

# ドキュメントストラクチャ

## Get Active Layer obj

ノードの説明  
ドキュメント内で現在アクティブなレイヤを返します。

ポートの説明  
obj :  
現在のファイル内のアクティブレイヤ。

## Get First Layer obj

ノードの説明  
Vectorworksファイルの最初のレイヤオブジェクトを返します。

ポートの説明  
obj : レイヤオブジェクト。

## Get Last Layer obj

ノードの説明  
Vectorworksファイルの最後のレイヤオブジェクトを返します。

ポートの説明  
obj : レイヤオブジェクト。

## obj Get Layer obj

ノードの説明  
指定した図形が存在するレイヤを返します。

ポートの説明  
object : オブジェクトの情報。  
obj :  
オブジェクトの情報によるレイヤオブジェクト。

## string Get Layer By Name obj

ノードの説明  
指定した名前をもつレイヤオブジェクトを返します。

ポートの説明  
string : レイヤ名。  
obj : レイヤオブジェクト。

## Get Layer Count numLay

ノードの説明  
アクティブドキュメント内のレイヤの現在の数を返します。

ポートの説明  
numLay :  
アクティブドキュメント内のレイヤの現在の数。

## obj Get Layer Elevation baseElev thickness

ノードの説明  
指定したレイヤの高さ (Z) と厚み (ΔZ) を返します。

ポートの説明  
obj : レイヤオブジェクトの情報。  
baseElev : レイヤの高さ (Z)。  
thickness : レイヤの厚み (ΔZ)。

## obj Get Layer Name string

ノードの説明  
対象レイヤの名前を返します。

ポートの説明  
obj : レイヤオブジェクトの情報。  
string : レイヤの名前。

## obj Get Layer Visibility visibility

ノードの説明  
対象レイヤの表示設定を返します。

ポートの説明  
obj : レイヤオブジェクトの情報。  
visibility : 表示設定のインデックス値。 表示 - 0, 非表示 - 1, グレイ表示 - 2

## obj Get Object Count numObj

ノードの説明  
対象レイヤの図形数を返します。

ポートの説明  
obj : レイヤオブジェクトの情報。  
numObj : レイヤ上の図形数。

## zVal deltaZVal Get Z Values

ノードの説明  
アクティブなレイヤの高さ (Z) と厚み (ΔZ) を返します。

ポートの説明  
zVal : レイヤの高さ (Z)。  
deltaZVal : レイヤの厚み (ΔZ)。

## string Layer

ノードの説明  
Vectorworksファイルに新しいレイヤを新規作成します。作成後、新しいレイヤがアクティブレイヤになります。関数に渡されたレイヤ名がすでに存在している場合は、指定したレイヤをアクティブレイヤにします。

ポートの説明  
string :  
作成するレイヤの名前です。シングルクォーテは一致しない文字列修飾子として扱われ、エラーの原因となるため、使用しないでください。

## string type obj Layer With Type

ノードの説明  
指定した種類のレイヤを作成します。

ポートの説明  
string : 作成するレイヤの名前。  
type : 作成するレイヤの種類。入力値: デザインレイヤ = 1, シートレイヤ = 2  
obj : 作成されたレイヤオブジェクト。

## obj Set Layer obj

ノードの説明  
指定したオブジェクトを指定したレイヤに配置します。

コントロールの説明  
Layers :  
選択可能なレイヤのリストです。

ポートの説明  
object : オブジェクトの情報。  
obj : 結果のオブジェクト。

## zVal deltaZVal Set Z Values obj

ノードの説明  
アクティブなレイヤの高さ (Z) と厚み (ΔZ) を設定します。

ポートの説明  
zVal :  
ファイルの基準平面に対するレイヤの高さ (Z)。  
deltaZVal : レイヤの厚み (ΔZ)。  
obj : 結果のアクティブレイヤ。

# ストリングとテキスト

## ASCII Letters string

ノードの説明  
アルファベット (ascii) の小文字と大文字を返します。アルファベットのみで地域言語に依存しません。

ポートの説明  
string : 結果の文字列。

## ASCII Lowercase string

ノードの説明  
アルファベット小文字'abcdefghijklmnopqrstu  
vwxyz'を返します。地域言語に依存しません。

ポートの説明  
string : 結果の文字列。

## ASCII Uppercase string

ノードの説明  
アルファベット大文字'ABCDEFGHIJKLMNOP  
QRSTUVWXYZ'を返します。地域言語に依存しません。

ポートの説明  
string : 結果の文字列。

## text Convert to Poly Group polyGroup

ノードの説明  
文字列図形を同様の形状を持つ多角形/曲線のグループ図形に変換します。

ポートの説明  
text : 変換する文字列図形。  
polyGroup : 結果の多角形/曲線グループ。

## Digits string

ノードの説明  
文字列を返します。'0123456789'

ポートの説明  
string : 結果の文字列。

## format Format string

ノードの説明  
文字列を結果として得られる書式文字列と引数の結果です。このノードはPythonのフォーマットを使用しています: <http://docs.python.jp/3.3/library/string.html?highlight=format#format-string-syntax>

ポートの説明  
format : 書式指定する文字列。  
args : フォーマット文字列関数で使用する引数。  
string : 結果の文字列。

## text fontID pos Get Char Properties size style

ノードの説明  
文字列図形内の指定した位置にある文字のプロパティを返します。

ポートの説明  
text : プロパティを取得する文字列図形。  
pos : 文字列内の文字の位置を表す整数。0のインデックスは文字列の最初の文字を指します。  
fontID : 文字のフォントIDを返します。  
size : 文字のポイントサイズ。(1ポイント = 1/72インチ)  
style : 文字のスタイル。結果は下位7ビット、[0, 127]範囲内整数で以下の値を組み合わせることによってエンコードされています: 0 = 標準, 1 = ボールド, 2 = イタリック, 4 = アンダーライン, 8 = アウトライン, 16 = シャドウ, 32 = 上付き, 64 = 下付き。

## name Get Font ID fontID

ノードの説明  
利用可能なフォント名の文字列を他のノードに渡すことができるフォントIDに変換します。要求されたフォントが使用できない場合は-1を返します。

ポートの説明  
name : フォントの名前。  
fontID : フォントID。

## Get Font List Size numFonts

ノードの説明  
ローカルシステム上で利用可能なフォントの数を返します。

ポートの説明  
numFonts : 利用可能なフォント数。

## fontID Get Font Name name

ノードの説明  
フォントの名前を返します。

ポートの説明  
fontID : 現在のオペレーティングシステムで、フォントを表す値のID (整数)。  
name : フォント名。

## text Get String str

ノードの説明  
対象の文字列図形内に含まれる文字列を返します。

ポートの説明  
text : 文字情報を取得する文字列図形。  
str : 結果の文字。

## text pos Get Style Ref styleRef byClass

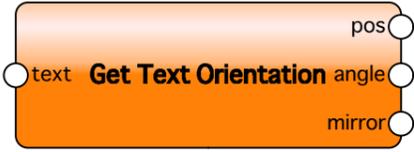
ノードの説明  
文字列図形の指定した位置の文字の文字スタイルを返します。

ポートの説明  
text : プロパティを取得する文字列図形。  
pos : 文字列内の文字の位置を表す整数。0のインデックスは文字列の最初の文字を指します。  
styleRef : 文字列図形内の指定した位置の文字スタイルを返します。0は文字スタイルがない事を意味します。文字列図形がクラスのテキストスタイルを使用している場合は該当するスタイルを返します。  
byClass : 文字列図形の指定した位置にクラスの文字スタイルが適応されているかを示すboolean値。

text **Get Text Length** len

ノードの説明  
ハンドルで指定した文字列図形の文字の数を返します。

ポートの説明  
text : 対象の文字列図形。  
len : 文字列図形の文字数。



ノードの説明  
ハンドルで指定した文字列図形の座標と角度を返します。

ポートの説明  
text : 対象の文字列図形。  
textOrigin : 文字列図形の位置を表す2D座標。  
textAng : 文字列図形の角度 (度数)。  
textIsMirrored :  
文字列図形のミラー反転状況を表すboolean値。



ノードの説明  
文字列図形のプロパティを返します。

ポートの説明  
text : プロパティを取得する文字列図形。  
txtJust :  
対象文字列図形の水平方向の文字の位置揃え。¥n  
1 = 左よせ, ¥n 2 = 中央, ¥n 3 = 右よせ, ¥n 4 = 両端揃え  
txtVJust :  
対象文字列図形の垂直方向の文字の位置揃え。¥n  
1 = 上揃え, ¥n 2 = ベースライン揃え, ¥n 3 = 中央揃え, ¥n 4 = 英字下揃え, ¥n 5 = 下揃え  
leadingVal :  
対象文字列図形の文字の行間 (ポイント単位: 1ポイント = 1/72")。  
spacing : 対象文字列図形の文字の行間。¥n 2 = 全角, ¥n 3 = 1.5角, ¥n 4 = 倍角  
wrap : 対象文字列図形の文字の折り返しモード。

text **Get Text Width** width

ノードの説明  
文字列図形の余白の幅を返します。文字の折り返しを設定している場合、マージン幅はユーザが設定します。文字の折り返しを設定していない場合、VWによって最も長い行の幅が計算されます。

ポートの説明  
text : 対象の文字列図形。  
marginWidth : 文字列図形の幅。

**Hex Digits** string

ノードの説明  
文字'0123456789abcdefABCDEF'を返します。

ポートの説明  
string : 結果の文字列。

**Oct Digits** string

ノードの説明  
文字列「01234567」を返します。

ポートの説明  
string : 結果の文字列。

list **Print List** text

ノードの説明  
指定したリストの内容で、新規の文字列図形を作成します。

ポートの説明  
list : 文字列にするリスト。  
text : 結果の文字列図形。

**Punctuation** string

ノードの説明  
現在のロケールで句読点とみなされるASCII文字の文字列を返します。

ポートの説明  
string : 結果の文字列。



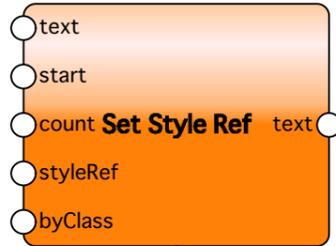
ノードの説明  
文字列図形内の指定した範囲の文字のプロパティを設定します。

ポートの説明  
textIn : 対象の文字列図形。  
start :  
文字列内で変更したい部分の開始文字位置を整数値で指定します。インデックスの0は全体の文字列内の一番最初の文字を示します。  
count : 設定を適用する文字数。  
fontID : 文字のフォントIDを返します。  
size : 文字のポイントサイズ。(1ポイント = 1/72インチ)  
style : 文字のスタイル。結果は下位7ビット、[0, 127]範囲内整数で以下の値を組み合わせることによってエンコードされています: 0 = 標準, 1 = ボールド, 2 = イタリック, 4 = アンダーライン, 8 = アウトライン, 16 = シャドウ, 32 = 上付き, 64 = 下付き。  
textOut : 変更後の文字列図形。



ノードの説明  
指定した文字列図形の文字列を設定します。新規の文字列には、古い文字列の最初の文字と同じフォント、サイズ、スタイルが割り当てられます。

ポートの説明  
inText :  
data :  
text :



ノードの説明  
文字列図形の指定した範囲の文字列に指定した文字スタイルを設定します。

ポートの説明  
text : 対象の文字列図形。  
start :  
文字列内で変更したい部分の開始文字位置を整数値で指定します。インデックスの0は全体の文字列内の一番最初の文字を示します。  
count : 設定を割り当てる文字数。  
styleRef :  
文字列図形の指定した位置と長さの文字列が、インデックスで指定した文字スタイルを使用するように設定します。文字スタイルを使用しない(解除する)場合は0を指定します。  
byClass :  
文字列図形の指定した位置と長さの文字列が、クラスの文字スタイルを使用するように設定します。  
textOut : 変更後の文字列図形。

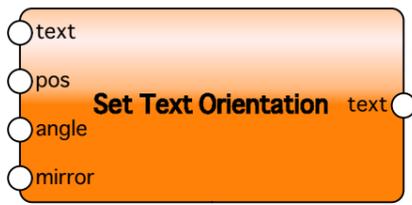


#### ノードの説明

指定した文字列図形と装飾オブジェクトとを関連付けます。例えば、ビューポートで文字列図形の縮率が変更されたら、装飾オブジェクトにも同様に縮率が適用されます。1つの文字列図形に対して複数のオブジェクトで装飾することもできます。

#### ポートの説明

textBlock : 装飾したい文字列図形。  
 textAdorner : 文字列を装飾するオブジェクト。  
 pt : 装飾オブジェクトを配置する文字列図形での位置。  
 textOut : 変更後の文字列図形。

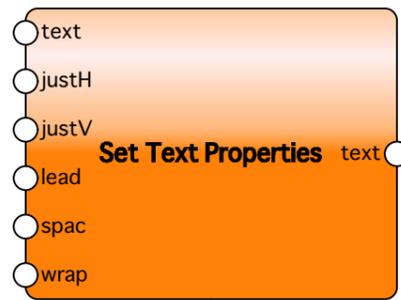


#### ノードの説明

指定した文字列図形の座標と角度を設定します。

#### ポートの説明

inText : 対象の文字列図形。  
 textOrigin : 文字列図形に指定したい2D座標。  
 textAng : 指定したい文字列の角度。  
 textIsMirrored : 文字列図形を反転させるかどうかをBoolean値で指定。  
 text : 変更後の文字列図形。



#### ノードの説明

文字列図形のプロパティを設定します。

#### ポートの説明

textIn : プロパティを設定する対象の文字列図形。  
 txtJust : 指定した文字列図形の水平方向の文字の位置揃えを設定します。¥n 1 = 左寄せ¥n 2 = 中央揃え¥n 3 = 右寄せ¥n 4 = 両端揃え  
 txtVJust : 指定した文字列図形の垂直方向の文字の位置揃えを設定します。¥n 1 = 上揃え¥n 2 = ベースライン揃え¥n 3 = 中央揃え¥n 4 = 英字下揃え¥n 5 = 下揃え  
 leadingVal : 指定した文字列図形の文字の行間隔を値で設定します (ポイント単位: 1ポイント= 1/72")。  
 spacing : 指定した文字列図形の文字の行間を設定します。¥n 2 = 行間全角¥n 3 = 行間1.5角¥n 4 = 行間倍角  
 wrap : 指定した文字列の文字の折り返しモードを設定します。  
 textOut : 変更後の文字列図形。



#### ノードの説明

指定した文字列図形の幅を設定します。文字の折り返しは自動的に有効になります。

#### ポートの説明

inText : 対象の文字列図形。  
 widthDistance : 文字列の新しい幅。  
 text : 変更後の文字列図形。



#### ノードの説明

任意の値から文字列を作成します。リストを文字列に変換する場合は、区切り文字を指定できます。

#### ポートの説明

any : 任意の入力値。  
 delim : 区切り文字。  
 string : 結果の文字列。



#### ノードの説明

新規の文字列図形を作成します。

#### ポートの説明

prefix : オプションとして追加できる前記号用文字列。  
 string : メインの文字列。  
 suffix : オプションとして追加できる後記号用文字列。  
 text : 結果の文字列図形。



#### ノードの説明

空白文字とみなされるASCII文字すべてを含む文字列を返します。スペース、タブ、改行、復帰、改ページ、垂直タブなどです。

#### ポートの説明

string : 結果の文字列。

# ユーティリティ

## wires **Dummy End**

**ノードの説明**  
ダミーノードです。サブネットワークの端部に接続するために使用されます。

**ポートの説明**  
wires : ネットワークからの接続ワイヤ。

## value **Test List** out

**ノードの説明**  
入力リストの長さが期待するリストの長さ  
と一致するかどうかをテストします。オプションとして、リストの内容もテスト  
ことができます。テストに失敗するとノ  
ードが赤色になり、警告ダイアログボックス  
が表示されます。

**コントロールの説明**  
Expected Length :  
期待するリストの長さです。  
Expect Value :  
リストの長さと共にリストの値も比較する  
場合にはチェックを入れます。  
Value :  
値もテストする場合は期待するリスト内容  
を入力します。

**ポートの説明**  
value : テストするリスト。  
out : 元のリスト。

## value **Test List At** out

**ノードの説明**  
リスト内の指定したインデックスの値をテ  
ストします。値が一致しない場合はノード  
が赤色になり、警告ダイアログボックスが  
表示されます。

**コントロールの説明**  
Test Index :  
テストする値のインデックスです。  
Expected Value : 期待値を入力します。  
epsilon :  
期待値からの許容偏差を入力します。

**ポートの説明**  
value : テストするリスト。  
out : 元のリスト。

## value **Test Value** out

**ノードの説明**  
入力値が期待値と一致するかどうかをテス  
トします。値が一致しない場合はノードが  
赤色になり、警告ダイアログボックスが表  
示されます。

**コントロールの説明**  
Expected Value : 期待値を入力します。  
epsilon :  
期待値からの許容偏差を入力します。

**ポートの説明**  
value : テストする値。  
out : 元の値。

# 図形データ



**ノードの説明**  
 オブジェクトへ既存のレコードフォーマットのインスタンスを割り当てます。

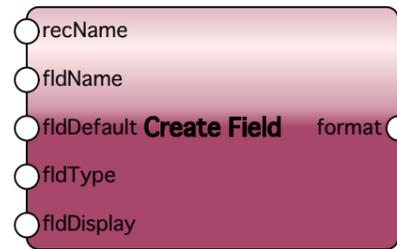
**ポートの説明**  
 inObj: レコードを割り当てるオブジェクト。  
 recName: オブジェクトに割り当てるレコードの名前。  
 obj: 結果のオブジェクト。  
 result: レコードが存在しない場合False、操作が成功するとtrue。



**ノードの説明**  
 データパレットにシンプルなIFCオブジェクトタイプの選択項目を作成します。

**コントロールの説明**  
 IFC Objects (Simple): IFCオブジェクトタイプのシンプルリストです。

**ポートの説明**  
 ifcName: IFCオブジェクトタイプ名。



**ノードの説明**  
 レコードフォーマットの定義フィールドを作成します。フォーマットが存在しない場合はフォーマットを作成します。

**ポートの説明**  
 recName: フィールドを追加するレコードの名前。  
 fldName: 新しいフィールドの名前。  
 fldDefault: 新しいフィールドの初期値。  
 fldType: 新しいフィールドのデータタイプ。  
 fldDisplay: フィールドの表示形式。  
 format: 結果のレコードフォーマット。



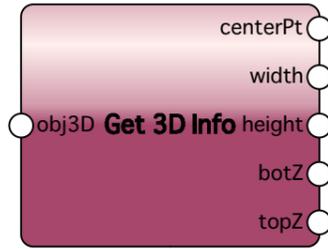
**ノードの説明**  
 指定した2Dオブジェクトの中心の座標を返します。

**ポートの説明**  
 obj2D: 指定する2Dオブジェクト。  
 cenPt: 取得したオブジェクトの中心の座標。



**ノードの説明**  
 図形の指定した頂点の位置を返します。

**ポートの説明**  
 obj2D: オブジェクトの情報。  
 index: 頂点のインデックス。  
 pnt2D: 頂点の座標。



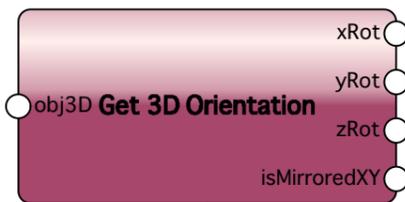
**ノードの説明**  
 対象のオブジェクトの高さ、幅、奥行きを返します。

**ポートの説明**  
 obj3D: 3Dオブジェクトの情報。  
 centerPt: オブジェクトの3D中心の座標。  
 width: オブジェクトの幅。  
 height: オブジェクトの高さ。  
 botZ: オブジェクトのバウンダリボックス底面の高さ。  
 topZ: オブジェクトのバウンダリボックス上面の高さ。



**ノードの説明**  
 オブジェクトの質量中心を通る軸での慣性モーメントを返します。

**ポートの説明**  
 obj3D: 3Dオブジェクトの情報。  
 iX: オブジェクトの質量中心を通るX軸に対する慣性モーメント。  
 iY: オブジェクトの質量中心を通るY軸に対する慣性モーメント。  
 iZ: オブジェクトの質量中心を通るZ軸に対する慣性モーメント。



**ノードの説明**  
 指定した3次元図形の回転角度を返します。オブジェクトをミラー反転している場合、オブジェクトの向きを再現するために、上記の角度によって回転する前に反転を適用する必要があります。

**ポートの説明**  
 obj3D: 3Dオブジェクトの情報。  
 xRot: X軸の回転角度。  
 yRot: Y軸の回転角度。  
 zRot: Z軸の回転角度。  
 mirror: オブジェクトのミラー反転状況。



**ノードの説明**  
 図形の質量中心を通る平面での慣性相乗モーメントを返します。

**ポートの説明**  
 obj3D: 3Dオブジェクトの情報。  
 success: ノードが成功するとTrue、それ以外の場合はfalse。  
 iXY: オブジェクトの質量の中心を通るYZとXZ平面に対する慣性相乗モーメント。  
 iYZ: オブジェクトの質量の中心を通るXZとXY平面に対する慣性相乗モーメント。  
 iZX: オブジェクトの質量の中心を通るXYとYZ平面に対する慣性相乗モーメント。

obj **Get Angle** angle

ノードの説明  
指定した線分や円弧の角度を返します。

ポートの説明  
obj : オブジェクトの情報。  
angle : オブジェクトの角度。

obj **Get Area** area

ノードの説明  
図形の面積を返します。

ポートの説明  
obj : オブジェクトの情報。  
area : 図形の面積。

obj **Get Bounding Box** topL2D botR2D

ノードの説明  
スクリーンプレーン上にオブジェクトが投影するバウンダリボックスの座標を返します。

ポートの説明  
obj : オブジェクトの情報。  
topL2D : バウンダリボックスの左上の座標。  
botR2D : バウンダリボックスの右下の座標。

obj3D **Get Gravity Center** pnt3D

ノードの説明  
3Dオブジェクトの重心を返します。

ポートの説明  
obj3D : 3Dオブジェクト。  
pnt3D : 3Dオブジェクトの重心。

obj **Get Height** height

ノードの説明  
指定したオブジェクトの高さを返します。

ポートの説明  
obj : オブジェクトの情報。  
height : オブジェクトの高さ。

obj **Get IFC Entity** ifcEntityName result

ノードの説明  
対象の図形に割り当てられたIFCエンティティの名前を返します。

ポートの説明  
inObj : オブジェクトの情報。  
obj : 元の図形。  
ifcEntityName : IFCエンティティの名前。  
result : 成功するとtrue、失敗するとfalse。 - 不適切なハンドルまたはオブジェクトにIFCデータが無いと失敗します。

obj ifcPropName **Get IFC Property** result value type

ノードの説明  
指定した基本IFCプロパティとタイプを返します。

ポートの説明  
inObj : オブジェクトの情報。  
ifcPropName : プロパティ名。  
obj : 元の図形。  
result : 成功するとTrue、失敗するとFalseを返します。 - 不適切なハンドルやオブジェクトにIFCデータがない、または名前が不適切だと失敗します。  
value : プロパティの値。  
type : プロパティのタイプ。

obj ifcPSetName ifcPropName **Get IFC pSet Value** result value type

ノードの説明  
指定したIFCプロパティセットフィールドの値とタイプを返します。  
プロパティセットが無効であるか、割り当てられていない場合はfalseを返します。

ポートの説明  
inObj : オブジェクトの情報。  
ifcPSetName : プロパティセットの名前。  
ifcPropName : プロパティ名。  
obj : 元の図形。  
result : 成功するとTrue、失敗するとFalseを返します。 - 不適切なハンドルやオブジェクトにIFCデータがない、またはプロパティやプロパティセットの名前が不適切だと失敗します。  
value : プロパティの値。  
type : プロパティのタイプ。

obj **Get Length** length

ノードの説明  
図形の長さを返します。

ポートの説明  
obj : オブジェクトの情報。  
length : 図形の長さ。

obj3D **Get Mesh Vertex** pnt3D index

ノードの説明  
メッシュオブジェクトの指定した頂点を返します。

ポートの説明  
obj3D : メッシュ図形の情報。  
index : 頂点のインデックス。  
pnt3D : 頂点の座標。

obj3D **Get Mesh Vertex Count** vertNum

ノードの説明  
メッシュ図形の頂点数を返します。

ポートの説明  
obj3D : メッシュ図形の情報。  
vertNum : メッシュ図形の頂点数。

obj **Get Name** string

ノードの説明  
指定したオブジェクトの名前を返します。  
オブジェクトに名前が割り当てられていない場合は「none」を返します。

ポートの説明  
obj : 指定するオブジェクト。  
name : 指定したオブジェクトの名前。



**ノードの説明**  
値またはVectorworksオブジェクトのプロパティの状態を返します。

**コントロールの説明**  
Property Type :  
Vectorworksオブジェクトのプロパティタイプを選択します。

**ポートの説明**  
obj : オブジェクトの情報。  
index :  
オブジェクトのプロパティのインデックスです。特定のオブジェクトセレクトインデックス値については、付録 (Appendix) を参照してください。  
value : Boolean:  
オブジェクトプロパティの状態がONならTrue、OFFならばfalse。 Point:  
操作が成功すると2Dまたは3Dの座標、それ以外の場合は何も返しません。 その他のプロパティ:  
そのタイプに対応する値。



**ノードの説明**  
対象オブジェクトの親コンテナオブジェクトを返します。

**ポートの説明**  
obj : オブジェクトの情報。  
parentObj :  
入力オブジェクトの親コンテナオブジェクト。入力オブジェクトがコンテナを持っていない場合は、レイヤが返されます。



**ノードの説明**  
図形の周囲を計算します。

**ポートの説明**  
obj : オブジェクトの情報。  
perim : 図形の周長。



**ノードの説明**  
指定した図形の平面参照番号を取得します。

**ポートの説明**  
obj : オブジェクトの情報。  
pRefId : 図形の平面参照番号。



**ノードの説明**  
データレコードのフィールドの値を返します。データレコードが存在しないか、または割り当てられていない場合はfalseを返します。

**ポートの説明**  
inObj : オブジェクトの情報。  
recName : レコードフォーマット名。  
fldName : フィールド名。  
obj : 元の図形。  
fldValue : フィールドの値。  
result :  
成功するとTrueを、それ以外はFalseを返します。



**ノードの説明**  
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。

**ポートの説明**  
obj : シンボルやプラグインオブジェクトの情報。  
point : オブジェクトの位置。



**ノードの説明**  
指定したシンボルやプラグインオブジェクトの回転角度 (度数) を返します。

**ポートの説明**  
obj : シンボルやプラグインオブジェクトの情報。  
rotAng :  
シンボルやプラグインオブジェクトの回転角度 (度数)。



**ノードの説明**  
指定したプレーナー又はスクリーン図形のタイプのインデックスを返します。サポートされているオブジェクトタイプの完全なリストは、付録 (Appendix) で見つけることができます。

**ポートの説明**  
obj : オブジェクトの情報。  
type : 図形のタイプ。



**ノードの説明**  
対象のメッシュ、3D多角形、NURBS曲線の指定した頂点の座標を返します。

**ポートの説明**  
obj3D : 3Dオブジェクトの情報。  
index : 頂点のインデックス。  
pnt3D : 頂点。



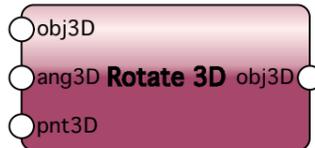
**ノードの説明**  
対象図形の幅を返します。

**ポートの説明**  
obj : オブジェクトの情報。  
width : オブジェクトの幅。



**ノードの説明**  
このノードの親オブジェクトを返します。

**ポートの説明**  
handle :



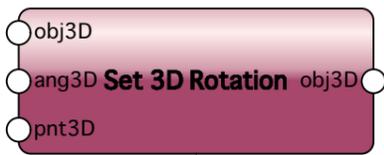
**ノードの説明**  
指定した3Dオブジェクトを指定した3D座標を軸に回転します。適用できる3Dオブジェクトは、柱状体、多段柱状体、メッシュ、3D多角形、ソリッド図形、CSGソリッド図形、グループ図形、シンボル、プラグインオブジェクト、NURBS曲線、NURBS曲面です。

**ポートの説明**  
obj3D : 3Dオブジェクトの情報。  
ang3D : X、Y、Z軸に対する回転角度。  
pnt3D : 回転軸の座標。  
object\_ : 回転したオブジェクト。



**ノードの説明**  
指定したオブジェクトの高さ、幅、奥行きを設定します。

**ポートの説明**  
obj3D : 3Dオブジェクトの情報。  
height : 設定する高さ。  
width : 設定する幅。  
depth : 設定する奥行き。  
object\_ : 結果のオブジェクト。



**ノードの説明**

指定したオブジェクトを指定した角度と軸で回転させます。次の3Dオブジェクトで使用できます：柱状体、多段柱状体、回転体。

**ポートの説明**

obj3D：3Dオブジェクトの情報。  
ang3D：X、Y、Z軸に対する回転角度。  
pnt3D：回転軸の座標。  
object\_：回転したオブジェクト。



**ノードの説明**

指定したオブジェクトの回転角度を設定します。

**ポートの説明**

inObj：オブジェクトの情報。  
angle：オブジェクトに指定する新しい回転角度。  
obj：結果のオブジェクト。



**ノードの説明**

指定したオブジェクトの高さを設定します。

**ポートの説明**

inObj：オブジェクトの情報。  
height：設定するオブジェクトの高さ。  
obj：新しい高さを適用したオブジェクト。



**ノードの説明**

指定したIFCエンティティとIFCレコードを作成して指定したオブジェクトに割り当てます。

**ポートの説明**

inObj：オブジェクトの情報。  
ifcObjName：IFCエンティティの名前。  
obj：結果のオブジェクト。  
result：成功するとTrue、失敗するとFalseを返します。 - ハンドルやIFCエンティティ名が不適切だと失敗します。



**ノードの説明**

基本的IFCプロパティセットを作成して指定したオブジェクトに割り当てます。

**ポートの説明**

inObj：オブジェクトの情報。  
ifcPropName：プロパティ名。  
ifcPropValue：プロパティの値。  
obj：結果のオブジェクト。  
result：成功するとTrue、失敗するとFalseを返します。 - 不適切なハンドルやオブジェクトにIFCデータがない、または名前が不適切だと失敗します。



**ノードの説明**

IFCのプロパティセットのプロパティの値を設定します。プロパティセットが割り当てられていない場合は、指定したIFCのプロパティセットも割り当てます。

**ポートの説明**

inObj：オブジェクトの情報。  
ifcPSetName：プロパティセットの名前。  
ifcPropName：プロパティ名。  
ifcPropValue：プロパティの値。  
obj：元の図形。  
result：成功するとTrue、失敗するとFalseを返します。 - 不適切なハンドルやオブジェクトにIFCデータがない、またはプロパティやプロパティセットの名前が不適切だと失敗します。



**ノードの説明**

指定したメッシュの頂点を、指定した新しい座標に設定します。

**ポートの説明**

inObj3D：3Dメッシュオブジェクト。  
index：変更する頂点のインデックス。  
pnt3D：新しい頂点座標。  
obj3D：作成されるメッシュオブジェクト。



**ノードの説明**

指定したオブジェクトの名前を設定します。

**ポートの説明**

inObj：オブジェクトの情報。  
string：オブジェクトに設定する名前。  
obj：結果のオブジェクト。



**ノードの説明**

指定したオブジェクトの平面参照番号を設定します。

**ポートの説明**

inObj：オブジェクトの情報。  
pRefID：平面の参照番号。  
obj：結果のオブジェクト。



**ノードの説明**

指定したオブジェクトを基準平面上に設定します。パラメトリックオブジェクトの中で使用します。

**ポートの説明**

inObj：オブジェクトの情報。  
obj：結果のオブジェクト。

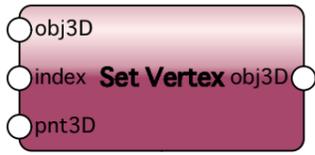


**ノードの説明**

レコードフィールドの値を設定します。レコードが存在していてもオブジェクトに割り当てられていない場合は、オブジェクトにレコードフォーマットを割り当てます。フィールド名が正しくない、またはレコードが存在しない場合は何も変更せずにFalseを返します。

**ポートの説明**

inObj：オブジェクトの情報。  
recName：レコードの名前。  
fldName：レコードフィールドの名前。  
fldValue：レコードフィールドの値。  
obj：結果のオブジェクト。  
result：成功するとTrue、それ以外の場合はFalse。



**ノードの説明**  
 3Dオブジェクトの指定した頂点の座標を設定します。

**ポートの説明**  
 obj3D : 3Dオブジェクトの情報。  
 index : 変更する頂点のインデックス。  
 pnt3D : 指定した頂点の新しい座標。  
 obj\_ : 結果のオブジェクト。



**ノードの説明**  
 指定したオブジェクトの幅を設定します。

**ポートの説明**  
 inObj : オブジェクトの情報。  
 width : オブジェクトに設定する幅。  
 obj : 指定した新しい幅を持つ結果のオブジェクト。



**ノードの説明**  
 ソリッドの表面積を計算します。

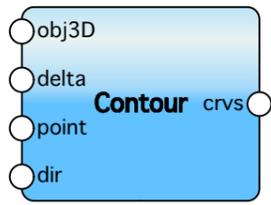
**ポートの説明**  
 obj3D : 対象のソリッドオブジェクト。  
 surfArea : ソリッドオブジェクトの表面積。



**ノードの説明**  
 ソリッドオブジェクトの体積を計算します。

**ポートの説明**  
 obj3D : 対象のソリッドオブジェクト。  
 volume : ソリッドオブジェクトの体積。

# NURBS



**ノードの説明**  
ソリッド図形に等高線を作成します。

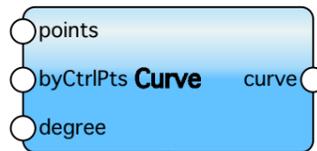
**ポートの説明**  
obj3D : ソリッド図形。  
delta : 等高線の間隔。  
point : 等高線を定義するために使われている平面上の点の座標。  
dir : 平面の法線ベクトル。  
crvs : 等高線を表すためのNURBS曲線を含むグループ。



**ノードの説明**  
入力された図形をNURBS図形やNURBS図形のグループに変換します。

**コントロールの説明**  
Keep Original Object : 元のオブジェクトを保持する場合はチェックを入れ、そうでない場合はチェックを外します。

**ポートの説明**  
obj : オブジェクトの情報。  
nurbs : 結果のNURBS図形。



**ノードの説明**  
座標のリストからNURBS曲線を作成します。

**ポートの説明**  
points : 3D座標のリスト。  
byCtrlPts : このノードが(補間しない)制御点を使用して曲線を作成する場合Trueを、そうでない場合はfalse。  
degree : NURBS曲線の次数。  
curve : 新規に作成されたNURBS曲線。



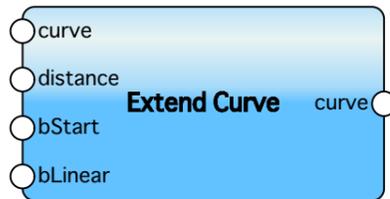
**ノードの説明**  
ハンドルで指定したNURBS曲線または曲面の頂点を削除します。

**ポートの説明**  
inNurbsObj : NURBS曲線またはNURBS曲面。  
index1 : NURBS曲線では、このインデックスはクエリする曲線の部分/セグメントを指定します。NURBS曲面では、このインデックスは曲面のU方向またはV方向を指定します(列、index1 = 1 : U方向、行、index1 = 0 : V方向)。インデックスはゼロから始まります。  
index2 : NURBS曲線では、このインデックスはインデックス1のセグメントにある頂点です。NURBS曲面では、このインデックスはインデックス1で定義した方向にある頂点の全体の行(または列)です。インデックスはゼロから始まります。  
nurbsObj : 削除後のNURBS図形。



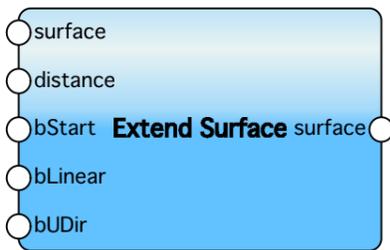
**ノードの説明**  
NURBS曲線に沿った3D座標のリストを返します。座標は同じ長さのセグメントに曲線を分割したものです。

**ポートの説明**  
curve : NURBS曲線。  
divs : 曲線の分割数。  
pnts3D : 結果の座標リスト。



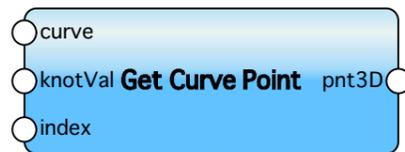
**ノードの説明**  
始点または終点からの距離を指定してNURBS曲線を延長します。直線状または曲率に沿って延長することができます。

**ポートの説明**  
inCurve : NURBS曲線。  
distance : 曲線の延長距離。  
bStart : 延長する点(TRUE : 始点/FALSE : 終点)。  
bLinear : 延長する形状(True : 直線、False : 終点と同じ曲率の曲線(曲面))。  
curve : 結果の曲線。



**ノードの説明**  
U方向やV方向の始点または終点までの距離を指定してNURBS曲面を延長します。

**ポートの説明**  
inSurface : NURBS曲面。  
distance : 曲面の延長距離。  
bStart : 延長する点(TRUE : 始点/FALSE : 終点)。  
bLinear : 延長する形状(True : 直線、False : 終点と同じ曲率の曲線(曲面))。  
bUDir : 延長する方向(TRUE : U方向/FALSE : V方向)。  
surface : 延長したNURBS曲面。



**ノードの説明**  
NURBS曲線上で指定したインデックスの辺の指定されたノット値の座標を返します。インデックスはゼロベースです(0からノット数-1)。ノット値は0から曲線セグメントの最後のノット値の範囲にすることができます。

**ポートの説明**  
curve : NURBS曲線。  
knotVal : ノット値の最小値と最大値の間になるパラメータ。  
index : 辺のインデックス。  
pnt3D : 結果の座標。



**ノードの説明**  
指定したNURBS曲線/曲面の、指定した辺の次数を返します。

**ポートの説明**  
nurbsObj : NURBS曲線またはNURBS曲面。  
index : NURBS曲線の場合、インデックスは曲線辺の番号を示します。NURBS曲面の図形表裏面の場合、インデックスは、U方向は1を、V方向は0を指定します。インデックスはゼロベースです。  
degree : 辺の次数(整数)。



#### ノードの説明

入力点からNURBSオブジェクトへの距離を返します。

#### ポートの説明

nurbsObj : NURBS曲線またはNURBS曲面。

pnt2D : 2Dの入力点 (座標)。

success :

ノードが成功したらTrue、そうでない場合はfalse。

distance :

入力点とNURBSオブジェクト間の距離。



#### ノードの説明

指定したNURBS曲線/曲面の、指定した頂点のノット値を返します。

#### ポートの説明

nurbsObj :

NURBS曲線またはNURBS曲面。

index1 :

NURBS曲線では、このインデックスはクエリする曲線の部分/セグメントを指定します。NURBS曲面では、このインデックスは曲面のU方向またはV方向を指定します (index1 = 1 : U方向、index1 = 0 : V方向)。インデックスはゼロから始まります。

index2 :

NURBS曲線では、このインデックスはインデックス1のセグメントにあるノットです。NURBS曲面では、このインデックスはインデックス1で定義した方向にあるノットです。インデックスはゼロから始まります。

knot : 結果のノット値。



#### ノードの説明

対象のNURBS曲線または曲面のノットの数を返します。

#### ポートの説明

nurbsObj : NURBS曲線またはNURBS曲面。

index :

オブジェクトが曲線の場合はインデックスは曲線辺の番号を示します。オブジェクトが曲面の場合には、インデックスは、U方向は1を、V方向は0を指定します。

numKnots : NURBS図形のノットの数。



#### ノードの説明

入力点を投影した点のパラメータを返します。また、点が投影されたNURBS曲線の番号も返します。

#### ポートの説明

curve : NURBS曲線。

pt : 座標の情報。

success : ノードが成功するとTrue。

param : 投影点のパラメータ。

index :

座標が投影するNURBS曲線のインデックス。



#### ノードの説明

対象のNURBS曲線を構成する辺の数を返します。

#### ポートの説明

curve : NURBS曲線。

numPieces : NURBS曲線を構成している辺の数。



#### ノードの説明

対象のNURBS曲線や曲面の頂点の座標を返します。

#### ポートの説明

nurbsObj :

NURBS曲線またはNURBS曲面。

index1 :

曲線の場合は、参照するセグメントを指定します。曲面の場合は、参照する制御点のU方向のインデックスを指定します。インデックスは0から始まります。

index2 :

曲線の場合は、index1のセグメント内の制御点のインデックスを指定します。曲面の場合は、参照する制御点のV方向のインデックスを指定します。インデックスは0から始まります。

point : 結果の3D座標。



#### ノードの説明

NURBS曲線のセグメント内の制御点の数、または、NURBS曲面のUまたはV方向の制御点の数を返します。

#### ポートの説明

nurbsObj : NURBS曲線またはNURBS曲面。

index :

NURBS曲線の場合、インデックスは曲線辺の番号を示します。

NURBS曲面の図形表裏面の場合、インデックスは、U方向は1を、V方向は0を指定します。インデックスはゼロベースです。

numPts : 結果の制御点の数。



#### ノードの説明

指定したNURBS曲線の指定した辺の種類を返します。インデックスはゼロから始まります (0から辺の数-1)。

返り値は曲線を作成した補間点(True)または制御点(False)がboolean値で示されます。

#### ポートの説明

curve : NURBS曲線。

index : 辺のインデックス。

isByFit : 辺の種類。



#### ノードの説明

指定した方向のNURBS曲面上の点を返します。パラメータ値は0から各方向における最後のノット値の範囲です。

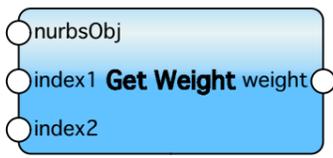
#### ポートの説明

surface : NURBS曲面。

dir : ポイントの指定した方向 (U、V)。

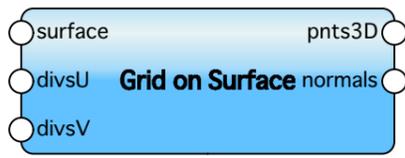
U及びVは、その方向での最小と最大のノット値の間になります。

point : 結果の座標。



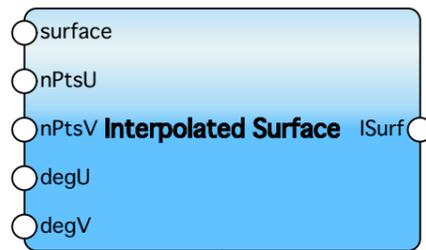
**ノードの説明**  
 NURBS曲線/曲面の指定した頂点の重みを返します。

**ポートの説明**  
 nurbsObj : NURBS曲線またはNURBS曲面。  
 index1 : 曲線の場合は、参照するセグメントを指定します。曲面の場合は、参照する制御点のU方向のインデックスを指定します。インデックスは0から始まります。  
 index2 : 曲線の場合は、index1のセグメント内の制御点のインデックスを指定します。曲面の場合は、参照する制御点のV方向のインデックスを指定します。インデックスは0から始まります。  
 weight : NURBS図形上の頂点の重み。



**ノードの説明**  
 NURBS曲面上の3D座標のグリッド (U x V) を返します。

**ポートの説明**  
 surface : NURBS曲面図形。  
 divsU : U方向の辺の数。  
 divsV : V方向の辺の数。  
 pnts3D : グリッド上の点のリスト。  
 normals : グリッドポイントでそれぞれ計算した曲面の法線ベクトルのリスト。



**ノードの説明**  
 指定した次数と補間点の数で既存のNURBS曲面の面を補間し3Dを作成します。NURBS曲面が指定されない場合は平面的な長方形の面が作成されます。

**ポートの説明**  
 surface : 補間するNURBS曲面。  
 nPtsU : NURBS曲面のUパラメータ方向の補間点の数。Uパラメータ方向の次数よりも大きくする必要があります。  
 nPtsV : NURBS曲面のVパラメータ方向の補間点の数。Vパラメータ方向の次数よりも大きくする必要があります。  
 degU : 補間点によるNURBS曲面のUパラメータ方向の次数。  
 degV : 補間点によるNURBS曲面のVパラメータ方向の次数。  
 ISurf : 作成される補間点によるNURBS曲面。



**ノードの説明**  
 NURBS曲線のグループから多段曲面を作成します。NURBS曲線は、それらがグループに追加された順序で多段曲面を生成します。

**コントロールの説明**  
 rule : チェックを入れると直線的に補完された図形を作成します。  
 close : チェックを入れると閉じた多段曲面を作成します。始点は多段曲面の終点として選択できないため、このオプションを選択すると、自動的に始点と終点を接続した図形が作成されます。  
 solid : チェックを入れると端点が閉じられ、すべての曲面が互いに閉じられたソリッド多段曲面を作成します。汎用ソリッドではなくNURBS曲面のグループを作成する場合は選択を解除します。

**ポートの説明**  
 groupCurves : NURBS曲線のグループ。  
 surface : 結果の多段曲面。



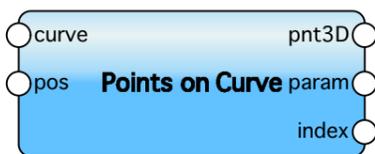
**ノードの説明**  
 指定したオフセット距離を使用して、元の位置からオフセットしたNURBSオブジェクトを作成します。

**ポートの説明**  
 oriNurbsObj : 元のNURBSオブジェクト。  
 offset : オフセットする距離の値。  
 nurbsObj : オフセットされたNURBSオブジェクト。



**ノードの説明**  
 NURBS曲面上の指定した方向の3D座標とその座標で計算した法線ベクトルを返します。

**ポートの説明**  
 surface : NURBS曲面。  
 dir : 座標のU方向とV方向です。UとVの値は0と1の間である必要があります。  
 pnt3D : 結果の座標。  
 normal : 結果の座標で計算したNURBS曲面の法線ベクトル。



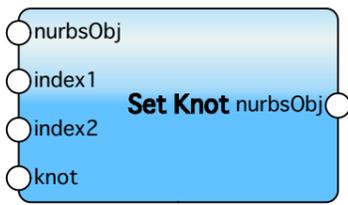
**ノードの説明**  
 NURBS曲線上の指定した位置の頂点、パラメータ、曲線のインデックス番号を返します。位置の値は0から1の間である必要があります。

**ポートの説明**  
 curve : NURBS曲線。  
 pos : NURBS曲線上の位置。  
 pnt3D : 結果の座標。  
 param : 結果のパラメータ。  
 index : 結果のインデックス。



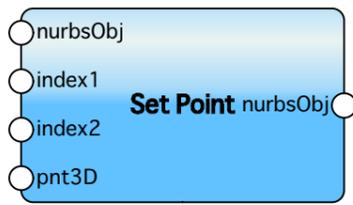
**ノードの説明**  
 輪郭を示す曲線を指定した軸と軌道に沿って回転させ、NURBS曲面またはNURBS曲面のグループを作成します。

**ポートの説明**  
 profile : 輪郭オブジェクトとして使用するNURBS曲線。  
 rail : 軌道オブジェクトとして使用するNURBS曲線。  
 axis : 輪郭を回転させる軸として使用する直線状のNURBS曲線。  
 surface : 結果のNURBS曲面。



**ノードの説明**  
指定したNURBS曲線/曲面のノットに、指定したノット値を設定します。

**ポートの説明**  
inNurbsObj : NURBS曲線またはNURBS曲面。  
index1 : NURBS曲線では、このインデックスはクエリする曲線の部分/セグメントを指定します。NURBS曲面では、このインデックスは曲面のU方向またはV方向を指定します (index1 = 1 : U方向、index1 = 0 : V方向)。インデックスはゼロから始まります。  
index2 : NURBS曲線では、このインデックスはインデックス1のセグメントにあるノットです。NURBS曲面では、このインデックスはインデックス1で定義した方向にあるノットです。インデックスはゼロから始まります。  
knot : 入力するノット値。  
nurbsObj : 結果のNURBS図形。



**ノードの説明**  
指定したNURBS曲線/曲面の頂点の座標を、指定した頂点の座標に設定します。

**ポートの説明**  
inNurbsObj : NURBS曲線またはNURBS曲面。  
index1 : 曲線の場合は、参照するセグメントを指定します。曲面の場合は、参照する制御点のU方向のインデックスを指定します。インデックスは0から始まります。  
index2 : 曲線の場合は、index1のセグメント内の制御点のインデックスを指定します。曲面の場合は、参照する制御点のV方向のインデックスを指定します。インデックスは0から始まります。  
pnt3D : 入力する3D座標。  
nurbsObj : 結果のNURBS図形。



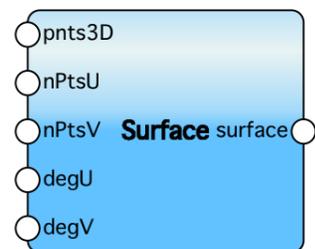
**ノードの説明**  
NURBS曲線/曲面の指定した頂点の重みを設定します。

**ポートの説明**  
inNurbsObj : NURBS曲線またはNURBS曲面。  
index1 : 曲線の場合は、参照するセグメントを指定します。曲面の場合は、参照する制御点のU方向のインデックスを指定します。インデックスは0から始まります。  
index2 : 曲線の場合は、index1のセグメント内の制御点のインデックスを指定します。曲面の場合は、参照する制御点のV方向のインデックスを指定します。インデックスは0から始まります。  
weight : 指定する頂点の重み。  
nurbsObj : 結果のNURBS図形。



**ノードの説明**  
NURBS曲面からシェルソリッドを作成します。正数値の厚みは曲面の法線ベクトル方向になります。

**ポートの説明**  
surface : NURBS曲面。  
thick : シェルの厚みを定義する正の数値。  
shell : 作成されるシェルソリッドオブジェクト



**ノードの説明**  
座標のリストからNURBS曲面を作成します。

**ポートの説明**  
pnts3D : NURBS曲面の3D座標のリスト。  
nPtsU : 曲面のU制御点の数。  
nPtsV : 曲面のV制御点の数。  
degU : 曲面のU方向の次数。  
degV : 曲面のV方向の次数。  
surface : 結果のNURBS曲面。



**ノードの説明**  
選択されている交差した曲線からNURBS曲面を作成します。

**ポートの説明**  
success : NURBS曲面が作成されるとTrueを返し、それ以外はFalseを返します。  
surface : 作成されるNURBS曲面。



**ノードの説明**  
NURBS曲線でNURBS曲面をトリミングします。

**ポートの説明**  
surface : NURBS曲面。  
curve : NURBS曲線。  
success : NURBS曲面がトリミングされ、NURBS曲線がNURBS曲面にトリミングされた曲線として連結された場合はTrueを返し、それ以外はFalseを返します。

# Document Structure

## Default Fill Color

ノードの説明  
アクティブクラスの面の色にドキュメント初期値の面の色を指定します。

## Default Fill Pattern

ノードの説明  
アクティブクラスの面の模様 to ドキュメント初期値の面の模様を指定します。

## Default Line Style

ノードの説明  
アクティブクラスの線種にドキュメント初期値の線種を指定します。

## Default Line Weight

ノードの説明  
アクティブクラスの線の太さにドキュメント初期値の線の太さを指定します。

## Default Opacity

ノードの説明  
アクティブクラスの不透明度にドキュメント初期値の不透明度を指定します。

## Default Pen Color

ノードの説明  
アクティブクラスの線の色にドキュメント初期値の線の色を指定します。

## string Delete Class

ノードの説明  
アクティブなドキュメントから指定した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。

ポートの説明  
string : 削除するクラスの名前。

## Get Active Class string

ノードの説明  
アクティブなクラスの名前を返します。

ポートの説明  
string : 現在のファイル内のアクティブクラス名。

## Get Class Count classNum

ノードの説明  
アクティブなドキュメント内のクラスの合計数を返します。

ポートの説明  
classNum : クラスの数。

## index Get Class List string

ノードの説明  
ドキュメントクラスリスト内で指定したインデックスのクラス名を返します。たとえば、「ClassList(4)」はリストで4番目の名前を返します。

ポートの説明  
index : クラスリスト内のクラスのインデックス (1 ~ N の範囲)。  
string : 指定したクラスの名前。



ノードの説明  
指定したクラスのマーカーのすべての設定値を返します。

コントロールの説明  
Setting : 変更するマーカーの位置を指定します。

ポートの説明  
string : クラスの名前。  
success : 成功するとTrue、それ以外の場合はFalse。  
style : マーカースタイル。  
angle : マーカーの角度 (度数単位で0から90) です。整数値で指定します。  
size : マーカーサイズ (インチ)。  
width : マーカーの幅 (インチ)。  
thickBasis : マーカーの太さの定数。整数値で指定します。  
thickness : マーカーの幅 (インチ)。

## string Get Class Opacity opacity

ノードの説明  
指定したクラスの不透明度を返します。

ポートの説明  
string : クラスの名前。  
opacity : クラスの不透明度。範囲内でパーセント [0から100] で返します。

## Get Class Options index

ノードの説明  
アクティブなドキュメントのクラスの他のクラスの表示方法を値で返します。

ポートの説明  
index : ドキュメントの現在のクラスの表示設定 (「他のクラスを」の設定) (整数値)。1 - 非表示, 2 - グレイ表示, 3 - 表示, 4 - 表示+スナップ, 5 - 表示+スナップ+編集, 6 - グレイ表示+スナップ

## string Get Fill Color fillColor

ノードの説明  
指定したクラスの塗りつぶし背景や前景の色の設定を返します。

コントロールの説明  
Side : 背景色または前景色を選択して設定します。

ポートの説明  
string : クラスの名前。  
fillColor : RGB3色の値 (範囲は0から65535)。

string **Get Fill Pattern** pattern

ノードの説明  
クラスに設定されている面の模様を返します。

ポートの説明

string : クラスの名前。

pattern :

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。パターンなどに関連する定数はFunction ReferenceのAppendixに記載されています。

string **Get Hatch Pattern** pattern

ノードの説明  
指定したクラスに設定しているハッチングの名前を返します。クラスにハッチングを設定していない場合は空の文字列が返されます。

ポートの説明

string : クラスの名前。

pattern : ハッチング名。

string **Get Line Style** style

ノードの説明  
指定したクラスの線種を返します。

ポートの説明

string : クラスの名前。

style : 線種。

string **Get Line Weight** weight

ノードの説明  
指定したクラスの線の太さを返します。

ポートの説明

string : クラスの名前。

weight : 線の太さ。

string **Get Pen Color** penColor

ノードの説明  
指定したクラスの線の色を設定を返します。

コントロールの説明

Side : 背景色または前景色を選択して設定します。

ポートの説明

string : クラスの名前。

penColor : RGB3色の値 (範囲は0から65535)。

string **Get Text Style** txtStyle

ノードの説明  
指定したクラスの文字スタイルを返します。

ポートの説明

string : クラスの名前。

txtStyle :

テキストスタイルの内部インデックス (整数)。

string **Get Visibility Status** visibility

ノードの説明  
クラスの表示状態を返します。

ポートの説明

string : クラスの名前。

visibility : 0 - 表¥示, -1 - 非表¥示, 2 - グレイ表¥示

string **Gray Class**

ノードの説明  
指定したクラスの表示設定を「グレイ表示」に設定します。

ポートの説明

string : クラスの名前。

string **Hide Class**

ノードの説明  
指定したクラスの表示状態を非表示にします。

ポートの説明

string : クラスの名前。

string **Is Graphic Used** bool

ノードの説明  
オブジェクトの作成時に指定したクラスのグラフィック属性を使用するよう設定されているかどうかを返します。

ポートの説明

string : クラスの名前。

bool :

オブジェクトの作成時にこのクラスのグラフィック属性を使用するように設定されている場合はtrueを返します。falseの場合、このクラスで作成されるオブジェクトは、デフォルトのグローバル属性設定からデフォルトの属性を取得することを示します。

string **Is Text Style Used** bool

ノードの説明  
オブジェクトの作成時に指定したクラスの文字スタイルを使用するよう設定されているかどうかを返します。

ポートの説明

string : クラスの名前。

bool :

オブジェクトの作成時に指定したクラスの文字スタイルを使用するように設定されていればtrueを返し、そうでない場合はfalseを返します。

string **Name Class**

ノードの説明  
Vectorworksファイルに新しいクラスを新規作成します。作成後、新しいクラスがアクティブクラスになります。指定したクラス名がすでに存在している場合は、そのクラスがアクティブクラスになります。注記 : クラス名は最大で63文字までです。

ポートの説明

string :

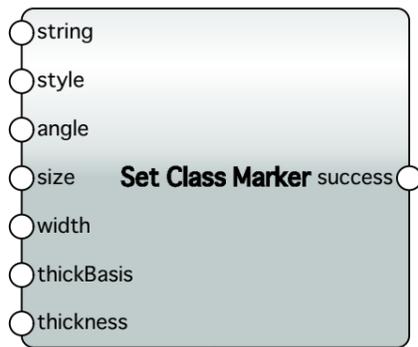
className **Rename Class** newName

ノードの説明  
指定した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。

ポートの説明

className : 現在のクラス名。

newName : 新しいクラス名。



**ノードの説明**  
指定したクラスのマーカーのすべてのプロパティを設定します。

**コントロールの説明**  
Setting : 変更するマーカーの位置を指定します。

**ポートの説明**  
string : クラスの名前。  
style : マーカースタイル。  
angle : マーカーの角度(0から90度)。  
size : マーカーサイズ (インチ)。  
width : マーカーの幅 (インチ)。  
thickBasis : マーカーの太さの定数。  
thickness : マーカーの幅 (インチ)。  
success :  
成功するとTrue、それ以外の場合はFalse。



**ノードの説明**  
指定したクラスの不透明度を設定します。不透明度は0から100の範囲のパーセンテージで指定します。

**ポートの説明**  
string : クラスの名前。  
opacity :  
0から100の範囲のパーセンテージの値で不透明度を指定します。整数値を使用します。



**ノードの説明**  
アクティブファイルの他のクラスの表示状態を設定します。

**ポートの説明**  
index :  
他のクラスの表示設定のインデックスです：  
1 - 非表示、2 - グレイ表示、3 - 表示、4 - 表示+スナップ、5 - 表示+スナップ+編集、6 - グレイ表示+スナップ



**ノードの説明**  
指定したクラスの面の色を設定します。色は赤、緑、青の成分 (RGB) で指定します。RGB値の範囲は0から65535までです。

**コントロールの説明**  
Side :  
背景色または前景色を選択して設定します。

**ポートの説明**  
string : クラスの名前。  
color :  
RGB3色の値 (範囲は0から65535)。



**ノードの説明**  
指定したクラスの面の模様を設定します。模様を指定するには、模様のインデックス番号 (正の数値) を使用します。その他の塗りつぶしパターンを使用するには、ベクトルフィルインデックスの負の数値で指定します (インデックス\* -1)。

**ポートの説明**  
string : クラスの名前。  
pattern : 模様番号。



**ノードの説明**  
指定したクラスにグラフィック属性を使用するように設定します。

**ポートの説明**  
string : クラスの名前。  
use :  
グラフィック属性を使用する場合はTrueを、使用しない場合はFalseを設定します。



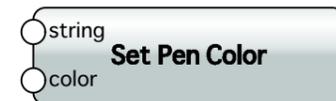
**ノードの説明**  
指定したクラスの線の種類を設定します。

**ポートの説明**  
string : クラスの名前。  
style :  
線の種類のインデックス番号です。整数値で指定します。



**ノードの説明**  
指定したクラスの線の太さを設定します。

**ポートの説明**  
string : クラスの名前。  
weight : 線の太さ (ミル単位)。



**ノードの説明**  
指定したクラスの線の色を設定します。

**コントロールの説明**  
Side :  
背景色または前景色を選択して設定します。

**ポートの説明**  
string :  
color :  
設定する線の色です。色は赤、緑、青 (RGB) の成分で指定します。値の範囲は0から65535までです。



**ノードの説明**  
クラスのハッチングを設定します。

**ポートの説明**  
string : クラスの名前。  
pattern : ハッチングの名前。  
success : 成功するとTrue、それ以外の場合はFalse。



**ノードの説明**  
指定したクラスの文字スタイルを設定します。

**ポートの説明**  
string : クラスの名前。  
style : 指定する文字スタイル番号。



**ノードの説明**  
指定したクラスで文字スタイルを使用するかどうかを設定します。

**ポートの説明**  
string : クラスの名前。  
use :  
文字スタイルを使用する (TRUE) /使用しない (FALSE)。



**ノードの説明**  
オブジェクトの作成時に指定したクラスのテクスチャを使用するかどうかを設定します。

**ポートの説明**  
string : クラスの名前。  
use :  
テクスチャを使用する (TRUE) /使用しない (FALSE)。



**ノードの説明**  
指定したクラスの表示設定を「表示」に設定します。

**ポートの説明**  
string : クラスの名前。

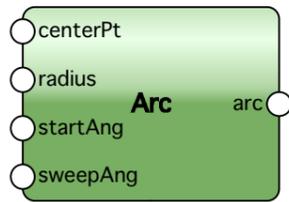
# ジオメトリ 2D



**ノードの説明**  
多角形/曲線に頂点を追加します。円弧指定の頂点を追加する時は半径を指定します。

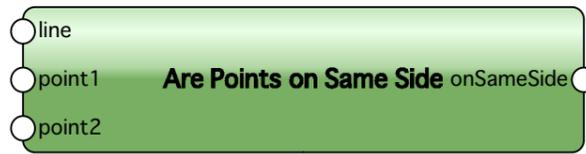
**コントロールの説明**  
Vertex Type :  
追加する新しい頂点の形式を選択します。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
vertex : 多角形/曲線に追加する頂点。  
radius :  
円弧指定の半径。この値は他の形式の頂点では不要です。  
poly : 結果の多角形/曲線。



**ノードの説明**  
提供された入力パラメータを使用して円弧を作成します。

**ポートの説明**  
centerPt : 円弧の2D中心点。  
radius : 円弧の半径。  
startAng :  
円弧の開始角度。度数で指定します。  
sweepAng : 円弧角。度数で指定します。  
arc : パラメータ適用後の円弧。



**ノードの説明**  
指定した2つの点が指定した直線を基準に同じ側にある場合にtrueを返します。

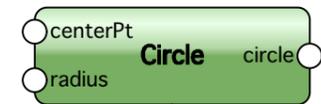
**ポートの説明**  
lineH : 2Dの直線図形。  
point1 : 2Dの座標。  
point2 : 2Dの座標。  
onSameSide :  
判定結果。2つの点が直線を基準に同じ側であればtrue、そうではない場合はfalseを返します。



**ノードの説明**  
指定したインデックスの頂点の種類を変更します。頂点が円弧指定でない場合は、半径の値は必要ありません。

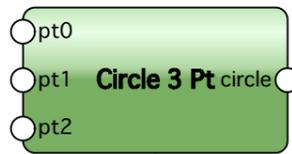
**コントロールの説明**  
Vertex Type :  
指定した頂点の変換後の形式を選択します。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
at : 変更する頂点のインデックス。  
radius :  
円弧指定の半径。この値は他の形式の頂点では不要です。  
poly : 結果の多角形/曲線。



**ノードの説明**  
指定した仕様の円を作成します。

**ポートの説明**  
centerPt : 円の中心点。  
radius : 円の半径。  
circle : 作成する円。



**ノードの説明**  
指定した3点を通る円を作成します。

**ポートの説明**  
pt0 : 1つ目の指定する点。  
pt1 : 2つ目の指定する点。  
pt2 : 3つ目の指定する点。  
circle : 作成される円図形。



**ノードの説明**  
多角形/曲線内で誤差内のすべての同じ頂点を削除します。

**コントロールの説明**  
Tolerance :  
計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
poly : 結果の多角形/曲線。



**ノードの説明**  
入力された多角形を誤差内で円弧指定の頂点を使用した曲線に変換します。

**コントロールの説明**  
Tolerance :  
計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
polyH : 多角形の情報。  
poly : 結果の曲線。



**ノードの説明**  
入力した図形を頂点のみの多角形化します。

**ポートの説明**  
obj2D : 2D図形。  
resolution :  
変換に使用される整数 (integer値)。数が大きいほど、より正確な多角形となります。最大値は32,767です。  
poly : 結果の多角形。



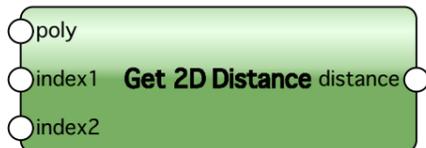
**ノードの説明**  
あらゆる閉じた図形 (円、四角形、楕円など) を多角形もしくは曲線に変換します。多角形化は行いません。

**ポートの説明**  
obj2D : オブジェクトの情報。  
poly : 結果の多角形/曲線。



**ノードの説明**  
多角形/曲線の面積を計算します。多角形/曲線が反時計回りの場合は正の、時計回りの場合は負の面積となります。

**ポートの説明**  
poly : 多角形/曲線の情報。  
area : 多角形/曲線の面積。



**ノードの説明**  
多角形/曲線の2つの頂点間の距離を返します。

**ポートの説明**  
poly : 多角形/曲線の情報。  
index1 : 最初の頂点のインデックス。  
index2 : 2番目の頂点のインデックス。  
distance : 2つの頂点間の距離。



**ノードの説明**  
指定されたインデックスから開始する多角形/曲線の辺の端点座標を返します。多角形/曲線の頂点配列の最後のインデックスの場合、辺の端点は多角形/曲線の頂点配列内の最初の頂点になります。辺は円弧、直線などになるはずですが。

**ポートの説明**  
poly : 多角形/曲線の情報。  
at : 辺の始点のインデックス。  
start : 辺の始点のインデックス。  
end : 辺の終点のインデックス。

poly **Get 2D Perimeter** perim

ノードの説明  
多角形/曲線の周長（頂点間の距離の和）を計算します。

ポートの説明  
poly : 多角形/曲線の情報。  
perim : 多角形/曲線の周長。

poly  
at **Get 2D Vertex** vertex

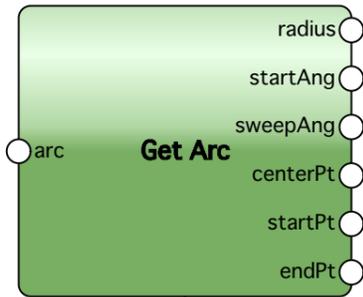
ノードの説明  
多角形/曲線の頂点配列内で指定したインデックスの頂点を返します。

ポートの説明  
poly : 多角形/曲線の情報。  
at : 頂点のインデックス。  
vertex : 頂点。

poly **Get 2D Vertex Count** vertNum

ノードの説明  
入力された多角形/曲線の頂点数を返します。

ポートの説明  
poly : 多角形/曲線の情報。  
vertNum : 多角形/曲線の頂点数。



arc  
point **Get Arc Distance** distance

ノードの説明  
指定した点から円弧までの距離を返します。

ポートの説明  
arc : 対象の円弧。  
point : 2Dの座標。  
distance : 点から円弧までの距離。

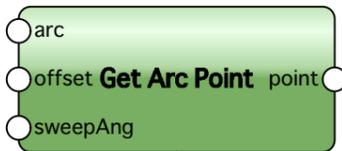
arc **Get Arc Length** length

ノードの説明  
円弧の弧の長さを返します。

ポートの説明  
arc : 対象の円弧。  
length : 円弧の弧の長さ。

ノードの説明  
円弧のパラメータを返します。

ポートの説明  
arc : 対象の円弧。  
radius : 円弧の半径。  
startAng : 円弧の開始角度（度数）。  
sweepAng : 円弧角（度数）。  
centerPt : 円弧の中心。  
startPt : 円弧の始発の座標。  
endPt : 円弧の終着の座標。



poly  
at **Get Arc Vertex Radius** radius

ノードの説明  
多角形/曲線の頂点配列内のインデックスで指定した円弧指定の頂点の半径を返します。

ポートの説明  
poly : 多角形/曲線の情報。  
at : 多角形/曲線の頂点配列内の円弧指定の頂点のインデックス。  
radius : 円弧指定の頂点の半径。



ノードの説明  
円の中心点と半径を返します。

ポートの説明  
circle : 円の情報。  
centerPt : 円の中心点。  
radius : 円の半径。

ポートの説明  
arc : 対象の円弧。  
offset : 円弧の始点から結果の座標までの距離。  
sweepAng : 円弧の中心を基準とした、円弧の始点から結果の座標までの角度。  
point : 結果の座標。

circle **Get Circle Length** length

ノードの説明  
円の周長を返します。

ポートの説明  
circle : 円の情報。  
length : 円の周長。



ノードの説明  
円上の点を計算して返します。注：オフセットまたは角度のどちらかを指定することができます。両方が指定された場合はオフセット値が使用されます。

ポートの説明  
circle : 円の情報。  
offset : 指定された点から結果の座標までの距離。  
angle : 円の中心を基準とした、指定された点から結果の座標までの角度。  
pt : 結果の座標。この点は、円の中心から水平線上にある円上の点を元に計算されます。



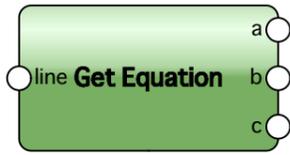
ノードの説明  
指定した弧上で入力した座標に最も近い座標を返します。

ポートの説明  
arc : 対象の円弧。  
point : 2Dの座標。  
dist : 結果の座標。



ノードの説明  
法線ベクトルによって決定される線の方向を返します。

ポートの説明  
lineH: 2Dの直線図形。  
normVec: 線の向き。



ノードの説明  
直線の方程式に使用する値を返します。

ポートの説明  
lineH: 2Dの直線図形。  
a: 式の第1の係数。  
b: 式の第2の係数。  
c: 式の最後の係数。



ノードの説明  
線の始点と終点の座標を返します。

ポートの説明  
lineH: 2Dの直線図形。  
start: 直線の始点。  
end: 直線の終点。



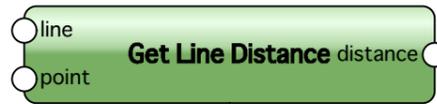
ノードの説明  
指定した2つの線の間の角度を0から90度の範囲で返します。

ポートの説明  
lineH1: 2Dの直線図形。  
lineH2: 2Dの直線図形。  
angDeg: 2本の線の間の角度 (度)。



ノードの説明  
線を中心点を返します。

ポートの説明  
lineH: 線図形。  
cntrPt: 線を中心点。



ノードの説明  
点から線までの距離を返します。

ポートの説明  
lineH: 2Dの直線図形。  
point: 2Dの座標。  
distance: 結果の距離。



ノードの説明  
線の長さを返します。

ポートの説明  
lineH: 2Dの直線図形。  
length: 線の長さ。



ノードの説明  
基準点の座標を返します。

ポートの説明  
locus: 基準点。  
location: 基準点の座標。



ノードの説明  
長円の高さと幅を返します。

ポートの説明  
oval: 対象の長円。  
width: 長円の幅。  
height: 長円の高さ。



ノードの説明  
長円の長さ (周長) を返します。

ポートの説明  
oval: 長円の情報。  
length\_out: 長円の長さ (周長)。



ノードの説明  
曲線の内側のポイントを検索します。

ポートの説明  
poly: 多角形/曲線の情報。  
point: 多角形/曲線内側の2D座標。



ノードの説明  
多角形/曲線の最初の頂点からの指定した距離にある点とその点での接戦ベクトルを返します。

コントロールの説明  
Tolerance: 計算値と近似値の許容可能な差を指定します。

ポートの説明  
poly: 多角形/曲線の情報。  
distance: 距離の指定。  
point: 結果の座標。  
tanVec: その点での接戦ベクトル。  
success: ノードが成功し、完了するとTrue、そうでない場合はfalse。



ノードの説明  
指定したポイントに直線上の最も近い点を返します。

ポートの説明  
lineH: 2Dの直線図形。  
point: 2Dの座標。  
perpPt: 結果の座標。



ノードの説明

四角形の高さ、幅、原点の座標を返します。

コントロールの説明

Point : 四角形の取得したい座標の位置を指定します。

ポートの説明

rectangle : 対象の四角形オブジェクト。  
width : 四角形の幅。  
height : 四角形の高さ。  
pt : 四角形の指定した位置の座標。



ノードの説明

直線によって定義される座標系に点を変換します。

ポートの説明

lineH : 直線図形。  
point : 2Dの座標。  
transPt : 変換後の座標。



ノードの説明

多角形/曲線に少なくとも1つの頂点指定ではない頂点がある場合は、trueを返します。

ポートの説明

poly : 多角形/曲線の情報。  
bool : 多角形/曲線に少なくとも1つの頂点指定ではない頂点がある場合はTrue、そうでない場合はfalse。

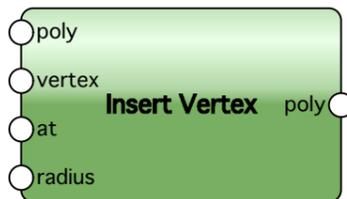


ノードの説明

2本の無限の線が交差する場合はtrueを返し、その場合、交差するポイントを返します。交点は線セグメントの外側であることができます。

ポートの説明

lineH1 : 2Dの直線図形。  
lineH2 : 2Dの直線図形。  
isInters : 線が交差する場合はtrueを返し、そうでない場合はfalseを返します。  
intersPt : 交差するポイント。



ノードの説明

新しい頂点を多角形/曲線の与えられた箇所に挿入します。頂点の種類が円弧指定でない場合は、半径の値は不要です。

コントロールの説明

Vertex Type : 追加する新しい頂点の形式を選択します。

ポートの説明

polyH : 多角形/曲線の情報。  
vertex : 新しく挿入する頂点。  
at : 多角形/曲線の頂点配列内の新しい頂点のインデックス。  
radius : 円弧指定の頂点の半径。この値は他のタイプの頂点には必要ありません。  
poly : 結果の多角形/曲線。



ノードの説明

指定した行列で多角形の点を逆変換します。

ポートの説明

polyH : 多角形/曲線の情報。  
matrix : 座標変換行列。  
poly : 結果の多角形/曲線。



ノードの説明

多角形/曲線が時計回りに配置（作図）されている場合はtrueを返します。

ポートの説明

poly : 多角形/曲線の情報。  
isCW : 多角形/曲線が時計回りに作図されている場合はtrueを返し、そうでない場合はfalseを返します。



ノードの説明

多角形/曲線が閉じている場合はtrueを返します。

ポートの説明

poly : 多角形/曲線の情報。  
isClosed : 多角形/曲線が閉じている場合はtrueを返し、そうでない場合はfalseを返します。



ノードの説明

指定した点が参照した多角形か曲線の内側または線上にある場合はTrueを返します。また座標も返します。

ポートの説明

poly : 多角形/曲線の情報。  
point : 2Dの入力点（座標）。  
bool : 入力した点が多角形/曲線の内側または線上にある場合はtrueを返します。  
truePt : 入力した点が多角形/曲線の内側または線上にある場合は、その点の座標を返します。そうでない場合は何も返しません。



ノードの説明

入力された図形が直線ならばtrueを、そうでない場合はfalseを返します。

ポートの説明

obj2D : 2D図形。  
isLine : 図形が直線であればtrueを返し、そうでない場合はfalseを返します。



ノードの説明

オブジェクトが2Dまたは3D基準点かどうかを判定します。

ポートの説明

obj : 2Dまたは3D図形。  
isLocus : 図形が基準点であればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
入力点が誤差内で円弧上にあるかどうかを判定します。

**コントロールの説明**  
Tolerance :  
計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
arc : 対象の円弧。  
point : 座標の情報。  
isOnArc :  
入力点が円弧上にあればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
入力点が円上にある場合はtrueを返します。ノードのパラメータとして指定した誤差を使用して判定します。

**コントロールの説明**  
Tolerance :  
計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
circle : 円の情報。  
pt : 座標の情報。  
isOnCirc :  
点が円上にあればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
入力点が誤差内で線分の近くにあるかどうかをチェックし、線分上の垂直点を返します。

**コントロールの説明**  
Tolerance :  
計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
lineH : 2Dの直線図形。  
point : 2Dの座標。  
isNear :  
点が誤差内で線分の近くにあればtrueを返し、そうでない場合はfalseを返します。  
perpPt : 垂直点。



**ノードの説明**  
指定した点が直線の左側にある場合はtrueを返します。

**ポートの説明**  
lineH : 2Dの直線図形。  
point : 2Dの座標。  
isOnLeft :  
点が直線の左側にあればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
指定した点が誤差内で直線上にある場合はtrueを返します。

**コントロールの説明**  
Tolerance :  
計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
lineH : 2Dの直線図形。  
point : 2Dの座標。  
isOnLine :  
点が誤差内で直線上にあればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
オブジェクトが四角形であるかどうかを判定します。

**ポートの説明**  
obj : オブジェクトの情報。  
isRect :  
オブジェクトが四角形であればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
指定した点が誤差内で直線の開始点である場合はtrueを返します。

**コントロールの説明**  
Tolerance :  
計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
lineH : 2Dの直線図形。  
point : 2Dの座標。  
isStart :  
点が誤差内で直線の開始点であればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
指定したインデックスの頂点のタイプが指定したタイプと同じである場合にtrueを返します。

**コントロールの説明**  
Vertex Type :  
頂点のタイプを指定します。

**ポートの説明**  
poly : 多角形/曲線の情報。  
at :  
多角形/曲線の頂点配列内の頂点のインデックス。  
bool :  
頂点のタイプが指定したタイプと同じであればtrueを返します。



**ノードの説明**  
指定したインデックスの頂点が表示されている場合にtrueを返します。

**ポートの説明**  
poly : 多角形/曲線の情報。  
at : 頂点のインデックス番号。  
isVisible :  
頂点が表示されていればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
図面上に新しい2D直線を作成します。

**ポートの説明**  
start : 直線の始点。  
end : 直線の終点。  
lineH : 新しく作成された直線。



**ノードの説明**  
2Dまたは3D基準点を作成します。

**ポートの説明**  
location : 基準点の座標。  
locus : 基準点。



**ノードの説明**  
指定した多角形/曲線を外側または内側にオフセットします。

**コントロールの説明**  
Smooth Corners :  
チェックを入れて有効にすると、結果の多角形/曲線にスムージングが適用されます。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
offset : オフセットの値。  
poly :  
オフセットされた結果の多角形/曲線。



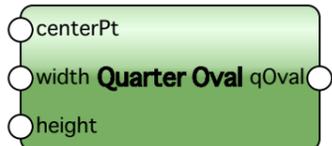
**ノードの説明**  
指定したパラメータで長円オブジェクトを作成します。

**ポートの説明**  
pt : 長円の原点。  
width : 長円の幅。  
height : 長円の高さ。  
oval : 作成される長円オブジェクト。



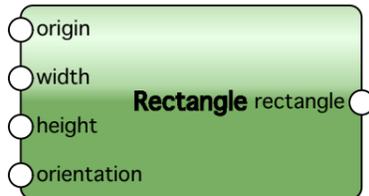
**ノードの説明**  
新しい2D多角形を指定した2D頂点で作成します。

**ポートの説明**  
vertices : 指定する頂点のセット。  
poly : 作成される多角形。



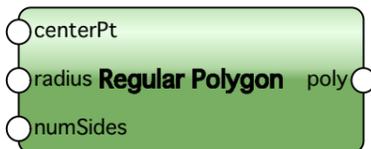
**ノードの説明**  
指定したパラメータの値で四分円を作成します。「width」と「height」には負数も使用できます。

**ポートの説明**  
pt : 四分円の2D中心点。  
width : 四分円の幅。  
height : 四分円の高さ。  
qArc : 作成される四分円。



**ノードの説明**  
指定した仕様の四角形を作成します。

**ポートの説明**  
pt : 四角形の原点。  
width : 四角形の幅。  
height : 四角形の高さ。  
orientation : 四角形の方向。  
rectangle : 作成される四角形。



**ノードの説明**  
指定したn辺から正多角形を作成します。

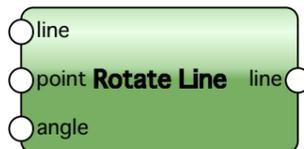
**コントロールの説明**  
Mode :  
指定する多角形の種類です。「Circumscribed」は、円の外側に頂点がある正多角形です。「Inscribed」は、円の内部に頂点がある正多角形です。

**ポートの説明**  
centerPt : 多角形の中心点。  
radius : 多角形に付随する円の半径。  
numSides : 多角形の辺の数。  
poly : 作成される正多角形。



**ノードの説明**  
多角形/曲線の頂点の順番を逆にします（時計回りを反時計回り、または反時計回りを時計回り）。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
poly : 結果の多角形/曲線。



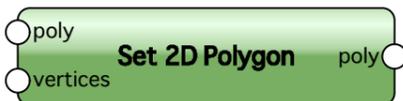
**ノードの説明**  
指定した座標を基点として、指定した角度（度数）で直線を回転します。

**ポートの説明**  
line : 2Dの直線図形。  
point : 2Dの座標。  
angle : 度数単位での角度。  
lineH : 回転した直線。



**ノードの説明**  
指定した2本の直線が並行している、またはセグメントの境界内で直線のセグメント同士が交差していない場合はFalseを返します。それ以外はTrueと交差するポイントの座標を返します。

**ポートの説明**  
lineH1 : 2Dの直線図形。  
lineH2 : 2Dの直線図形。  
isInters :  
2本の直線が交差している場合はTrueを、それ以外はFalseを返します。  
intersPt :  
2本の直線が交差するポイントの座標を返します。



**ノードの説明**  
指定した多角形/曲線に指定した頂点の座標を設定します。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
vertices : 設定する頂点の座標。  
poly : 結果の多角形/曲線。



**ノードの説明**  
指定した頂点インデックスの座標を設定します。

**コントロールの説明**  
Vertex Type :  
指定した頂点の変換後の形式を選択します。  
Recalculate Bounds :  
図形の領域の再計算を実行するかどうかを指定します。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
at : 頂点のインデックス番号。  
point : 座標の情報。  
radius :  
円弧指定の半径。この値は他の形式の頂点では不要です。  
poly : 結果の多角形/曲線。



**ノードの説明**  
指定した多角形/曲線の頂点の表示/非表示を設定します。

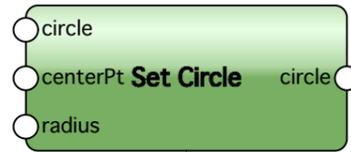
**コントロールの説明**  
Is Visible :  
すべての頂点を表示に設定する場合はチェックを入れます。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
poly : 結果の多角形/曲線。



**ノードの説明**  
指定したパラメータで円弧を設定します。

**ポートの説明**  
arc : 対象の円弧。  
startAng : 円弧の開始角度。  
sweepAng : 円弧の角度。  
arc\_out : 作成される円弧。



**ノードの説明**  
指定したパラメータで円オブジェクトを設定します。

**ポートの説明**  
circle : 円の情報。  
centerPt : 設定する中心点の2D座標。  
radius : 設定する半径の値。  
circle\_out : 作成される円オブジェクト。



**ノードの説明**  
多角形/曲線の閉じる/開く状態を設定します。

**コントロールの説明**  
Is Closed :  
多角形/曲線を閉じる場合はチェックボックスにチェックを入れます。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
poly : 結果の多角形/曲線。



**ノードの説明**  
直線の終点の座標を設定します。

**ポートの説明**  
line : 2Dの直線図形。  
point : 2Dの座標。  
lineH : 作成される直線。



**ノードの説明**  
直線の始点と終点の座標を設定します。

**ポートの説明**  
line : 2Dの直線図形。  
start : 直線の始点。  
end : 直線の終点。  
lineH : 作成される直線。



**ノードの説明**  
基準点の座標を設定します。

**ポートの説明**  
locus : 対象の基準点。  
location : 基準点の座標。  
res : 結果の基準点。



**ノードの説明**  
指定したパラメータで長円を設定します。

**ポートの説明**  
oval : 対象の長円。  
width : 長円の幅。  
height : 長円の高さ。  
oval\_out : 作成される長円。



**ノードの説明**  
指定した四角形の高さ、幅、原点座標を変更します。

**コントロールの説明**  
Control Point :  
設定したい四角形の座標の位置を指定します。

**ポートの説明**  
rectangle : 対象の四角形オブジェクト。  
width : 指定する幅。  
height : 指定する高さ。  
pt : 指定する位置の座標。  
rect : 作成される四角形。



**ノードの説明**  
直線の開始位置の座標を設定します。

**ポートの説明**  
line : 2Dの直線図形。  
point : 2Dの座標。  
lineH : 作成される直線。



**ノードの説明**  
指定した頂点の表示/非表示を設定します。

**コントロールの説明**  
Is Visible :  
表示に設定する場合はチェックを入れます。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
at : 頂点のインデックス番号。  
poly : 結果の多角形/曲線。



**ノードの説明**  
円弧の開始点から指定したポイントまでの円弧角を計算します。

**ポートの説明**  
arc : 対象の円弧。  
point : 2Dの座標。  
sweepAng : 結果の円弧角。



**ノードの説明**  
指定した行列で多角形/曲線を変形します。

**ポートの説明**  
polyH : 多角形/曲線の情報。  
matrix : 座標変換行列。  
poly : 結果の多角形/曲線。

# ジオメトリ 3D



**ノードの説明**  
3D多角形に頂点を1つ追加します。

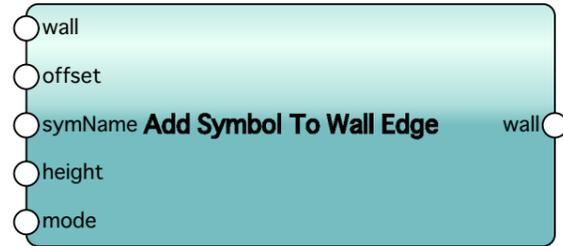
**ポートの説明**  
polyH : 3D多角形の情報。  
point : 3D頂点の座標。  
poly : 結果の3D多角形。



**ノードの説明**  
指定した壁にシンボルを挿入します。

**コントロールの説明**  
Flip : シンボルを反転する場合はチェックを入れます。  
Right : 右側に挿入する場合はチェックを入れます。

**ポートの説明**  
inWall : 指定する壁オブジェクト。  
offset : 壁の始点からのオフセット距離。  
symName : 壁に挿入するシンボルの名前。  
height : シンボル挿入位置の高さ。  
wall : 結果の壁オブジェクト。



**ノードの説明**  
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。

**コントロールの説明**  
Flip : シンボルを反転する場合はチェックを入れます。  
Right : 右側に挿入する場合はチェックを入れます。

**ポートの説明**  
inWall : 指定する壁オブジェクト。  
offset : 壁の始点から挿入点までのオフセット距離。  
symName : 挿入するシンボルの名前。  
height : シンボル挿入位置の高さ。  
mode : 挿入モード (0 : 中心/1 : 左端/2 : 右端)。  
wall : 結果の壁オブジェクト。



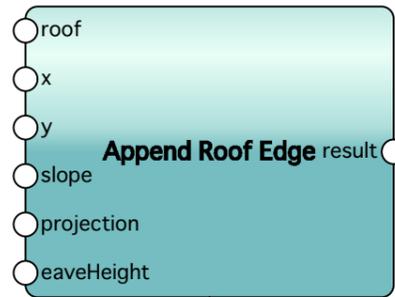
**ノードの説明**  
指定した壁の底面に頂点を追加します。

**ポートの説明**  
inWall : 指定する壁オブジェクト。  
offset : 壁の始点から頂点までのオフセット距離。  
height : 頂点の高さ。  
wall : 結果の壁オブジェクト。



**ノードの説明**  
指定した壁に頂点を追加します。

**ポートの説明**  
inWall : 指定する壁オブジェクト。  
offset : 壁の始点からのオフセット距離。  
height : 頂点の高さ。  
wall : 結果の壁オブジェクト。



**ノードの説明**  
指定した屋根オブジェクトに、新しい屋根の端部を追加します。

**ポートの説明**  
roof : 指定する屋根。  
x : 指定する屋根端部のX座標。  
y : 指定する屋根端部のY座標。  
slope : 屋根の勾配。  
projection : 壁外面から軒先の距離。  
eaveHeight : 軒先下端の高さ。  
result : 変更後の屋根。



**ノードの説明**  
3D多角形内で誤差内のすべての同じ頂点を削除します。

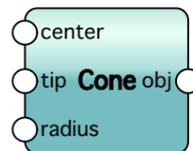
**コントロールの説明**  
Tolerance :  
計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
polyH : 3D多角形の情報。  
poly : 結果の3D多角形。



**ノードの説明**  
指定した壁の頂点をすべて消去します。

**ポートの説明**  
wall : 指定する壁オブジェクト。



**ノードの説明**  
3Dソリッドの円錐を作成します。

**ポートの説明**  
center :  
円錐底面の中心に設定する3D座標。  
tip :  
円錐の先端の位置を設定する3D座標、または、底面から円錐の頂点までの高さを設定する単一の値。  
radius : 円錐底面の半径の値。  
obj : 結果の円錐。



**ノードの説明**  
コンテナ図形 (グループ、シンボル、レイヤなど) 内に光源が含まれているかどうかをチェックします。指定したオブジェクト自体が光源図形の場合はFalseを返します。

**ポートの説明**  
handle : 指定するハンドル。  
bool :  
光源が含まれている場合はtrueを、それ以外はfalseを返します。



**ノードの説明**  
オブジェクトを3D多角形のグループに変換します。四角形、円、円弧、曲線、多角形、長円、線、壁、円弧壁、柱状体、及び屋根を変換します。

**ポートの説明**  
obj : 変換するオブジェクト。  
polys : 変換により作成される3D多角形。



**ノードの説明**

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/非表示も設定できます。

**ポートの説明**

name : プラグインオブジェクトの名前。  
 p : オブジェクトを配置する座標。  
 angle : オブジェクトの回転角度 (度数)。  
 showPref : 設定ダイアログの表示設定 (TRUE : 表示、FALSE : 非表示)。  
 h : 作成したオブジェクト。



**ノードの説明**

引数で指定した、パスを用いたカスタムプラグインオブジェクトを作成します。パスの最初の頂点は、プラグインの座標空間の原点に配置されます。

**ポートの説明**

name : プラグインオブジェクトの名前。  
 path : パス図形 (曲線/NURBS) のハンドル。  
 profileGroup : 輪郭図形のハンドル。  
 h : 作成したオブジェクト。



**ノードの説明**

座標を指定して光源を作成します。

**コントロールの説明**

lightType : 光源の種類を選択します。

**ポートの説明**

location : 光源を配置する位置 (座標)。  
 isOn : 光源のON/OFF (OFF : 0、ON : 1)。  
 castShadow : 影を付ける設定の指定 (FALSE : 0、TRUE : 1)。  
 light : 作成した光源。



**ノードの説明**

新しい屋根を作成し、そのハンドルを返します。屋根オブジェクトのテンプレートを定義する場合は、AppendRoofEdgeを使用してください。

**ポートの説明**

genGableWall : 切り妻屋根の有無。  
 bearingInset : 支持部分の差し込みの距離。  
 roofThick : 屋根の厚み。  
 miterType : 屋根端部 (軒) の形状 : 1 = 垂直、2 = 水平、3 = 水平・垂直、4 = 直角  
 vertMiterDistance : 水平・垂直の形状選択時の垂直部分の長さ。  
 roof : 作成した屋根。



**ノードの説明**

スラブを作成します。

**ポートの説明**

profile : スラブを作る元となる輪郭形状。  
 slab : 作成したスラブ。



**ノードの説明**

壁フィーチャを輪郭形状から作成します。壁フィーチャは突出と窪みを選択できます。

**コントロールの説明**

operation : 壁フィーチャのタイプを選択します。

**ポートの説明**

wall : 壁フィーチャを作成する壁オブジェクト。  
 profile : 壁フィーチャの輪郭として使用するオブジェクト。  
 wFeature : 作成した壁フィーチャ。



**ノードの説明**

指定した番号の壁の頂点を削除します。

**ポートの説明**

inWall : 指定する壁オブジェクト。  
 index : 削除する壁の頂点のインデックス番号。  
 wall : 結果の壁オブジェクト。

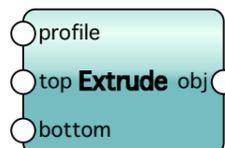


**ノードの説明**

指定したシンボルを壁から削除します。

**ポートの説明**

symbol : 削除するシンボル。  
 success : 削除が成功するとTrueを、それ以外はFalseを返します。



**ノードの説明**

2Dの断面から底面と上面の高さを指定して柱状体を作成します。

**ポートの説明**

profile : 柱状体にする2D断面。  
 top : 柱状体の上面の高さ。  
 bottom : 柱状体の底面の高さ。  
 xtrds : 結果の柱状体。



**ノードの説明**

水平または垂直方向に3Dオブジェクトを反転します。サポートされているタイプは次のとおりです。柱状体、回転体、多段柱状体、シンボル、メッシュおよび3Dソリッド。

**コントロールの説明**

Flip Axis : 反転の軸を選択します。

**ポートの説明**

object : 反転させるオブジェクト。  
 flip : 結果のオブジェクト。



**ノードの説明**

3D多角形の面積を計算します。3D多角形の頂点が反時計回りの場合は正の、時計回りの場合は負の面積となります。

**ポートの説明**

poly : 3D多角形の情報。  
 area : 指定した多角形の面積。



**ノードの説明**

指定されたインデックスから開始する3D多角形の辺の端点座標を返します。3D多角形の頂点配列の最後のインデックスの場合、辺の端点は3D多角形の頂点配列内の最初の頂点になります。

**ポートの説明**

poly : 3D多角形の情報。  
 startIdx : 辺の始点のインデックス。  
 start : 辺の始点のインデックス。  
 end : 辺の終点のインデックス。



**ノードの説明**

3D多角形の周長 (各辺の長さの合計) を計算します。

**ポートの説明**

poly : 3D多角形の情報。  
 perim : 多角形の周長。



**ノードの説明**  
3D多角形の頂点配列内で指定したインデックスの頂点を返します。

**ポートの説明**  
poly : 3D多角形の情報。  
at : 頂点のインデックス。  
vertex : 頂点。



**ノードの説明**  
入力された3D多角形の頂点数を返します。

**ポートの説明**  
poly : 3D多角形の情報。  
vertNum : 多角形の頂点数。



**ノードの説明**  
環境光の色成分を返します。

**ポートの説明**  
red : RGBの赤の値。  
green : RGBの緑の値。  
blue : RGBの青の値。



**ノードの説明**  
環境光の情報を返します。

**ポートの説明**  
isOn : 環境光のON/OFFステータス (On=True、Off=False)。  
brightness : 環境光の明るさ。



**ノードの説明**  
指定した光源 (スポットライト) の光束の角度を返します。

**ポートの説明**  
handle : 指定する光源のハンドル。  
BeamAngleR : 光源の光束の角度。



**ノードの説明**  
3D多角形のジオメトリ中心を返します。

**ポートの説明**  
poly : 3D多角形の情報。  
centerPt : 対象の多角形の中心座標。



**ノードの説明**  
指定したオブジェクトの指定したインデックス番号の構成要素の名前を取得します。

**ポートの説明**  
obj : 指定するオブジェクト (壁、円弧壁、スラブ、屋根、壁スタイル、スラブスタイル、屋根スタイル、壁設定、スラブ設定、または屋根設定)。  
index : 構成要素のインデックス番号。  
name : 構成要素の名前。



**ノードの説明**  
オブジェクトの構成要素の面積 (ネット) を取得します。

**ポートの説明**  
obj : 指定するオブジェクト (壁、円弧壁、スラブ、屋根、壁スタイル、スラブスタイル、屋根スタイル、壁設定、スラブ設定、または屋根設定)。  
index : 構成要素のインデックス番号。  
area : 構成要素の面積 (ネット)。



**ノードの説明**  
オブジェクトの構成要素の体積 (ネット) を取得します。

**ポートの説明**  
obj : 指定するオブジェクト (壁、円弧壁、スラブ、屋根、壁スタイル、スラブスタイル、屋根スタイル、壁設定、スラブ設定、または屋根設定)。  
index : 構成要素のインデックス番号。  
vol : 構成要素の体積 (ネット)。



**ノードの説明**  
デフォルトの壁の厚みの設定値を返します。

**ポートの説明**  
width : ファイルのデフォルトの壁の厚み。



**ノードの説明**  
指定した光源の色のRGB値を返します。

**ポートの説明**  
handle : 指定する光源のハンドル。  
R : RGBの赤の値。  
G : RGBの緑の値。  
B : RGBの青の値。



**ノードの説明**  
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。

**ポートの説明**  
handle : 指定する光源のハンドル。  
PanAngle : 光源のパン角度。  
TiltAngle : 光源の傾き角度。



**ノードの説明**  
指定した光源の距離減衰のタイプを返します。なし  
0 標準 1 スムーズ 2 シャープ  
3

**ポートの説明**  
handle : 指定する光源のハンドル。  
distFalloff : 光源の距離減衰タイプ。



**ノードの説明**  
指定した光源の情報を返します。

**ポートの説明**  
handle : 指定する光源のハンドル。  
lightType : 光源の種類 (0 : 平行光源、1 : 点光源、2 : スポット)。  
brightness : 光源の明るさ。  
isOn : 光源のON/OFF (False : OFF、True : ON)。  
castShadow : 影を付ける設定の有無 (False : なし、True : あり)。



**ノードの説明**  
指定した光源の位置を座標で返します。

**ポートの説明**  
handle : 指定する光源のハンドル。  
pt : 光源の位置の座標。

obj **Get Number of Components** n

ノードの説明  
指定したオブジェクトの構成要素の数を取得します。

ポートの説明  
obj : 作成したオブジェクト。  
n : 構成要素の数。

roof **Get Roof Vertices** n

ノードの説明  
指定した屋根オブジェクトが持つ屋根の端部の数を返します。

ポートの説明  
roof : 指定する屋根。  
n : 屋根の端部の数。

slab **Get Slab Style** style

ノードの説明  
指定したスラブのスラブスタイルを取得します。

ポートの説明  
slab : 指定するスラブ。  
style :  
スラブのスラブスタイルの参照番号。スタイルなしの場合は0。

handle **Get Spread Angle** SpreadAngle

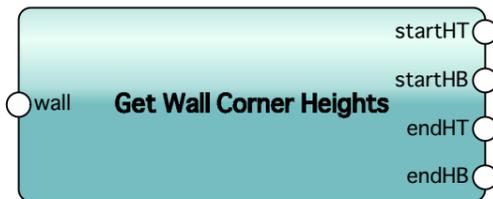
ノードの説明  
指定した光源 (スポットライト) の拡散光の角度を返します。

ポートの説明  
handle : 指定する光源のハンドル。  
SpreadAngle : 光源の拡散光の角度。

wall **Get Wall Break Num** success numBreaks

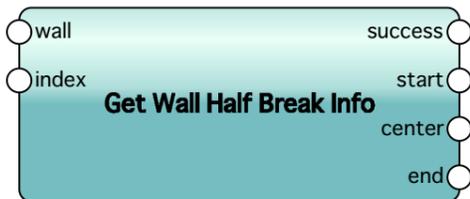
ノードの説明  
指定した壁の切断箇所数を返します。

ポートの説明  
wall : 指定する壁オブジェクト。  
success : 成功するとTrue、それ以外の場合はFalse。  
numBreaks : 壁の切断箇所数。



ノードの説明  
指定した壁、または円弧壁のコーナーの高さを取得します。

ポートの説明  
wall : 指定する壁、または円弧壁。  
startHT : 始点のオフセット高さ (上)。  
startHB : 始点のオフセット高さ (下)。  
endHT : 終点のオフセット高さ (上)。  
endHB : 終点のオフセット高さ (下)。



ノードの説明  
壁の線に沿った、壁内の半破断箇所の始点、中心点、終点を取得します。

ポートの説明  
wall : 指定する壁オブジェクト。  
index : 指定した壁の半破断箇所のインデックス番号。  
success : 成功するとTrue、それ以外の場合はFalse。  
start : 半破断箇所の始点。  
center : 半破断箇所の中心点。  
end : 半破断箇所の終点。



ノードの説明  
指定した壁、または円弧壁の高さを取得します。

ポートの説明  
wall : 指定する壁オブジェクト。  
overallHT : 壁のオフセット高さ (上)。  
overallHB : 壁のオフセット高さ (下)。



ノードの説明  
指定した壁の頂点の座標を返します。

ポートの説明  
wall : 指定する壁オブジェクト。  
index : 壁の頂点のインデックス番号。  
wallPeak : 壁の頂点の座標。



ノードの説明  
指定した壁の頂点の数を返します。

ポートの説明  
wall : 指定する壁オブジェクト。  
numPeaks : 壁の頂点の数。



ノードの説明  
指定した壁の厚みを返します。

ポートの説明  
wall : 指定する壁オブジェクト。  
success : 成功するとTrue、それ以外の場合はFalse。  
thickness : 壁の厚み。



ノードの説明  
3D多角形のX軸の範囲を返します。

ポートの説明  
poly : 3D多角形の情報。  
minX : 多角形のX軸上範囲の下限。  
maxX : 多角形のX軸上範囲の上限。



ノードの説明  
3D多角形のY軸の範囲を返します。

ポートの説明  
poly : 3D多角形の情報。  
minY : 多角形のY軸上範囲の下限。  
maxY : 多角形のY軸上範囲の上限。



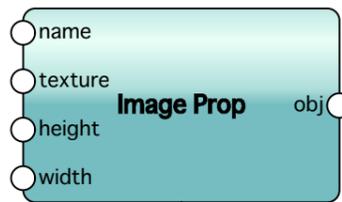
**ノードの説明**  
3D多角形のZ軸の範囲を返します。

**ポートの説明**  
poly : 3D多角形の情報。  
minZ : 多角形のZ軸上範囲の下限。  
maxZ : 多角形のZ軸上範囲の上限。



**ノードの説明**  
3Dソリッドの半球を作成します。

**ポートの説明**  
center : 半球の底面中心の3D座標。  
top : 阪急の先端の位置を設定する3D座標、または、底面から円錐の頂点までの高さを設定する単一の値。  
obj : 結果の半球。



**ノードの説明**  
添景オブジェクトを作成します。

**コントロールの説明**  
Enforce aspect ratio : 添景のアスペクト比をテクスチャと一致させるオプションを使用する場合は、チェックを入れます。  
Crossed planes : 添景のイメージ面を十¥文字に配置するオプションを使用する場合は、チェックを入れます。  
Create plugin object : プラグインオブジェクトとして作成するオプションを使用する場合は、チェックを入れます。  
Auto rotate : 添景を自動回転（視点に追従）させるオプションを使用する場合は、チェックを入れます。  
Create symbol : シンボルとして作成するオプションを使用する場合は、チェックを入れます。

**ポートの説明**  
name : 作成されたシンボルで使用する名前。その名前がすでに使われている場合は、固有の名前が割り当てられます。  
text\_ref : 添景で使用するテクスチャ。  
height : アスペクト比が固定されていない場合の添景の高さ。  
width : アスペクト比が固定されていない場合の添景の幅。  
i\_prop : 結果の添景。



**ノードの説明**  
マトリックスを使用して3D多角形の頂点に逆変換を適用します。

**ポートの説明**  
polyH : 3D多角形の情報。  
mat : 座標変換行列。  
poly : 変換した結果の頂点を持つ多角形。



**ノードの説明**  
多角形が時計回りに配置（作図）されているかどうかを判定します。

**ポートの説明**  
poly : 3D多角形の情報。  
isCW : 多角形が時計回りに作図されている場合はtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
多角形が閉じている場合はtrueを返します。

**ポートの説明**  
poly : 3D多角形の情報。  
isClosed : 多角形が閉じている場合はtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
3D多角形のすべての頂点と同じ平面にあるかを判定します。

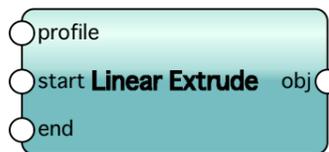
**コントロールの説明**  
Tolerance : 計算値と近似値の許容可能な差を指定します。

**ポートの説明**  
poly : 3D多角形の情報。  
isFlat : 3D多角形のすべての頂点と同じ平面にあればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**  
インデックス番号で指定した壁の頂点がある場合はTRUEを返します。

**ポートの説明**  
wall : 指定する壁オブジェクト。  
index : 壁の頂点のインデックス番号。  
isWPTop : 指定した壁の頂点がある場合はTrueを、そうでない場合はFalseを返します。



**ノードの説明**  
このノードは断面を元に始点から終点までの3D柱状体を作成します。

**ポートの説明**  
profile : 柱状体を作成する2D断面オブジェクト。  
start : 柱状体の始点を表¥す3Dの座標。  
end : 柱状体の終点を表¥す3Dの座標。  
xtrds : 結果の柱状体。



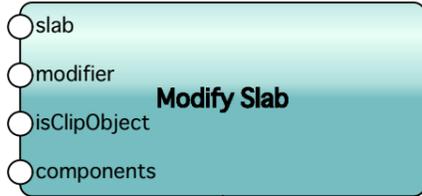
**ノードの説明**  
指定した3D多角形のリストからメッシュ図形を作成します。

**ポートの説明**  
obj : 3D多角形のリスト。  
mesh : 結果のメッシュ図形。

mesh **Mesh to Group** group

ノードの説明  
メッシュ図形を3D多角形のグループに変換します。

ポートの説明  
mesh : 変換するメッシュ図形。  
group : 結果の3D多角形のグループ。



ノードの説明  
スラブを貼り合わせ、または切り欠きします。

ポートの説明  
slab : 指定するスラブ。  
modifier :  
貼り合わせ、または切り欠きに使用するオブジェクト。  
isClipObject :  
実行する変更が切り欠きの場合はTrueを、貼り合わせの場合はFalseを指定。  
components :  
この変更によって影響を受ける構成要素のインデックス番号のリスト。



ノードの説明  
2つ以上の2D断面から柱状体を作成します。入力したそれぞれの断面のリストに対して、柱状体を作成されます。

ポートの説明  
profiles :  
柱状体を作成する2D図形のリスト。  
top : 柱状体の上面の高さ。  
bottom : 柱状体の底面の高さ。  
xtrds : 結果の柱状体。



ノードの説明  
断面図形をパスに沿って押し出して、3Dパス図形を作成します。

ポートの説明  
profiles : 3Dパス図形に使用する断面図形。  
paths : 3Dパス図形に使用するパス図形。  
xtrds : 結果のオブジェクト。



ノードの説明  
新しい3D多角形を指定した3D頂点の座標で作成します。

ポートの説明  
vertices : 3D座標のリスト。  
poly : 新しく作成される多角形。



ノードの説明  
3D多角形の頂点の順番を逆にします（時計回りを反時計回り、または反時計回りを時計回り）。

ポートの説明  
polyH : 3D多角形の情報。  
poly : 結果の多角形。



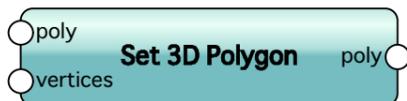
ノードの説明  
指定した壁の向きを逆にすることで、壁の左右を切り替えます。これはデータパレット上にある「壁の向きを反転」ボタンと同じ動作です。

ポートの説明  
wall : 指定する壁オブジェクト。



ノードの説明  
反時計回りの円弧壁を作成します。

ポートの説明  
center : 円弧壁の中心の座標。  
start : 円弧壁の始点の座標。  
end : 円弧壁の終点の座標。  
wall :  
指定した3点をもとに作成した円弧壁。



ノードの説明  
指定した頂点で3D多角形を設定します。

ポートの説明  
polyH : 3D多角形の情報。  
vertices : 3D座標のリスト。  
poly : 結果の多角形。



ノードの説明  
3D多角形の指定した頂点インデックスの座標を設定します。

ポートの説明  
polyH : 3D多角形の情報。  
at :  
多角形に指定する新しい頂点のインデックス。  
point : 3D頂点の座標。  
poly : 結果の多角形。



ノードの説明  
環境光の色をRGB値で設定します。

ポートの説明  
rgb : 指定するRGB値。



ノードの説明  
環境光の情報（On/Off、明るさ）を設定します。

ポートの説明  
isOn :  
環境光のOn/Off（Onの場合はTrueを、Offの場合はFalseを指定）。  
brightness : 環境光の明るさ。



ノードの説明  
光源（スポットライト）の光束の角度を設定します。

ポートの説明  
handle : 指定する光源のハンドル。  
BeamAngle : 指定する光束の角度。  
light : 設定した光源のハンドル。



ノードの説明  
3D多角形の閉じる/開く状態を設定します。

コントロールの説明  
Is Closed :  
多角形を閉じる場合はチェックボックスにチェックを入れます。

ポートの説明  
polyH : 3D多角形の情報。  
poly : 結果の多角形。



ノードの説明  
ファイルでデフォルトとする壁の厚みを設定します。

ポートの説明  
width : 指定する新しいデフォルトの壁の厚み。



**ノードの説明**  
指定した光源の色をRGB値で設定します。

**ポートの説明**  
handle : 指定する光源のハンドル。  
rgb : 指定するRGB値。  
light : 設定した光源のハンドル。



**ノードの説明**  
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。

**ポートの説明**  
handle : 指定する光源のハンドル。  
panAngle : 光源のパン角度。  
tiltAngle : 光源の傾き角度。  
light : 設定した光源のハンドル。



**ノードの説明**  
指定した光源の距離減衰のタイプを設定します。

**コントロールの説明**  
Falloff : 距離減衰のタイプを選択します。

**ポートの説明**  
handle : 指定する光源のハンドル。  
light : 設定した光源のハンドル。



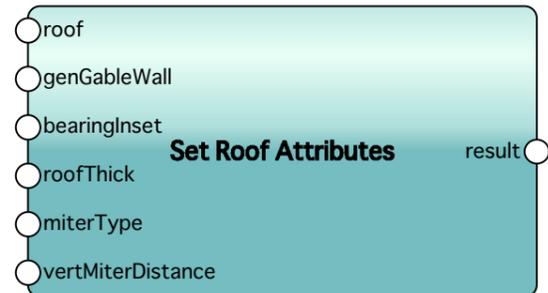
**ノードの説明**  
指定した光源の情報を設定します。

**ポートの説明**  
handle : 指定する光源のハンドル。  
lightType : 光源の種類 (0 : 平行光源、1 : 点光源、2 : スポット)。  
brightness : 光源の明るさ。  
isOn : 光源のON/OFF (False : OFF、True : ON)。  
castShadow : 影を付ける設定 (False : なし、True : あり)。  
light : 設定した光源のハンドル。



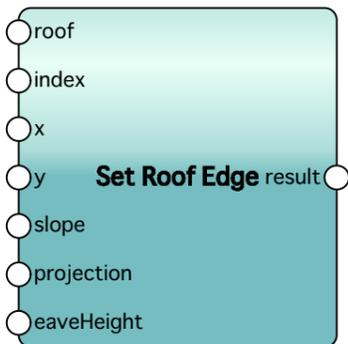
**ノードの説明**  
指定した光源の位置を座標で設定します。

**ポートの説明**  
handle : 指定する光源のハンドル。  
location : 光源の座標。  
light : 設定した光源のハンドル。



**ノードの説明**  
指定した屋根の定義を設定します。

**ポートの説明**  
roof : 指定する屋根。  
genGableWall : 切り妻屋根の有無。  
bearingInset : 支持部分の差し込みの距離。  
roofThick : 屋根の厚み。  
miterType : 屋根端部（軒）の形状 : 1 = 垂直、2 = 水平、3 = 水平・垂直、4 = 直角  
vertMiterDistance : 水平・垂直の形状選択時の垂直部分の長さ。  
result : 変更後の屋根。



**ノードの説明**  
指定した屋根の屋根の端部の定義を設定します。

**ポートの説明**  
roof : 指定する屋根。  
index : 屋根の端部のインデックス番号。  
x : 指定する屋根端部のX座標。  
y : 指定する屋根端部のY座標。  
slope : 屋根の勾配。  
projection : 壁外面から軒先の距離。  
eaveHeight : 軒先下端の高さ。  
result : 変更後の屋根。



**ノードの説明**  
スラブの高さを設定します。

**ポートの説明**  
slab : 指定するスラブ。  
height : 指定するスラブの高さ。



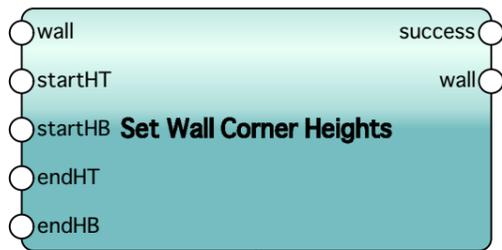
**ノードの説明**  
指定したスラブのスラブスタイルを設定します。

**ポートの説明**  
slab : 指定するスラブ。  
style : 指定するスラブスタイルの参照番号。スタイルなしの場合は0。



**ノードの説明**  
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。

**ポートの説明**  
handle : 指定する光源のハンドル。  
spreadAngleR : 光源の拡散光の角度。  
light : 設定した光源のハンドル。



ノードの説明

壁、または円弧壁のコーナーの高さを設定します。

ポートの説明

inWall : 指定する壁、または円弧壁。  
 startHT : 始点のオフセット高さ (上)。  
 startHB : 始点のオフセット高さ (下)。  
 endHT : 終点のオフセット高さ (上)。  
 endHB : 終点のオフセット高さ (下)。  
 success : 成功するとTrue、それ以外の場合はFalse。  
 wall : 結果の壁、または円弧壁。



ノードの説明

指定した壁、または円弧壁の高さを設定します。

ポートの説明

inWall : 指定する壁オブジェクト。  
 botBType : タイプ (下)。  
 botBStory : 高さ基準 (下)。  
 botLLType : レイヤのレベルタイプ (下)。  
 botOffset : オフセット (下)。  
 topBType : タイプ (上)。  
 topBStory : 高さ基準 (上)。  
 topLLType : レイヤのレベルタイプ (上)。  
 topOffset : オフセット (上)。  
 success : 成功するとTrue、それ以外の場合はFalse。  
 wall : 結果の壁オブジェクト。

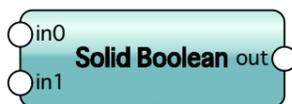


ノードの説明

スタイルなしの壁で、構成要素もない壁の厚みを設定します。スタイルが設定されている、または構成要素がある壁の場合は、Falseを返します。

ポートの説明

inWall : 指定する壁オブジェクト。  
 thickness : 壁の厚み。  
 success : 成功するとTrue、それ以外の場合はFalse。  
 wall : 結果の壁オブジェクト。



ノードの説明

オブジェクトリスト間でブール演算を実行します。操作は提供されているリスト内のオブジェクトの各組合せごとに実行されます。

コントロールの説明

operation : オブジェクト間で実行する操作です。¥n Add: 「噛み合わせる」メニューコマンドと同じです。¥n Subtract: 「削り取る」メニューコマンドと同じです。¥n Intersect: 「重なった部分を残す」メニューコマンドと同じです。

ポートの説明

in0 : 操作を実行される側のオブジェクトのリスト (対象物)。  
 in1 : 「in0」に対して使用するオブジェクトのリスト (ツール)。  
 out : 結果のオブジェクト。

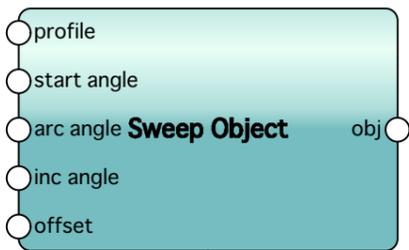


ノードの説明

3Dの球オブジェクトを作成します。

ポートの説明

center : 球体の中心を示す3D座標。  
 radius : 球体の半径。  
 obj : 作成される球体。

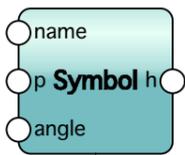


ノードの説明

3D回転体オブジェクトを作成します。回転体オブジェクトとは2D断面が指定したポイントの周りを回転して3Dオブジェクトを作成したものです。

ポートの説明

object : 回転体の2D断面。  
 start : 回転を開始する角度。  
 arc : 回転体の円弧角です。  
 inc : 結果のオブジェクトのフ分割角。  
 pitch :  
 回転体の中心から断面をオフセットする距離。  
 sweep : 回転体オブジェクト。



ノードの説明

指定したシンボルを指定した座標に配置します。

ポートの説明

name : シンボルの名前。  
 p : シンボルを配置する座標。  
 angle : シンボルの回転角度。  
 h :



ノードの説明

錐状体を作成します。

ポートの説明

profile : 錐状体にする2D図形。  
 angle : 錐状体のテーパ面の斜度。  
 height : 錐状体の高さ。  
 xtrd : 作成される錐状体。

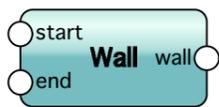


ノードの説明

座標変換行列を使用して3D多角形の頂点の座標を変換します。

ポートの説明

polyH : 3D多角形の情報。  
 mat : 座標変換行列。  
 poly : 変換した結果の頂点を持つ多角形。



ノードの説明

指定した座標に壁を作成します。作成した壁には、壁の現在のデフォルトの設定が使用されます。

ポートの説明

start : 壁の始点の座標。  
 end : 壁の終点の座標。  
 wall : 作成された壁。



ノードの説明

壁の輪郭を表す曲線を作成し、そのハンドルを返します。

ポートの説明

wall : 指定する壁オブジェクト。  
 footprint : 壁の輪郭を表す曲線。

# 操作



**ノードの説明**  
コンテナ項目に含まれるオブジェクトを返します。次のサポートされているコンテナタイプが含まれます：グループ、レイヤ、メッシュ、柱状体、多段柱状体、回転体、シンボル定義、フォルダ、パラメトリックオブジェクト、データノード、屋根。

**ポートの説明**  
group : コンテナタイプ  
objs : コンテナ内のオブジェクト。



**ノードの説明**  
3D多角形のグループ図形をメッシュ図形に変換します。

**ポートの説明**  
group : 3D多角形のグループ図形。  
mesh : 結果のメッシュ図形。



**ノードの説明**  
オブジェクトの単一コピーを作成します。

**ポートの説明**  
objIn : コピーするオブジェクト。  
orig : 元の図形。  
copy : コピーした図形。



**ノードの説明**  
オブジェクトを削除します。注：削除するオブジェクトがリストに保存され、ネットワークの実行の終了時に削除されるため、存在しないオブジェクトを参照する危険性はありません。

**ポートの説明**  
obj : 削除するオブジェクト。



**ノードの説明**  
特定図形の複製を指定個数作成します。

**ポートの説明**  
obj : 任意のVectorworksオブジェクト。  
n : 作成するコピー数。  
objs : コピー。



**ノードの説明**  
既存のグループにオブジェクトを追加します。グループがない場合は新しいものが作成されます。

**ポートの説明**  
inGroup : 図形を追加する既存グループ。  
handle : グループに追加する図形。  
group : 図形が追加された既存グループまたは追加したい図形を含む新規グループ。



**ノードの説明**  
2Dオブジェクトを指定した基準線を軸にミラー反転します。

**コントロールの説明**  
Create a duplicate object : チェックを入れない場合は、元の図形を移動します。チェックを入れると、図形を複製します。

**ポートの説明**  
input : ミラー反転させるオブジェクト。  
p1 : 反転に使用する基準線の始点の座標。  
p2 : 反転に使用する基準線の終点の座標。  
output : ミラー反転させたオブジェクト。



**ノードの説明**  
図形または点を指定した2Dまたは3Dのオフセット値で移動します。図形がプレイン図形の場合は、その面の外側までは移動しません。

**ポートの説明**  
input : 移動する図形、ベクトル、または点。  
d : 図形を移動させる距離。2D/3Dベクトルまたは座標2/座標3が入力できます。  
output : 結果。



**ノードの説明**  
指定した3DオブジェクトのZ軸を設定します。

**ポートの説明**  
inObj : 対象の3Dオブジェクト。  
orientation : オブジェクトのZ軸の方向を示す、3Dベクトル。  
obj : 結果。



**ノードの説明**  
オブジェクトリスト間でブール演算を実行します。操作は提供されているリスト内のオブジェクトの各組合せごとに実行されます。

**コントロールの説明**  
operation : オブジェクト間で実行する操作を指定します。¥n Add : 「貼寄り合わせ」コマンドと同等です。¥n Subtract : 「切り欠き」コマンドと同等です。¥n Intersect : 「抜き取り」コマンドと同等です。

**ポートの説明**  
in0 : 操作を実行される側のオブジェクトのリスト (対象物)。  
in1 : 「in0」に対して使用するオブジェクトのリスト (ツール)。  
out : 結果のオブジェクト。



**ノードの説明**  
2Dまたは3Dオブジェクトを回転させます。回転の「中心」を指定することもできます。指定しない場合はオブジェクトの中心を基点にして回転させます。

**ポートの説明**  
input : 回転させるオブジェクト。  
rVec : 2D/3D回転ベクトル、または入力カテゴリの座標2/座標3。角度単位は度数。  
center : 回転の基点座標指定オプション。  
output : 回転したオブジェクト。



**ノードの説明**  
オブジェクトやベクトルのスケール (倍率) を変更します。

**ポートの説明**  
input : 倍率を変更するオブジェクトまたはベクトル。  
s : 2D/3Dベクトルでの倍率 (sx,sy,sz)、または入力カテゴリの座標2/座標3。  
output : 倍率を変更したオブジェクト。



**ノードの説明**  
2Dオブジェクトを指定する平面に設定します。

**コントロールの説明**  
plane :  
オブジェクトを配置する平面を選択します。「Custom」を選択する場合は、平面のマトリックスを指定します。

**ポートの説明**  
obj : 平面に設定するオブジェクト。  
mat :  
平面を表すマトリックス。12の要素のリストまたはタプルとして指定します。  
output : 指定した平面上のオブジェクト。



**ノードの説明**  
2Dオブジェクトまたは2次元ベクトルを変形します。

**ポートの説明**  
input : 変形する2Dオブジェクト。  
translate : 2Dオフセットベクトル。  
scale : 2D拡大/縮小ベクトル。  
rotate : Z軸に対する単回転角度。  
anchor :  
回転の軸となる2Dポイントです。指定されていない場合は、オブジェクトは自身の中心点を軸として回転します。  
output : 変形したオブジェクト。



**ノードの説明**  
グループ図形を解除します。

**ポートの説明**  
group : グループ解除するオブジェクト。  
objs :  
グループ内に含まれていたオブジェクト。

# 計算



ノードの説明  
xの絶対値を返します。

ポートの説明  
x: 数値。  
res: xの絶対値



ノードの説明  
xの絶対値を返します。返る値は浮動小数点です。

ポートの説明  
x: 数値。  
y: xの絶対値



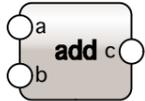
ノードの説明  
 $y = \text{acos}(x)$ :  
xの逆余弦（アークコサイン）を返します。  
yの単位はラジアンです。

ポートの説明  
x: -1から1の範囲の数値。  
y: xの逆コサイン（ラジアン値）。



ノードの説明  
xの逆双曲線余弦関数値を返します。

ポートの説明  
x: 数値。  
y: xの逆双曲線余弦関数値（ラジアン値）。



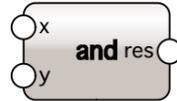
ノードの説明  
2つの値を加算します:  $c = a + b$

ポートの説明  
a: 被加数。  
b: 加数。  
c: 加算の結果。



ノードの説明  
すべての値を加算します。複数にリスト化されている値は順番毎にブロックで加算します:  $\text{out} = \text{list}[0] + \text{list}[1] + \dots$

ポートの説明  
list: 加算したい値のリスト。  
out: 加算の結果。



ノードの説明  
ビット単位のxとyの論理積を返します。

ポートの説明  
x: 第一値。  
y: 第二値。  
res: 結果。



ノードの説明  
 $y = \text{asin}(x)$ :  
xの逆正弦（アークサイン）を返します。yの単位はラジアンです。

ポートの説明  
x: -1から1の範囲の数値。  
y: xの逆サイン（ラジアン値）。



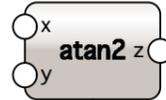
ノードの説明  
xの逆双曲線正弦関数値を返します。

ポートの説明  
x: 数値。  
y: xの逆双曲線正弦関数値（ラジアン値）。



ノードの説明  
 $y = \text{atan}(x)$ :  
xの逆正接（アークタンジェント）を返します。yの単位はラジアンです。

ポートの説明  
x: 数値。  
y: xの逆正接（ラジアン値）。



ノードの説明  
 $z = \text{atan2}(y, x) = \text{atan}(y / x)$ : ポイント(y, x)の逆正接（アークタンジェント2）を返します。zの単位はラジアンです。

ポートの説明  
x: 分母。  
y: 分子。  
z:  $y / x$ の逆正接（ラジアン値）。



ノードの説明  
xの逆双曲線正接関数値を返します。

ポートの説明  
x: 数値。  
y: xの逆双曲線正接関数値（ラジアン値）。



ノードの説明  
xをTrueまたはFalseブール値に変換します。

ポートの説明  
x: 任意の値。  
res: TrueまたはFalse。



ノードの説明  
x以上の最小の整数で、xの天井値を返します。

ポートの説明  
x: 数値。  
y: xの小数点以下の数値をプラス側に切り上げた整数。



ノードの説明  
xの逆余弦（アークコサイン）を返します。2つの分岐があります:  
1つは下から連続し、実軸に沿って1から8に右側に伸びます。もう1つは上から連続し、実数軸に沿って-1から-8に左側に伸びています。

ポートの説明  
x: 複素数。  
y: xの複素数アークコサイン。



ノードの説明  
xの双曲線逆余弦（アークコサイン）を返します。

ポートの説明  
x: 複素数。  
y: xの逆双曲線余弦関数値。



ノードの説明  
xの逆正弦を返します。acos()と同じ分岐を持ちます。

ポートの説明  
x: 複素数。  
y: xの複素数アークサイン。



ノードの説明  
xの双曲線逆正弦（アークサイン）を返します。

ポートの説明  
x: 複素数。  
y: xの逆双曲線正弦関数値。



ノードの説明  
xの逆正接（アークタンジェント）を返します。2つの分岐があります:  
1つは右から連続し1jから8jに虚数軸に沿って伸びています。もう1つは左から連続し、虚軸に沿って-1jから-8jに伸びています。

ポートの説明  
x: 複素数。  
y: xの複素数アークタンジェント。



ノードの説明  
xの双曲線逆正接（アークタンジェント）を返します。

ポートの説明  
x: 複素数。  
y: xの逆双曲線正接関数値。

x complex cos y

ノードの説明
xの余弦値（コサイン）を返します。

ポートの説明
x: 複素数。
y: xの複素数コサイン。

x complex cosh y

ノードの説明
xの双曲線余弦関数値を返します。

ポートの説明
x: 複素数。
y: xの双曲線余弦関数値。

complex e out

ノードの説明
浮動小数点として数学定数「e」を出力します。

ポートの説明
out:

x complex exp y

ノードの説明
指数値e^xを返します。

ポートの説明
x: 複素数。
y: 結果。



ノードの説明
指定された複素数の実数と虚数を抽出します。

ポートの説明
num: 複素数。
real: 実数部分の数値。
imag: 虚数部分の数値。

x complex isfinite y

ノードの説明
xの実数と虚数の両方が有限であればTrueを、そうでなければFalseを返します。

ポートの説明
x: 複素数。
y: TrueまたはFalse。

x complex isinf y

ノードの説明
xの実数または虚数のどちらかが無限であればTrueを、そうでなければFalseを返します。

ポートの説明
x: 複素数。
y: TrueまたはFalse。

x complex isnan y

ノードの説明
xの実数または虚数のどちらかが非数であればTrueを、そうでなければFalseを返します。

ポートの説明
x: 複素数。
y: TrueまたはFalse。



ノードの説明
指定した基数のxの対数を返します。基数が指定されていない場合は、xの自然対数を返します。1つの分岐があり、負の実軸に沿って0から-8に上から連続します。

ポートの説明
x: 複素数。
base: 基数
res: 指定した基数のxの対数。

x complex log10 y

ノードの説明
xの基数10の対数を返します。log()と同じ分岐を持ちます。

ポートの説明
x: 複素数。
y: xの基数10の対数。



ノードの説明
指定した実数と虚数から複素数を作成します。

ポートの説明
real: 実数。
imag: 虚数。
res: 複素数。

x complex phase y

ノードの説明
xの位相（別名xの偏角）を浮動小数点として返します。phase(x)はmath.atan2と同等です(x.imag, x.real)。

ポートの説明
x: 複素数。
y: xの位相（浮動小数点）。

complex pi out

ノードの説明
浮動小数点として数学定数pi（円周率）を出力します。

ポートの説明
out: 複素数のpi（円周率）。(虚数部分は0。)

x complex polar y

ノードの説明
xの極座標表現を返します。xの半径rとxの位相であるphiの組(r, phi)を返します。polar(x)は(abs(x), phase(x))と同等です。

ポートの説明
x: 複素数。
y: rとphiの値(r, phi)。



ノードの説明
極座標rとphiを含む複素数xを返します。

ポートの説明
r: Modulus（半径）。
phi: 位相（極座標）。
res: 複素数。

x complex sin y

ノードの説明
xの正弦値（サイン）を返します。

ポートの説明
x: 複素数。
y: xの複素数サイン。

x complex sinh y

ノードの説明
xの双曲線正弦（サイン）を返します。

ポートの説明
x: 複素数。
y: xの双曲線正弦関数値。

x complex sqrt y

ノードの説明
xの平方根を返します。

ポートの説明
x: 複素数。
y: xの平方根。

x complex tan y

ノードの説明
xの正接値（タンジェント）を返します。

ポートの説明
x: 複素数。
y: xの複素数タンジェント。

x complex tanh y

ノードの説明
xの双曲線正接（タンジェント）を返します。

ポートの説明
x: 複素数。
y: xの双曲線正接関数値。

x conjugate res

ノードの説明
xの複素共役を返します。

ポートの説明
x: 複素数。
res: xの複素共役。



ノードの説明
xの絶対値を特定の符号を付けた浮動小数点で返します。

ポートの説明
x: 数値。
y: コピーされるべき符号がある数値。
res: yの符号のx。

x cos y

ノードの説明
y = cos(x):
xの余弦（コサイン）を返します。yの単位はラジアンです。

ポートの説明
x: ラジアン単位の角度。
y: xのコサイン。

x cosh y

ノードの説明
xの双曲線余弦関数値を返します。

ポートの説明
x: ラジアン単位の角度。
y: xの双曲線余弦関数値。



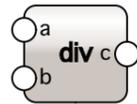
ノードの説明  
角度xを度数値からラジアン値に変換します。

ポートの説明  
x: 度数単位での角度。  
y: ラジアン値に変換されたx。



ノードの説明  
2点間の距離を計算します。

ポートの説明  
a: 2Dまたは3Dの座標。  
b: 2Dまたは3Dの座標。  
l: a点とb点の距離。



ノードの説明  
2つの値を割り算します:  $c = a / b$

ポートの説明  
a: 除数。  
b: 被除数。  
c: 商。



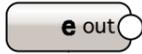
ノードの説明  
切り捨て除算です。整数で返します。  $c = a / b$

ポートの説明  
a: 除数。  
b: 被除数。  
c: 商。小数点以下切り捨て (整数)。



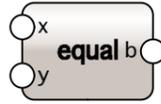
ノードの説明  
xとyの切り捨て除算の商と剰余をペアを整数で返します。

ポートの説明  
x: 除数。  
y: 被除数。  
div: 商。小数点以下切り捨て (整数)。  
mod: 余り。



ノードの説明  
数学定数「e」を出力します。

ポートの説明  
out: eの値。



ノードの説明  
 $b = (x == y)$

コントロールの説明  
epsilon:  
比較の許容エラー誤差 (または、精度しきい値) を指定します。

ポートの説明  
x: 第一値。  
y: 第二値。  
b: TrueまたはFalse。



ノードの説明  
xでの誤差関数を返します。この機能は、累積標準正規分布のように、伝統的な統計関数を計算するために使用することができます:  $\text{phi}(x) = (1.0 + \text{erf}(x / \sqrt{2.0})) / 2.0$

ポートの説明  
x: 数値。  
y: 結果。



ノードの説明  
xでの相補誤差関数を返します。相補誤差関数とは  $1.0 - \text{erf}(x)$  で定義されます。この関数はxが大きな数の場合に効果を発揮します。

ポートの説明  
x: 数値。  
y: 結果。



ノードの説明  
 $y = e^x$   
x: 指定した数値の指数を返します。

ポートの説明  
x: 指数。  
y: 結果。



ノードの説明  
 $y = e^x - 1$   
小さな実数xについて、減算は、精度の大幅な損失をもたらしますが、この関数は完全に精度を保ち値を計算します。

ポートの説明  
x: 指数。  
y: 結果。



ノードの説明  
xの階乗を返します。

ポートの説明  
x: 正の整数。  
y: x以下のすべての正の整数から得た値。



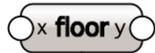
ノードの説明  
入力行列を反転します。

ポートの説明  
inMat: 行列。  
mat: 反転した行列。



ノードの説明  
xを浮動小数点に変換します。

ポートの説明  
x: 数値。  
res: 浮動小数点。



ノードの説明  
xの床値 (x以下の最大の整数) を返します。

ポートの説明  
x: 数値。  
y: xの小数点以下の数値をマイナス側に切り捨てた整数。



ノードの説明  
(m, e) のペアとしてxの仮数と指数を返します。可搬性を保ったまま浮動小数点型の内部表現を「分解 (pick apart)」するために使用します。

ポートの説明  
x: 浮動小数点数。この数がゼロの場合は、(0, 0, 0)を返し、それ以外の場合は、 $0.5 \leq \text{abs}(m) < 1$ を返します。  
m: 仮数。「m」は浮動小数点数 (float型) です。  
e: 指数。「e」は厳密に  $x == m * 2^e$  であるような整数型です。



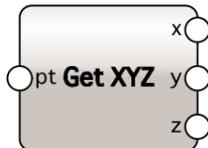
ノードの説明  
xのガンマ関数を返します。

ポートの説明  
x: 数値。  
y: 結果。



ノードの説明  
ポイントまたはベクトルのx、y成分を返します。

ポートの説明  
pt: 2Dポイントまたはベクトル。  
x: X座標。  
y: Y座標。



ノードの説明  
ポイントまたはベクトルのx,y,z成分を返します。

ポートの説明  
pt: 3Dポイントまたはベクトル。  
x: X座標。  
y: Y座標。  
z: Z座標。



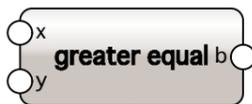
ノードの説明  
ポイントまたはベクトルのx,y,z成分と原点からのオフセットを返します。

ポートの説明  
pt : 3Dポイントまたはベクトル。  
x : X座標。  
y : Y座標。  
z : Z座標。  
off : 原点からの距離。



ノードの説明  
 $b = (x > y)$

ポートの説明  
x : 第一値。  
y : 第二値。  
b : TrueまたはFalse。



ノードの説明  
 $b = (x \geq y)$

コントロールの説明  
epsilon :  
比較の許容エラー誤差 (または、精度しきい値) を指定します。

ポートの説明  
x : 第一値。  
y : 第二値。  
b : TrueまたはFalse。



ノードの説明  
同じx,yの位置にある最高のポイント (座標) を返します。

ポートの説明  
pts : 3D座標のリスト。  
high :  
3D座標のリストです。ptsで指定したリスト内に同じx,yの位置にある座標セットがある場合、Z値が最も高い座標をそれぞれ返します。



ノードの説明  
ユークリッドノルム $\sqrt{x^2 + y^2}$ を返します。(例: 直角三角形の斜辺)

ポートの説明  
x : 隣辺の長さ。  
y : 別の隣辺の長さ。  
z : 斜辺。



ノードの説明  
xを整数値に変換します。

ポートの説明  
x : 数値。  
res : 整数値。



ノードの説明  
指定した行列で2D/3Dの入力点を逆変換します。

ポートの説明  
inMat : 座標変換行列。  
pt : 2Dまたは3Dの座標。  
res : 変換された点。



ノードの説明  
xの逆行列を返します。

ポートの説明  
x : 数値または行列。  
res :  
数値の場合はそのビット単位の逆数、行列の場合はその逆行列。



ノードの説明  
指定した行列の反転行列を返します。

ポートの説明  
inMat : 行列。  
mat : 反転行列。



ノードの説明  
指定した行列が既に反転しているかどうかをチェックします。

ポートの説明  
mat : 行列。  
bool : TrueまたはFalse。



ノードの説明  
現在の行列が単位行列であるかどうかをチェックします。

ポートの説明  
mat : 行列。  
bool : TrueまたはFalse。



ノードの説明  
xが無限でも数字以外の値でもない場合はTrue、それ以外の場合はFalseを返します(注: 0.0は有限数として扱われます)。

ポートの説明  
x : 浮動小数点値。  
res : TrueまたはFalse。



ノードの説明  
xが正または負の無限数の場合はTrue、そうでない場合はFalseを返します。

ポートの説明  
x : 浮動小数点値。  
res : TrueまたはFalse。



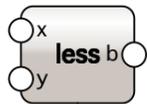
ノードの説明  
xが数字以外の値の場合はTrue、そうでない場合はFalseを返します。

ポートの説明  
x : 浮動小数点値。  
res : TrueまたはFalse。



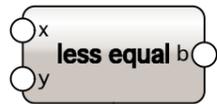
ノードの説明  
 $x * (2^i)$ を返します。frexp()の逆数です。

ポートの説明  
x : 仮数。  
i : 指数。  
y :  $x * (2^i)$ の値 (浮動小数点)。



ノードの説明  
 $b = (x < y)$

ポートの説明  
x : 第一値。  
y : 第二値。  
b : TrueまたはFalse。



ノードの説明  
 $b = (x \leq y)$

コントロールの説明  
epsilon :  
比較の許容エラー誤差 (または、精度しきい値) を指定します。

ポートの説明  
x : 第一値。  
y : 第二値。  
b : TrueまたはFalse。



ノードの説明  
xのガンマ関数における絶対値の自然対数を返します。

ポートの説明  
x : 数値。  
y : 結果。



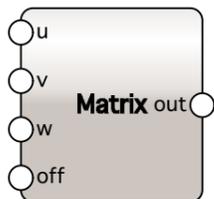
ノードの説明  
引数が1つの場合は、数値xの (eを底とする) 自然対数を返します。引数が2つの場合は、 $\log(x)/\log(\text{base})$ として計算するbaseを底としたxの対数を返します。

ポートの説明  
x : 数値。  
base :  
対数の底。特に指定しない場合はe。  
y : 結果。



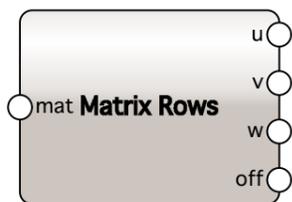
ノードの説明  
1+xの自然対数を返します（底はe）。結果は0近傍のxに対して正確になるように計算されます。

ポートの説明  
x: 数値。  
y: 結果。



ノードの説明  
行列を作成します。

ポートの説明  
u: ベクトルu。  
v: ベクトルv。  
w: ベクトルw。  
off: オフセットベクトル。  
out: 出力する行列。



ノードの説明  
指定した行の行列(x,y,z)を取得します。

ポートの説明  
mat: 行列。  
u: ベクトルu。  
v: ベクトルv。  
w: ベクトルw。  
off: オフセットベクトル。



ノードの説明  
最小値を返します。

ポートの説明  
x: 第一値。  
y: 第二値。  
res: xとyを比較した場合の最小値。



ノードの説明  
リストのすべての値の乗算: out = list[0] \* list[1] \* ...

ポートの説明  
list: 乗算を行う値のリスト。  
out: リストのすべての値の積。



ノードの説明  
2を底とするxの対数を返します。通常log(x, 2)よりも正確です。

ポートの説明  
x: 数値。  
y: 結果。



ノードの説明  
指定した列の行列(x,y,z,off)を取得します。

ポートの説明  
mat: 行列。  
col0: 1番目の列ベクトル。  
col1: 2番目の列ベクトル。  
col2: 3番目の列ベクトル。



ノードの説明  
拡大縮小を行うための行列を新規作成します。

ポートの説明  
x: x方向への倍率。  
y: y方向への倍率。  
z: z方向への倍率。  
mat: 拡大縮小行列。



ノードの説明  
res = x % y 除算の余りを返します。

ポートの説明  
x: 除数。  
y: 被除数。  
out: 余り。



ノードの説明  
y = -x

ポートの説明  
x: 数値。  
res: xの反数。



ノードの説明  
xの常用対数（底が10.）を返します。通常log(x, 10)よりも正確です。

ポートの説明  
x: 数値。  
y: 結果。



ノードの説明  
回転を実行するための回転行列を新規作成します。

ポートの説明  
angle: 度数単位での回転角度。  
axisVec3D: 回転の軸を表す3Dベクトル。  
mat: 回転行列。



ノードの説明  
位置の変更（平行移動）を行うための行列を新規作成します。

ポートの説明  
pt: オフセットベクトル。  
mat: 平行移動行列。



ノードの説明  
xの整数値部分と小数値部分を返します。

ポートの説明  
x: 浮動小数点値。  
int: xの整数値部分。  
frac: xの小数値部分。



ノードの説明  
ベクトルを正規化します。

ポートの説明  
inVec: ベクトル。  
vec: 正規化されたベクトル。



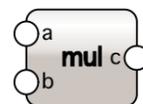
ノードの説明  
同じx,yの位置にある最も低いポイント（座標）を返します。

ポートの説明  
pts: 3D座標のリスト。  
low: 3D座標のリスト。ptsのリストの中で同じx,yの位置にある最も低いポイントのZ値を持ちます。



ノードの説明  
最大値を返します。

ポートの説明  
x: 第一値。  
y: 第二値。  
res: xとyを比較した場合の最大値。



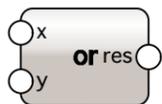
ノードの説明  
aとbの乗算: c = a \* b

ポートの説明  
a: 被乗数。  
b: 乗数。  
c: aとbの積。



ノードの説明  
比較演算です。値が等しくない場合にTrueを返します。b = (x != y)

ポートの説明  
x: 第一値。  
y: 第二値。  
b: TrueまたはFalse。



ノードの説明  
xとyのビット単位論理和です。

ポートの説明  
x : 第一値。  
y : 第二値。  
res : 結果。



ノードの説明  
数学定数phi (ファイ) を出力します。黄金比とも言います。

ポートの説明  
out : phi (ファイ)。



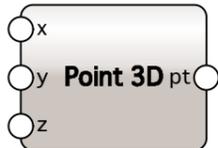
ノードの説明  
数学定数pi (円周率) を出力します。

ポートの説明  
out : 円周率。



ノードの説明  
x、yの値から2D座標を作成します。

ポートの説明  
x : 原点から x 軸方向への距離の値。  
y : 原点から y 軸方向への距離の値。  
pt : 2Dの座標。



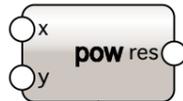
ノードの説明  
x、y、zの値から3D座標を作成します。

ポートの説明  
x : 原点から x 軸方向への距離の値。  
y : 原点から y 軸方向への距離の値。  
z : 原点から z 軸方向への距離の値。  
pt : 3Dの座標。



ノードの説明  
指定した行列で2Dまたは3Dの座標を変換します。

ポートの説明  
inMat : 行列。  
pt : 2Dまたは3Dの座標。  
res : 指定した行列で変換された座標。



ノードの説明  
xのy乗を返します。例外ケース : pow(1.0, x)とpow(x, 0.0)の場合は、たとえxがゼロまたは数値以外だとしても常に1.0を返します。xとyの両方が有限で、xは負数、yは整数ではない場合、pow(x, y)は未定義になりValueErrorを返します。

ポートの説明  
x : 基数  
y : 指数。  
res : 結果。



ノードの説明  
3D空間上のプレーナー図形に2D座標を投影します。

ポートの説明  
pts : 2Dの座標。  
obj : 2Dオブジェクト。  
pts3D : 3Dの座標。



ノードの説明  
ラジアン単位の角度xを度数単位の角度に変換します。

ポートの説明  
x : ラジアン単位の角度。  
y : 度数単位に変換されたx。



ノードの説明  
「high」と「low」の間のランダムな数値を返します。

ポートの説明  
high : 数値。  
low : 他の数値。  
r : ランダムな数値 (low < r < high) 。



ノードの説明  
指定した行列に回転を適用します。回転は中心点と3つの軸まわりでの回転角度で定義します。

ポートの説明  
inMat : 行列。  
centerPt : 回転の中心  
angXDeg : X軸まわりでの回転角度 (度数)。  
angYDeg : Y軸まわりでの回転角度 (度数)。  
angZDeg : Z軸まわりでの回転角度 (度数)。  
mat : 回転行列。



ノードの説明  
指定した角度と指定した軸まわり (X軸、Y軸、またはZ軸) で、指定した行列に回転を適用します。回転をどのように行列に適用するかを選択できるオプションがあります : after (後) またはbefore (前) 。

コントロールの説明  
Axis : まわりを回転させる軸の名前を指定します。  
Mode : Before (前) またはAfter (後) に実行するかを指定します。

ポートの説明  
inMat : 行列。  
angDeg : 度数単位での回転角度。  
mat : 回転行列。



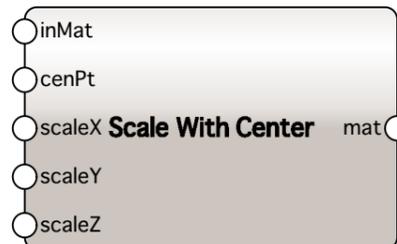
ノードの説明  
xの小数点以下の値をnで指定した桁数で丸めます。値を指定しない場合は、0がデフォルトで指定されます。

ポートの説明  
x : 浮動小数点数。  
n : 整数値。  
res : 数値xがn桁で丸められた浮動小数点数。



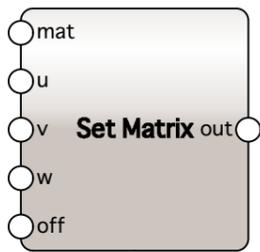
ノードの説明  
指定した行列に指定した倍率を適用します。

ポートの説明  
inMat : 行列。  
scaleX : X方向への倍率。  
scaleY : Y方向への倍率。  
scaleZ : Z方向への倍率。  
mat : 倍率を変更した行列。



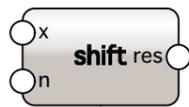
ノードの説明  
指定した座標で指定した行列に指定した倍率を適用します。

ポートの説明  
inMat : 行列。  
cenPt : 中心点。  
scaleX : x方向への倍率。  
scaleY : y方向への倍率。  
scaleZ : z方向への倍率。  
mat : 倍率を変更した行列。



**ノードの説明**  
 行列の行を設定します。連結している値のみを設定して、結果の行列を返します。

**ポートの説明**  
 mat : 行列。  
 u : ベクトルu。  
 v : ベクトルv。  
 w : ベクトルw。  
 off : オフセットベクトル。  
 out : 出力する行列。



**ノードの説明**  
 xを"n"ビットシフトさせます。シフトする方向はパラメータで指定します。

**コントロールの説明**  
 Direction :  
 シフトする方向を指定します: Left (左方向) またはright (右方向)。

**ポートの説明**  
 x :  
 2の補数のビットストリングとして扱われる数値。  
 n : シフトさせるビット数。  
 res :  
 指定した方向に n ビットシフトしたx。



**ノードの説明**  
 $y = \sin(x)$ :  
 正弦 (サイン) を返します。xの単位はラジアンです。

**ポートの説明**  
 x : ラジアン単位の角度。  
 y : xの正弦値 (サイン)。



**ノードの説明**  
 xの双曲線正弦 (サイン) を返します。

**ポートの説明**  
 x : ラジアン単位の角度。  
 y : xの双曲線正弦関数値。



**ノードの説明**  
 xの平方根を返します。

**ポートの説明**  
 x : 数値。  
 y : 結果。



**ノードの説明**  
 2つの数字で引き算をします:  $c = a - b$

**ポートの説明**  
 a : 被減数。  
 b : 減数。  
 c : 差分。



**ノードの説明**  
 すべての値を引き算します:  $out = list[0] - list[1] - \dots$

**ポートの説明**  
 list :  
 最初の値は被減数を、次に来る値は減数を示す数字のリスト。  
 out : 差分。



**ノードの説明**  
 $y = \tan(x)$ :  
 正接値 (タンジェント) を返します。xの単位はラジアンです。

**ポートの説明**  
 x : ラジアン単位の角度。  
 y : xの正接値。



**ノードの説明**  
 xの双曲線正接関数値を返します。

**ポートの説明**  
 x : ラジアン単位の角度。  
 y : xの双曲線正接関数値。



**ノードの説明**  
 入力行列を変換します。変換をいつ行列に適用するかを選択できるオプションがあります: after (後) またはbefore (前)。

**コントロールの説明**  
 Mode :  
 Before (前) またはAfter (後) に実行するかを指定します。

**ポートの説明**  
 inMat : 行列。  
 pt : オフセットベクトル。  
 mat : 変換された行列。



**ノードの説明**  
 行列の転置を返します。転置行列とは、オリジナルの列が行となる新しい行列です。この操作では並進行列はそのまま回転行列のみが転置します。

**ポートの説明**  
 inMat : 行列。  
 mat : 転置行列。



**ノードの説明**  
 小数点以下を切り捨てて積分値 (通常は整数) にした実数値 x を返します。

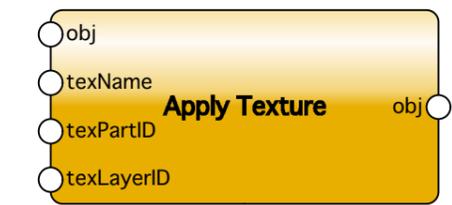
**ポートの説明**  
 x : 浮動小数点値。  
 y : xの整数値部分。



**ノードの説明**  
 xとyのビット排他論理和です。

**ポートの説明**  
 x : 第一値。  
 y : 第二値。  
 res : 結果。

## 表示関連



### ノードの説明

指定したオブジェクトに指定したテクスチャを適用します。

### ポートの説明

inObj : テクスチャを適用したいオブジェクト。  
 texName : 適用するテクスチャ名。  
 texPartID : テクスチャを貼る範囲のID。Script ReferenceのAppendixに詳細情報があります。  
 texLayerID : テクスチャのレイヤID。0で基準、0を超えるとデカールとして扱います。  
 obj : テクスチャ適用後のオブジェクト結果。



### ノードの説明

入力したオブジェクト情報に、このノードの属性を割り当てて出力します。

### ポートの説明

IN : オブジェクトの情報。  
 OUT : 割り当てられた属性を持つオブジェクト。

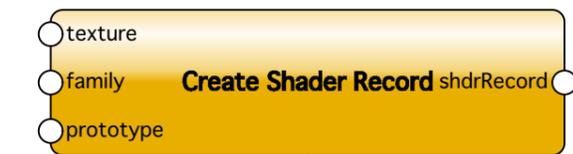


### ノードの説明

位置と回転を指定し、画像リソースからビットマップ（ペイント）オブジェクトを画面上に作成します。

### ポートの説明

image : ビットマップを作成するイメージリソース。  
 pnt2D : ビットマップの位置。  
 rotAng : ビットマップの回転角度（度）。  
 paint : 作成した新しいビットマップ（ペイント）オブジェクト。



### ノードの説明

指定した属性をもつ、テクスチャを作成します。

### ポートの説明

texture : テクスチャのハンドル。  
 family : 属性番号(1 = 色属性, 2 = 反射属性, 3 = 透明属性, 4 = バンプ属性, 5 = バックグラウンド, 6 = フォアグラウンド, 7 = スケッチ(RW-アート))。  
 prototype : 属性内の特定の設定（定数は属性固有の値に依存）。  
 shdrRecord : シェーダレコード。  
 Vectorworks内のテクスチャはシェーダレコードと呼ばれるレコードを使用して格納されます。レコード値はVectorworks内で他のレコードのようにRecordHandlerルーチンを使用してアクセスされません。



### ノードの説明

初期値の新規テクスチャを作成します。

### ポートの説明

texture : 作成されたテクスチャ。



### ノードの説明

選択したシェーダレコードのテクスチャ用のビットマップオブジェクトを作成します。画像ファイルを選択するダイアログボックスが表示されます。「キャンセル」をクリックするか、シェーダがイメージベースのシェーダではない場合はNULLを返します。

### ポートの説明

shdrRecord : シェーダレコードオブジェクト。  
 texBitmap : 作成されたテクスチャ用ビットマップオブジェクト。



### ノードの説明

デフォルトのテクスチャマッピング情報を持つようにオブジェクトを設定します。

### ポートの説明

inObj : オブジェクトの情報。  
 texPartID : テクスチャを貼る範囲のID。Script ReferenceのAppendixに詳細情報があります。  
 texLayerID : テクスチャのレイヤID。0で基準、0を超えるとデカールとして扱います。  
 obj : 結果のオブジェクト。



### ノードの説明

シェーダを編集するための編集ダイアログボックスが表示されます。

### ポートの説明

inShaderRecord : 編集するシェーダレコード。  
 dismiss : 「OK」ボタンを押してダイアログボックスを閉じるとTrue、そうでない場合はfalse。  
 shdrRecord : 編集されたシェーダレコード。



### ノードの説明

指定したテクスチャについてのテクスチャ編集ダイアログボックスを表示します。

### ポートの説明

inTexture : 編集したいテクスチャ。  
 isChanged : テクスチャを変更するとTrue、そうでない場合はfalse。  
 texture : 編集したテクスチャ。

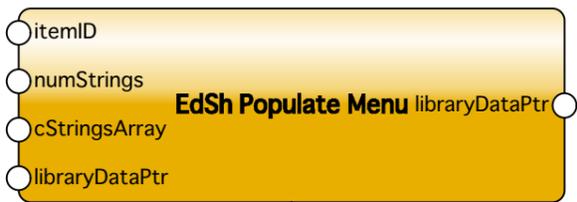


### ノードの説明

シェーダのパラメータ値を編集するためのダイアログボックスレイアウトを作成します。

### ポートの説明

shaderNameCStr : シェーダ名。  
 paramPtr : シェーダのパラメータ。  
 libraryDataPtr : シェーダの編集ダイアログボックスのレイアウト。



**ノードの説明**  
シェーダ編集ダイアログボックスのポップアップにメニュー項目を追加します。

**ポートの説明**  
itemID : ポップアップのID。  
numStrings : ポップアップの項目数。  
cStringsArray : ポップアップ内項目の位置揃え。  
inLibraryDataPtr : シェーダの編集ダイアログボックスのレイアウト。  
libraryDataPtr : 結果のダイアログボックスレイアウト。



**ノードの説明**  
シェーダの編集ダイアログボックスのレイアウトを実行します。

**ポートの説明**  
libraryDataPtr : シェーダの編集ダイアログボックスのレイアウト。  
userHitOK : ユーザーが「OK」をクリックするとTrue、そうでない場合はfalse。



**ノードの説明**  
対象の図形に割り当てられたクラスを返します。図形に割り当てられたクラスがない場合は"一般"を返します。

**ポートの説明**  
obj : オブジェクトの情報。  
styleClass : 図形に割り当てられているクラス。



**ノードの説明**  
指定したクラスに設定しているテクスチャを返します。

**コントロールの説明**  
Property : テクスチャの適用タイプを指定します。

**ポートの説明**  
className : クラスの名前。  
texture : Center - 指定したクラスの壁中央のテクスチャ。Dormer - 屋根のドーマーのテクスチャ。Generic - その他のテクスチャ。Left - 壁左側のテクスチャ。Right - 壁右側のテクスチャ。Top - 屋根のテクスチャ。



**ノードの説明**  
図形作成時に指定したクラスの属性のテクスチャが使用可能かどうかを返します。

**ポートの説明**  
className : 対象のクラス名。  
bool : クラスの属性のテクスチャが使用出来る場合はTrue、そうでない場合はfalse。



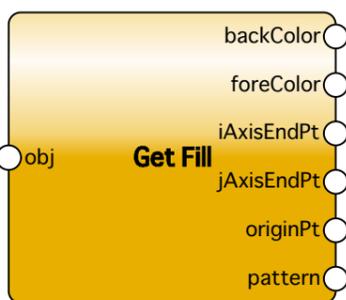
**ノードの説明**  
指定したオブジェクトのために存在する任意の説明文を取得します。

**ポートの説明**  
obj : オブジェクトの情報。  
descText : 説明文。オブジェクトの説明文がない場合は「No descriptions」を返します。



**ノードの説明**  
プレーナー図形の平面マトリックスを返します。

**ポートの説明**  
obj : プレーナー図形の情報。  
success : 対象がプレーナー図形で操作が成功した場合はTrue、そうでない場合はfalse。  
offset3D : 現在のファイルの単位で指定する平面のオフセット。  
rotAng3D : X, Y, Z-軸についての平面の回転角。



**ノードの説明**  
図形の面（塗りつぶし）のプロパティを返します。

**ポートの説明**  
obj : 指定するオブジェクト。  
backColor : 図形の背景色。RGB値は0から65535の範囲内。  
foreColor : 図形の前景色。RGB値は0から65535の範囲内。  
iAxisEndPt : 面のI軸終点（座標）。注：グラデーションやイメージ属性をもつ2次元図形でのみ利用できます。  
jAxisEndPt : 面のJ軸終点（座標）。注：グラデーションやイメージ属性をもつ2次元図形でのみ利用できます。  
originPt : 面の原点（座標）。注：グラデーションやイメージ属性をもつ2次元図形でのみ利用できます。  
pattern : 図形の塗りつぶしパターン。正の値は、模様パレットの模様インデックスに対応します。負の値は、オブジェクトに適用されるベクトルフィルパターンの内部インデックスに対応しています。塗りつぶしパターンとそれに関連する定数はVectorScript付録を参照してください。



**ノードの説明**  
原点やI、J軸の終点座標を返します。注：グラデーションやイメージ属性をもつ2次元図形でのみ利用できます。

**ポートの説明**  
obj : 面を持つ図形の情報。  
originPt : 原点の座標。  
iAxisEndPt : I軸終点の座標。  
jAxisEndPt : J軸終点の座標。

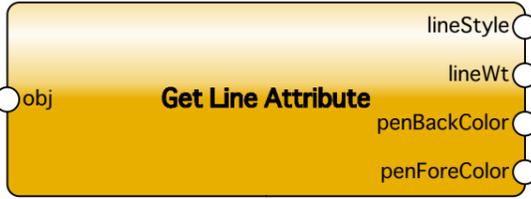
image **Get Image Cropper** cropper

ノードの説明

クロップされた画像の枠クロップを返します。

ポートの説明

image : クロップされた画像。  
cropper : 画像のクロップ図形 (枠)。

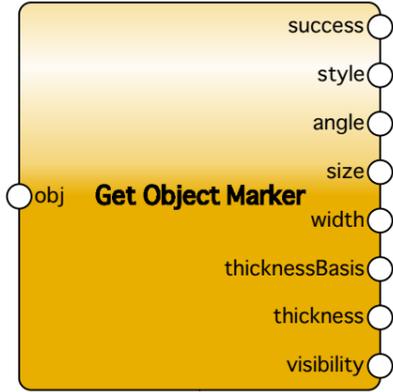


ノードの説明

対象図形の線の属性を返します。

ポートの説明

obj : 指定するオブジェクト。  
lineStyle : 図形の線種。  
lineWt : 図形の線の太さ (ミル)。  
penBackColor :  
線の背景色を指定します。RGB値の範囲は0から65535までです。  
penForeColor :  
線の前景色を指定します。RGB値の範囲は0から65535までです。



ノードの説明

オブジェクトの指定したマーカのすべてのプロパティを取得します。

コントロールの説明

Which Marker :  
対象とするマーカの位置を指定します。

ポートの説明

obj : オブジェクトの情報。  
success :  
この操作が成功したらTrue、そうでなければfalse。  
style : マーカースタイル。  
angle : マーカの角度(0から90度)。  
size : マーカースイズ (インチ)。  
width : マーカの幅 (インチ)。  
thicknessBasis : マーカの太さの定数。  
thickness : マーカの幅 (インチ)。  
visibility :  
マーカの表示設定を表示にする場合はTrue、それ以外はFalse。

obj **Get Opacity** opacity

ノードの説明

オブジェクトの不透明度を返します。不透明度は0から100の範囲内のパーセント値です。

ポートの説明

obj : オブジェクトの情報。  
opacity : オブジェクトの不透明度。

obj **Get Opacity By Class** isByClass

ノードの説明

オブジェクト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。

ポートの説明

obj : オブジェクトの情報。  
isByClass :  
クラスの不透明度を使用している場合はTrue、そうでない場合はfalse。



ノードの説明

テクスチャが割り当てられている場合、指定したファミリのシェーダレコードを返します。

ポートの説明

texture : テクスチャの情報。  
family : 指定したシェーダレコードの属性を指定。1 = 色属性, 2 = 反射属性, 3 = 透明属性, 4 = パンプ属性。  
shdrRecord : 結果のシェーダレコード。

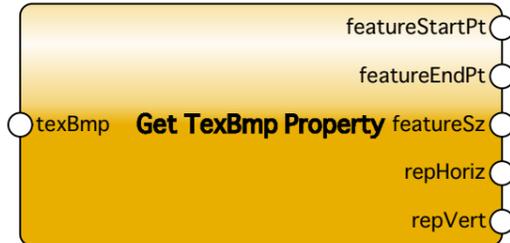
shdrRecord **Get TexBmp** texBmp

ノードの説明

対象のシェーダレコードに設定してあるビットマップオブジェクトを返します。オブジェクトに適用されるテクスチャがない場合は何も返されません。

ポートの説明

shdrRecord : シェーダレコード。  
texBmp :  
シェーダレコードに設定されているビットマップオブジェクト。



ノードの説明

指定したテクスチャビットマップのプロパティを返します。

ポートの説明

texBmp : 対象のテクスチャビットマップオブジェクト。  
featureStartPt :  
テクスチャビットマップの「始点座標」です。始点の座標はビットマップの左上コーナーからのピクセルで設定します。  
featureEndPt :  
テクスチャビットマップの「終点座標」です。終点の座標はビットマップの左上コーナーからのピクセルで設定します。  
featureSz : 実寸でのビットマップのサイズ。  
repHoriz :  
指定したテクスチャビットマップで水平方向の繰り返しを有効にする場合はTrue、無効にする場合はFalseを設定。  
repVert :  
指定したテクスチャビットマップで垂直方向の繰り返しを有効にする場合はTrue、無効にする場合はFalseを設定。

**Get Texture** texName

ノードの説明

ドロップダウンリストで選択したテクスチャを返します。

コントロールの説明

Texture :  
ドキュメント上で利用可能なテクスチャのリストです。

ポートの説明

texName : 選択したテクスチャの名前。

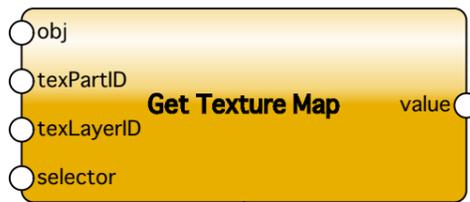


**ノードの説明**

オブジェクトの指定した部分のためのテクスチャレイヤ（ベース+デカル）の数を返します。

**ポートの説明**

obj : オブジェクトの情報。  
 partID : テクスチャを貼る範囲のID。Script ReferenceのAppendixに詳細情報があります。  
 numTexLay : テクスチャレイヤの数。



**ノードの説明**

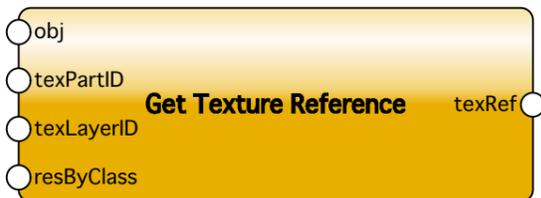
オブジェクトの指定した部分のためのマッピング情報を返します。

**コントロールの説明**

Type : テクスチャマッピングのタイプを選択します。

**ポートの説明**

obj : オブジェクトの情報。  
 texPartID : テクスチャを貼る範囲のID（全体の場合は3）。  
 texLayerID : テクスチャのレイヤID。0で基準、0を超えるとデカルとして扱います。  
 selector : Booleanの場合: Init - 1, flip - 2, repH - 3, repV - 4, long edge - 5, worldZ - 6, auto align - 7  
 Integerの場合: セレクタは1に設定  
 Realの場合: offsetX:1, offsetY:2, scale2D:3, rotate2D:4, radius:5, matrix mat00 through mat32: 6-17  
 value : 結果の値。



**ノードの説明**

指定した図形のテクスチャ番号を返します。

**ポートの説明**

obj : オブジェクトの情報。  
 texPartID : テクスチャを貼る範囲のID。Script ReferenceのAppendixに詳細情報があります。  
 texLayerID : テクスチャのレイヤID。0で基準、0を超えるとデカルとして扱います。  
 resByClass : クラス設定により決定される場合はTrue、そうでない場合はfalse。  
 texRef : オブジェクトのテクスチャ番号。



**ノードの説明**

オブジェクトのテクスチャセットを返します。

**ポートの説明**

obj : オブジェクトの情報。  
 texSet : 0 - オブジェクトのテクスチャ、 1 - 構成要素のテクスチャ。

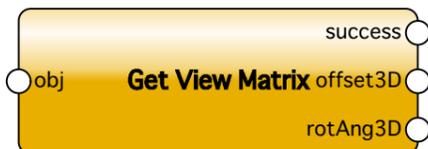


**ノードの説明**

テクスチャのサイズを実寸で返します。

**ポートの説明**

texture : テクスチャの情報。  
 size : テクスチャのサイズ。



**ノードの説明**

レイヤやビューポート図形の見的方向を返します。

**ポートの説明**

obj : オブジェクトの情報。  
 success : ノードが成功したらTrue、そうでない場合はfalse。  
 offset3D : 現在のファイルの単位で指定する平面のオフセット。  
 rotAng3D : X, Y, Z軸についての平面の回転角。



**ノードの説明**

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を返します。

**ポートの説明**

obj : テクスチャの情報。  
 wallTexPart : オブジェクトを使用して壁にあげた開口部の各面に割り当てる壁のテクスチャです。返り値: 0 - 壁の開口部のテクスチャ、 1 - 壁の左側のテクスチャ、 2 - 壁の右側のテクスチャ



**ノードの説明**

指定したオブジェクトがガラスの面の色を使用しているかを返します。

**ポートの説明**

obj : オブジェクトの情報。  
 isByClass : 指定したオブジェクトがガラスの面の色を使用していればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**

指定したオブジェクトがガラスの面の模様を使用しているかを返します。

**ポートの説明**

obj : オブジェクトの情報。  
 isByClass : 指定したオブジェクトがガラスの面の模様を使用していればtrueを返します。



**ノードの説明**

イメージの枠が表示されているかをチェックします。

**ポートの説明**

image : イメージのクロップ枠オブジェクト。  
 isVisible : 枠が表示されていればtrueを返し、そうでない場合はfalseを返します。



**ノードの説明**

指定したイメージがクロップされているかをチェックします。

**ポートの説明**

image : イメージオブジェクト。  
 isCropped : イメージがクロップされていればtrueを返し、そうでない場合はfalseを返します。

obj **Is Line Style By Class** isByClass

ノードの説明  
指定したオブジェクトがクラスの線種を使用しているかどうかを返します。

ポートの説明  
obj: オブジェクトの情報。  
isByClass: オブジェクトがクラスの線種を使用していればtrueを返します。

obj **Is Line Weight By Class** isByClass

ノードの説明  
指定したオブジェクトがクラスの線の太さを使用しているかどうかを返します。

ポートの説明  
obj: オブジェクトの情報。  
isByClass: オブジェクトがクラスの線の太さを使用していればtrueを返します。

obj **Is Marker By Class** isByClass

ノードの説明  
指定したオブジェクトがクラスのマーカーのスタイルを使用しているかどうかを返します。

ポートの説明  
obj: オブジェクトの情報。  
isByClass: オブジェクトがクラスのマーカーのスタイルを使用していればtrueを返します。

obj **Is Object Flipped** isFlipped

ノードの説明  
指定した3Dオブジェクトが反転しているかを返します。この機能は回転体、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に有効です。

ポートの説明  
obj: オブジェクトの情報。  
isFlipped: オブジェクトが反転していればtrueを返し、そうでない場合はfalseを返します。

obj **Is Pen Color By Class** isByClass

ノードの説明  
指定したオブジェクトがクラスの線の色を使用しているかどうかを返します。

ポートの説明  
obj: オブジェクトの情報。  
isByClass: 指定したオブジェクトがクラスの線の色を使用していればtrueを返します。

**Is RndrWrks Available** isAvail

ノードの説明  
Renderworksが利用可能かどうかを判定します。

ポートの説明  
isAvail: Renderworksが利用可能であればtrueを返します。

obj **Is Text Style By Class** isByClass

ノードの説明  
指定したオブジェクトがクラスの文字スタイルを使用しているかを判定します。

ポートの説明  
obj: オブジェクトの情報。  
isByClass: オブジェクトがクラスの文字スタイルを使用している場合はtrueを返します。

obj **Is Textureable Object** isTex

ノードの説明  
指定したオブジェクトがテクスチャマッピングをサポートしているかを判定します。

ポートの説明  
obj: オブジェクトの情報。  
isTex: オブジェクトがテクスチャマッピングをサポートしていればtrueを返し、そうでない場合はfalseを返します。

obj **Set Class** styleClass obj

ノードの説明  
指定したオブジェクトにクラスを割り当てます。

ポートの説明  
inObj: オブジェクトの情報。  
styleClass: オブジェクトに割り当てるクラスの名前。  
obj: 結果のオブジェクト。

className **Set Class Texture** texRef

ノードの説明  
指定したクラスのテクスチャを設定します。

コントロールの説明  
Property: テクスチャの適用タイプを指定します。

ポートの説明  
className: クラスの名前。  
texRef: テクスチャ番号です: Center - 壁の中央のテクスチャです。 Dormer - 屋根のドーマーのテクスチャです。 Generic - テクスチャです。 Left - 壁の左側のテクスチャです。 Right - 壁の右側のテクスチャです。 Top - 屋根上面のテクスチャです。

obj **Set Description** descText success

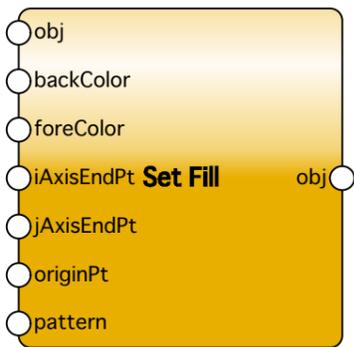
ノードの説明  
ユーザ定義の説明文をオブジェクトに設定します。説明文が未設定の場合はデータノードを追加します。

ポートの説明  
inObj: オブジェクトの情報。  
descText: オブジェクトに設定する説明文。  
obj: 新しい説明文が追加されたオブジェクト。  
success: 説明文が正しく設定された場合はTrueを、エラーが発生した場合はFalseを返します。

obj **Set Entity Matrix** offset3D rotAng3D success

ノードの説明  
プレーナー図形の平面マトリックスを設定します。既に同じマトリックスの平面がファイルに存在する場合は、オブジェクトはその平面に設定されます。ない場合は新しい平面がファイルに追加されます。

ポートの説明  
inObj: オブジェクトの情報。  
offset3D: 現在のファイルの単位で指定する平面のオフセット。  
rotAng3D: X、Y、Z軸に対する平面の回転角度。  
success: 成功した場合はTrueを、それ以外はFalseを返します。  
obj: 結果のオブジェクト。



ノードの説明

指定したオブジェクトにクラス属性の面の色を使用するように設定します。

ノードの説明

指定したオブジェクトにクラス属性の面の模様を使用するように設定します。

ポートの説明

inObj : オブジェクトの情報。  
obj : 結果のオブジェクト。

ポートの説明

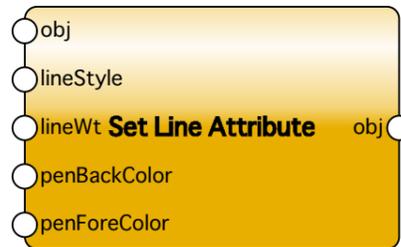
inObj : オブジェクトの情報。  
obj : 結果のオブジェクト。

ノードの説明

指定したオブジェクトの面の属性を設定します。

ポートの説明

inObj : オブジェクトの情報。  
backColor : 背景色を指定します。RGB値の範囲は0から65535までです。  
foreColor : 前景色を指定します。RGB値の範囲は0から65535までです。  
iAxisEndPt : 面のI軸終点 (座標)。注: グラデーションやイメージ属性をもつ2次元図形でのみ利用できます。  
jAxisEndPt : 面のJ軸終点 (座標)。注: グラデーションやイメージ属性をもつ2次元図形でのみ利用できます。  
originPt : 面の原点 (座標)。注: グラデーションやイメージ属性をもつ2次元図形でのみ利用できます。  
pattern : オブジェクトに指定する模様番号です。模様を指定するには、模様のインデックス番号 (正の数値) を使用します。その他の塗りつぶしパターンを使用するには、ベクトルフィルインデックスの負の数値で指定します (インデックス \* -1)。  
obj : 結果のオブジェクト。



ノードの説明

イメージのクロップ枠の表示設定を設定します。

ノードの説明

イメージにクロップ枠を設定します。

ノードの説明

指定したオブジェクトに線の属性を設定します。

コントロールの説明

is visible : クロップ枠を表¥示に設定する場合はチェックを入れます。

ポートの説明

inImage : 対象のイメージオブジェクト。  
cropper : クロップ枠に使用するオブジェクト。  
success : クロップが成功するとTrueを、それ以外はFalseを返します。  
image : 結果のイメージ。

ポートの説明

inCropper : イメージのクロップ枠オブジェクト。  
cropper : 結果のイメージクロップ枠。

ポートの説明

inObj : オブジェクトの情報。  
lineStyle : 値が0から71の範囲の場合は、指定した模様が線の種類として割り当てられます。値が-1から-8の範囲の場合は、指定した線の種類が割り当てられます。模様の定数はVectorScriptのAppendixを参照してください。  
lineWt : オブジェクトに割り当てる線の太さ (ミル単位)。  
penBackColor : 線の背景色を指定します。RGB値の範囲は0から65535までです。  
penForeColor : 線の前景色を指定します。RGB値の範囲は0から65535までです。  
obj : 結果のオブジェクト。

### obj Set Line Style By Class obj

#### ノードの説明

指定したオブジェクトに指定したクラス属性の線の種類を設定します。

#### ポートの説明

inObj: オブジェクトの情報。  
obj: 結果のオブジェクト。



#### ノードの説明

マーカのすべてのプロパティを設定します。

#### コントロールの説明

Which Marker:

変更するマーカの位置を指定します。

#### ポートの説明

inObj: オブジェクトの情報。  
style: マーカースタイル。  
angle: マーカの角度 (0°から90°) です。Integer値のみ使用できます。  
size: マーカの長さ (インチ)。  
width: マーカの幅 (インチ)。  
thicknessBasis: マーカの太さの定数。  
thickness: マーカの幅 (インチ)。  
visibility: マーカの表示設定を表す場合はTrue、それ以外はFalse。  
success: 成功するとTrueを、それ以外はFalseを返します。  
obj: 結果のオブジェクト。

### obj Set Pen Color By Class obj

#### ノードの説明

指定したオブジェクトにクラス属性の線の色を設定します。

#### ポートの説明

inObj: オブジェクトの情報。  
obj: 結果のオブジェクト。

### obj Set Line Weight By Class obj

#### ノードの説明

指定したオブジェクトに指定したクラス属性の線の太さを設定します。

#### ポートの説明

inObj: オブジェクトの情報。  
obj: 結果のオブジェクト。



#### ノードの説明

指定したオブジェクトの不透明度を設定します。

#### ポートの説明

inObj: オブジェクトの情報。  
opacity: 指定する不透明度です。パラメトリック内のオブジェクトに不透明度を設定する場合は、実際の不透明度はパラメトリックオブジェクト自身の不透明度と統合されます。  
obj: 結果のオブジェクト。

### obj Set Marker By Class obj

#### ノードの説明

指定したオブジェクトにクラス属性のマーカースタイルを設定します。設定すると、クラスのマーカースタイルがオブジェクトに使用されません。

#### ポートの説明

inObj: オブジェクトの情報。  
obj: 結果のオブジェクト。

### obj Set Opacity By Class obj

#### ノードの説明

指定したオブジェクトの不透明度に、クラス属性の不透明度を設定します。パラメトリック内のオブジェクトに不透明度を設定する場合は、実際の不透明度はパラメトリックオブジェクト自身の不透明度と統合されます。

#### ポートの説明

inObj: オブジェクトの情報。  
obj: 結果のオブジェクト。

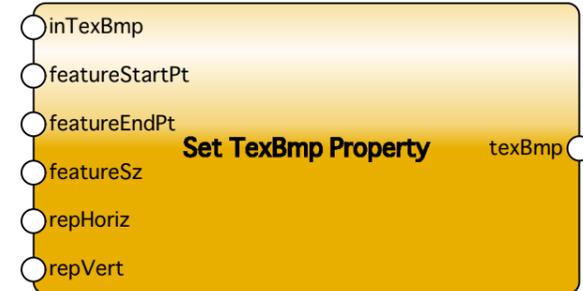


#### ノードの説明

指定したテクスチャのビットマップオブジェクトを設定します。テクスチャが存在しない場合は何も起こりません。

#### ポートの説明

inShaderRecord: シェーダレコード。  
texBmp: テクスチャのビットマップオブジェクト。  
shdrRecord: 結果のシェーダレコード。



#### ノードの説明

指定したテクスチャビットマップのプロパティを設定します。

#### ポートの説明

inTexBmp: 対象のテクスチャビットマップオブジェクト。  
featureStartPt: テクスチャビットマップの「始点座標」です。始点の座標はビットマップの左上コーナーからのピクセルで設定します。  
featureEndPt: テクスチャビットマップの「終点座標」です。終点の座標はビットマップの左上コーナーからのピクセルで設定します。  
featureSz: 実寸でのビットマップのサイズ。  
repHoriz: 指定したテクスチャビットマップで水平方向の繰り返しを有効にする場合はTrue、無効にする場合はFalseを設定。  
repVert: 指定したテクスチャビットマップで垂直方向の繰り返しを有効にする場合はTrue、無効にする場合はFalseを設定。  
texBmp: 結果のテクスチャビットマップ。

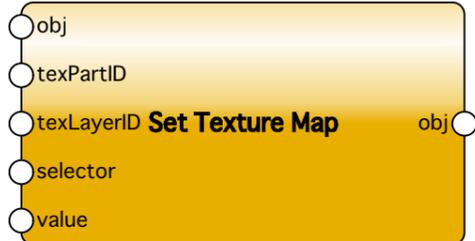
### obj Set Text Style By Class obj

#### ノードの説明

指定したオブジェクトにクラスの文字スタイルを使用するように設定します。設定を取り消すには、「SetTextStyleRef」を使用します。

#### ポートの説明

inObj: オブジェクトの情報。  
obj: 結果のオブジェクト。



#### ノードの説明

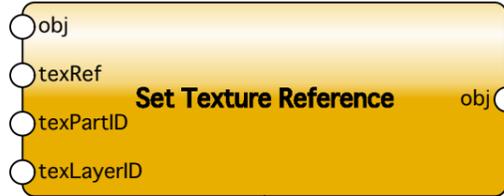
オブジェクトの指定した範囲にテクスチャのマッピング情報を設定します。

#### コントロールの説明

Type: テクスチャマッピングのタイプを選択します。

#### ポートの説明

inObj: オブジェクトの情報。  
texPartID: テクスチャを貼る範囲です。3を指定すると全体になります。  
texLayerID: テクスチャのレイヤID。0で基準、0を超えるとデカールとして扱います。  
selector: Booleanの場合: Init - 1, flip - 2, repH - 3, repV - 4, long edge - 5, worldZ - 6, auto align - 7  
Integerの場合: セレクタは1に設定  
Realの場合: offsetX:1, offsetY:2, scale2D:3, rotate2D:4, radius:5, matrix mat00 through mat32: 6-17  
value: 設定する値。  
obj: 結果のオブジェクト。



#### ノードの説明

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。

#### ポートの説明

inObj: オブジェクトの情報。  
texRef: オブジェクトのテクスチャ番号。  
texPartID: テクスチャを貼る範囲のID。Script ReferenceのAppendixに詳細情報があります。  
texLayerID: テクスチャのレイヤID。0で基準、0を超えるとデカールとして扱います。  
obj: 結果のオブジェクト。



#### ノードの説明

オブジェクトのテクスチャセットを設定します。

#### ポートの説明

inObj: オブジェクトの情報。  
texSet: テクスチャセットを指定します。0 - オブジェクトのテクスチャ、1 - 構成要素のテクスチャ。  
obj: 結果のオブジェクト。



#### ノードの説明

テクスチャのサイズを実寸で設定します。

#### ポートの説明

inTexture: テクスチャの情報。  
size: テクスチャの新しいサイズ。  
texture: 結果のテクスチャ。



#### ノードの説明

レイヤやビューポートの見る方向を設定します。

#### ポートの説明

inObj: オブジェクトの情報。  
offset3D: 現在のファイルの単位で指定する平面のオフセット。  
rotAng3D: X、Y、Z軸に対する平面の回転角度。  
success: 成功した場合はTrueを、それ以外はFalseを返します。  
obj: 結果のオブジェクト。



#### ノードの説明

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を設定します。

#### ポートの説明

inObj: オブジェクトの情報。  
wallTexPart: オブジェクトを使用して壁にあげた開口部の各面に割り当てる壁のテクスチャです。  
値: 0 - 壁の開口部のテクスチャ、1 - 壁の左側のテクスチャ、2 - 壁の右側のテクスチャ  
obj: 結果のオブジェクト。



#### ノードの説明

指定したオブジェクトに指定したテクスチャを適用します。

#### コントロールの説明

Texture Part ID: テクスチャを貼る範囲のID。Script ReferenceのAppendixに詳細情報があります。  
Texture: ファイル内で利用可能なテクスチャのリストです。

#### ポートの説明

inObj: テクスチャを適用するオブジェクト。  
obj: テクスチャ適用後のオブジェクト結果。



#### ノードの説明

指定したオブジェクトのテクスチャタイプのIDを返します。

#### コントロールの説明

Component Part:

#### ポートの説明

texPartID: テクスチャタイプのID。



#### ノードの説明

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。

#### ポートの説明

inObj：オブジェクトの情報。

success：成功するとTrue、それ以外の場合はFalse。

obj：結果のオブジェクト。

# データフロー



ノードの説明  
リストの最後に追加したい単独の項目を指定します。

ポートの説明  
inList: 追加したいリスト。  
item: リストに追加したい項目。  
list: 結果のリスト。



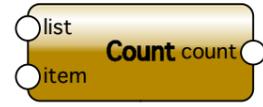
ノードの説明  
リストを指定したアイテム数「n」ずつのリストに分割します。

ポートの説明  
inList: 対象のリスト。  
n: リストごとに分割するアイテムの数。  
list: 結果のリスト。



ノードの説明  
リストのコピーを作成します。

ポートの説明  
list: コピーするリスト。  
res: コピーされたリスト。



ノードの説明  
リスト内の項目の発生回数の合計を返します。

ポートの説明  
list: 与えたリスト。  
item: カウントしたい項目。  
count: リスト内の項目の発生回数。



ノードの説明  
リスト内のインデックスが偶数か奇数かすべての値をフィルタリングします。

コントロールの説明  
pattern: フィルタパターン:  
even (偶数) または odd (奇数) を指定します。

ポートの説明  
inList: リスト。  
list: 結果のリスト。



ノードの説明  
入力したブール値がtrueの場合に入力項目を返します。それ以外の場合は何も返しません。

ポートの説明  
item: 入力項目。  
bool: ブール値。  
out: 結果の項目。  
out2: False項目のリスト。



ノードの説明  
リスト、または複数リストのリストをフラット化します。

ポートの説明  
list: 対象のリスト。  
listOut: フラット化したリスト。



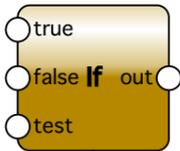
ノードの説明  
リスト内の指定したインデックスにある要素を返します。

ポートの説明  
list:  
index:  
item:



ノードの説明  
リストの長さを返します。

ポートの説明  
list: 対象のリスト。  
length: リストの長さ。



ノードの説明  
条件に合致すればtrueを、そうでない場合はfalse値を渡します。

ポートの説明  
true: 項目。  
false: 項目。  
test: ブーリアン値。  
out: 結果の項目。



ノードの説明  
指定した値の項目の値に等しい最初の項目のリストのインデックスを返します。そのような項目が存在しない場合は-1を返します。項目のリストが指定された場合、出力は見つかったインデックスのリストになります。

ポートの説明  
list: 対象のリスト。  
item: 検索する項目。  
index: 結果のインデックス。



ノードの説明  
値が開始と終了のインデックス間の入力項目値に等しい最初の要素のリスト内のインデックスを返します。何も接続されていない場合はスタート=0、終了はリストの最後です。要素が存在しない場合はエラーになります。

ポートの説明  
list: 対象のリスト。  
item: 入力項目。  
start: 開始インデックス。  
end: 終了インデックス。  
index: 見つかった要素のインデックス。



ノードの説明  
リスト内の指定した位置に複数の項目を挿入します。

ポートの説明  
inList: 対象のリスト。  
item: 挿入する項目。  
index: 項目を追加する位置。  
list: 結果のリスト。



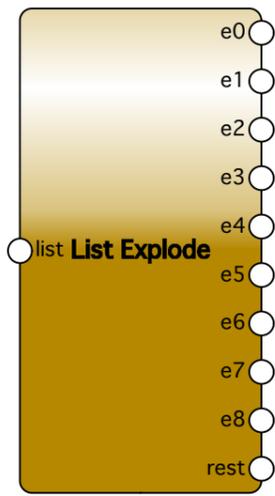
ノードの説明  
項目が既にリストに存在する場合はtrueを返します。

ポートの説明  
list: 入力するリスト。  
item: 検索する項目。  
isMember: 項目がリストに存在すればtrueを返し、そうでない場合はfalseを返します。



ノードの説明  
入力された値からリストを作成します。入力値がリストであれば連結します。

ポートの説明  
a: 1番目の入力。  
b: 2番目の入力。  
res: 結果のリスト。



**ノードの説明**  
リストをその構成する項目に分類します。

**ポートの説明**  
list : リスト。  
e0 : インデックス0の項目 (最初の項目)。  
e1 : インデックス1の項目 (2番目の項目)。  
e2 : インデックス2の項目 (3番目の項目)。  
e3 : インデックス3の項目 (4番目の項目)。  
e4 : インデックス4の項目 (5番目の項目)。  
e5 : インデックス5の項目 (6番目の項目)。  
e6 : インデックス6の項目 (7番目の項目)。  
e7 : インデックス7の項目 (8番目の項目)。  
e8 : インデックス8の項目 (9番目の項目)。  
rest : リストが9より多い場合の、残りの項目のリスト。



**ノードの説明**  
リスト中で最も大きな値の項目を返します。

**ポートの説明**  
list : 対象のリスト。  
item : リスト中で最も大きな値の項目。



**ノードの説明**  
リスト中で最も小さな値の項目を返します。

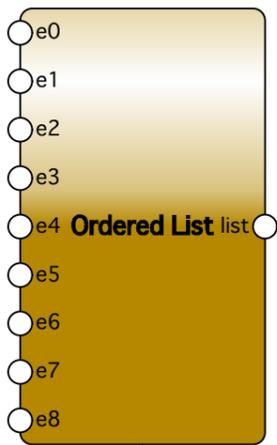
**ポートの説明**  
list : 対象のリスト。  
item : リスト中で最も小さな値の項目。



**ノードの説明**  
指定したモードでリストをミックスし、すべてが同じ長さのリストを同じ数だけ出力します。

**コントロールの説明**  
List Matching : Short List:  
リストを最短の長さに揃えます。¥nLongest List:  
最後の値を繰り返して最長のリストを作成します。¥nCycle:  
短い方のリストを繰り返して最長のリストを作成します。¥nCross Reference:  
すべてを並べ替えたリストを作成します。

**ポートの説明**  
list1 : 第一入力リスト。  
list2 : 第二入力リスト。  
resList1 : 第一出力リスト。  
resList2 : 第二出力リスト。



**ノードの説明**  
複数の要素から番号付きリストを作成します。

**ポートの説明**  
e0 : インデックス0の項目 (最初の項目)。  
e1 : インデックス1の項目 (2番目の項目)。  
e2 : インデックス2の項目 (3番目の項目)。  
e3 : インデックス3の項目 (4番目の項目)。  
e4 : インデックス4の項目 (5番目の項目)。  
e5 : インデックス5の項目 (6番目の項目)。  
e6 : インデックス6の項目 (7番目の項目)。  
e7 : インデックス7の項目 (8番目の項目)。  
e8 : インデックス8の項目 (9番目の項目)。  
list : 作成されるリスト。



**ノードの説明**  
何もしないノードです。単純に入力したものをそのまま出力します:  $y = x$

**ポートの説明**  
x : 入力。  
y : 出力。



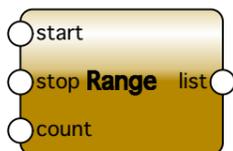
**ノードの説明**  
リスト内の最後のアイテムを削除し、削除したアイテムを返します。

**ポートの説明**  
inList : 対象のリスト。  
list : 結果のリスト。  
item : 指定したリストから削除されたアイテム。



**ノードの説明**  
リスト内の最初のアイテムを削除し、削除したアイテムを返します。

**ポートの説明**  
inList : 対象のリスト。  
list : 結果のリスト。  
item : 指定したリストから削除されたアイテム。



**ノードの説明**  
指定した開始番号と終了番号間で数値範囲を作成します。値を指定しない場合は以下の値がデフォルトで指定されます: start = 0 stop = 1 count = 1

**ポートの説明**  
start : 開始番号。  
stop : 終了番号。  
count : 開始番号と終了番号間のステップ数。  
list : 結果の範囲。



**ノードの説明**  
指定したリストに比例する値のリストを、指定した最大/最小値とともに返します。

**ポートの説明**  
list : 新しい数値の範囲にマッピングする値のリスト。すべての値は最大/最小値に基づいて変更されます。  
max : 新しいリストの最大値。  
min : 新しいリストの最小値。  
remap : 新しいドメインに再マッピングされた入力リスト。



**ノードの説明**  
リスト内から、指定した値を持つ最初の要素を削除します。指定する値がリストの場合は、リスト内のそれぞれの項目に対して削除を実行します。

**ポートの説明**  
inList : 対象のリスト。  
item : 入力項目。  
list : 結果のリスト。



**ノードの説明**  
指定したインデックス番号のアイテムをリスト内から削除し、削除したアイテムを返します。インデックス番号をリストで指定した場合は、それぞれのインデックス番号に該当するアイテムを削除し、削除したアイテムのリストを返します。

**ポートの説明**  
inList : 対象のリスト。  
index : 削除するアイテムのインデックス番号。  
list : 結果のリスト。  
item : リスト内から削除されたアイテム。



**ノードの説明**  
指定したリストを指定したn回数分繰り返し作成します。

**ポートの説明**  
item : 入力項目。  
n : 複製を繰り返し替えず回数。  
list : 結果のリスト。



**ノードの説明**  
リスト内のアイテムを逆順にします。

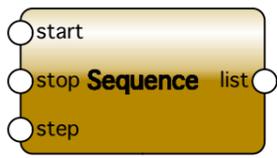
**ポートの説明**  
inList : 対象のリスト。  
list : 結果のリスト。



**ノードの説明**  
リスト内のアイテムを指定した回数で回転します。

**コントロールの説明**  
Direction : 回転方向を指定します。

**ポートの説明**  
inList : 対象のリスト。  
cnt : リストを回転させる合計回数。  
list : 結果のリスト。



**ノードの説明**  
 数列xを作成します:  $x = \text{start} : (\text{x} < \text{stop}) \text{x} + \text{step}$   
 step (startからstopまでの範囲で、stepで指定した数値刻みで増分する数列) 値を指定しない場合は以下の値がデフォルトで指定されます: start = 0 stop = 1 step = 1

**ポートの説明**  
 start: 開始番号。  
 stop: 終了番号。  
 step: 数列内の2つの連続する数値間の差異。スキップ数。  
 list: 結果の数列。



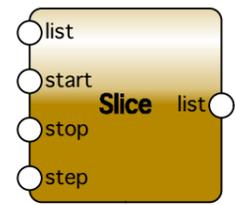
**ノードの説明**  
 指定した開始番号から始まり、指定したスキップ間隔と繰り返し回数で数列を作成します。値を指定しない場合は以下の値がデフォルトで指定されます: start = 0 step = 1 count = 1

**ポートの説明**  
 start: 開始番号。  
 step: 2つの連続する数値間の差異。スキップ数。  
 count: stepの繰り返し回数。  
 list: 結果の範囲。



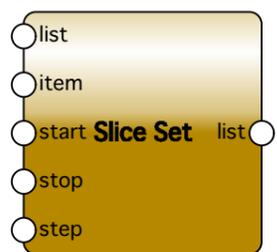
**ノードの説明**  
 入力リスト内で指定したインデックスの要素を指定した入力項目の値に設定します。項目リストとインデックスの両方が指定されている場合、項目すべてをトラバースし、入力リストへの変更を実行します。

**ポートの説明**  
 inList: 対象のリスト。  
 item: 入力項目。  
 index: 入力リスト内で置き換える要素のインデックス。  
 list: 結果のリスト。



**ノードの説明**  
 指定した開始インデックスから終了インデックスでリストをスライスします。スキップ数を指定することもできます。値を指定しない場合は以下の値がデフォルトで指定されます: start = 0 stop = 全リスト step = 1

**ポートの説明**  
 inList: 対象のリスト。  
 start: 開始インデックス。  
 stop: 終了インデックス。  
 step: 結果のリスト内で2つの連続する数値間の「距離」。スキップ数。  
 list: 結果のリスト。



**ノードの説明**  
 指定した開始インデックスから終了インデックスでスライスしたリストに、指定した値を追加します。スキップ数を指定することもできます。値を指定しない場合は以下の値がデフォルトで指定されます: start = 0 stop = 全リスト step = 1

**ポートの説明**  
 inList: 対象のリスト。  
 item: 入力項目。  
 start: 開始インデックス。  
 stop: 終了インデックス。  
 step: 結果のリスト内で2つの連続する数値間の「距離」。スキップ数。  
 list: 結果のリスト。



**ノードの説明**  
 リスト内の要素をソートします。

**ポートの説明**  
 inList: 対象のリスト。  
 list: 結果のリスト。



**ノードの説明**  
 リストを指定したインデックス番号の位置で分割します。

**ポートの説明**  
 list: 分割したいリスト。  
 index: リストを分割する位置。  
 list1: 結果の分割されたリスト1。  
 list2: 結果の分割されたリスト2。



**ノードの説明**  
 行列の転置を返します。転置行列とは、オリジナルの列が行となる新しい行列です。この操作では並進行列はそのまま回転行列のみが転置します。

**ポートの説明**  
 list: 対象のリスト。  
 listOut: 転置した行列。



**ノードの説明**  
 リストを各出力リストに交互に展開します。

**ポートの説明**  
 listin: 対象のリスト。  
 out0: 第一出力リスト。  
 out1: 第二出力リスト。



**ノードの説明**  
 2つのリストを交互に行き来して1つにまとめます。

**ポートの説明**  
 list1: 第一入力リスト。  
 list2: 第二入力リスト。  
 list: 結果のリスト。

# 入力

## Any v

ノードの説明  
Pythonのeval関数を使用して任意の値を作成します。

コントロールの説明  
eval : 評価する文字列を入力します。

ポートの説明  
v : 結果となる値。

## Bool b

ノードの説明  
データパレットのチェックボックスから、TrueまたはFalseの値を生成します。

コントロールの説明  
True : Trueの値を設定する場合はチェックを入れます。

ポートの説明  
b : 値。

## Button

ノードの説明  
ボタンウィジェットをどのように使用するか例です。ボタンを押すと指定した定義の中に含まれているスクリプトを実行します。

コントロールの説明  
Button : スクリプトを実行します。

## Control Geometry obj

ノードの説明  
マリオンネットオブジェクトの定義ネットワークで使われる時、コントロールジオメトリグループの最初のオブジェクトを返します。オブジェクトの外で使用可能ではありません。

ポートの説明  
cntrl : コントロールジオメトリオブジェクト。

## Dim d

ノードの説明  
寸法としての実数を設定します。

コントロールの説明  
Dim : データパレットで数値を指定します。単位を入力すると自動的にファイル単位に変換されます。

ポートの説明  
d : ファイルの単位に合わせた結果の値。

## Int i

ノードの説明  
整数値を返します。

コントロールの説明  
Integer : データパレットから整数値を指定します。

ポートの説明  
i : 値。

## Name obj

ノードの説明  
名前の付いた図形を図面上から取得します。

コントロールの説明  
Object Name : 指定する図形の名前です。Create a duplicate object : 図形の複製を作成するオプションです。マリオンネットネットワークが、名前の付いた図形を変更または使用する場合は、このオプションが必要になることがあります。

ポートの説明  
obj : 名前の付いた図形。

## Objs by Crit objs

ノードの説明  
指定した検索条件に合致するオブジェクトのリスト返します。

コントロールの説明  
Criteria : 検索条件を定義する文字列を入力します。

ポートの説明  
obj : 検索条件に一致するファイル内のオブジェクトのリスト。

## Point2 p

ノードの説明  
データパレットで定義した2D座標を返します。

コントロールの説明  
x : x座標を指定します。  
y : y座標を指定します。

ポートの説明  
p : 2D座標。

## Point3 p

ノードの説明  
データパレットで定義した3D座標を返します。

コントロールの説明  
x : x座標を指定します。  
y : y座標を指定します。  
z : z座標を指定します。

ポートの説明  
p : 3D座標。

## Popup int

ノードの説明  
ポップアップのデータパレットコントロールの使い方をデモンストレーションするノードです。このノードによって返される値は整数です。値は選択したオプションによって異なります：1つ目のオプションを選択した場合は0、その後1ずつ増分します。

コントロールの説明  
Popup : スクリプトエディタで設定した選択オプションのデータパレットコントロールです。

ポートの説明  
output : データパレットで選択したオプションを示す整数値です。返り値の範囲は0から始まり、設定したオプション数-1です。

## Radio Button int

ノードの説明  
データパレットで選択したアイテムに従って、整数値を返します。返り値の範囲は0から始まり、設定したオプション数-1です。

コントロールの説明  
Options : 整数値を返すデータパレットコントロールです。

ポートの説明  
i : 値。

## Real r

ノードの説明  
データパレットから定義するファイルの単位での実数です。

コントロールの説明  
real : ファイルの単位で指定する実数値です。

ポートの説明  
r : 値。

## Slider int

ノードの説明  
スライダの位置に従って整数値を返します。スライダの範囲は、スクリプト内の最後の2つの値（整数）で指定します。

コントロールの説明  
Integer : データパレットスライダです。

ポートの説明  
i : 値。

## String s

ノードの説明  
データパレットで定義する文字列ストリングです。

コントロールの説明  
string : 文字列ストリングです。

ポートの説明  
s : 文字列。

## Vec2 v

ノードの説明  
2次元ベクトルを作成します。

コントロールの説明  
x : x値。  
y : y値。

ポートの説明  
v : 二つ組(x,y)。

## Vec3 v

### ノードの説明

3次元ベクトルを作成します。

### コントロールの説明

x : x値。

y : y値。

z : z 値。

### ポートの説明

v : 三つ組(x,y,z)。