変えずに位置揃えを設定します。 位置揃え(1:標準／2:中央／3:右寄せ)
SetTextLeading(theText : HANDLE; leading : REAL);	8 ～	指定した文字図形の文字の行間隔を設定します。
SetTextOrientation(theText : HANDLE; textOriginX, textOriginY : REAL; textAngle : REAL; textIsMirrored : BOOLEAN);	8 ～	指定した文字図形の座標と角度を設定します。
SetTextSize(objectHd : HANDLE; Start, Count : INTEGER; Size : REAL);	6 ～	ハンドルで指定した文字図形の指定した位置から、指定した長さまでのフォントの大きさを設定します。
SetTextSpace(theText : HANDLE; spacing : INTEGER);	8 ～	ハンドルで指定した文字図形の行間隔を設定します。 行間(2:全角／3:1.5角／4:倍角)
SetTextStyle(objectHd : HANDLE; Start, Count, Style : INTEGER);	6 ～	ハンドルで指定した文字図形の指定した位置から、指定した長さまでのフォントのスタイルを設定します。
SetTextVerticalAlign(TextHd : HANDLE; verticalAlignment : INTEGER);	8 ～	ハンドルで指定した文字図形の垂直方向の位置揃えを設定します。 位置揃え(1:上／2:最上行のBL／3:中央／4:最下行のBL／5:下)

Objects - Text

SetTextVertAlignN(TextHd : HANDLE; verticalAlignment : INTEGER);	2011 ～	ハンドルで指定した文字図形の位置を変えずに垂直方向の配置を設定します。 位置揃え(1:上／2:最上行のBL／3:中央／4:最下行のBL／5:下)
SetTextWidth(theText : HANDLE; widthDistance : REAL);	8 ～	ハンドルで指定した文字図形の幅を設定します。
SetTextWrap(theText : HANDLE; wrap : BOOLEAN);	8 ～	ハンドルで指定した文字図形でラップテキストの有効／無効を設定します。
TextFace(s : TEXTSTYLE);	1 ～	文字のスタイルを設定します。入力パラメータは以下の定数名で指定します。 複数組み合わせる場合は,で繋ぎます。 文字のスタイル(Bold:太文字/Italic:斜体/Underline:下線文字/Outline:袋文字/Shadow:影文字) 例)TextFace([Bold,Italic])
TextFlip(FlipType : INTEGER);	1 ～	以降に作成する文字を垂直か水平に反転します。 反転の種類(1:水平反転／2＝垂直反転)
TextFont(fontID : INTEGER);	1 ～	文字のフォントを番号で設定します。
TextJust(justify : INTEGER);	1 ～	文字の位置揃えを設定します。 位置揃え(1:標準／2:左寄せ／3:中央／4:右寄せ)
TextLeading(leading : REAL);	8 ～	文字の行間隔を設定します。
TextOrigin(pX, pY : REAL);	1 ～	作成する文字図形の左上の座標を設定します。
TextRotate(Rotation : REAL);	1 ～	以降に作成する文字を回転させます。
TextSize(size : REAL);	1 ～	文字の大きさを設定します。
TextSpace(spacing : INTEGER);	1 ～	文字の行間隔を設定します。 文字の行間隔(2:全角／3:1.5倍角／4:2倍角)
TextVerticalAlign(verticalAlignment : INTEGER);	8 ～	垂直方向の位置揃えを設定します。

Objects - Walls

AddCavity(pair : BOOLEAN; leftOffDistance, rightOffDistance : REAL; pairFill : LONGINT);	4 ～ 11.5	壁に構成を追加します。この関数はもう使われません。 代わりにInsertNewComponentを使用してください。
AddSymToWall(wallHd : HANDLE; offDistance, heightDistance : REAL; flip, right : BOOLEAN; symbolName : STRING);	6 ～	ハンドルで指定した壁に、シンボルを挿入します。
AddSymToWallEdge(h : HANDLE; alongDistance, heightDistance : REAL; flip, right : BOOLEAN; symbolName : STRING; insertMode : INTEGER);	8 ～	ハンドルで指定した壁に、挿入モードを指定してシンボルを挿入します。 挿入モード(0:中心／1:左端／2:右端)
AddWallBottomPeak(wallHd : HANDLE; offDistance, heightDistance : REAL);	9 ～	ハンドルで指定した壁の底面に、頂点を追加します。
AddWallPeak(wallHd : HANDLE; offDistance, heightDistance : REAL);	6 ～	ハンドルで指定した壁に頂点を追加します。
BreakWall(offsetDistance, breakWidthDistance : REAL; right : BOOLEAN);	4 ～	直前に作成された壁を、指定した位置と長さで切り欠きます。
ClearCavities;	4 ～ 12.0.1	壁の中心線をすべて消去します。
ClearWallPeaks(h : HANDLE);	9 ～	ハンドルで指定した壁の頂点を、すべて消去します。
ConvertToUnstyledWall(h : HANDLE): BOOLEAN;	12 ～	壁を<unstyled>に設定します。成功した場合はTRUEを返します。
CreateWallFeature(wall, profile : HANDLE; wallFeatureType : INTEGER): HANDLE;	2010 ～	壁フィーチャーを断面形状から作成します。
CreateWallStyle(wallStyleName : STRING): HANDLE;	12.5 ～	新しい壁スタイルを指定した名前で作成します。 指定した名前の壁スタイルが存在する場合は、指定した名前に文字を追加し利用できる名前が使われます。
DeleteAllComponents(wall : HANDLE): BOOLEAN;	12.5 ～	ハンドルで指定した壁または壁スタイルの全構成要素を削除します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
DeleteComponent(wall : HANDLE; index : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁または壁スタイルの、指定した番号の構成要素を削除します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。

Objects - Walls

DeleteWallSym(symbolHd : HANDLE): BOOLEAN;	6 ～	ハンドルで指定したシンボルを、選択された壁から削除します。 削除に成功した場合はTRUEを返します。
GetComponentClass(wall : HANDLE; index : INTEGER; VAR componentClass : LONGINT): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素からクラスを取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentFill(wall : HANDLE; index : INTEGER; VAR fill : LONGINT): BOOLEAN;	12 ～	ハンドルで指定した壁の、指定した番号の構成要素から面の模様番号を取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentFillColors(wall : HANDLE; index : INTEGER; VAR fillForeColor, fillBackColor : INTEGER): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素から面の色と地色を取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentName(wall : HANDLE; index : INTEGER): STRING;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素から名前を取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentPenColors(wall : HANDLE; index : INTEGER; VAR leftPenForeColor, leftPenBackColor, rightPenForeColor, rightPenBackColor : INTEGER): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素から線の色を取得します。
GetComponentPenStyles(wall : HANDLE; index : INTEGER; VAR penStyleLeft, penStyleRight : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁の、指定した番号の構成要素から左右の線の種類を取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentPenWeights(wall : HANDLE; index : INTEGER; VAR penWeightLeft, penWeightRight : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁の、指定した番号の構成要素から左右の線の太さを取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetComponentUseFillClassAttr(wall : HANDLE; index : INTEGER; VAR useClassAttr : BOOLEAN): BOOLEAN;	2008 ～	指定した壁の構成要素で、面がクラス属性を使用しているかどうかを返します。 ハンドルがNILならば、関数は壁の設定に作用します。

Objects - Walls

GetComponentUsePenClassAttr(wall : HANDLE; index : INTEGER; VAR leftPenUseClassAttr, rightPenUseClassAttr : BOOLEAN): BOOLEAN;	2008 ～	指定した壁の構成要素で、線種がクラス属性を使用しているかどうかを返します。 ハンドルがNILならば、関数は壁の設定に作用します。
GetComponentWidth(wall : HANDLE; index : INTEGER; VAR width : REAL): BOOLEAN;	12 ～	ハンドルで指定した壁の、指定した番号の構成要素から幅を取得します。 ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetCoreWallComponent(wall : HANDLE): INTEGER;	2010 ～	壁の構成要素の中のコア指定されているIDを返します。
GetLayerDeltaZOffset(theWall : HANDLE): REAL;	12.5 ～	ハンドルで指定した壁の高さのレイヤの厚み（ΔZ）からのオフセット値を返します。
GetLinkHeightToLayerDeltaZ(theWall : HANDLE): BOOLEAN;	12.5 ～	ハンドルで指定した壁の高さがレイヤの厚み（ΔZ）にリンクしている／いないを返します。
GetNumberOfComponents(wall : HANDLE; VAR numComponents : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁の構成要素数を返します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
GetNumOfWallBreaks(wallH : HANDLE; VAR numWallBreaks : INTEGER): BOOLEAN;	2008 ～	ハンドルで指定した壁の、切断箇所数を返します。
GetNumWallPeaks(h : HANDLE): INTEGER;	9 ～	ハンドルで指定した壁の頂点の数を返します。
GetObjExtentsInWall(symH, wallH : HANDLE; VAR startPtX, startPtY, endPtX, endPtY : REAL): BOOLEAN;	2008 ～	ハンドルで指定した壁に挿入されているプラグインオブジェクトまたはシンボルの、挿入位置の始点座標と終点座標を返します。
GetWallCapAttributesType(wall : HANDLE): INTEGER;	2010 ～	壁の端の属性を返します。
GetWallCaps(theWall : HANDLE; VAR leftCap, rightCap, round : BOOLEAN);	12.5 ～	壁の端部の設定を取得します。
GetWallCapsOffsets(theWall : HANDLE; VAR leftCapLeftOffset, leftCapRightOffset, rightCapLeftOffset, rightCapRightOffset : REAL);	12.5 ～	壁の端部のオフセットを取得します。
GetWallControlOffset : REAL;	8.5 ～ 12.0.1	壁のオフセットの値を返します。

Objects - Walls

GetWallHalfBreakInfo(wallH : HANDLE; breakIndex : INTEGER; VAR startPtX, startPtY, centerPtX, centerPtY, endPtX, endPtY : REAL): BOOLEAN;	2008 ～	壁の線に沿った、壁内の半破断線の始点、中心点、終点を取得します。
GetWallPeak(h : HANDLE; index : INTEGER; VAR xPeak, yPeak, zPeak : REAL);	9 ～	ハンドルで指定した壁の頂点座標を返します。
GetWallStyle(theWall : HANDLE): STRING;	12 ～	壁のStyleの名前を返します。
GetWallThickness(h : HANDLE; VAR thicknessDist : REAL): BOOLEAN;	12 ～	壁の厚みを取得します。成功した場合はTRUEを返します。
GetWallWidth : REAL;	6 ～	デフォルトの壁の幅を返します。
HWallHeight(wallHd : HANDLE; startHeightDistance, endHeightDistance : REAL);	6 ～	ハンドルで指定した壁の始点と終点の高さを設定します。
HWallWidth(wallHd : HANDLE; widthDistance : REAL);	6 ～	ハンドルで指定した壁の幅を設定します。
InsertNewComponent(wall : HANDLE; index : INTEGER; widthDistance : REAL; fill : LONGINT; pwLeft, pwRight, psLeft, psRight : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁または壁スタイルの、指定した番号の構成要素の前に新しい構成要素を挿入します。
InsertSymbol(offsetDistance, heightDistance : REAL; flipped, right, capped : BOOLEAN; symbolName : STRING);	4 ～	壁の中の指定した位置にシンボルを配置します。
JoinWalls(firstWall, secondWall : HANDLE; firstWallX, firstWallY, secondWallX, secondWallY : REAL; joinModifier : INTEGER; capped, showAlerts : BOOLEAN): BOOLEAN;	10 ～	壁結合ツールです。 結合の種類（1:T型結合／2:L型結合／3:X型結合／4:自動結合）
MoveWallByOffset(theWall : HANDLE; VAR offset : REAL);	10 ～	ARCHITECTでのみ利用可能です。 すべての壁の連結を維持したまま、壁を指定した値で定義した線に垂直に動かします。 壁の移動の幾何学的に拘束されているので、実際に壁が動いた値を返します。

Objects - Walls

ReverseWallSides(theWall : HANDLE);	10 ～	ハンドルで指定した壁の向きを逆にすることで壁の左右を切り替えます。
RoundWall(centerPtX, centerPtY, startPtX, startPtY, endPtX, endPtY : REAL);	7 ～	壁の中の指定した位置にシンボルを配置します。
SetComponentClass(wall : HANDLE; index : INTEGER; componentClass : LONGINT): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素のクラスを設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentFill(wall : HANDLE; index : INTEGER; fill : LONGINT): BOOLEAN;	12 ～	ハンドルで指定した壁の、指定した番号の構成要素の面を模様番号で設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentFillColors(wall : HANDLE; index, fillForeColor, fillBackColor : INTEGER): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の面の色と地色を設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentName(wall : HANDLE; index : INTEGER; componentName : STRING): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の名前を設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentPenColors(wall : HANDLE; index, leftPenForeColor, leftPenBackColor, rightPenForeColor, rightPenBackColor : INTEGER): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の線の色を設定します。
SetComponentPenStyles(wall : HANDLE; index, penStyleLeft, penStyleRight : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁の、指定した番号の構成要素の左右の線の種類を設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentPenWeights(wall : HANDLE; index, penWeightLeft, penWeightRight : INTEGER): BOOLEAN;	12 ～	ハンドルで指定した壁の、指定した番号の構成要素の左右の線の太さを設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetComponentUseFillClassAttr(wall : HANDLE; index : INTEGER; useClassAttr : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の面の模様に、クラス属性を設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。

Objects - Walls

SetComponentUsePenClassAttr(wall : HANDLE; index : INTEGER; leftPenUseClassAttr, rightPenUseClassAttr : BOOLEAN): BOOLEAN;	2008 ～	ハンドルで指定した壁の、指定した番号の構成要素の構成要素の左右の線種に、クラス属性を設定します。
SetComponentWidth(wall : HANDLE; index : INTEGER; widthDistance : REAL): BOOLEAN;	12 ～	ハンドルで指定した壁の、指定した番号の構成要素から幅を設定します。ハンドルがNILならば、この関数はデフォルト壁スタイルに作用します。
SetCoreWallComponent(wall : HANDLE; coreWallComponent : INTEGER);	2010 ～	壁、円弧壁、壁スタイル、壁設定のコアを設定します。
SetLayerDeltaZOffset(theWall : HANDLE; layerDeltaZOffset : REAL): BOOLEAN;	12.5 ～	ハンドルで指定した壁の高さのレイヤの厚み（ΔZ）からのオフセット値を設定します。
SetLinkHeightToLayerDeltaZ(theWall : HANDLE; linkToLayerDeltaZ : BOOLEAN): BOOLEAN;	12.5 ～	ハンドルで指定した壁の高さをレイヤの厚み（ΔZ）にリンクする／しないを設定します。
SetObjectAsCornerBreak(objH, wallH : HANDLE; cornerBreak : BOOLEAN): BOOLEAN;	2010 ～	図形の頂点レコードのコーナー頂点フラグを設定します。
SetObjectWallOffset(objH, wallH : HANDLE; offset : REAL): BOOLEAN;	2010 ～	壁の中の図形のオフセットを設定します。
SetWallCapAttributesType(wall : HANDLE; wallCapAttributesType : INTEGER);	2010 ～	壁の端の属性を指定します。
SetWallCaps(theWall : HANDLE; leftCap, rightCap, round : BOOLEAN): BOOLEAN;	12.5 ～	壁の端部を設定します。
SetWallCapsOffsets(theWall : HANDLE; leftCapLeftDistance, leftCapRightDistance, rightCapLeftDistance, rightCapRightDistance : REAL): BOOLEAN;	12.5 ～	壁の端部のオフセットを設定します。
SetWallControlOffset(offset : REAL);	8.5 ～ 12.0.1	壁のオフセットの値を設定します。

Objects - Walls

SetWallHeights(h : HANDLE; startHtDistance, endHtDistance : REAL): BOOLEAN;	12 ～	壁の高さを設定します。
SetWallStyle(theWall : HANDLE; wallStyle : STRING; selectedOffDistance, replacingOffDistance : REAL): BOOLEAN;	12.5 ～	ハンドルで指定した壁に、壁スタイルを設定し、指定したオフセットに揃えます。
SetWallThickness(h : HANDLE; thicknessDistance : REAL): BOOLEAN;	12 ～	壁の厚みを設定します。
SetWallWidth(widthDistance : REAL);	6 ～ 11.5	ドキュメントのアクティブな壁の幅を設定します。 この関数はもう使われません。 代わりにSetWallPrefStyleを使用してください。
Wall(p1X, p1Y, p2X, p2Y : REAL);	4 ～	指定した座標に壁を作成します。
WallCap(atStart, closed, round : BOOLEA; rightOffDistance, leftOffDistance : REAL);	4 ～	壁の端の処理を設定します。
WallFootPrint(wallHandle : HANDLE): HANDLE;	10 ～	壁の設置跡を表す曲線のハンドルを返します。
WallHeight(wallHd : HANDLE; VAR startHt, endHt : REAL);	6 ～	ハンドルで指定した壁の始点と終点の高さを返します。
WallPeak(alongDistance, heightDistance : REAL);	4 ～	壁の上部に頂点を追加します。
WallTo(pX, pY : REAL);	4 ～	現在のペン位置から指定した座標まで壁を作成します。
WallWidth(wallHd : HANDLE): REAL;	6 ～	ハンドルで指定した壁の幅を返します。

ODBC

DBDocAddConn(dsn, userName, password : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	現在のドキュメントにデータベース接続を追加します。
DBDocGetColumns(database, table : STRING; VAR outNames, outTypes : DYNARRAY[] of CHAR; VAR outCanBeKey, outIsKey : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	指定されたテーブルデータのコロンで区切られたリストをストリング表示で返します。
DBDocGetConn(databaseName : DYNARRAY[] of CHAR; VAR outUserName, outPassword : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	データベース接続情報を取得します。
DBDocGetDB(VAR outDatabases : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	指定された書類に関連づけられたデータベースのコロンで区切られたリストをストリング表示で返します。
DBDocGetTables(database : STRING; VAR outTables : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	指定されたデータベースのコロンで区切られたテーブルのリストをストリング表示で返します。
DBDocHasConn : BOOLEAN;	2011 ～	現在のドキュメントにODBCデータソースの接続があるかどうかをチェックします。
DBDocRemoveConn(databaseName : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	現在のドキュメントからデータベース接続を削除します。
DBDocSetColKey(databaseName, tableName, columnName : DYNARRAY[] of CHAR; setIsKey : BOOLEAN): BOOLEAN;	2011 ～	データベースのテーブルカラムのキーを取得します。
DBGetFormatConn(formatName : STRING; VAR outDatabase, outTable : STRING): BOOLEAN;	2011 ～	指定された形式のためのODBC接続を返します。
DBGetFormatFieldConn(formatName :STRING; VAR fieldName, columnName :STRING; VAR linkType :INTEGER): BOOLEAN;	2011 ～	指定された形式フィールドのためのODBC接続を取得します。
DBObjSQLGetRead(hObject : HANDLE; VAR SQLSentence : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	ODBCを読むためのオブジェクトのSQL文を取得します。
DBObjSQLGetWrite(hObject : HANDLE; VAR SQLSentence : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	ODBCを書くためのオブジェクトのSQL文を取得します。

ODBC

DBObjSQLSetRead(hObject : HANDLE; SQLSentence : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	ODBCを読むためのオブジェクトのSQL文を設定します。
DBObjSQLSetWrite(hObject : HANDLE; SQLSentence : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	ODBCを書くためのオブジェクトのSQL文を設定します。
DBSetFormatConn(formatName, database, tableName : STRING): BOOLEAN;	2011 ～	指定された形式のためのODBC接続を設定します。
DBSetFormatFieldConn(formatName, fieldName, columnName : STRING; linkType : INTEGER): BOOLEAN;	2011 ～	指定された形式フィールドのためのODBC接続を設定します。
DBShowDBTableDlg(database, table : STRING): BOOLEAN;	2011 ～	指定したデータベースのテーブルダイアログを表示します。
DBShowManageDBsDlg;	2011 ～	データベースの管理ダイアログを表示します。
DBShowObjConnDlg : BOOLEAN;	2011 ～	選択されたオブジェクトのオブジェクト接続ダイアログを表示します。
DBSQLExecute(database : STRING; SQLQuery : DYNARRAY[] of CHAR; VAR outColumnCnt, outResultSetInst : LONGINT): BOOLEAN;	2011 ～	現在のドキュメントに関連づけられた指定されたデータベースのSQLを実行します。 注意！ resultSetInstanceは「DBSQLExecuteDelete」を使って削除します。
DBSQLExecuteDelete(VAR resultSetInst : LONGINT);	2011 ～	「DBSQLExecute」か「DBSQLExecuteDSN」に作成されたresultSetInstanceを削除します。
DBSQLExecuteDSN(dsn, userName, password, SQLQuery : DYNARRAY[] of CHAR; VAR outColumnCnt, outResultSetInst : LONGINT): BOOLEAN;	2011 ～	ODBCマネージャに登録された指定されたDSNのSQLを実行します。 注意！ resultSetInstanceは「DBSQLExecuteDelete」を使って削除します。
DBSQLExecuteGet(resultSetInst, colIndex : LONGINT; VAR outColumnName, outValue : DYNARRAY[] of CHAR): BOOLEAN;	2011 ～	「DBSQLExecute」か「DBSQLExecuteDSN」に作成されたresultSetInstanceからの情報を検索します。
DBSQLExecuteNext(resultSetInst : LONGINT): BOOLEAN;	2011 ～	現在ポインタを次のエントリーに動かします。

PlantObjectCoreTools

Plant_CreateDuplicatePlant(plantToCreateFrom : HANDLE);	2011 ～	今選択されている植栽から新しい植栽を作る。
Plant_EditPlantDefinitionViaResourceBrowser(plantToEdit : HANDLE);	2011 ～	ユーザーがリソースブラウザ上で植栽定義を編集した際に定義を更新する。
Plant_GetToolInitialized : INTEGER;	2011 ～	植栽が初期化されたときにtrueを返す。
Plant_GetToolPlacementMode : INTEGER;	2011 ～	植栽ツールの配置モードを取得する。
Plant_GetToolPlantName : PROCEDURE;	2011 ～	ツールが呼び出している植栽を取得する。
Plant_GetToolSpacing : REAL;	2011 ～	植栽ツールの配置間隔を取得する。
Plant_ReplacePlant(plantToReplace : HANDLE);	2011 ～	選択されている植栽を置き換える。
Plant_ResetPlantInstances(plantSymbolName : PROCEDURE);	2011 ～	定義の編集された植栽シンボルをすべてリセットする。
Plant_UpdatePlacePlantTool(plantToUpdateWith : HANDLE);	2011 ～	植栽がリソースブラウザ上でダブルクリックされたら植栽配置ツールを更新する。
Plant_UpdateTranslatedPlantIDRecord(newID : PROCEDURE);	2011 ～	植栽レコードを新しいIDで更新する。

Parametric Constraints

BuildConstraintModelForObject(h : HANDLE; traverseContainers : BOOLEAN);	2011 ～	拘束マネージャーの指定した図形の拘束モデルを作成します。
DeleteConstraint(obj, constraint : HANDLE);	9 ～	ハンドルで指定した図形から、指定した拘束を解除します。
GetBinaryConstraint(constrType : INTEGER; obj1, obj2 : HANDLE; obj1VertA, obj1VertB, obj2VertA, obj2VertB : INTEGER; containedObj1, containedObj2 : LONGINT): HANDLE;	9 ～	ハンドルで指定した図形間に設定されている拘束のハンドルを返します。
GetSingularConstraint(typeOfConstraint : INTEGER; obj : HANDLE; vertexA, vertexB : INTEGER): HANDLE;	9 ～	ハンドルで指定した図形に設定されている拘束のハンドルを返します。
HasConstraint(h : HANDLE): BOOLEAN;	9 ～	ハンドルで指定した図形が拘束されていた場合は、TRUEを返します。
SetBinaryConstraint(typeOfConstraint : INTEGER; h1, h2 : HANDLE; obj1VertA, obj1VertB, obj2VertA, obj2VertB : INTEGER; containedObj1, containedObj2 : LONGINT): BOOLEAN;	9 ～	2つの図形に対して拘束を設定します。
SetConstraintValue(constraint : HANDLE; value : REAL);	9 ～	拘束の値を設定します。
SetSingularConstraint(typeOfConstraint : INTEGER; h : HANDLE; vertexA, vertexB : INTEGER): BOOLEAN;	9 ～	1つの図形に対して拘束を設定します。

Selection

SelectAll;	1 ～	アクティブなレイヤ上の、すべての図形の選択状態を解除します。
NumSelectedObjects : LONGINT;	2010 ～	選択されている図形数を返します。
NumSObj(h : HANDLE): LONGINT;	1 ～	ハンドルで指定したレイヤ上の、選択されている図形の数を返します。
SelectAll;	1 ～	アクティブなレイヤ上の、すべての図形を選択します。
Selected(h : HANDLE): BOOLEAN;	1 ～	ハンドルで指定した図形の選択状態を返します。 図形が選択されている場合は、TRUEを返します。
SetDSelect(h : HANDLE);	1 ～	ハンドルで指定した図形の選択を解除します。
SetSelect(h : HANDLE);	1 ～	ハンドルで指定した図形を選択します。

SiteModel Interface Library

DTM6_ClearModelCache(hObject : HANDLE);	2011 ～	サイトモデルのキャッシュをクリアします。
DTM6_GetDTMObject(hLayer : HANDLE; bPickUpModel : BOOLEAN): HANDLE;	2011 ～	DTMを取得します。 それがドキュメントのレイヤの上の1であるだけであれば、それを返します。 多くのDTMsがあれば、1つを選ぶようにユーザに頼んでください。
DTM6_GetDTMOver(hObject : HANDLE): HANDLE;	2011 ～	指定されたオブジェクトの上にDTMを取得します。
DTM6_GetZatXY(hDTMObject : HANDLE; TINType : INTEGER; x, y : REAL; VAR outZ : REAL): BOOLEAN;	2011 ～	指定されたDTMでx、yを指定されるところの高度を取得します。 DTMが存在していないか、またはDTMの外にポイントがあるなら、FALSEが返ります。 (TINタイプ0: 現況地形 1: 計画地形 2: 計画+現況地形)
DTM6_IsDTM6Object(hDTMObject : HANDLE): BOOLEAN;	2011 ～	オブジェクトハンドルがSiteModelオブジェクトタイプであるかチェックします。
DTM6_IsObjectReady(hDTMObject : HANDLE): BOOLEAN;	2011 ～	DTMオブジェクトが使用できるかどうかチェックします。 もしFALSEが戻るなら、ハンドルに「ResetObject」を実行するべきです。
DTM6_IsTypeVisible(hDTMObject : HANDLE; TINType : INTEGER): BOOLEAN;	2011 ～	指定されたDTMタイプが目に見えるなら、TRUEが返ります。
DTM6_SendToSurface(hDTMObject : HANDLE; hObject : HANDLE; TINType : INTEGER): BOOLEAN;	2011 ～	渡されたオブジェクトをDTMの表面に載せます。

Special - QuickTime

QTCloseMovieFile(movieRef : INTEGER);	8.5 ～	指定したQuickTimeファイルを閉じます。
QTGetMovieOptions(movieRef : INTEGER; VAR frameRate : REAL; VAR keyFrameRate : LONGINT);	8.5 ～	指定したQuickTimeファイルに設定されているフレームと基本フレーム単位の情報を返します。
QTInitialize : INTEGER;	8.5 ～	QuickTimeを初期化し、QuickTimeのバージョンを返します。 0が返ってきた場合QuickTimeが有効でないことを表します。
QTOpenMovieFile(fileName : STRING): INTEGER;	8.5 ～	QuickTimeファイルを作成または開きます。
QTSetMovieOptions(movieRef : INTEGER; frameRate : REAL; keyFrameRate : LONGINT; useDlg, useDlgPreview : BOOLEAN);	8.5 ～	QuickTimeファイルのフレームと基本フレーム単位を設定します。
QTTerminate;	8.5 ～	QuickTimeが動作しないようにします。
QTWriteFrame(movieRef : INTEGER);	8.5 ～	現在の画面をQuickTimeファイルへ書き出します。

Spotlight

DBeam_Begin;	2011 ～	照射ルーチンの描画スタート
DBeam_BeginShttGet;	2011 ～	3次元シャッターオブジェクトの生成のスタート
DBeam_End;	2011 ～	照射ルーチンの描画エンド
DBeam_EndShttGet;	2011 ～	3次元シャッターオブジェクトの生成のエンド
DBeam_Get2DLines : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_Get2DLn2FOff : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_Get2DObjAtFs : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_Get2DObjFOff : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_Get3DShutter : HANDLE;	2011 ～	3次元シャッターオブジェクトのハンドルを返します。
DBeam_GetLast2DObj : HANDLE;	2011 ～	最も最近作った、2次元照明オブジェクトのハンドルを返します。
DBeam_GetLastObject : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_GetLines : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_GetLines2FOff : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_GetObjAtFocus : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_GetObjFallOff : HANDLE;	2011 ～	最も最近作った、照明オブジェクトのハンドルを返します。
DBeam_SetBeamAngle(angle : REAL);	2011 ～	
DBeam_SetBeamAngle2(angle : REAL);	2011 ～	
DBeam_SetBShutAngle(angle : REAL);	2011 ～	
DBeam_SetBShutDepth(depth : INTEGER);	2011 ～	
DBeam_SetFallOffDist(dist : REAL);	2011 ～	
DBeam_SetFocusPoint(ptX : REAL; ptY : REAL; ptZ : REAL);	2011 ～	照明器具のフォーカスポイントを設定します。
DBeam_SetLampRot(rot : REAL);	2011 ～	
DBeam_SetLightOrigin(originX : REAL; originY : REAL; originZ : REAL);	2011 ～	照明器具の原点を設定します。
DBeam_SetLShutAngle(angle : REAL);	2011 ～	
DBeam_SetLShutDepth(depth : INTEGER);	2011 ～	
DBeam_SetRShutAngle(angle : REAL);	2011 ～	
DBeam_SetRShutDepth(depth : INTEGER);	2011 ～	
DBeam_SetShow3DType(typeFlag : INTEGER);	2011 ～	

Spotlight

DBeam_SetShowAtPoint(showFlag : INTEGER);	2011 ～	
DBeam_SetTShutAngle(angle : REAL);	2011 ～	
DBeam_SetTShutDepth(depth : INTEGER);	2011 ～	
DBeam_ShowBeamLines(showFlag : BOOLEAN);	2011 ～	
LDevice_ClearCache(h : HANDLE);	2011 ～	指定された照明器具オブジェクトにあわせて描画キャッシュをきれいにする。
LDevice_ExtractCache(h : HANDLE; geometryID : INTEGER);	2011 ～	レイヤ上の指定された照明器具オブジェクトにあわせて描画キャッシュを作成する。
SL_Export(exportType : INTEGER; instHand : HANDLE; fieldName : DYNARRAY[] of CHAR);	2011 ～	スポットライトデータをXMLファイルに取り出す。
SL_Import(paramSelfHandle : HANDLE);	2011 ～	XMLファイルからスポットライトデータを取り込む。

Strings

Chr(v : INTEGER): CHAR;	1 ～	指定したASCIIコードのキャラクタを返します。 ASCIIコードの値の範囲は1から255です。
Concat(str1,..., strN : DYNARRAY[] of CHAR): DYNARRAY[] of CHAR;	1 ～	指定したすべての文字列を連結します。
Copy(source : DYNARRAY[] of CHAR; index, count : INTEGER): DYNARRAY[] of CHAR;	1 ～	指定した文字列から、指定した位置と長さの文字列を返します。
Delete(VAR source : DYNARRAY[] of CHAR; index, count : INTEGER);	1 ～	指定した文字列から、指定した位置と長さで文字列を削除します。
GetResourceString(VAR theString : STRING; id, index : INTEGER);	9 ～	ﾘｿｰｽﾌｧｲﾙから文字列を返します。
Insert(source : DYNARRAY[] of CHAR; VAR dest : DYNARRAY[] of CHAR; index : INTEGER);	1 ～	指定した文字列を、指定した位置に挿入します。
Len(v : DYNARRAY[] of CHAR): INTEGER;	1 ～	指定した文字列の文字数を返します。
Num2Str(decPlace : INTEGER; v : REAL): STRING;	1 ～	実数を文字列に変換します。
Num2StrF(vDistance : REAL): STRING;	1 ～	実数を文字列に変換します。 数値は現在の画面上の単位で表します。
Ord(v : CHAR): INTEGER;	1 ～	指定したキャラクタのASCIIコードを返します。
Pos(subStr, str : DYNARRAY[] of CHAR): INTEGER;	1 ～	指定した文字列から指定した文字を検索し、その位置を返します。 該当する文字が含まれていない場合は0が返されます。
Str2Num(s : STRING): REAL;	1 ～	文字列を実数に変換します。
UprString(VAR str : DYNARRAY[] of CHAR);	1 ～	指定した文字列をすべて大文字に変換します。

Textures

AttachDefaultTextureSpace(obj : HANDLE; partID : INTEGER);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。 SetDefaultTexMapNをご利用ください。
CreatePaintFromImage(image : HANDLE): HANDLE;	10 ～	イメージからペイントノードを作成します。
CreatePaintFromImgN(image : HANDLE; locPtX : REAL; locPtY : REAL; rotDeg : REAL): HANDLE;	2011 ～	指定された位置と回転で画像リソースからペイントノードを作成します。
CreateShaderRecord(texture : HANDLE; family, prototype : LONGINT): HANDLE;	10.1 ～	指定した属性をもつ、テクスチャを作成します。 (1:色属性／2:反射属性／3:透明属性／4:バンプ属性)
CreateTexture : HANDLE;	8 ～	新しいテクスチャを作成します。
CreateTextureBitmap : HANDLE;	8 ～ 9.5	テクスチャ用のビットマップを作成します。
CreateTextureBitmapN(shaderRecord : HANDLE): HANDLE;	10.1 ～	指定した属性で、ビットマップテクスチャを作成します。 イメージファイルを選択するダイアログが表示されます。 「キャンセル」ボタンが押されたり、属性がイメージでない場合はNILを返します。
DeleteTextureSpace(obj : HANDLE; partID : INTEGER);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。 Set/Get TexMapをご利用ください。
EditShaderRecord(shaderRecord : HANDLE): BOOLEAN;	10.1 ～	属性を設定するための編集ダイアログが表示されます。 「OK」ボタンが押された場合は、TRUEを返します。 属性番号:1＝色属性／2＝反射属性／3＝透明属性／4＝バンプ属性
EditTexture(texture : HANDLE): BOOLEAN;	8 ～	ハンドルで指定したテクスチャを編集するダイアログを表示します。
EditTextureBitmap(textureBitmap : HANDLE): BOOLEAN;	8 ～ 8.5	ハンドルで指定したビットマップテクスチャを編集するダイアログを表示します。 ダイアログでビットマップテクスチャが編集された場合は、TRUEを返します。
EditTextureSpace(obj : HANDLE; partID : INTEGER): BOOLEAN;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。
GetClTextureC(className : STRING): LONGINT;	8 ～	指定したクラスの属性で、壁の中央に設定されているテクスチャの番号を返します。
GetClTextureD(className : STRING): LONGINT;	8 ～	指定したクラスの属性で、ドーマーに設定されているテクスチャの番号を返します。

Textures

GetClTextureG(className : STRING): LONGINT;	8 ～	指定したクラスの属性に設定されているテクスチャの番号を返します。
GetClTextureL(className : STRING): LONGINT;	8 ～	指定したクラスの属性で、壁の左側に設定されているテクスチャの番号を返します。
GetClTextureR(className : STRING): LONGINT;	8 ～	指定したクラスの属性で、壁の右側に設定されているテクスチャの番号を返します。
GetClTextureT(className : STRING): LONGINT;	8 ～	指定したクラスの属性で、屋根に設定されているテクスチャの番号を返します。
GetClUseTexture(className : STRING): BOOLEAN;	8 ～	指定したクラスの属性で、テクスチャの使用が有効の場合は、TRUEを返します。
GetNumTexLayers(obj : HANDLE; texPartID : LONGINT): LONGINT;	2010 ～	テクスチャーのレイヤ数（基準+デカール）を返します。
GetObjExpandTexture(obj : HANDLE): BOOLEAN;	8 ～ 2008	ハンドルで指定した3次元図形のテクスチャが全体に貼り付けられている場合は、TRUEを返します。
GetShaderRecord(texture : HANDLE; family : LONGINT): HANDLE;	10.1 ～	指定した属性のハンドルを返します。（1:色属性／2:反射属性／3:透明属性／4:バンプ属性）
GetTexBFeatureEnd(textureBitmap : HANDLE; VAR featureEndX, featureEndY : INTEGER);	8 ～	ハンドルで指定したビットマップのノードの終点座標を返します。
GetTexBFeatureStart(textureBitmap : HANDLE; VAR featureStartX, featureStartY : INTEGER);	8 ～	ハンドルで指定したビットマップのノードの始点座標を返します。
GetTexBitFeatureSize(textureBitmap : HANDLE): REAL;	8 ～	ハンドルで指定したビットマップのノードの大きさを返します。
GetTexBitmapOrigin(textureBitmap : HANDLE; VAR originX, originY : INTEGER);	8 ～ 8.5	ハンドルで指定したビットマップのノードの原点座標を返します。
GetTexBitPaintNode(textureBitmap : HANDLE): HANDLE;	8 ～ 11.5	バージョン12では使用できません。 GetObjectVariableHandle(textureBitmap, 528)を使って、代わりにイメージノードを返します。
GetTexBitRepHoriz(textureBitmap : HANDLE): BOOLEAN;	8 ～	ハンドルで指定したビットマップで、水平方向の繰り返しの有効／無効を返します。

Textures

GetTexBitRepVert(textureBitmap : HANDLE): BOOLEAN;	8 ～	ハンドルで指定したビットマップで、垂直方向の繰り返しが有効な場合は、TRUEを返します。
GetTexMapBool(h : HANDLE; partID : LONGINT; selector : INTEGER): BOOLEAN;	2009 ～ 2009	VW2010以降はGetTexMapxxxNを使用してください。
GetTexMapBoolN(obj : HANDLE; texPartID, texLayerID : LONGINT; selector : INTEGER): BOOLEAN;	2010 ～	ハンドルで指定した図形の指定した部分番号でのマッピング情報をBOOLEAN値で返します。部分番号を3に指定すると全体になります。
GetTexMapInt(h : HANDLE; partID : LONGINT; selector : INTEGER): INTEGER;	2009 ～ 2009	VW2010以降はGetTexMapxxxNを使用してください。
GetTexMapIntN(obj : HANDLE; texPartID, texLayerID : LONGINT; selector : INTEGER): INTEGER;	2010 ～	ハンドルで指定した図形の指定した部分番号でのマッピング情報をINTEGER値で返します。部分番号を3に指定すると全体になります。
GetTexMapReal(h : HANDLE; partID : LONGINT; selector : INTEGER): REAL;	2009 ～ 2009	VW2010以降はGetTexMapxxxNを使用してください。
GetTexMapRealN(obj : HANDLE; texPartID, texLayerID : LONGINT; selector : INTEGER): REAL;	2010 ～	ハンドルで指定した図形の指定した部分番号でのマッピング情報をREAL値で返します。 部分番号を3に指定すると全体になります。
GetTexSpace2DOffset(textureSpace : HANDLE; VAR offsetU, offsetV : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpace2DRadius(textureSpace : HANDLE): REAL;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpace2DRot(textureSpace : HANDLE): REAL;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpace2DScale(textureSpace : HANDLE): REAL;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。

Textures

GetTexSpaceEndCap(textureSpace : HANDLE): BOOLEAN;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpaceKind(textureSpace : HANDLE): INTEGER;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpaceOrientU(textureSpace : HANDLE; VAR uXAxis, uYAxis, uZAxis : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpaceOrientV(textureSpace : HANDLE; VAR vXAxis, vYAxis, vZAxis : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpaceOrientW(textureSpace : HANDLE; VAR wXAxis, wYAxis, wZAxis : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpaceOrigin(textureSpace : HANDLE; VAR offsetX, offsetY, offsetZ : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpacePartID(textureSpace : HANDLE): INTEGER;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTexSpaceStartCap(textureSpace : HANDLE): BOOLEAN;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
GetTextureBitmap(shaderRecord : HANDLE): HANDLE;	8 ～	ハンドルで指定したテクスチャのビットマップを返します。
GetTextureRef(obj : HANDLE; partID : INTEGER; resolveByClass : BOOLEAN): LONGINT;	8 ～ 2009	VW2010以降はGetTextureRefNを使用してください。
GetTextureRefN(obj : HANDLE; texPartID, texLayerID : LONGINT; resolveByClass : BOOLEAN): LONGINT;	2010 ～	ハンドルで指定した3次元図形のテクスチャ番号を返します。
GetTextureSet(object : HANDLE): INTEGER;	2011 ～	オブジェクトのテクスチャーセットを取得します。
GetTextureShader(texture : HANDLE): LONGINT;	8 ～ 8.5	ハンドルで指定したテクスチャのシェーダ番号を返します。

Textures

GetTextureShininess(texture : HANDLE): INTEGER;	8 ～ 8.5	ハンドルで指定したテクスチャの反射率をパーセンテージで返します。
GetTextureSize(texture : HANDLE): REAL;	10.1 ～	テクスチャのサイズを実寸で返します。
GetTextureSpace(obj : HANDLE; partID : INTEGER): HANDLE;	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。 Set/Get TexMapをご利用ください。
GetTextureTransp(texture : HANDLE): INTEGER;	8 ～ 8.5	ハンドルで指定したテクスチャの透明度をパーセンテージで返します。
GetWallHoleTexturePart(object : HANDLE): INTEGER;	2010 ～	プラグインオブジェクトや、シンボル定義の壁穴グループの図形の壁テクスチャーを返します。
IsRW : BOOLEAN;	10 ～	Renderworksが利用可能な場合はTRUEを返します。
IsTextureableObject(obj : HANDLE): BOOLEAN;	8 ～	ハンドルで指定した3次元図形にテクスチャが設定されていた場合はTRUEを返します。
ResolveByClassTextureRef(obj : HANDLE; partID : INTEGER): LONGINT;	8 ～	ハンドルで指定した3次元図形のテクスチャがクラス定義だった場合（部分番号が-1の場合）は、クラスのテクスチャ番号を返します。
SetClTextureC(className : STRING; textureRef : LONGINT);	8 ～	指定したクラス属性の、壁の中央のテクスチャを設定します。
SetClTextureD(className : STRING; textureRef : LONGINT);	8 ～	指定したクラス属性の、屋根のテクスチャを設定します。
SetClTextureG(className : STRING; textureRef : LONGINT);	8 ～	指定したクラス属性の、テクスチャを設定します。
SetClTextureL(className : STRING; textureRef : LONGINT);	8 ～	指定したクラス属性の、壁の左側のテクスチャを設定します。
SetClTextureR(className : STRING; textureRef : LONGINT);	8 ～	指定したクラス属性の、壁の右側のテクスチャを設定します。
SetClTextureT(className : STRING; textureRef : LONGINT);	8 ～	指定したクラス属性の、屋根のテクスチャを設定します。
SetDefaultTexMap(h : HANDLE);	2009 ～ 2009	VW2010以降はSetDefaultTexMapNを使用してください。

Textures

SetDefaultTexMapN(h : HANDLE; texPartID, texLayerID : LONGINT);	2010 ～	図形にデフォルトのテクスチャーマッピング情報を割り当てます。テクスチャーはSetTextureRefで設定します。
SetDefaultTextureSpace(obj, texSpace : HANDLE; partID : INTEGER);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。 SetDefaultTexMapNをご利用ください。
SetObjExpandTexture(obj : HANDLE; expanded : BOOLEAN);	8 ～ 2008	ハンドルで指定した3次元図形のテクスチャを全体に貼り付けるように設定します。
SetTexBFeatureEnd(textureBitmap : HANDLE; featureEndX, featureEndY : INTEGER);	8 ～	ハンドルで指定したビットマップのノードの終点座標を設定します。
SetTexBFeatureStart(textureBitmap : HANDLE; featureStartX, featureStartY : INTEGER);	8 ～	ハンドルで指定したビットマップのノードの始点座標を設定します。
SetTexBitFeatureSize(textureBitmap : HANDLE; featureSize : REAL);	8 ～	ハンドルで指定したビットマップのノードの大きさを設定します。
SetTexBitmapOrigin(textureBitmap : HANDLE; originX, originY : INTEGER);	8 ～ 8.5	ハンドルで指定したビットマップのノードの原点座標を設定します。
SetTexBitPaintNode(textureBitmap, paintNode : HANDLE);	8 ～ 11.5	バージョン12では使われない関数です。 イメージノードの設定には、SetObjectVariableHandle(textureBitmap, 528, imageNode)を使用してください。
SetTexBitRepHoriz(textureBitmap : HANDLE; repeatHoriz : BOOLEAN);	8 ～	ハンドルで指定したビットマップの水平方向の繰り返しの有無を設定します。
SetTexBitRepVert(textureBitmap : HANDLE; repeatVert : BOOLEAN);	8 ～	ハンドルで指定したビットマップの垂直方向の繰り返しの有無を設定します。
SetTexMapBool(h : HANDLE; partID : LONGINT; selector : INTEGER; value : BOOLEAN);	2009 ～ 2009	VW2010以降はGetTexMapxxxNを使用してください。
SetTexMapBoolN(obj : HANDLE; texPartID, texLayerID : LONGINT; selector : INTEGER; value : BOOLEAN);	2010 ～	ハンドルで指定した図形の指定した部分番号でのマッピング情報をBOOLEAN値で設定します。部分番号を3に指定すると全体になります。
SetTexMapInt(h : HANDLE; partID : LONGINT; selector, value : INTEGER);	2009 ～ 2009	VW2010以降はGetTexMapxxxNを使用してください。

Textures

SetTexMapIntN(obj : HANDLE; texPartID, texLayerID : LONGINT; selector, value : INTEGER);	2010 ～	ハンドルで指定した図形の指定した部分番号でのマッピング情報をINTEGER値で設定します。 部分番号を3に指定すると全体になります。
SetTexMapReal(h : HANDLE; partID : LONGINT; selector : INTEGER; value : REAL);	2009 ～ 2009	VW2010以降はGetTexMapxxxNを使用してください。
SetTexMapRealN(obj : HANDLE; texPartID, texLayerID : LONGINT; selector : INTEGER; value : REAL);	2010 ～	ハンドルで指定した図形の指定した部分番号でのマッピング情報をREAL値で設定します。 部分番号を3に指定すると全体になります。
SetTexSpace2DOffset(textureSpace : HANDLE; offsetU, offsetV : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpace2DRadius(textureSpace : HANDLE; radius : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpace2DRot(textureSpace : HANDLE; rotationDegrees : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpace2DScale(textureSpace : HANDLE; scale : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpaceEndCap(textureSpace : HANDLE; endCapTextured : BOOLEAN);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpaceKind(textureSpace : HANDLE; kind : INTEGER);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpaceOrientU(textureSpace : HANDLE; uXAxis, uYAxis, uZAxis : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpaceOrientV(textureSpace : HANDLE; vXAxis, vYAxis, vZAxis : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpaceOrientW(textureSpace : HANDLE; wXAxis, wYAxis, wZAxis : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpaceOrigin(textureSpace : HANDLE; offsetX, offsetY, offsetZ : REAL);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。

Textures

SetTexSpacePartID(textureSpace : HANDLE; partID : INTEGER);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTexSpaceStartCap(textureSpace : HANDLE; startCapTextured : BOOLEAN);	8 ～ 2008	VW2009で使用できなくなった関数／手続きです。Set/Get TexMapをご利用ください。
SetTextureBitmap(shaderRecord, textureBitmap : HANDLE);	8 ～	ハンドルで指定したテクスチャのビットマップを設定します。
SetTextureRef(obj : HANDLE; textureRef : LONGINT; partID : INTEGER);	8 ～ 2009	VW2010以降はSetTextureRefNを使用してください。
SetTextureRefN(obj : HANDLE; textureRef, texPartID, texLayerID : LONGINT);	2010 ～	ハンドルで指定した3次元図形のテクスチャ番号を設定します。
SetTextureSet(object : HANDLE; textureSet : INTEGER);	2011 ～	オブジェクトのテクスチャーセットを設定します。
SetTextureShader(texture : HANDLE; shaderIndex : LONGINT);	8 ～ 8.5	ハンドルで指定したテクスチャのシェーダ番号を設定します。
SetTextureShininess(texture : HANDLE; shininess : INTEGER);	8 ～ 8.5	ハンドルで指定したテクスチャの反射率をパーセンテージで設定します。
SetTextureSize(texture : HANDLE; newSize : REAL);	10.1 ～	テクスチャのサイズを実寸で設定します。
SetTextureTransp(texture : HANDLE; transparency : INTEGER);	8 ～ 8.5	ハンドルで指定したテクスチャの反射率をパーセンテージで設定します。
SetWallHoleTexturePart(object : HANDLE; part : INTEGER);	2010 ～	プラグインオブジェクトや、シンボル定義の壁穴グループの図形の壁テクスチャーを設定します。

Units

GetPrimaryUnitInfo(VAR style : INTEGER; VAR prec, dimPrec : LONGINT; VAR format, angPrec : INTEGER; VAR showMark, dispFrac : BOOLEAN);	8 ～	現在設定されている主単位の情報を返します。
GetRoundingBase(VAR display, primary, secondary : INTEGER);	10 ～	主単位、寸法(主単位)、寸法(補助単位)の端数丸めの基準(1、2.5、5)を返します。
GetSecondaryUnitInfo(VAR style : INTEGER; VAR dimPrec : LONGINT; VAR format : INTEGER; VAR showMark, dispFrac : BOOLEAN);	8 ～	現在設定されている補助単位の情報を返します
GetUnits(VAR fraction, display : LONGINT; VAR format : INTEGER; VAR upi : REAL; VAR name, squareName : STRING);	1 ～	現在の単位設定の情報を返します。
PrimaryUnits(style : INTEGER; prec, dimPrec : LONGINT; format, angPrec : INTEGER; showMark, dispFrac : BOOLEAN);	7 ～	主単位を設定します。
SecondaryUnits(style : INTEGER; dimPrec : LONGINT; format : INTEGER; showMark, dispFrac : BOOLEAN);	7 ～	補助単位を設定します。
Units(i : INTEGER);	1 ～	現在の単位を設定します。

User Interactive

AutoKey(VAR asciiCode : LONGINT): BOOLEAN;	1 ～	ファンクションキー以外のキーが押し続けられていれば、TRUEを返し、同時に押されたキーのASCIIコードを返します。
CapsLock : BOOLEAN;	1 ～	CapsLockキーが押されている場合は、TRUEを返します。
Command : BOOLEAN;	1 ～	Command（Control）キーが押されている場合は、TRUEを返します。
GetKeyDown(VAR asciiCode : LONGINT);	1 ～	キーを押されるまで待機し、押されたキーのASCIIコードを返します。
GetLine(VAR p1X, p1Y, p2X, p2Y : REAL);	1 ～	マウスで2点を指示されるまで待機し、始点と終点の座標を返します。
GetLine3D(VAR p1X, p1Y, p1Z, p2X, p2Y, p2Z : REAL; useWP : BOOLEAN);	2010 ～	マウスで2点を指示されるまで待機し、始点と終点の3D座標を返します。
GetMouse(VAR pX, pY : REAL);	1 ～	現在のマウスの座標を返します。
GetPt(VAR pX, pY : REAL);	1 ～	マウスがクリックされるまで待機し、クリックされた点の座標を返します。
GetPt3D(VAR pX, pY, pZ : REAL; useWPOnly : BOOLEAN);	2010 ～	マウスがクリックされるまで待機し、クリックされた点の3D座標を返します。
GetPtL(p1X, p1Y : REAL; VAR p2X, p2Y : REAL);	1 ～	指定した始点から直線が描かれ、マウスで終点が指示されるまで待機します。
GetPtL3D(p1X, p1Y, p1Z : REAL; VAR p2X, p2Y, p2Z : REAL; useWPOnly : BOOLEAN);	2010 ～	指定した始点から四角形が描かれ、マウスで終点が指示されるまで待機します。
GetRect(VAR p1X, p1Y, p2X, p2Y : REAL);	1 ～	マウスで四角形の座標を指示されるまで待機し、四角形の座標を返します。
GetRect3D(VAR p1X, p1Y, p1Z, p2X, p2Y, p2Z : REAL; useWP : BOOLEAN);	2010 ～	マウスで四角形の座標を指示されるまで待機し、四角形の3D座標を返します。
KeyDown(VAR asciiCode : LONGINT): BOOLEAN;	1 ～	ファンクションキー以外のキーが押された場合、TRUEを返すと同時に押されたキーのASCIIコードを返します。
MouseDown(VAR pX, pY : REAL): BOOLEAN;	1 ～	マウスボタンが押されている場合は、TRUEと座標を返します。
Option : BOOLEAN;	1 ～	MacではOptionキー、WindowsではAltキーが押されている場合は、TRUEを返します。
RunTempTool(toolCallback : PROCEDURE; initialScroll : BOOLEAN);	2010 ～	テンポラリツール関数を呼びます。 ツール関数が終了するまで待機します。コールバック関数へはツールイベントが通知されます。
SetCursor(cursor : INTEGER);	1 ～	カーソルの形状を設定します。 カーソルの種類（LgCrossC:大きい十字形／SmCrossC:小さい十字形／WatchC:時計型／TextBarC:アイビーム形／ArrowC:矢印形／HandC:手のひら形）例）

User Interactive

SetTempToolHelpStr(helpString : STRING);	2010 ～	テンポラリツール関数のヘルプを設定します。
Shift : BOOLEAN;	1 ～	Shiftキーが押されている場合は、TRUEを返します。
TrackObject(callback : PROCEDURE; VAR outObj : HANDLE; VAR pX, pY, pZ : REAL);	2010 ～	インタラクティブにクライテリアに合致するオブジェクトを選択します。

Utility

BeginContext;	12.5 ～	この手続きを実行した後、EndContextを実行するまでの変化を記録します。 EndContextとともに使用します。
ClrMessage;	1 ～	メッセージウインドウを消去します。
ColorIndexToRGB(color : INTEGER; VAR red, green, blue : LONGINT);	6 ～	カラーパレットの色番号から各色の成分を取り出します。値の範囲は0から65535までです。
ColorIndexToRGBN(color : INTEGER; VAR red, green, blue : LONGINT; ignoreBlackBackground : BOOLEAN);	2010 ～	カラーパレットの色番号から各色の成分を取り出します。値の範囲は0から65535までです。
Date(dateFormat, infoFormat : INTEGER): STRING;	1 ～	現在の日付と時間を文字列で返します。 日付の種類（0:日付のみ／1:日付と時間／2:時間のみ）情報の種類（0:完全形式／1:完全形式の省略形／2:短縮形式）
DisableModules(modules : LONGINT);	10 ～	動作させないモジュールを設定します。モジュールパラメータは、どのモジュールで動作させないかを指定するビットフィールドです。
DisplayContextHelpOfCurrentPlugin;	12 ～	アクティブなプラグインのヘルプをウェブブラウザに表示します。
DisplayContextualHelp(Identifier : STRING): BOOLEAN;	12 ～	文字列を使用して、関連するコンテクストヘルプを表示します。WebWorksのウェブページ、インターネットのウェブページ、またはローカルファイルを指定できます。
DisplayContexualHelp(helpIdentifier : STRING): BOOLEAN;	2008 ～	パラメータで指定したコンテクストヘルプを表示します。
DisplayOrganizationDialog(tabToSelect : INTEGER);	12 ～	最初に選択されたタブとして指定された整数を持つ「オーガナイザ」ダイアログを表示します。
EndContext(acceptOrReject : INTEGER);	12.5 ～	Begin/Endブロックの間に、自動的になされたどんな変更にも順応する／しないを設定します。BeginContextとともに使用します。
FndError : BOOLEAN;	1 ～	直前で実行したプログラムにエラーが発生した場合はTRUEを返します。
ForEachObjectAtPoint(actionFunc : PROCEDURE; objOptions, travOptions : INTEGER; locX, locY : REAL; pickRadius : REAL);	2008 ～	指定した座標の下にある、条件に合致した図形のうち、最上位の図形を、指定した手続きで処理します。
ForEachObjectInLayer(actionFunc : PROCEDURE; objOptions, travOptions, layerOptions : INTEGER);	8.5 ～	指定した条件に合致した図形を、指定した手続きで処理します。
ForEachObjectInList(actionFunc : PROCEDURE; objOptions, travOptions : INTEGER; list : HANDLE);	8.5 ～	指定した条件に合致した図形を、指定した手続きで処理します。
GetActiveSerialNumber : STRING;	10 ～	アクティブなシリアル番号を返します。
GetArrayDimensions(arrayname : ARRAY; VAR rowStart, rowEnd, columnStart, columnEnd : INTEGER);	9 ～	指定した配列の大きさを返します。

Utility

GetClosestPt(VAR obj : HANDLE; ptX, ptY : REAL; VAR index : INTEGER; VAR containedObj : LONGINT);	8.5 ～	ハンドルで指定した図形の頂点を返します。 頂点が見つからない場合は、0を返します。
GetClosestSide(obj : HANDLE; ptX, ptY : REAL; VAR index1, index2 : INTEGER);	8.5 ～	ハンドルで指定した2次元図形で、指定した座標に最も近い頂点の番号を返します。
GetCurrentLocalization(VAR language, subLanguage : STRING);	2010 ～	Vectorworksの言語環境をISO 639-3 ドラフト形式の言語IDで返します。 sublanguageは使われていません。
GetCurrentMode : INTEGER;	10 ～	アプリケーションの現在のプロテクトモードを返します。
GetEnabledModules : LONGINT;	10 ～	現在利用可能なプロダクトモジュールの組み合わせを返します。
GetOSVersion(VAR major, minor, incr : LONGINT);	10 ～	動作しているOSのバージョンを返します。
GetPaletteVisibility(paletteName : STRING): BOOLEAN;	10 ～	パレットの表示／非表示を返します。
GetPickObjectInfo(pX, pY : REAL; VAR h, subH : HANDLE; VAR message : INTEGER): BOOLEAN;	8 ～	指定した座標の下にある図形のハンドルを返します。
GetPlantToolInitialized : BOOLEAN;	2008 ～	「植栽シンボル配置」ツールが初期化されているかどうかを返します。
GetPlantToolPlacementMode : INTEGER;	2008 ～	「植栽シンボル配置」ツールで設定されている配置モードを返します。
GetPlantToolPlantName : STRING;	2008 ～	「植栽シンボル配置の設定」ダイアログで設定した植栽名を返します。
GetPlantToolSpacing : REAL;	2008 ～	「植栽シンボル配置の設定」ダイアログで設定した間隔を返します。
GetProduct(VAR product : INTEGER; VAR modules : LONGINT);	9 ～	製品番号とパッケージ番号を返します。
GetSavedSetting(category, setting : STRING; VAR value : STRING): BOOLEAN;	12 ～	保存された設定ファイルから値を読み込みます。
GetScreen(VAR x1, y1, x2, y2 : INTEGER);	1 ～	モニタ画面の左上と右下の座標を返します。
GetTickCount : LONGINT;	8.5 ～	システムの時間を返します。
GetVersion(VAR major, minor, maintenance, platform : INTEGER);	7 ～	Vectorworksのバージョンと OSの種類を返します。 OSの種類（1:Macintosh／2:Windows）
GetVersionEx(VAR major, minor, maintenance, platform : INTEGER; VAR buildNum : LONGINT);	2010 ～	Vectorworksのバージョンとビルド番号とOSの種類を返します。
GetWorkingPlane(VAR x, y, z, xRotation, yRotation, zRotation : REAL);	10 ～	ワーキングプレーンの位置と方向を返します。

Utility

GetWorkingPlaneMat(refID : LONGINT; VAR outCenterPtX, outCenterPtY, outCenterPtZ REAL; VAR outNormalX, outNormalY, outNormalZ REAL; VAR outUVecX, outUVecY, outUVecZ REAL);	2011 ～	指定したワーキングプレーンマトリックスを取得します。
GetWorkingPlaneN(VAR outCenterPtX, outCenterPtY, outCenterPtZ :REAL; VAR outNormalX, outNormalY, outNormalZ :REAL; VAR outUVecX, outUVecY, outUVecZ :REAL);	2011 ～	アクティブなワーキングプレーンを取得します。
IsCoPlanar(refID1, refID2 : LONGINT): BOOLEAN;	2011 ～	2つの平面が同一平面の場合、TRUEを返します。
IsPerpPlane(refID1, refID2 : LONGINT): BOOLEAN;	2011 ～	2つの平面が直交している場合、TRUEを返します。
Message(z1,z2,...,zN : ANY);	1 ～	メッセージウインドウに文字列を表示します。
NameUndoEvent(eventName : STRING);	8 ～	取り消し処理のイベントを設定します。
OpenURL(URLname : DYNARRAY[] of CHAR): BOOLEAN;	10 ～	デフォルトブラウザやAdobe Acrobatを使ってURLを開きます。 URLが開けた場合はTRUEを返します。
PickObject(pX, pY : REAL): HANDLE;	1 ～	指定した座標の下にある図形のうち、最上位の図形のハンドルを返します。
PlanarPtTo3DModelPt(refID : LONGINT; pt2DX, pt2DY : REAL; VAR outPt3DX, outPt3DY, outPt3DZ : REAL): BOOLEAN;	2011 ～	指定した平面の2D座標を3D座標に変換します。
PlanarPtToScreenPlanePt(refID : LONGINT; pt2DX, pt2DY : REAL; VAR outPtX, outPtY : REAL): BOOLEAN ;	2011 ～	特定の平面上の2D座標をスクリーンプレーンに投影します。
PrepRelatedObjectForChange(objectAboutToBeChange : HANDLE);	12.5 ～	行われようとしている変更に関連した図形を準備します。
ReDraw;	1 ～	直前に作成または変更された図形を再描画します。
ReDrawAll;	1 ～	すべての図形を再描画します。
RedrawSelection;	2010 ～	選択状態を示す強調表示を再描画します。
ResetObject(objectHandle : HANDLE);	10 ～	現在の設定やパラメータの値を使って指定した図形を更新します。 図形のバウンドボックスをリセットします。壁の中の図形であれば、壁もリセットします。
RGBToColorIndex(red, green, blue : LONGINT; VAR color : INTEGER);	6 ～	各色の成分に一番近いカラーパレットの色番号を返します。 値の範囲は0から65535までです。

Utility

RGBToColorIndexN(red, green, blue : LONGINT; VAR color : INTEGER; ignoreBlackBackground : BOOLEAN);	2010 ～	各色の成分に一番近いカラーパレットの色番号を返します。
ScreenPlanePtToPlanarPt(refID : LONGINT; pt2DX, pt2DY : REAL; VAR outPtX, outPtY : REAL);	2011 ～	スクリーンプレーン上の2D座標を特定の平面上に投影します。
SetCurrentObject(h : HANDLE);	4 ～	ハンドルで指定した図形を、最後に作成された図形とします。
SetDrawingRect(paperWidth, paperHeight : REAL);	10 ～	用紙の大きさを設定します。
SetMaximumUndoEvents(events : INTEGER);	8 ～	取り消し可能な回数を設定します。
SetPaletteVisibility(paletteName : STRING; vis : BOOLEAN);	10 ～	パレットの表示／非表示を設定します。
SetSavedSetting(category, setting, value : STRING);	12 ～	保存された設定ファイルに値を書き込みます。
SetWorkingPlane(x, y, z, xRotation, yRotation, zRotation : REAL);	10 ～	ワーキングプレーンのx,y,zオフセット値とx,y,z平面の回転角度を設定します。
SetWorkingPlaneN(centerPtX, centerPtY, centerPtZ :REAL; normalX, normalY, normalZ :REAL; uVecX, uVecY, uVecZ :REAL);	2011 ～	アクティブなワーキングプレーンを設定します。
SortArray(VAR arraytosort : ARRAY; numtosort, fieldnumber : INTEGER);	9 ～	指定した1次配列を並び替え（ソート）します。 配列がレコードのハンドルの場合、指定したフィールド番号で並び替え（ソート）します。
SysBeep;	1 ～	警告音を鳴らします。
UndoOff;	8 ～	記憶している取り消し操作をすべて消去します。
ValidAngStr(str : STRING; VAR value : REAL): BOOLEAN;	1 ～	文字列を角度に変換します。変換できた場合はTRUEを返します。
ValidNumStr(str : STRING; VAR value : REAL): BOOLEAN;	1 ～	文字列を数値に変換します。 変換できた場合はTRUEを返します。
VerifyLibraryRoutine(routineName : STRING): BOOLEAN;	9 ～	指定した名前の外部関数が有効の場合はTRUEを返します。
Wait(seconds : INTEGER);	1 ～	指定した秒数が経過するまで、プログラムを停止します。

View / Zoom

GetProjection(theyLayer : HANDLE): INTEGER;	2008 ～	特定レイヤの投影番号を返します。
GetVCenter(VAR centerX, centerY : REAL);	8 ～	現在見えている用紙の中心座標を返します。
GetView(VAR xAngleR, yAngelR, zAngleR : REAL; VAR offsetX, offsetY, offsetZ : REAL);	8 ～	現在の視点の情報を返します。
GetZoom : REAL;	8 ～	現在の画面の拡大／縮小率を返します。
Projection(proj, rMode : INTEGER; viewDistance : REAL; clip1X, clip1Y, clip2X, clip2Y : REAL);	4 ～	3次元の見え方やレンダリングの種類を設定します。
SaveSheet(name : STRING; saveView, saveClass, saveLayer : BOOLEAN);	8 ～	現在の画面を、指定した名前で登録します。
SetVCenter(viewCenterX, viewCenterY : REAL);	6 ～	画面の表示位置を、指定した座標に移動します。
SetView(xAngle, yAngle, zAngle : REAL; xDistance, yDistance, zDistance : REAL);	1 ～	視点を、指定した3次元軸を中心に、指定した角度で回転させます。
SetViewVector(locationX, locationY, locationZ, targetX, targetY, targetZ, upX, upY, upZ : REAL);	9 ～	3D視点の座標を設定します。
SetZoom(zoomfactor : LONGINT);	6 ～	画面の拡大／縮小率を設定します。
VDelete(name : STRING);	1 ～	登録されている画面を削除します。
VRestore(name : STRING);	1 ～	登録されている画面を呼び出して表示します。
VSave(name : STRING);	1 ～	現在の画面を、指定した名前で登録します。

Worksheets

ActSSheet : HANDLE;	1 ～ 8.5	アクティブなワークシートのハンドルを返します。
AreWorksheetGridLinesVisible(h:HANDLE): BOOLEAN;	2011 ～	グリッドラインが、指定したワークシートで現在有効であるかどうかを返します。
AutoFitWSRowHeights(worksheet : HANDLE; fromRow, toRow : INTEGER);	12 ～	ハンドルで指定したワークシート上の指定した行を高さを自動調整します。
CellHasNum(h : HANDLE; row, col : INTEGER): BOOLEAN;	1 ～ 8.5	ハンドルで指定したワークシート上のセルに数値が入っている場合は、TRUEを返します。
CellHasStr(h : HANDLE; row, col : INTEGER): BOOLEAN;	1 ～ 8.5	ハンドルで指定したワークシート上のセルに文字列が入っている場合は、TRUEを返します。
CellString(row, column : INTEGER): STRING;	1 ～ 8.5	アクティブなワークシート上の指定したセルの値を文字列で返します。
CellValue(row, column : INTEGER): REAL;	1 ～ 8.5	アクティブなワークシート上の指定したセルの値を数値で返します。
ClearWSCell(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定したセル範囲の内容を消去します。
CloseSS(h : HANDLE);	6 ～ 8.5	ハンドルで指定したワークシートを閉じます。
CreateWS(name : STRING; rows, columns : INTEGER): HANDLE;	9 ～	ワークシートを作成します。
CreateWSImage(worksheet : HANDLE; locationX, locationY : REAL): HANDLE;	9 ～	ハンドルで指定したワークシートを、図形モードとして用紙の上に作成します。
DeleteWSColumns(worksheet : HANDLE; startColumn, numColumns : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定した列を削除します。
DeleteWSRows(worksheet : HANDLE; startRow, numRows : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定した行を削除します。
EnableDrawingWorksheetPalette(enable : BOOLEAN; worksheet : HANDLE);	2009 ～	ハンドルで指定したワークシートまたはすべてのワークシートを描画する／しないを設定します。

Worksheets

GetCAlign(h : HANDLE; row, col : INTEGER): INTEGER;	1 ～ 8.5	ハンドルで指定したワークシート上のセルの位置揃えを返します。
GetCellNum(h : HANDLE; row, col : INTEGER): REAL;	1 ～ 8.5	ハンドルで指定したワークシート上のセルの数値を返します。
GetCellStr(h : HANDLE; row, col : INTEGER): STRING;	1 ～ 8.5	ハンドルで指定したワークシート上のセルの文字列を返します。
GetCWidth(h : HANDLE; row, col : INTEGER): INTEGER;	1 ～ 8.5	ハンドルで指定したワークシート上のセルの幅を返します。
GetSprdSortSum(sheetHd : HANDLE; row : INTEGER; VAR sortCol1, sortCol2, sortCol3, sumCol : INTEGER);	8 ～ 8.5	ハンドルで指定したワークシートの並び替え（ソート）の設定を返します。
GetSprdSortSumColumns(sheetHd : HANDLE; row : INTEGER; VAR sortCol1, sortCol2, sortCol3, sumCol1, sumCol2, sumCol3 : INTEGER);	8.5 ～ 8.5	ハンドルで指定したワークシートの並び替え（ソート）の設定を返します。
GetTopVisibleWS : HANDLE;	9 ～	表示されている最上位のワークシートのハンドルを返します。
GetWSAutoRecalcState(worksheet : HANDLE): BOOLEAN;	2009 ～	ハンドルで指定したワークシートがセル編集時に自動再計算する設定かどうかを返します。
GetWSCellAlignment(worksheet : HANDLE; row, column : INTEGER; VAR cellAlignment : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定したセルの位置揃えを返します。
GetWSCellBorder(worksheet : HANDLE; row, column : INTEGER; VAR top, left, bottom, right : BOOLEAN);	9 ～	ハンドルで指定したワークシートの、指定したセルの枠線情報を返します。
GetWSCellFill(worksheet : HANDLE; row, column : INTEGER; VAR style : INTEGER; VAR bgcolor, fgcolor : LONGINT; VAR fillpattern : INTEGER);	12 ～	ハンドルで指定したワークシートの、指定したセルの模様、及び色属性を返します。

Worksheets

GetWSCellFormula(worksheet : HANDLE; row, column : INTEGER; VAR formula : STRING);	9 ～	ハンドルで指定したワークシートの、指定したセルの式を返します。
GetWSCellNumberFormat(worksheet : HANDLE; row, column : INTEGER; VAR style, accuracy : INTEGER; VAR leaderString, trailerString : STRING);	9 ～	ハンドルで指定したワークシートの、指定したセルの数字の表示形式を返します。
GetWSCellString(worksheet : HANDLE; row, column : INTEGER; VAR cellString : STRING);	9 ～	ハンドルで指定したワークシートの、指定したセルの文字列を返します。
GetWSCellTextAngle(worksheet : HANDLE; row, column : INTEGER; VAR angle : INTEGER);	12 ～	ハンドルで指定したワークシートの、指定したセルの文字の角度を返します。
GetWSCellTextColor(worksheet : HANDLE; row, column : INTEGER; VAR color : LONGINT);	12 ～	ハンドルで指定したワークシートの、指定したセルの文字色を返します。
GetWSCellTextFormat(worksheet : HANDLE; row, column : INTEGER; VAR fontIndex, size, style : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定したセルの文字属性を返します。
GetWSCellValue(worksheet : HANDLE; row, column : INTEGER; VAR cellValue : REAL);	9 ～	ハンドルで指定したワークシートの、指定したセルの数値を返します。
GetWSCellVertAlignment(worksheet : HANDLE; row, column : INTEGER; VAR vAlignment : INTEGER);	12 ～	ハンドルで指定したワークシートの、指定したセルの垂直方向の位置揃えを返します。
GetWSCellWrapTextFlag(worksheet : HANDLE; row, column : INTEGER; VAR wrapTextFlag : BOOLEAN);	12 ～	ハンドルで指定したワークシートの、指定したセルのラップテキストの状態を返します。
GetWSColumnOperators(worksheet : HANDLE; row : INTEGER; VAR sort1, sort2, sort3, sum1, sum2, sum3 : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定した行の並び替え（ソート）と合計を返します。
GetWSColumnWidth(worksheet : HANDLE; column : INTEGER; VAR width : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定した列の幅を返します。

Worksheets

GetWSFromImage(worksheetImage : HANDLE): HANDLE;	9 ～	図形モードのワークシートのハンドルを与えると、そのワークシートのハンドルを返します。
GetWSImage(worksheet : HANDLE): HANDLE;	9 ～	ワークシートのハンドルを与えると、その図形モードのワークシートのハンドルを返します。
GetWSMergedCellRange(worksheet : HANDLE; row, column : INTEGER; VAR topRow, leftColumn, bottomRow, rightColumn : INTEGER): BOOLEAN;	12.5 ～	ハンドルで指定したワークシートの、指定したセルをカバーするセルの範囲を取得します。 統合セルならば、TRUEを返します。
GetWSPlacement(worksheet : HANDLE; VAR top, left, bottom, right : INTEGER);	9 ～	ハンドルで指定したワークシートのウインドウ位置を返します。
GetWSRowColumnCount(worksheet : HANDLE; VAR numRows, numColumns : INTEGER);	9 ～	ハンドルで指定したワークシートの行数と列数を返します。
GetWSRowHeight(worksheet : HANDLE; row : INTEGER; VAR height : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定した行の高さを返します。
GetWSRowHLockState(worksheet : HANDLE; row : INTEGER; VAR lockState : BOOLEAN);	12 ～	ハンドルで指定したワークシートの指定した行のロック状態を返します。
GetWSSelection(worksheet : HANDLE; VAR currentCellRow, currentCellColumn, topRangeRow, leftRangeColumn, topRangeSubrow, bottomRangeRow, rightRangeColumn, bottomRangeSubrow : INTEGER);	9 ～	ハンドルで指定したワークシートの選択されているセルの範囲を返します。
GetWSSubrowActualCellString(worksheet : HANDLE; row, column, subrow : INTEGER; VAR cellString : STRING);	2008 ～	ハンドルで指定したワークシートの、データベース設定されている行の文字列を返します。
GetWSSubrowCellString(worksheet : HANDLE; row, column, subrow : INTEGER; VAR cellString : STRING);	9 ～	ハンドルで指定したワークシートの、データベース設定されている行の文字列を返します。
GetWSSubrowCellValue(worksheet : HANDLE; row, column, subrow : INTEGER; VAR cellValue : REAL);	9 ～	ハンドルで指定したワークシートの、データベース設定されている行の数値を返します。

Worksheets

GetWSSubrowCount(worksheet : HANDLE; databaseRow : INTEGER; VAR numSubrows : INTEGER);	9 ～	ハンドルで指定したワークシートの、データベース設定されている行の補助行数を返します。
GetWSSubrowHeight(worksheet : HANDLE; databaserow, subrow : INTEGER; VAR height : INTEGER);	2009 ～	ハンドルで指定したワークシートのデータベース行の高さを取得します。
InsertWSColumns(worksheet : HANDLE; beforeColumn, numColumns : INTEGER);	9 ～	ハンドルで指定したワークシートに、指定した列数を挿入します。
InsertWSRows(worksheet : HANDLE; beforeRow, numRows : INTEGER);	9 ～	ハンドルで指定したワークシートに、指定した行数を挿入します。
IsValidWSCell(worksheet : HANDLE; row, column : INTEGER): BOOLEAN;	9 ～	ハンドルで指定したワークシートの、指定したセルが有効の場合はTRUEを返します。
IsValidWSRange(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER): BOOLEAN;	9 ～	ハンドルで指定したワークシートの、指定したセル範囲が有効の場合はTRUEを返します。
IsValidWSSubrowCell(worksheet : HANDLE; row, column, subrow : INTEGER): BOOLEAN;	9 ～	ハンドルで指定したワークシートの、指定したデータベース行が有効の場合はTRUEを返します。
IsWSCellNumber(worksheet : HANDLE; row, column : INTEGER): BOOLEAN;	9 ～	ハンドルで指定したワークシートの、指定したセルの内容が数値の場合はTRUEを返します。
IsWSCellString(worksheet : HANDLE; row, column : INTEGER): BOOLEAN;	9 ～	ハンドルで指定したワークシートの、指定したセルの内容が文字列の場合はTRUEを返します。
IsWSDatabaseRow(worksheet : HANDLE; databaseRow : INTEGER): BOOLEAN;	9 ～	ハンドルで指定したワークシートの、指定した行がデータベース行の場合はTRUEを返します。
IsWSSubrowCellNumber(worksheet : HANDLE; row, column, subrow : INTEGER): BOOLEAN;	9 ～	ハンドルで指定したワークシートの、指定したデータベース行が数値の場合はTRUEを返します。

Worksheets

IsWSSubrowCellString(worksheet : HANDLE; row, column, subrow : INTEGER): BOOLEAN;	9 ～	ハンドルで指定したワークシートの、指定したデータベース行が数値の場合はTRUEを返します。
IsWSVisible(worksheet : HANDLE): BOOLEAN;	9 ～	ハンドルで指定したワークシートが表示されている場合はTRUEを返します。
LoadCell(ro, col : INTEGER; entry : STRING);	1 ～ 8.5	アクティブなワークシートの指定したセルに文字列を入力します。
NewSprdSheet(name : STRING; locationX, locationY : REAL; rows, columns : INTEGER; showOnDrawing, openAfterCreate : BOOLEAN);	8 ～ 8.5	ワークシートを作成します。
RecalculateWS(worksheet : HANDLE);	9 ～	ハンドルで指定したワークシートを再計算します。
SelectSS(h : HANDLE);	1 ～ 8.5	ハンドルで指定したワークシートをアクティブにします。
SetSprdSortSum(sheetHd : HANDLE; row, sortCol1, sortCol2, sortCol3, sumCol : INTEGER);	8 ～ 8.5	ハンドルで指定したワークシートの内容を並び替え（ソート）します。
SetSprdSortSumColumns(sheetHd : HANDLE; row, sortCol1, sortCol2, sortCol3, sumCol1, sumCol2, sumCol3 : INTEGER);	8.5 ～ 8.5	ハンドルで指定したワークシートの並び替え（ソート）の設定します。
SetTopVisibleWS(worksheet : HANDLE);	9 ～	ハンドルで指定したワークシートを最上位に設定します。
SetWorksheetGridLinesVisibility(h : HANDLE; visible : BOOLEAN);	2011 ～	指定したワークシート内のグリッドラインの表示形式を設定します。
SetWSAutoRecalcState(worksheet : HANDLE; state : BOOLEAN);	2009 ～	ハンドルで指定したワークシートがセル編集時に自動再計算する／しないを設定します。
SetWSCellAlignment(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, cellAlignment : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定したセルの位置揃えを設定します。
SetWSCellBorder(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER; top, left, bottom, right, outline : BOOLEAN);	9 ～ 11.5	ハンドルで指定したワークシートの、指定したセルの枠線情報を設定します。

Worksheets

SetWSCellBorders(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER; top, left, bottom, right : BOOLEAN; OutlineInside : INTEGER);	12 ～	ハンドルで指定したワークシートの、指定したセルの枠線情報を設定します。
SetWSCellBottomBorder(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style, weight : INTEGER; color : LONGINT);	12.5 ～	ハンドルで指定したワークシートの、指定したセルの下の枠線情報を設定します。
SetWSCellFill(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style : INTEGER; bgcolor, fgcolor : LONGINT; fillpattern : INTEGER);	12 ～	ハンドルで指定したワークシートの、指定したセルの模様、及び色属性を設定します。
SetWSCellFormula(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER; formula : STRING);	9 ～	ハンドルで指定したワークシートの、指定したセルの式を設定します。
SetWSCellInsideHorizBorder(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style, weight : INTEGER; color : LONGINT);	12.5 ～	ハンドルで指定したワークシートの、指定したセルの上下の枠線情報を設定します。
SetWSCellInsideVertBorder(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style, weight : INTEGER; color : LONGINT);	12.5 ～	ハンドルで指定したワークシートの、指定したセルの左右の枠線情報を設定します。
SetWSCellLeftBorder(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style, weight : INTEGER; color : LONGINT);	12.5 ～	ハンドルで指定したワークシートの、指定したセルの左の枠線情報を設定します。
SetWSCellNumberFormat(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style, accuracy : INTEGER; leaderString, trailerString : STRING);	9 ～	ハンドルで指定したワークシートの、指定したセルの数字の表示形式を設定します。
SetWSCellOutlineBorder(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style, weight : INTEGER; color : LONGINT);	12.5 ～	ハンドルで指定したワークシートの、指定したセルの外側の枠線情報を設定します。

Worksheets

SetWSCellRightBorder(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style, weight : INTEGER; color : LONGINT);	12.5 ～	ハンドルで指定したワークシートの、指定したセルの右の枠線情報を設定します。
SetWSCellTextColor(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER; color : LONGINT);	12 ～	ハンドルで指定したワークシートの、指定したセルの文字色を設定します。
SetWSCellTextFormat(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, fontIndex, size, style : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定したセルの文字属性を設定します。
SetWSCellTopBorder(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, style, weight : INTEGER; color : LONGINT);	12.5 ～	ハンドルで指定したワークシートの、指定したセルの上の枠線情報を設定します。
SetWSCellVertAlignment(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, vAlignment : INTEGER);	12 ～	ハンドルで指定したワークシートの、指定したセルの垂直方向の位置揃えを設定します。
SetWSCellWrapTextFlag(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER; wrapTextFlag : BOOLEAN);	12 ～	ハンドルで指定したワークシートの、指定したセルのラップテキストの状態を設定します。
SetWSColumnOperators(worksheet : HANDLE; row, sort1, sort2, sort3, sum1, sum2, sum3 : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定した行の並び替え（ソート）と合計を設定します。
SetWSColumnWidth(worksheet : HANDLE; fromColumn, toColumn, width : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定した列の幅を設定します。
SetWSCurrentCell(worksheet : HANDLE; currentCellRow, currentCellColumn : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定したセルをアクティブにします。
SetWSPlacement(worksheet : HANDLE; top, left, bottom, right : INTEGER);	9 ～	ハンドルで指定したワークシートのウインドウ位置を設定にします。
SetWSRowHeight(worksheet : HANDLE; fromRow, toRow, height : INTEGER; updatePalette, lockHeight : BOOLEAN);	12 ～	ハンドルで指定したワークシートの、指定した行の高さを設定します。

Worksheets

SetWSSelection(worksheet : HANDLE; currentCellRow, currentCellColumn, topRangeRow, leftRangeColumn, topRangeSubrow, bottomRangeRow, rightRangeColumn, bottomRangeSubrow : INTEGER);	9 ～	ハンドルで指定したワークシートの、指定したセル範囲を選択します。
SetWSTextAngle(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn, angle : INTEGER);	12 ～	ハンドルで指定したワークシートの、指定したセルの文字の角度を設定します。
ShowWS(worksheet : HANDLE; show : BOOLEAN);	9 ～	ハンドルで指定したワークシートの、表示／非表示を設定します。
ShowWSDialog(dialogType : INTEGER);	9 ～	アクティブなワークシートに対する設定ダイアログを表示します。
SprdAlign(align : INTEGER);	1 ～ 8.5	ワークシートのセルの位置揃えを設定します。 この手続きは、LoadCellを実行する前に利用します。 位置揃え:1＝標準／2＝左寄せ／3＝中央／4＝右寄せ
SprdBorder(top, left, bot, right : BOOLEAN);	1 ～ 8.5	ワークシートのセルに枠線を設定します。この手続きは、LoadCellを実行する前に利用します。
SprdFormat(numForm, acc : INTEGER; ldr, trailr : STRING);	1 ～ 8.5	ワークシートのセルの数字の表示形式を設定します。この手続きは、LoadCellを実行する前に利用します。
SprdSize(h : HANDLE; VAR row, col : INTEGER);	1 ～ 8.5	ハンドルで指定したワークシートの行と列の数を返します。
SprdWidth(width : REAL);	1 ～ 8.5	ワークシートのセル幅を設定します。この手続きは、LoadCellを実行する前に利用します。
TargetSprdSheet(h : HANDLE);	8 ～ 8.5	ハンドルで指定したワークシートをアクティブにします。ワークシート自体は開きません。
WorksheetMergeCells(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER): BOOLEAN;	12.5 ～	ハンドルで指定したワークシートの、指定した複数のセルを1つのセルに統合します。
WorksheetSplitCells(worksheet : HANDLE; topRow, leftColumn, bottomRow, rightColumn : INTEGER): BOOLEAN;	12.5 ～	ハンドルで指定したワークシートの、指定したセルを分離して個別のセルに戻します。

XML		
CreateNewXMLDocument(XMLHandle : LONGINT; rootElementName : STRING): INTEGER;	2011 ～	新しいXML書類を作成します。
DeleteAttribute(XMLHandle : LONGINT; elementPath, attribute : STRING): INTEGER;	2011 ～	属性を一つ削除します。
DeleteCDATA(XMLHandle : LONGINT; elementPath : STRING): INTEGER;	2011 ～	CDATAセクションを削除します。
DeleteElement(XMLHandle : LONGINT; elementPath : STRING): INTEGER;	2011 ～	パスで指定された位置の要素を削除します。
FindAttribute(XMLHandle : LONGINT; startElementPath, searchAttribute : STRING; VAR foundPath, attributeValue : STRING): INTEGER;	2011 ～	名前で特定の属性を検索します。
FindElement(XMLHandle : LONGINT; startElementPath, searchElement : STRING; VAR foundPath : STRING): INTEGER;	2011 ～	名前で特定の要素を検索します。
GetAttributeValue(XMLHandle : LONGINT; elementPath, attribute : STRING; VAR value : STRING): INTEGER;	2011 ～	属性の値を取得する。
GetCDATA(XMLHandle : LONGINT; elementPath : STRING; VAR returnVal : DYNARRAY[] of CHAR): INTEGER;	2011 ～	CDATAセクションを取得する。
GetElementValue(XMLHandle : LONGINT; elementPath : STRING; VAR value : STRING): INTEGER;	2011 ～	パスで示される要素の値を取得する。

XML		
GetFirstChild(XMLHandle : LONGINT; elementPath : STRING; VAR value : STRING): INTEGER;	2011 ～	与えられた要素の最初の子を返す。
GetNextElement(XMLHandle : LONGINT; elementPath : STRING; VAR value : STRING): INTEGER;	2011 ～	与えられた要素の次（兄弟）の要素を返す。
GetPreviousElement(XMLHandle : LONGINT; elementPath : STRING; VAR value : STRING): INTEGER;	2011 ～	与えられた要素の前（兄弟）の要素を返す。
InitXML : LONGINT;	2011 ～	XMLシステムを初期化する。
ReadXMLFile(XMLHandle : LONGINT; whichPath : INTEGER; filename : STRING): INTEGER;	2011 ～	XMLファイルを開いて読み込む。
ReadXMLMemory(XMLHandle : LONGINT; XMLData : DYNARRAY[] of CHAR): INTEGER;	2011 ～	メモリ上のXMLデータをパースする。
ReleaseXML(XMLHandle : LONGINT): INTEGER;	2011 ～	XMLパーサが利用する内部メモリを解放する。
SetAttributeValue(XMLHandle : LONGINT; elementPath, attribute, value : STRING): INTEGER;	2011 ～	属性の値を設定する。
SetCDATA(XMLHandle : LONGINT; elementPath : STRING; data : DYNARRAY[] of CHAR): INTEGER;	2011 ～	CDATAセクションを設定する。
SetElementValue(XMLHandle : LONGINT; elementPath, value : STRING): INTEGER;	2011 ～	与えられたパスの要素の値を設定する。

XML		
WriteXMLFile(XMLHandle : LONGINT; whichPath : INTEGER; filename : STRING): INTEGER;	**2011 ～**	XMLファイルを書き出す。
WriteXMLMemory(XMLHandle : LONGINT; VAR XMLData : DYNARRAY[] of CHAR): INTEGER;	**2011 ～**	内部のDOMツリーからXMLデータを作成する

XML SAX

XMLSAXAddNodeAttr(XMLHandle : LONGINT; nodeAttrName, nodeAttrValue : STRING): INTEGER;	2011 ～	SAXを利用してXMLを書く。 XMLSAXBeginNodeで始まるノードに属性ノードを追加する。
XMLSAXAddNodeValue(XMLHandle : LONGINT; nodeValue : STRING): INTEGER;	2011 ～	SAXを利用してXMLを書く。 XMLSAXBeginNodeで始まるノードに要素ノードを追加する。
XMLSAXBeginDocFile(XMLHandle : LONGINT; whichPath : INTEGER; filename : STRING): INTEGER;	2011 ～	SAXを利用してXMLを書く。ファイル上の書類の開始。 XMLSAXEndDocで終了する。
XMLSAXBeginDocMemory(XMLHandle : LONGINT): INTEGER;	2011 ～	SAXを利用してXMLを書く。メモリ上の書類を開始する。 XMLSAXEndDocMemoryで終了する。
XMLSAXBeginNode(XMLHandle : LONGINT; nodeName : STRING): INTEGER;	2011 ～	SAXを利用してXMLを書く。ノードの開始。 XMLSAXEndNodeで終了する。
XMLSAXEndDoc(XMLHandle : LONGINT): INTEGER;	2011 ～	SAXを利用してXMLを書く。書類の終了。 書類の開始はXMLSAXBeginDocFile
XMLSAXEndDocMemory(XMLHandle : LONGINT; VAR XMLData : DYNARRAY[] of CHAR): INTEGER;	2011 ～	SAXを利用してXMLを書く。書類の終了。 書類の開始はXMLSAXBeginDocMemory
XMLSAXEndNode(XMLHandle : LONGINT): INTEGER;	2011 ～	SAXを利用してXMLを書く。ノードの終了。 ノードの開始はXMLSAXBeginNode
XMLSAXParseFile(XMLHandle : LONGINT; whichPath : INTEGER; filename : STRING; nodeCallback : PROCEDURE): INTEGER;	2011 ～	SAXを利用してXMLをパースする。
XMLSAXParseMemory(XMLHandle : LONGINT; XMLData : DYNARRAY[] of CHAR; nodeCallback : PROCEDURE): INTEGER;	2011 ～	

Notes

2011/02/02更新。
VW2011の機能追加。
Objects - Architecturalで未掲載の項目を追加。