

---

***dw-2000<sup>TM</sup>***

---

**Derived Layers**

**日本語解説マニュアル**

---

**for version 8.x**



dw-2000 version 8.x  
by Design Workshop Technologies

© 2004 by MEDIX International Corp.

All right reserved.

No part of this manual may be copied, reproduced, translated, or reduced to any electronic medium or machine-readable form without the prior written consent of MEDIX International Corp.

MEDIX International Corp. make no warranties, expressed or implied, with respect to this manual, regarding its quality, performance, merchantability, or fitness for a particular purpose, except as expressed herein.

MEDIX International Corp. reserve the right to make improvements to the product described in this manual at any time and without any notice.

Dw-2000 is a trademark of Design Workshop Technologies.  
UNIX is registered trademark of UNIX System Laboratories, Inc.  
Windows is a trademark of Microsoft Corp.

本書の著作権は株式会社メディックスにあります。

本書の内容の一部、または全部を株式会社メディックスの書面による許可なく複製、送信、複写、情報検索のために保存すること及び他の言語に翻訳することを禁じます。

本書の内容は予告なく変更する場合があります。

その他本書に関するお問い合わせは以下までお願い申し上げます：

**株式会社メディックス CAD 課**

〒176-0012 東京都練馬区豊玉北 5-13-1 シティ練馬 6F

電話：03-5984-5555 FAX：03-5984-5556

**e-mail**：[eda@medix.co.jp](mailto:eda@medix.co.jp) **URL**：<http://www.medix.co.jp/Eda/>

# 目次】

<b><u>Introduction</u></b>	1
<b>General Concepts</b>	2
•Physical レイヤーとは	2
•Derived レイヤーとは	2
•Derived レイヤーの働き	3
•Derived レイヤーの種類	4
- Shape Derived Layers	4
- Segment Derived Layers	6
•Terminology	6
<b><u>Using Common Functions</u></b>	9
<b>Derived レイヤーの初期化</b>	10
<b>Derived レイヤーの生成</b>	11
•統合Derivedレイヤーを生成する	11
•非統合Derivedレイヤーを生成する	12
Derivedレイヤーにデータを加える	13
<b>Derived レイヤーを保存する</b>	14
Derivedレイヤーにデータが含まれているか決定する	17
Derivedレイヤーの総面積を計算する	17
Derivedレイヤーをリサイズする	18
非統合Derivedレイヤーを統合する	21
<b>ブール演算</b>	22
•論理積演算 (AND)	24
•論理和演算 (OR)	25
•排他的論理和演算 (XOR)	26
•論理差演算 (SUB)	27
•否定演算 (NOT)	28
•非統合Derivedレイヤーへの論理積演算 (AND)	30

<b>Using Selection Operations</b> .....	<b>31</b>
<b>関連位置による形状 (shape) 選択演算</b> .....	<b>32</b>
形状選択演算 Avoid .....	<b>35</b>
形状選択演算 Butting .....	<b>36</b>
形状選択演算 Butting Only .....	<b>38</b>
形状選択演算 Butting or Overlap .....	<b>39</b>
形状選択演算 Coincident .....	<b>40</b>
形状選択演算 Coincident Include .....	<b>42</b>
形状選択演算 Coincident Inside .....	<b>43</b>
形状選択演算 Include .....	<b>44</b>
形状選択演算 Inside .....	<b>45</b>
形状選択演算 Inside Only .....	<b>46</b>
形状選択演算 Outside .....	<b>47</b>
形状選択演算 Overlap .....	<b>48</b>
形状選択演算 Straddle .....	<b>49</b>
形状選択演算 Touching .....	<b>50</b>
<b>特定のプロパティによる形状 (shape) 選択演算</b> .....	<b>51</b>
形状選択演算 Acute Angles .....	<b>53</b>
形状選択演算 45 Degree Angles .....	<b>55</b>
形状選択演算 Edge Length .....	<b>57</b>
形状選択演算 Number of Edges .....	<b>59</b>
形状選択演算 Number of Holes .....	<b>61</b>
形状選択演算 Octagonal Angles .....	<b>62</b>
形状選択演算 Orthogonal Angles .....	<b>64</b>
形状選択演算 Perimeter .....	<b>66</b>
形状選択演算 Rectangles .....	<b>68</b>

## Using Selection Operations

<b>関連位置によるエッジ選択演算</b> .....	<b>69</b>
•エッジ選択演算 Butting .....	71
•エッジ選択演算 Coincident .....	73
•エッジ選択演算 Touching .....	75
<b>特定のプロパティによるエッジ選択演算</b> .....	<b>76</b>
•エッジ選択演算 Acute Angle .....	78
•エッジ選択演算 Bend Radius .....	79
•エッジ選択演算 45 Degree Angle .....	80
•エッジ選択演算 Length .....	81
•エッジ選択演算 Number of Edges .....	82
•エッジ選択演算 Octagonal .....	83
•エッジ選択演算 Orthogonal .....	84
•エッジ選択演算 Relative Angle .....	85
•エッジ選択演算 Total Length .....	86
<b>関連位置による断片選択演算</b> .....	<b>87</b>
•断片選択演算 Avoid .....	89
•断片選択演算 Butting .....	90
•断片選択演算 Butting or Overlap .....	92
•断片選択演算 Coincident .....	93
•断片選択演算 Inside .....	95
•断片選択演算 Inside Only .....	96
•断片選択演算 Outside .....	97
•断片選択演算 Touching .....	98

<b><u>Using Net Functions</u></b> .....	<b>99</b>
<b>電氣的に接続しているレイヤー</b> .....	<b>100</b>
• 順次接続レイヤー (Sequentially Connecting Layers) .....	<b>100</b>
• 非順次接続レイヤー (Non-Sequentially Connecting Layers) .....	<b>103</b>
<b>レイヤーにネット番号を割り当てる</b> .....	<b>105</b>
<b>ネットに名称を割り当てる</b> .....	<b>107</b>
<b>ネットプロパティリストの作成</b> .....	<b>108</b>
• プロパティリストから計算を実行する .....	<b>109</b>
<b>ネットを使用している形状を選択する</b> .....	<b>110</b>

# Introduction

---

本章では Derived レイヤーの作成・使用に関する、dw-2000 操作の基本コンセプトを解説していきます。

Derived レイヤーは通常のレイアウトで使用される物理的なポリゴンと異なり、より実際のマスクデータとの整合性を考慮し、同一の元素属性を持つ重なり合ったポリゴンデータを一つの形状として論理的なデータとして保持し、様々なレイアウト検証を行えるようにした dw-2000 レイアウトエディタに標準装備されているモジュールです。

**DLA モジュールはレイアウトエディタに標準装備されておりますが、31 ページ以降の選択演算及び99 ページ以降のネット機能に関してはオプションの XDRC モジュールもしくは HLVS モジュールが必要となります。**

本章で解説している項目は以下の通りです：

- Physical レイヤーとは
- Derived レイヤーとは
- Derived レイヤーの働き
- Derived レイヤーの種類
- Terminology



# General Concepts

---

Derived レイヤーを用いて作業を行う前に、まずは「Derived レイヤー」と「Physical レイヤー」との違いについて理解して頂きます。

## Physical レイヤーとは

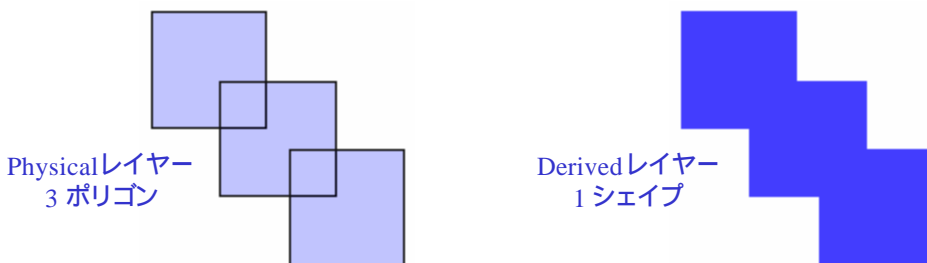
Physical レイヤーは dw-2000 でデータを組みあげていく上で使用され、ストラクチャを作成する様々なマテリアル (基礎素材) のことをいいます。この Physical レイヤーには 0 ~ 255 までの 256 通りの番号、または意味のある名称、例えば Metal や Poly と云った名称を割り振ることができます。

dw-2000 にはこの Physical レイヤーを操作できる多くのコマンドが用意されています。例えば、レイヤー毎に表示色を変更したり 特定のレイヤーのみ表示できるようにしたり、複数のレイヤーに特定の一個の名称を割り振ったりすることなども可能です (例えば、「Metal」と言う一つの素材に 5 番、16 番、30 番と言う三つのレイヤー番号を割り当てる、など)。

## Derived Layers レイヤーとは

Derived レイヤーはレイアウトデータベースからパスやバウンダリを抽出し生成される形状またはポリゴンの集まりのことを言います。また Derived レイヤー同士を論理的に結合することにより生成することも可能です。通常 Derived レイヤーはメモリ領域に置かれますが、データベース上に保存することも可能です。

Derived レイヤーを生成するプロセスでは、以下のように Physical レイヤーにおける重なり合っている部分やカットライン、接合部分は全て省かれます：

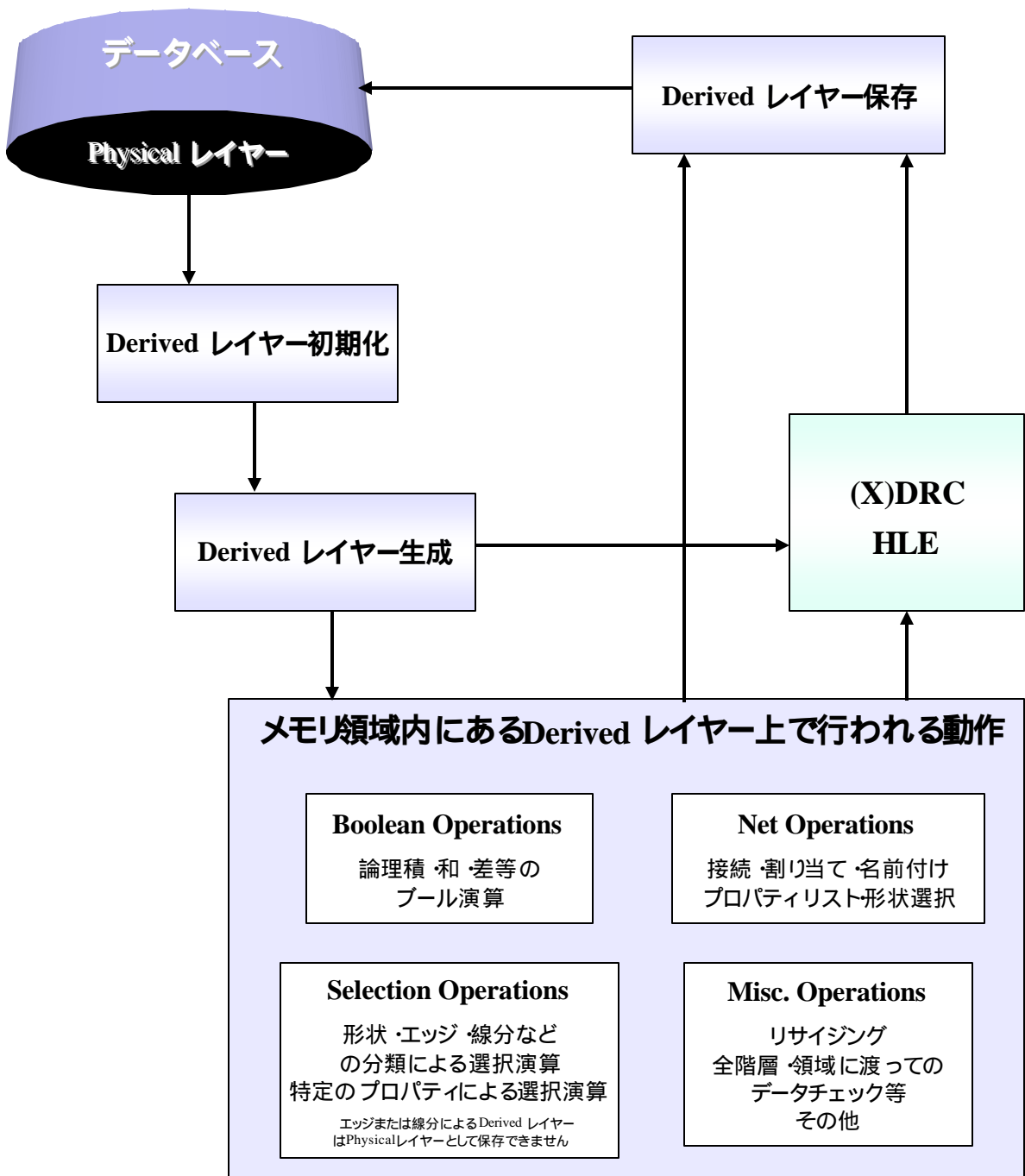


Derived レイヤーは GPE マクロスクリプトで変数として使用することも可能です。これにより様々な GPE 機能でダイレクトにお使い頂けます。

# General Concepts

## Derived レイヤーの働き

Derived レイヤーでの一連の作業の流れは下図のようになります。まずデータベース上から選択した Physical レイヤーを抽出し初期化、生成後ブール演算などの様々な操作を行っていきます。



# General Concepts

---

前項の流れを大まかにまとめると以下の通りとなります：

- ・ データを含むデータベース上の指定したエリアから、ユーザー情報を元に derived レイヤーを初期化する。
- ・ derived レイヤーを生成し、それらにレイヤータイプの GPE 変数を割り当てる。
- ・ ブール演算や形状選択などの動作をその derived レイヤーに対して行う。
- ・ それらの動作 (演算)結果が GPE 変数として返される。
- ・ その変数を XDRC や HLE と行った操作における入力情報として使用する。
- ・ 動作 (演算)結果をデータベースに戻す (保存する)。

## Derived レイヤーの種類

Derived レイヤーは以下のどちらかに分離されます：

- ・ shape derived layer (形状による Derived レイヤー)
- ・ segment derived layer (断片による Derived レイヤー)

### Shape Derived Layers

shape derived layer には閉ポリゴンのみが含まれます。またそれらは統合・非統合どちらの形状も持つことができます。この derived レイヤーは以下のいずれかの操作によって生成されます：

- ・ DiaCreate コマンド (統合レイヤーの生成) または DiaCreateUnmerged コマンド (非統合レイヤーの生成) を使用する  
( 11 ページ 「Derived レイヤーの生成」の項参照 )
- ・ DiaSelShape コマンドを使用して既存の derived レイヤーから形状を選択する  
( 32 ページ 「関連位置による形状選択」及び 51 ページ 「特定のプロパティによる形状選択」の項参照 )
- ・ shape レイヤーを返す XDRC コマンドを使用する
- ・ ブール演算コマンド (DiaAND, DiaOR, DiaXOR 及び DiaSUB) を使用する  
( 22 ページ 「ブール演算を使用する」の項参照 )
- ・ 閉ポリゴンを空のレイヤーまたは shape レイヤーに加える DiaCoordsToLayer コマンドを使用する  
( 13 ページ 「Derived レイヤーにデータを加える」の項参照 )

shape derived layer は DiaSave コマンドを使用することにより Physical レイヤーとしてデータベースに保存することができます (14 ページ 「Derived レイヤーの保存」の項参照)。

## General Concepts

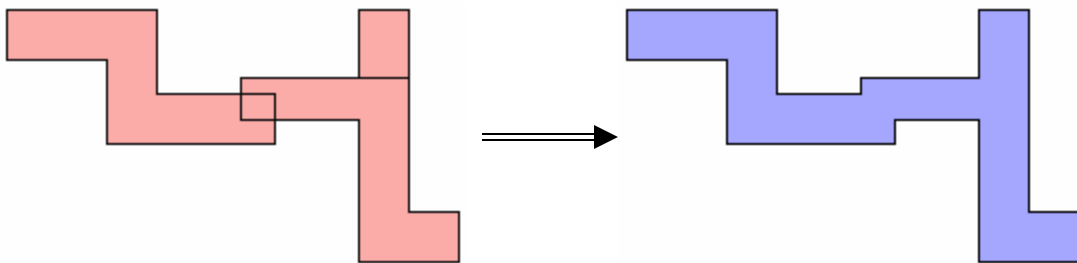
---

### 統合レイヤー (Merged Layers)

統合レイヤーは以下のいずれかの方法で生成することができます：

- DllCreate コマンドを使用する。重なっている全ての形状は統合されたDerived レイヤーとなります。この統合レイヤーは Derived レイヤーを生成する際のデフォルトとなっております。詳細につきましては 11 ページ 統合 Derived レイヤーの生成」の項をご参照下さい。
- DllMerge コマンドを使用して非統合レイヤー (Unmerged layer) を統合する。詳細につきましては 21 ページ 非統合 Derived レイヤーの統合」の項をご参照下さい。

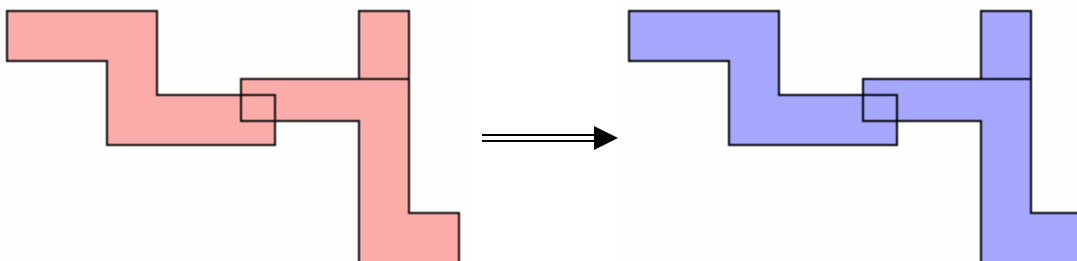
#### 統合レイヤーの例



### 非統合レイヤー (Unmerged Layers)

非統合レイヤーは DllCreateUnmerged コマンドを使用して生成されます。このコマンドを使用すると、それぞれの形状は維持されたままの状態でも Derived レイヤーが生成されます。また重なったポリゴンは非統合レイヤー内でも重なった状態のままとなります。詳細につきましては 12 ページ 非統合 Derived レイヤーを生成する」の項をご参照下さい。

#### 非統合レイヤーの例



# General Concepts

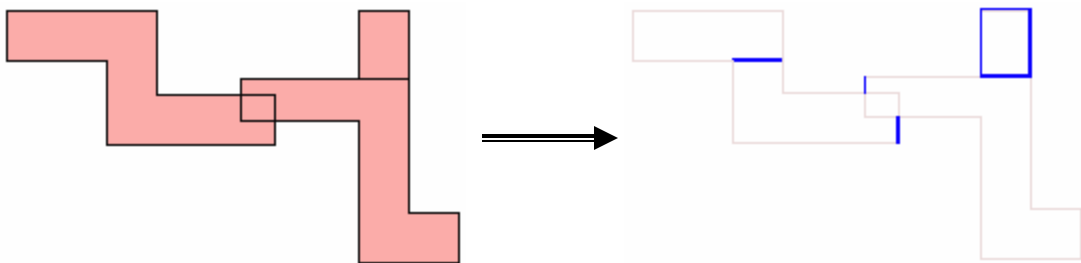
### Segment Derived Layers

segment derived layer には開ポリゴン、エッジまたは断片が含まれます。この derived レイヤーは以下のいずれかの操作で生成されます：

- ・ 既存の derived レイヤーからエッジまたは断片を選択し、DlaSelEdge または DlaSelSeg コマンドを使用する  
( 69 ページ 「関連位置によるエッジの選択」の項、76 ページ 「特定のプロパティによるエッジの選択」、及び 87 ページ 「関連位置による断片の選択」の項参照 )
- ・ segment レイヤーを返す XDRC コマンドを使用する
- ・ 既存の derived レイヤーに断片を加える DlaCoordsToLayer コマンドを使用する  
( 13 ページ 「Derived レイヤーにデータを加える」の項参照 )

前項の shape derived layer と異なり、この segment derived layer は Physical レイヤーとしてデータベースに保存することはできません。segment レイヤーの内容を見る必要がある場合は、結果である derived レイヤーを XDRCTagLayer などのコマンドを使用して「エラータグ」としてエラーマネージャーにインストールすることにより見ることが可能です (要 XDRC オプション)。XDRC に関する詳細につきましては「Working with the XDRC」マニュアルをご参照下さい。

### segment derived レイヤーの例



### Terminology

Derived レイヤーでの実際の作業 (ブール演算、選択、リサイジングまたはネット操作) に移って頂く前に、それらの機能を定義する用語を理解して置いて頂く必要があります。次章以降で解説する機能定義に最も頻繁に使用される用語を次項に記しておりますので、是非ご参照下さい。

## General Concepts

「**shape**」とはその内側のポリゴン (またはホール) を含むポリゴンの輪郭のことを指します。「**outline polygon**」とはエッジにより形成される素材を表す輪郭 (外形) のことを指します。「**inner polygon**」とはエッジにより形成される素材でない輪郭 (外形) のことを指します。「**segment**」とはエッジ上の異なる二点間を結んでいる断片 (線) のことを指します。

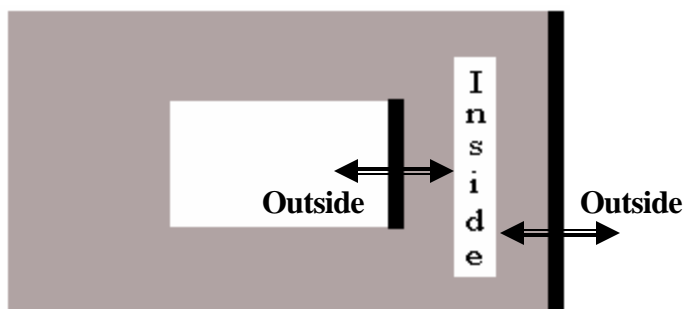
(いずれも下図参照して下さい)



「**area of shape**」とは「outline polygon」から「inner polygon」を引いた領域のことを指します (下図参照)。



それぞれのエッジや断片から見て、「area of shape」に向かっている方向をエッジの「**inside direction**」、その反対側に向かっている方向をエッジの「**outside direction**」とそれぞれ呼ぶことにします (下図参照)。



## **General Concepts**

---

# Using Common Functions

---

本章では Derived レイヤーを使用していく上で必須となる初期化、生成、保存と言った基礎コマンドや、頻繁に使用されると思われるブール演算に関するコマンドなど、Derived レイヤー操作の基本的なコマンド群を解説していきます。

## 本章での解説項目一覧：

- ・ Derived レイヤーの初期化
- ・ Derived レイヤーの生成
  - Derived レイヤーヘデータを加える
- ・ Derived レイヤーを保存する
  - Derived レイヤーにデータが含まれているかを決定する
  - Derived レイヤーの総面積を計算する
- ・ Derived レイヤーをリサイズする
- ・ 非統合 Derived レイヤーを統合する
- ・ ブール演算
  - 論理積演算 (AND )
  - 論理和演算 (OR )
  - 排他的論理和演算 (XOR )
  - 論理差演算 (SUB )
  - 否定演算 (NOT )
  - 非統合 Derived レイヤーへの論理積演算 (AND )



## Derived レイヤーの初期化

---

### Derived レイヤーの初期化

Derived レイヤーを生成する際、まず行って頂くのが Derived レイヤー状態の初期化です。これは初めて Derived レイヤーを使用する際にも必須の作業となります。初期化には抽出されるデータが含まれているストラクチャ名、範囲及び階層レベルの各情報を指定して実行します。

XDRC コマンド用に Derived レイヤーを生成する際は本項で解説する Dlalnit コマンドの代わりに XDRCInit というコマンドを用いますのでご注意ください。詳細につきましては XDRC マニュアルをご参照下さい。

### コマンド形式

Derived レイヤーの初期化には以下のコマンド形式を使用します：

```
Dlalnit "StructureName" [;Extent][;Level]
```

### コマンドパラメータ

Dlalnit	derived レイヤーを初期化するコマンドです。
StructureName	抽出されるデータが含まれているストラクチャの名称です。
Extent	抽出されるデータが含まれる領域です。X,Y 座標で指定します。null (デフォルト)に指定した場合、ストラクチャ全体が指定領域となります。

正しい領域指定の為に、編集中のストラクチャで Derived レイヤーを生成する際には必ず、Dlalnit コマンド実行前にストラクチャを保存して下さい。

Level	0またはnull (デフォルト)で全階層から抽出、1でトップレベルからのみ抽出となります。
-------	---

# Derived レイヤーの生成

---

## Derived レイヤーの生成

前項の初期化でデータがどのストラクチャのどの領域から抽出されるかを定義したら、次にどの素材 (エレメント属性、またはレイヤー及びデータタイプ) を抽出するのかを決定して頂きます。

抽出素材の設定はエレメント属性 (Criterion) またはレイヤー番号 データタイプ番号で行います。

エレメント属性の指定で抽出する場合は、そのエレメント属性で定義されている全てのレイヤー・データタイプ番号のセットが抽出されますのでご注意ください。

### 統合 Derived レイヤーを生成する

本項では統合 derived レイヤーの生成方法を解説致します。統合 derived レイヤーの解説は 4 ページ「Derived レイヤーの種類」の項をご参照下さい。

### コマンド形式

統合 Derived レイヤーの生成には以下のコマンド形式を使用します：

```
dlayer := DlaCreate "Criterion"
```

または

```
dlayer := DlaCreate LayerSet; DataTypeSet
```

### コマンドパラメータ

dlayer	derived レイヤーに割り当てられる GPE 変数です。ここでは便宜上「dlayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
DlaCreate	derived レイヤーを生成するためのコマンドです。
Criterion	エレメント属性で指定する場合にはこの部分に属性名が入ります (例 : "Metal", "Poly" など)。
LayerSet	エレメント属性の代わりにレイヤー番号で抽出データを指定する場合にはこの部分にレイヤー番号が入ります。スペースまたはカンマで区切ることにより複数指定することも可能です (例 : DlaCreate 6 2)。
DataTypeSet	エレメント属性の代わりにレイヤー番号で抽出データを指定する場合にはこの部分にデータタイプ番号が入ります。LayerSet 同様、複数の番号を指定することも可能です。

# Derived レイヤーの生成

---

### 非統合 Derived レイヤーを生成する

本項では非統合 derived レイヤーの生成方法を解説致します。非統合 derived レイヤーの解説は 4 ページ「Derived レイヤーの種類」の項をご参照下さい。

### コマンド形式

非統合 Derived レイヤーの生成には以下のコマンド形式を使用します：

```
dlayer := DlaCreateUnmerged "Criterion"
```

または

```
dlayer := DlaCreateUnmerged LayerSet;DataTypeSet
```

### コマンドパラメータ

dlayer	derived レイヤーに割り当てられる GPE 変数です。ここでは便宜上「dlayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
DlaCreateUnmerged	非統合 derived レイヤーを生成するためのコマンドです。
Criterion	エレメント属性で指定する場合にはこの部分に属性名が入ります (例 : "Metal"、"Poly" など)。
LayerSet	エレメント属性の代わりにレイヤー番号で抽出データを指定する場合にはこの部分にレイヤー番号が入ります。スペースまたはカンマで区切ることにより複数指定することも可能です。
DataTypeSet	エレメント属性の代わりにレイヤー番号で抽出データを指定する場合にはこの部分にデータタイプ番号が入ります。LayerSet 同様、複数の番号を指定することも可能です。

## Derived レイヤーにデータを加える

---

### Derived レイヤーにデータを加える

既存の derived レイヤーや座標指定で生成した新規 derived レイヤーに segment や shape と言ったデータを加えたい場合には以下のコマンド形式を使用します：

```
ResLayer := SlaCoordsToLayer dLayer; Coords
```

#### コマンドパラメータ

dLaCoordsToLayer	derived レイヤーにデータを加えるためのコマンドです。
dLayer	既に GPE 変数化されている derived レイヤーの変数名です。ここでは便宜上 dLayer という名称を使用しておりますが、実際にはユーザーが指定した任意の文字列となります。 また、既存の derived レイヤーで無く、新規の derived レイヤーに加える場合にはこの部分は null (“ ”)表記となります。
Coords	加えるデータの座標情報です。segment の場合は最低二点の座標、shape の場合は最低五点 (閉ポリゴンのため)の座標を指定します。

#### 返される値

ResLayer	データが追加された後の結果となる derived レイヤーの変数名です。ここでは便宜上 ResLayer という名称を使用しておりますが、実際にはユーザーが指定した任意の文字列となります。
----------	--

追加される derived レイヤーもしくは追加するデータのいずれか、もしくはどちらも segment である場合、結果となる derived レイヤー (上記例での ResLayer) は常に segment derived レイヤーとなります (segment derived レイヤーに shape データを追加、または shape derived レイヤーに segment データを追加した場合)。

## Derived レイヤーを保存する

---

### Derived レイヤーを保存する

Derived レイヤーは本項で紹介するコマンドで保存しない限り、その Derived レイヤーはレイアウトデータベース内に存在しません。つまりキャッシュ内には存在しても実際のストラクチャに反映されてはいないということです。本コマンドを使用すると、Derived レイヤーコマンドで行った様々な編集結果が指定したストラクチャに指定したエレメント属性にて物理的な形で (Physical レイヤーとして)保存、反映されます。

#### コマンド形式

Derived レイヤーを保存には以下のコマンド形式を使用します：

```
DlaSave dlayer; "Criterion"; "Structure" [; BreakMode FractMode MaxCoord]
```

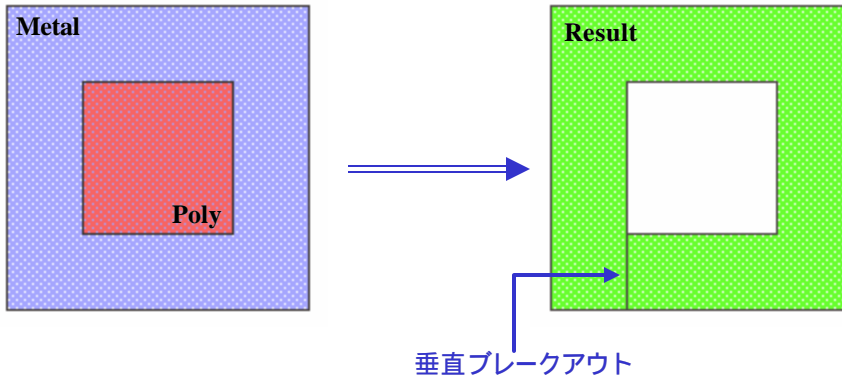
#### コマンドパラメータ

DlaSave	derived レイヤーを保存するためのコマンドです。
dlayer	保存される derived レイヤーの GPE 変数名です。ここでは便宜上「dlayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
Criterion	derived レイヤーを Physical レイヤーとして保存する際に使用されるエレメント属性です。
StructureName	derived レイヤーが保存される先のストラクチャ名です。この保存されるストラクチャは既存のものでなければいけません。
BreakMode Option	ブレイクアウトの設定です。0 または null (デフォルト)で X 軸に対して垂直なブレイクアウトを、1 で X 軸に対して水平なブレイクアウトを生成します。

ブレイクアウトとは、内部に空白を持つポリゴンが一つの閉ポリゴンとなるように、内辺と外辺を結んでいる線のことです。

## Derived レイヤーを保存する

以下はブレイクアウトの例です。本例では Metal のポリゴン (青) からその上に配置された Poly のポリゴン (赤) を抜き取った結果であるポリゴン (緑) にできた垂直ブレイクアウト(前項における BreakMode = 0 または null )を紹介しています。



### FractMode Option

Derived レイヤーが次で解説する MaxCoord オプションで指定した最大頂点角数 (MaxCoord オプションが null の場合は 4096 角) を超えてしまう場合、プログラムは自動的にその結果となるポリゴンを最大頂点角数を超えないように複数に分割します。その際に縦に分割するか横に分割するかを決定するのがこの FractMode Option です。

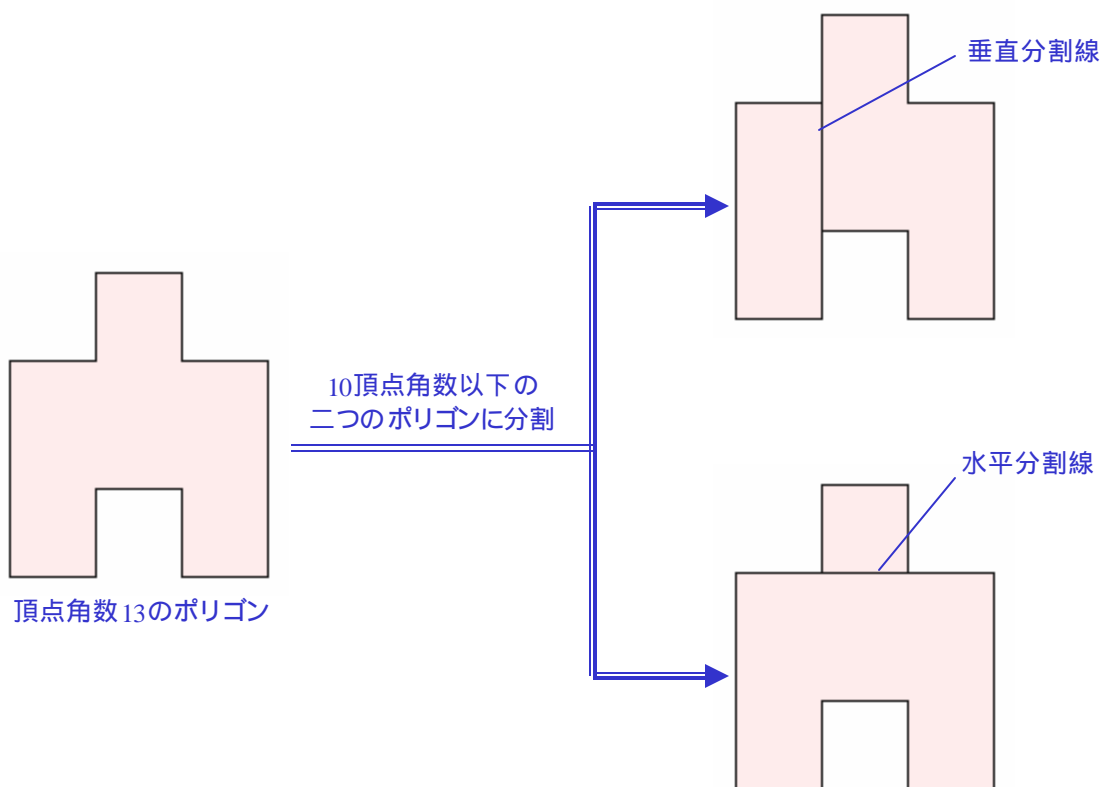
0 で X 軸に対する垂直線にて分割 (横)、1 または null (デフォルト) で X 軸に対する水平線にて分割 (縦) します。詳しくは次項の図をご参照下さい。

### MaxCoord Option

Derived レイヤーの最大頂点角数を 5 ~ 4096 までの数値で指定します。null (デフォルト) の場合この数値は 4096 に設定されます。詳しくは次項の図をご参照下さい。

## Derived レイヤーを保存する

以下はフラクチャライン (分割線) の例です。本例では左の 13 頂点角を持つポリゴン (12 角形ですが、閉ポリゴンのため頂点角数は +1 となります) を MaxCoord (最大頂点角数) = 10 として derived レイヤーを生成した際に、垂直・水平それぞれの分割線にて設定した最大頂点角数定義を守るように二つのポリゴンに分割された例を紹介しています。



## Derived レイヤーを保存する

---

### Derived レイヤーにデータが含まれているかを決定する

生成した derived レイヤーにデータが含まれているかどうかを調べるには以下のコマンドを使用します：

```
var := DlalsEmpty dLayer
```

#### コマンドパラメータ

DlalsEmpty	derived レイヤーにデータが含まれているかどうかを調べるためのコマンドです。
dLayer	調べる対象となる derived レイヤーに割り当てられている GPE 変数です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。

#### 返される値

derived レイヤーにデータが含まれている場合には 0 を、何もデータが含まれていない場合には 1 がそれぞれ返されます。

### Derived レイヤーの総面積を計算する

生成した derived レイヤーの総面積を調べるには以下のコマンドを使用します：

```
var := DlaGlobalArea dLayer [; Extent]
```

#### コマンドパラメータ

DlaGlobalArea	derived レイヤーの総面積を調べるためのコマンドです。
dLayer	調べる対象となる derived レイヤーに割り当てられている GPE 変数です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
Extent	0 または null (デフォルト) で対象となる derived レイヤー全体の領域について、頂点角座標のセットを入力することでその座標値範囲内の領域について、それぞれ面積計算します。

#### 返される値

面積の値が数値で返されます。



## Derived レイヤーをリサイズする

### Derived レイヤーをリサイズする

Derived レイヤーの形状 (shape) をリサイズします。ここで言うリサイズとは、設定した値だけ全ての辺をそのままの角度で外側・内側 (マイナス値を設定した場合) に伸ばします。拡大縮小と異なり設定した値分そのまま座標値を伸ばしますのでご注意ください。

#### コマンド形式

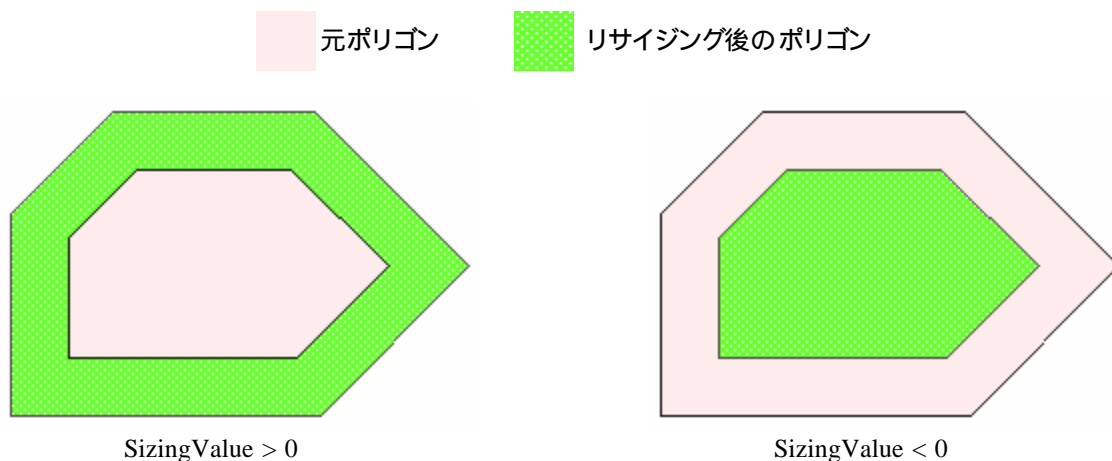
Derived レイヤーの形状をリサイズするには以下のコマンド形式を使用します：

```
dLayer := DlaResize dLayer1; SizingValue [; CornerType MaxSegs]
```

#### コマンドパラメータ

dLayer	リサイズ後の結果となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
DlaResize	derived レイヤーの形状をリサイズするためのコマンドです。
dLayer1	リサイズ前の元となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer1」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
SizingValue	リサイズする値です。外側に伸ばす場合には正の値を、内側に伸ばす場合には負の値を入力します。

#### リサイジングの例



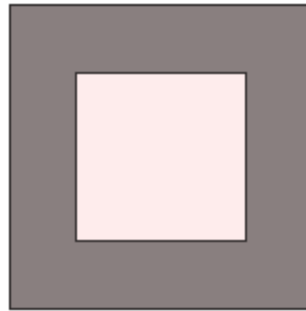
## Derived レイヤーをリサイズする

---

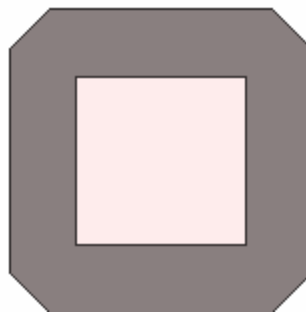
CornerType Option      リサイズした際に頂点角の形状を選択できます。0 または null (デフォルト)でそのままの形状 (角度)を維持、1 で 45 度 (mitered)、2 でラウンドにそれぞれ変形します。但し負の値でのリサイズの場合は 45 度、ラウンド処理とも実行されませんのでご注意ください。

### コーナータイプ選択例

 元ポリゴン       リサイズ後のポリゴン



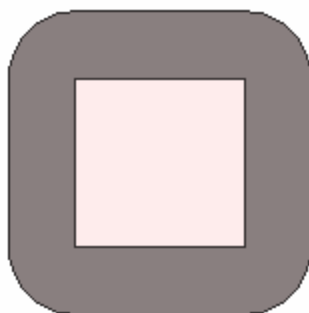
CornerType Option = 0



CornerType Option = 1

## Derived レイヤーをリサイズする

---



CornerType Option = 2

MaxSegs Option

リサイズした際に頂点角の形状をラウンド (CornerType Option = 2) に設定した際に、そのラウンドの 360 度あたりの頂点角数を 4 ~ 4095 の間で指定します。指定なし (null) でデフォルト値である 64 となります。

## 非統合Derived レイヤーを統合する

---

### 非統合 Derived レイヤーを統合する

DlaMerge というコメントを使用することにより非統合 Derived レイヤーを統合 Derived レイヤーに変換することができます。

#### コマンド形式

非統合 Derived レイヤーを統合するには以下のコマンド形式を使用します：

```
dLayer := DlaMerge dLayer1
```

#### コマンドパラメータ

dLayer	統合後の結果となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
DlaMerge	非統合 derived レイヤーを統合するためのコマンドです。
dLayer1	統合前の元となる非統合 derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer1」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。

# ブール演算

## ブール演算

指定したエレメント属性または既存の Drived レイヤーに対して様々な種類のブール演算を行い Derived レイヤーを生成します。dw-2000 では以下の五種類のブール演算を使用することができます :

- ・ 論理積演算 (AND)
- ・ 論理和演算 (OR)
- ・ 排他的論理和演算 (XOR)
- ・ 論理差演算 (SUB)
- ・ 否定演算 (NOT)

## コマンド形式

Derived レイヤーのブール演算には以下のコマンドを使用します。

```
dLayer := Layer1 <BOOLEAN OPERATOR> Layer2
```

または

```
dLayer := <DiaBOOLEAN OPERATOR> Layer1;Layer2;...
```

## コマンドパラメータ

dLayer 演算後の結果となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer」と言う名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。

<BOOLEAN OPERATOR>及び<DiaBOOLEAN OPERATOR>

ブール演算のコマンドです。この<>内に入る文字列は演算内容によりそれぞれ異なります。詳しくは 24 ページ以降をご参照下さい。

Layer1 Layer2 演算対象となるエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。

各演算別の具体的なコマンドの解説及び具体例の紹介は 24 ページ以降で行っております。必ずあわせてご参照下さい。

## ブール演算

---

基本的なブール演算演習用のサンプルライブラリー及び GPE スクリプトを以下のフォルダにご用意致しております：

¥samples¥dla\_examples¥boolean

各スクリプトは 'boolean.dwk' ライブラリーに用意されたストラクチャ、エレメント属性用に記述されており、それぞれ読み込むと Structure メニュー下に追加されたメニューコマンドから実行することができます。

前項で解説した < DlaBoolean OPERATOR > の項目をユーザーサイドで様々なパターンに書き換えて繰り返し演習を行ってみてください。

## ブール演算

## 論理積演算 (AND)

Derived レイヤーの論理積演算 (AND) には以下のコマンドを使用します。

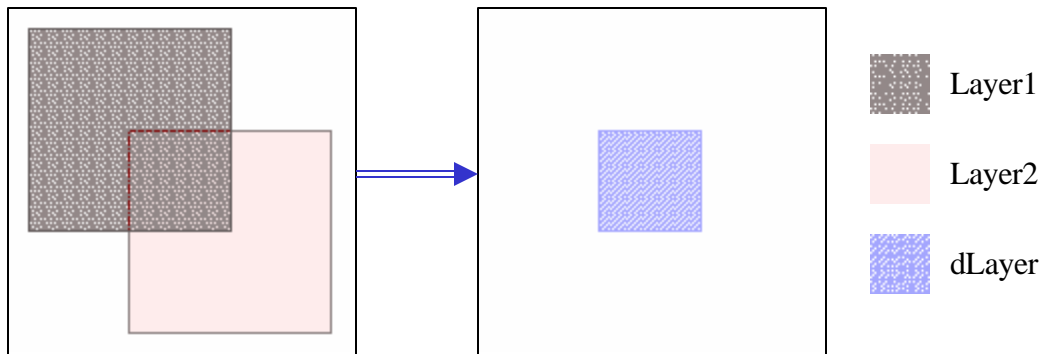
```
dLayer := Layer1 AND Layer2
```

または

```
dLayer := DiaAND Layer1;Layer2; ...
```

本演算に使用できるのは統合レイヤーのみです。非統合レイヤーや segment レイヤーはご使用できませんのでご注意ください。

## AND演算例



一番目の入力レイヤー (上記例で言うとLayer1) が接続 derived レイヤーの場合、ネット番号は結果となるレイヤー内で使用される形状 (shape) に関連して保持されることとなります。詳しくは 99 ページ「ネット機能を使用する」の項をご参照下さい。

## ブール演算

---

### 論理和演算 (OR)

Derived レイヤーの論理和演算 (OR) には以下のコマンドを使用します。

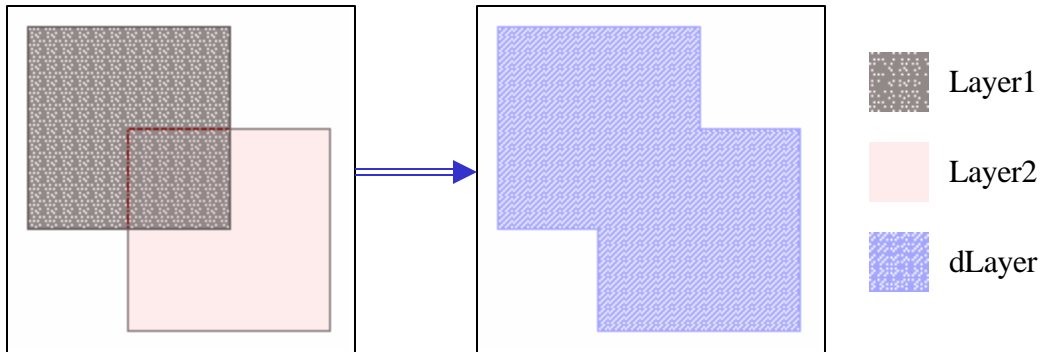
```
dLayer := Layer1 OR Layer2
```

または

```
dLayer := D1aOR Layer1;Layer2; ...
```

本演算に使用できるのは統合レイヤーのみです。非統合レイヤーや segment レイヤーはご使用できませんのでご注意ください。

### OR演算例





## ブール演算

### 排他的論理和演算 (XOR)

Derived レイヤーの排他的論理和演算 (XOR) には以下のコマンドを使用します。

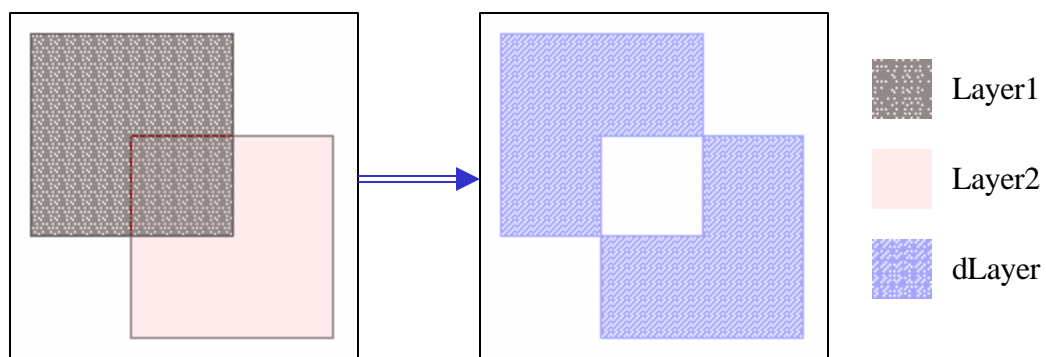
```
dLayer := Layer1 XOR Layer2
```

または

```
dLayer := DlxXOR Layer1;Layer2;...
```

本演算に使用できるのは統合レイヤーのみです。非統合レイヤーや segment レイヤーはご使用できませんのでご注意ください。

### XOR演算例



## ブール演算

### 論理差演算 (SUB)

Derived レイヤーの論理差演算 (SUB)には以下のコマンドを使用します。

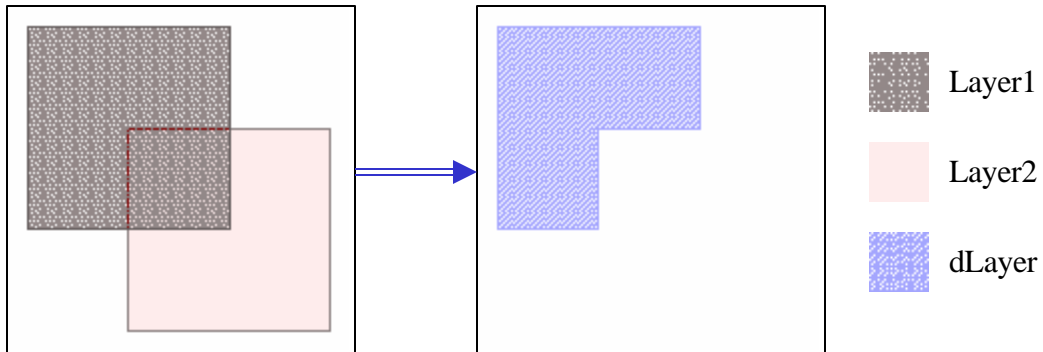
```
dLayer := Layer1 - Layer2
```

または

```
dLayer := DlsUB Layer1;Layer2;...
```

本演算に使用できるのは統合レイヤーのみです。非統合レイヤーや segment レイヤーはご使用できませんのでご注意ください。

### SUB演算例



一番目の入力レイヤー (上記例で言うとLayer1)が接続 derived レイヤーの場合、ネット番号は結果となるレイヤー内で使用される形状 (shape)に関連して保持されることとなります。詳しくは 99 ページ「ネット機能を使用する」の項をご参照下さい。

## ブール演算

### 否定演算 (NOT)

Derived レイヤーの否定演算 (NOT)には以下のコマンドを使用します。

```
dLayer := NOT Layer1 [;Extent]
```

または

```
dLayer := DlnOT Layer1 [;Extent]
```

本演算に使用できるのは統合レイヤーのみです。非統合レイヤーはご使用できませんのでご注意下さい。

### Extent Option

Extent

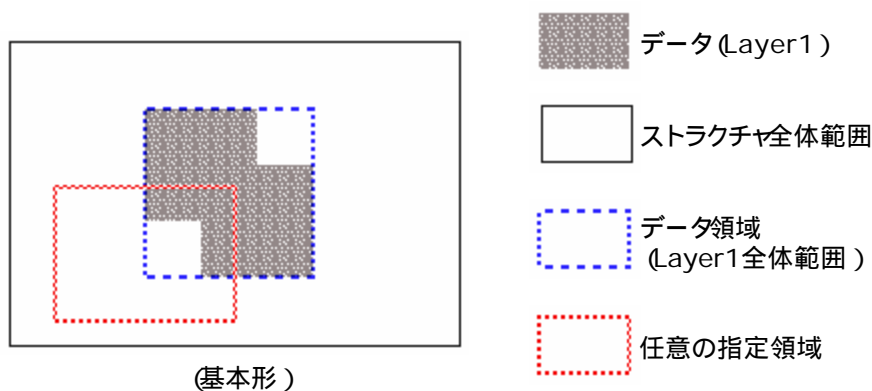
0 または null (デフォルト)でストラクチャ全体範囲に対して否定演算を行います。

1 で Layer1 全体範囲に対してのみ否定演算を行います。

また任意の X,Y 座標値セットによる領域指定も行うことが可能です。この場合は指定した領域内のみに対して否定演算を行います。

( 以下の否定演算例参照 )

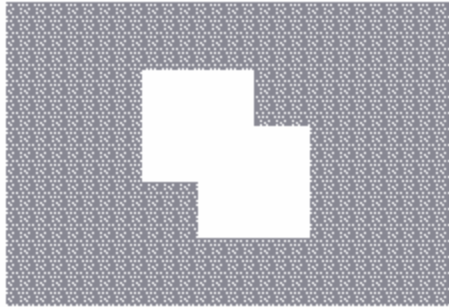
### NOT演算例



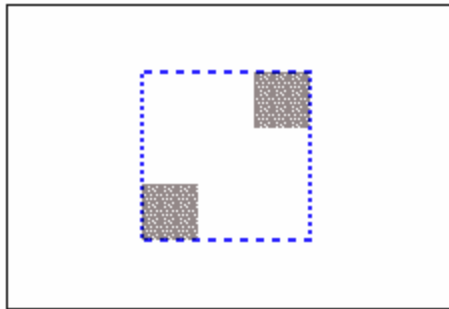
本例では上記基本形で Extent オプション 0、1 及び赤の点線で描かれた任意領域に対して否定演算をかけた場合の 3 パターンの演算結果を次項にてそれぞれ紹介しています。

## ブール演算

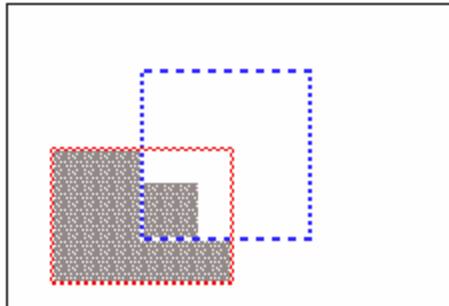
---



**Extent Option = 0**  
ストラクチャ全体範囲に対して否定演算



**Extent Option = 1**  
データ領域 (Layer1全体範囲) に対して否定演算



**Extent Option = 任意の座標値セット**  
ユーザー指定任意領域に対して否定演算

## ブール演算

### 非統合 Derived レイヤーへの論理積演算 (AND)

非統合 Derived レイヤーに含まれている各ポリゴン間で論理積演算 (AND) を行います。この作業には以下のコマンドを使用します。

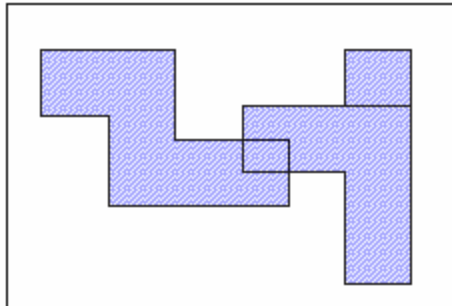
```
dLayer := DiaSingleLayerAnd Layer1 [;MergeFlag]
```

### MergeFlag Option

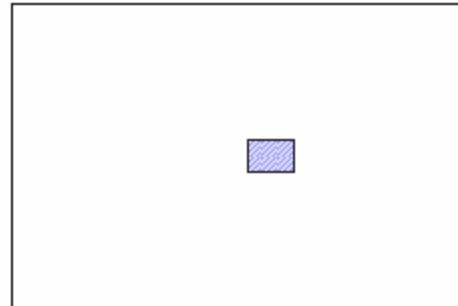
MergeFlag 0 または null (デフォルト) で結果となる dLayer が非統合 derived レイヤーに、1 で結果となる dLayer が統合 derived レイヤーにそれぞれなります。

本演算に使用できるのは非統合レイヤーのみです。統合レイヤーを使用した場合、空の (消滅した) derived レイヤーが返されます。

### 非統合 Derived レイヤーへの AND 演算例



三つのポリゴンからなる  
非統合 Derived レイヤー「Layer1」



三つのポリゴンが相互に  
重なっている部分のみが返された  
Derived レイヤー「dLayer」

# Using Selection Operations

---

**選択演算機能を使用するにはオプションの XDRC モジュールまたは HLVS モジュールが必要となります。**

本章では derived レイヤーから形状 (shape)、エッジまたは断片 (segment) を選択し、新しい derived レイヤーを生成する選択演算について解説していきます。

選択演算の種類は基本的に以下の三種類に分類されます：

- ・ 形状 (shape) 選択演算  
他の derived レイヤーとの位置関係指定による演算と、特定のプロパティ(エッジやホールの数、外周の長さなど)により選択された演算の二種類があります。
- ・ エッジ選択演算  
他の derived レイヤーとの位置関係指定による演算と、特定のプロパティ(アングルの角度、長さなど)により選択された演算の二種類があります。
- ・ 断片 (segment) 選択演算  
断片選択演算は他の derived レイヤーとの位置関係指定による演算のみとなります。

本章での解説項目一覧：

- ・ 関連位置による形状 (shape) 選択演算
- ・ 特定のプロパティによる形状 (shape) 選択演算
- ・ 関連位置によるエッジ選択演算
- ・ 特定のプロパティによるエッジ選択演算
- ・ 関連位置による断片 (segment) 選択演算

## 関連位置による形状 (shape) 選択演算

### 関連位置による形状 (shape) 選択演算

dw-2000 でサポートしている、derived レイヤーからの形状 (shape) 選択に使用される関連位置オプションは以下の通りです：

- Avoid
- Butting
- Butting Only
- Butting or Overlap
- Coincident
- Coincident Include
- Coincident Inside
- Include
- Inside
- Inside Only
- Outside
- Overlap
- Straddle
- Touching

コマンド入力では上記それぞれのオプションに対応したコマンドスイッチを選択演算子である「DlaSelShape」コマンドの後に入力します：

```
dLayer := DlaSelShape dLayer1;dLayer2;" [Val1 CO1]SEL_OP[CO2 Val2] [INVERT]"
```

#### コマンドパラメータ

**dLayer** 演算後の結果となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。

**DlaSelShape** 選択演算のメインコマンドです。

**dLayer1** 形状 (shape) 選択されるエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。

**dLayer1** が接続 (connected) derived layer である場合、そのネット番号は結果となるレイヤー (dLayer) で使用される形状 (shape) にそのまま受け継がれます。詳しくは 99 ページ「ネット機能を使用する」の項をご参照下さい。

**dLayer2** dLayer1 の抽出形状 (shape) の位置に関連して使用されるエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。

## 関連位置による形状 (shape) 選択演算

Val1 及び Val2           抽出する形状 (shape) の数量を条件付けする為のパラメータです。  
CO1 及び CO2           Val1 及び Val2 で定義された数値につく等式、不等式です。つまりこのパラメータは  $>$   $\cdot >=$   $\cdot =$   $\cdot <$   $\cdot <=$  のいずれかとなります。

上記の二種類のパラメータは、演算結果が定義した数値 (数式) を満たす場合のみ結果を返す、と言う場合に使用します。例えば複数のセルに同じ選択演算を掛け、抽出形状 (shape) が 5 個以上 10 個未満ある場合のみ結果を返す、と言う場合 :

```
dLayer := DlaSelShape dLayer1; dLayer2; "5 <= SEL_OP > 10"
```

と言うコマンドとなります。またここでは形状の個数に対して条件付けしておりますが、例えば断片の長さなどに対して条件付けをすることも可能です。

ちなみにこの数値定義オプションは全ての演算に使用されるわけではございません。また使用できる演算でも必ずしも定義しなければいけないということもありません。あくまで条件が必要な時のみ使用します。

SEL\_OP                    選択演算の種類を決定するコマンドスイッチです。  
INVERT                    指定した選択演算と逆の結果を返します。

本章では前項に掲載したコマンドスイッチ全てに対して二種類 (例 1 : Obvious、例 2 : Non Obvious) の演算例を紹介しています。また紹介する全ての例の基本形となるセルは以下のフォルダに収録してあります :

例 1 : ¥samples¥dla\_examples¥sel\_shapes¥position¥obvious

例 2 : ¥ samples¥dla\_examples¥sel\_shapes¥position¥non\_obvious

それぞれの例に対するスクリプトも同フォルダに収録しておりますが、例 1 例 2 とともに『INSIDE』の選択演算のみ記述されております。その他の演算をお試しになりたい場合には、スクリプト内の『Inside Input』及び『INSIDE』の部分进行他のストラクチャ及びコマンドスイッチに書き換えた上で、File:Save and Load... コマンドを選択して再読み込みし実行して下さい。

また実際の演算を行う前に、それぞれのサンプルライブラリで演算結果として収録されているストラクチャ『Selection Result』のバックアップを取って置いて下さい。バックアップの取り方は当該ストラクチャを開いた後、Structure:Backup コマンドを実行します。これにより一つの演算が終了し、次の演算に進む際に Structure:Revert を実行することで『Selection Result』ストラクチャをデフォルトの状態に戻すことができます。



## 関連位置による形状 (shape) 選択演算

## 関連位置による形状 (shape) 選択演算

### 形状選択演算 Avoid

Avoid は dLayer2 の形状 (shape) から完全に外部に位置する dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelShape dLayer1;dLayer2;"AVOID [INVERT]"
```

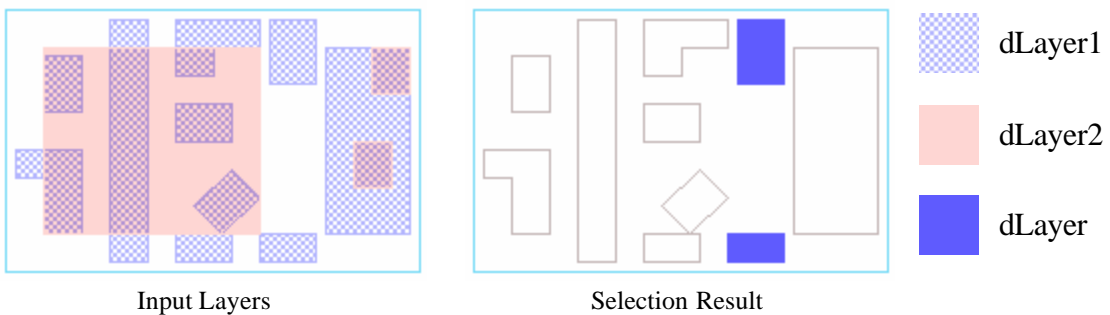
### 定義

dLayer2 から完全に孤立した形状 (shape) のみが選択されます。内部にあたり重なり合ったりまた断片だけが重なり合っている場合でも全て排除されます。但し頂点角のみが接触している場合は排除されず選択されることとなります。

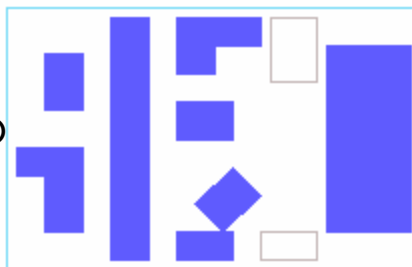
### 演算例

```
dLayer := DLaSelShape dLayer1;dLayer2;"AVOID"
```

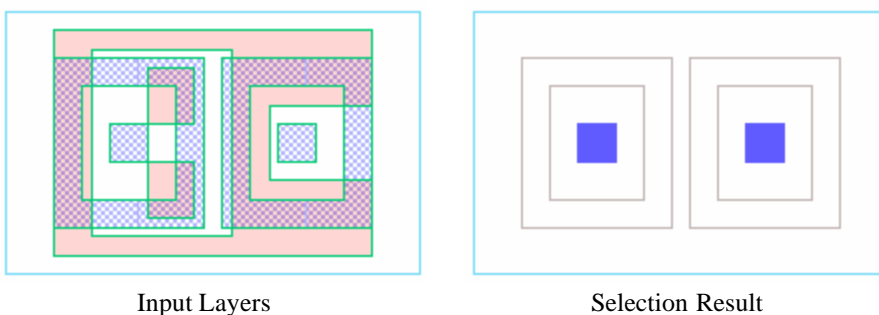
#### 例1 : Obvious



#### (例1をINVERTにした場合)



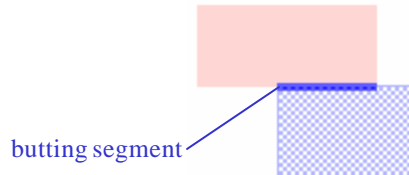
#### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 Butting

Butting は dLayer2 の形状 (shape) の外側の断片と接している断片を持つ dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。



### コマンド形式

```
dLayer := DiaSelShape dLayer1;dLayer2;"[Val1 CO1]BUTTING[CO2 Val2] [INVERT]"
```

### 定義

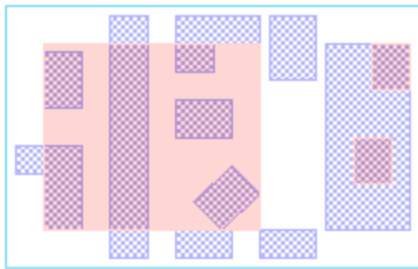
dLayer2 を形成しているポリゴンの外側の断片と dLayer2 の外部で接している断片を持つ形状 (shape) のみが選択されます。次項に記してある例 2 :Non Obvious の左側の形状のように、一見 dLayer2 の内側の断片に見えるような場合でも、ポリゴン内部の断片では無いので「外側」の断片となります ( 7 ページ参照 )。

## 関連位置による形状 (shape) 選択演算

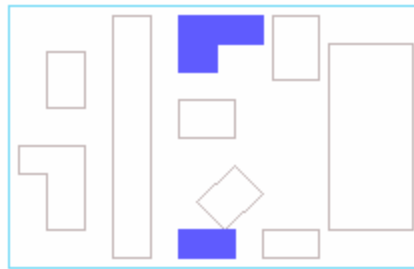
### 演算例

```
dLayer := DiaSelShape dLayer1;dLayer2;"BUTTING"
```

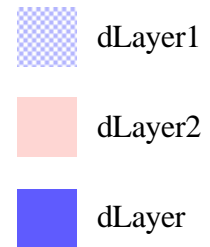
#### 例1 : Obvious



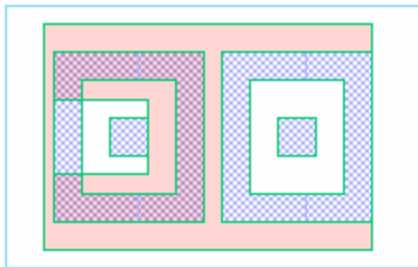
Input Layers



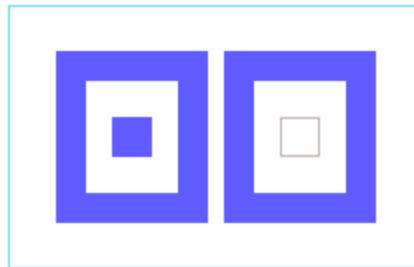
Selection Result



#### 例2 : Non Obvious



Input Layers



Selection Result

## 関連位置による形状 (shape) 選択演算

### 形状選択演算 **Butting Only**

Butonly は dLayer2 の形状 (shape) の外側の断片と接している断片を持ち、かつ dLayer2 の完全に外部に位置する dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DLaSelShape dLayer1;dLayer2;"[Val1 CO1]BUTTONONLY[CO2 Val2] [INVERT]"
```

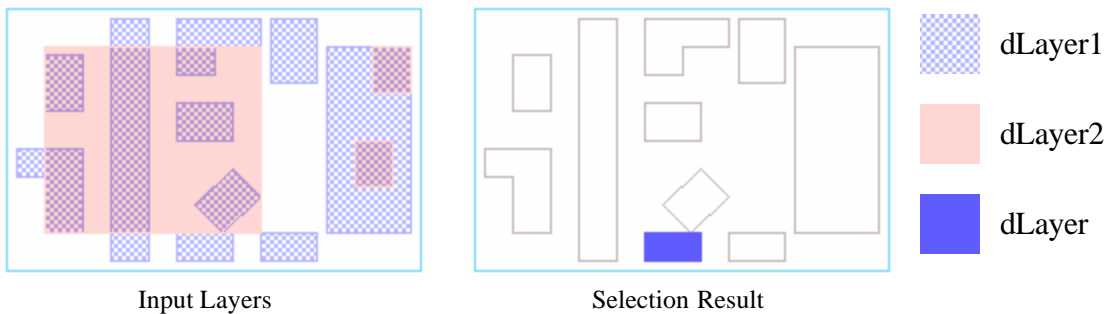
#### 定義

前項の Butting では外側の断片と接している形状 (shape) 全てが選択されましたが、この Butting Only ではそれにプラスして dLayer2 の完全に外部に位置する形状 (shape) のみが選択されます。

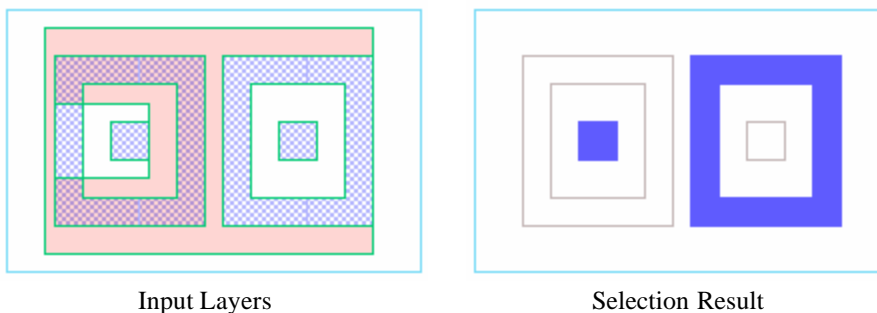
#### 演算例

```
dLayer := DLaSelShape dLayer1;dLayer2;"BUTTONONLY"
```

##### 例1 : Obvious



##### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 **Butting or Overlap**

ButtOrOver は dLayer2 の形状 (shape) の外側の断片と接している断片を持つか、dLayer2 のどこか一部でも重なっている dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := D1aSelShape dLayer1;dLayer2;"[Val1 CO1]BUTTOROVER[CO2 Val2] [INVERT]"
```

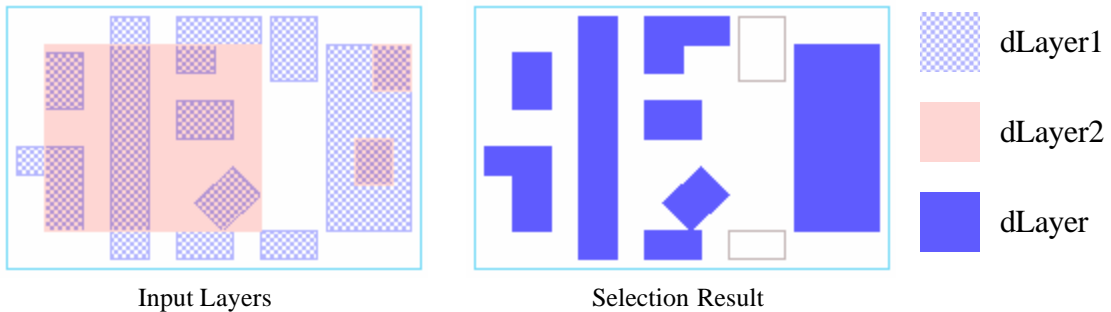
#### 定義

dLayer2 の外側の断片と接している断片を持つか、dLayer2 のどこか一部でも重なっている形状 (shape) のみが選択されます。但し、頂点角のみが接している形状については排除されます。

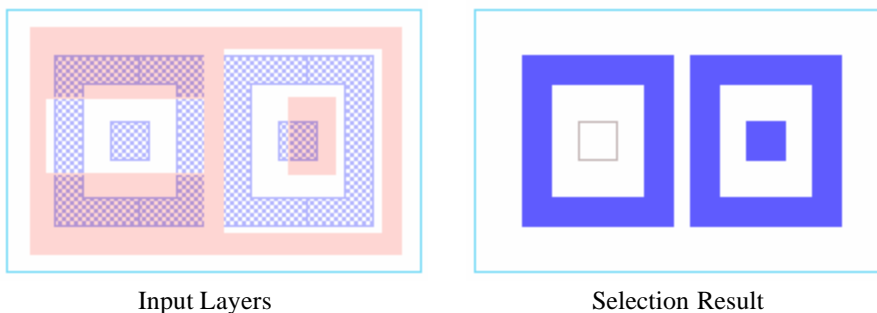
#### 演算例

```
dLayer := D1aSelShape dLayer1;dLayer2;"BUTTOROVER"
```

##### 例1 : Obvious



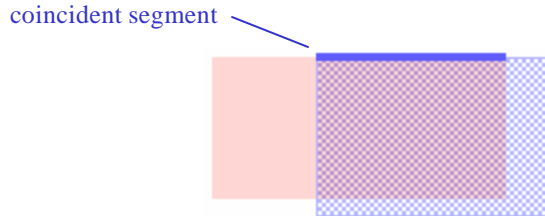
##### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 **Coincident**

Coincident は dLayer2 の形状 (shape) の断片と接している断片を持ち、かつポリゴンエリアが重なっている dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。



### コマンド形式

```
dLayer := D1aSelShape dLayer1;dLayer2;"[Val1 CO1]COINCIDENT[CO2 Val2] [INVERT]"
```

### 定義

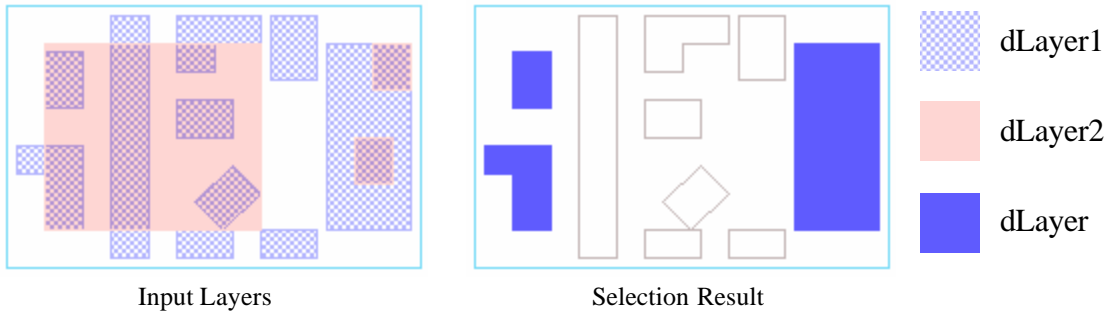
dLayer2 の形状 (shape) の断片と接している断片を持ち、かつポリゴンエリアが重なっている形状 (shape) が選択されます。つまり内側同士の断片が重なり合っている形状のみが選択されるということです。

## 関連位置による形状 (shape) 選択演算

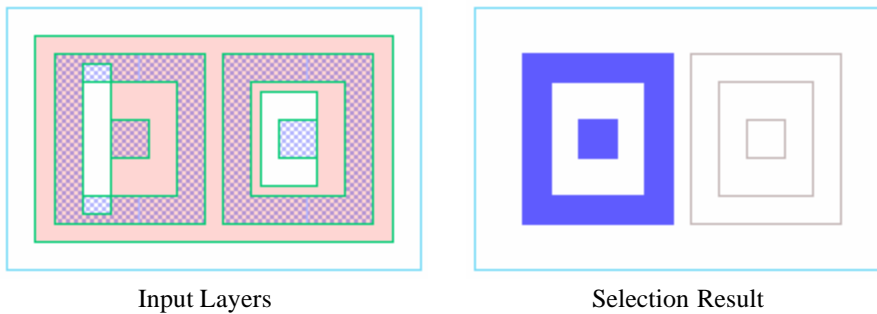
演算例

```
dLayer := DiaSelShape dLayer1;dLayer2;"COINCIDENT"
```

例1 : Obvious



例2 : Non Obvious





## 関連位置による形状 (shape) 選択演算

### 形状選択演算 **Coincident Include**

CoincInclude は dLayer2 の形状 (shape) の内側の断片と接し、かつ dLayer2 のどれかが完全に含まれる dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelShape dLayer1;dLayer2;"[Val1 CO1]COINCINCLUDE[CO2 Val2] [INVERT]"
```

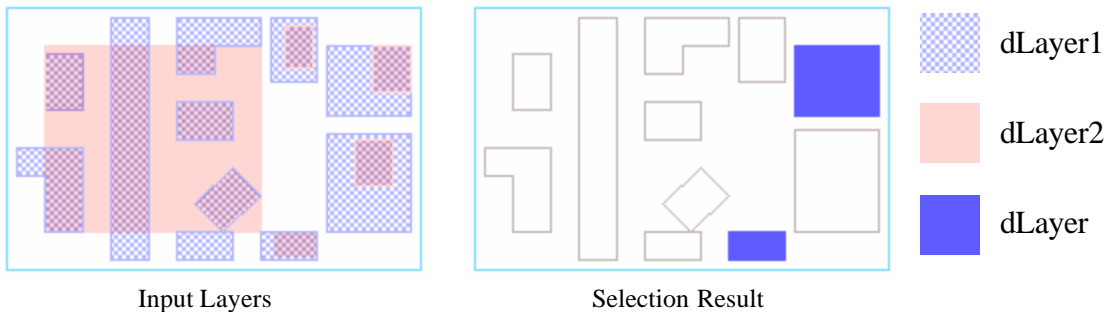
### 定義

dLayer2 の内側の断片と接し、かつ dLayer2 のどれかが完全に含まれる形状 (shape) が選択されます。但し dLayer1 の一部のポリゴンと重なっている dLayer2 の全てのポリゴンが完全に含まれている必要は無く、一つでも dLayer1 の一部のポリゴンに完全に含まれていれば選択されます。

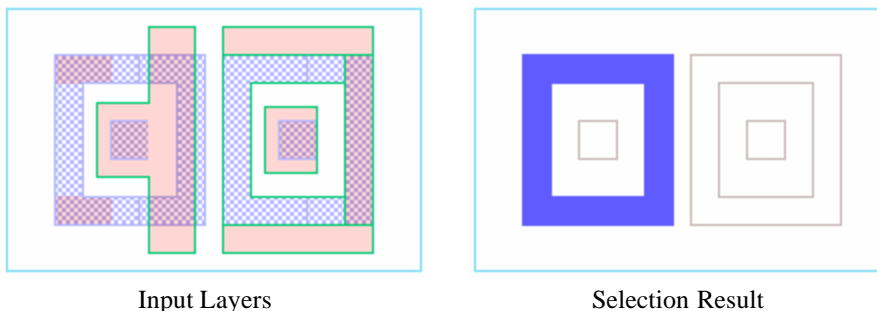
### 演算例

```
dLayer := DLaSelShape dLayer1;dLayer2;"COINCINCLUDE"
```

#### 例1 : Obvious



#### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 **Coincident Inside**

CoincInsideはdLayer2の形状 (shape) の内側の断片と接し、かつdLayer2のどれかに完全に含まれるdLayer1の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DLaSelShape dLayer1;dLayer2;"COINCINSIDE [INVERT]"
```

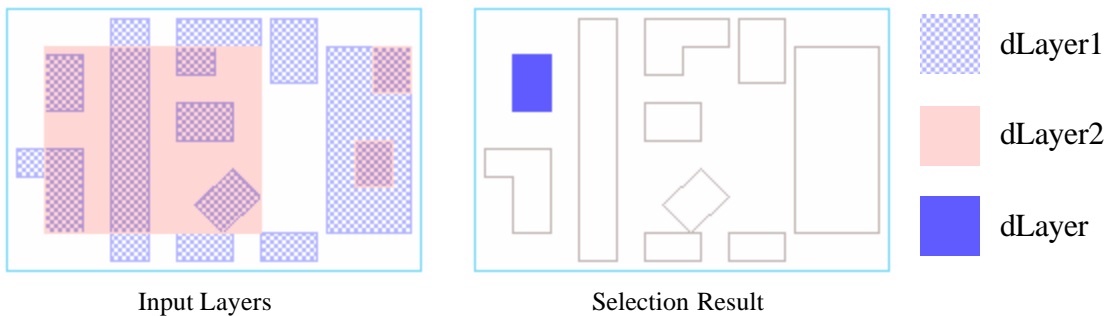
#### 定義

前項の Coincident Include とdLayer1 とdLayer2 の定義が全く逆で、dLayer2の内側の断片と接し、かつdLayer2のどれかに完全に含まれる形状 (shape) が選択されます。

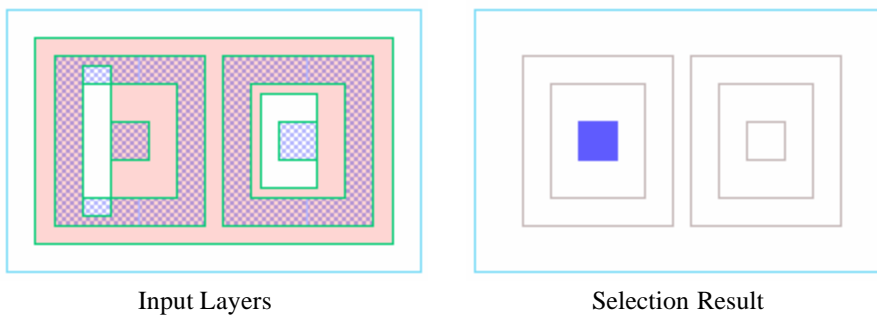
#### 演算例

```
dLayer := DLaSelShape dLayer1;dLayer2;"COINCINSIDE"
```

##### 例1 : Obvious



##### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 Include

Include は dLayer2 の形状 (shape) のどれかが完全に含まれ、かつ一部が重なっているものが無い dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DiaSelShape dLayer1;dLayer2;"[Val1 CO1]INCLUDE[CO2 Val2] [INVERT]"
```

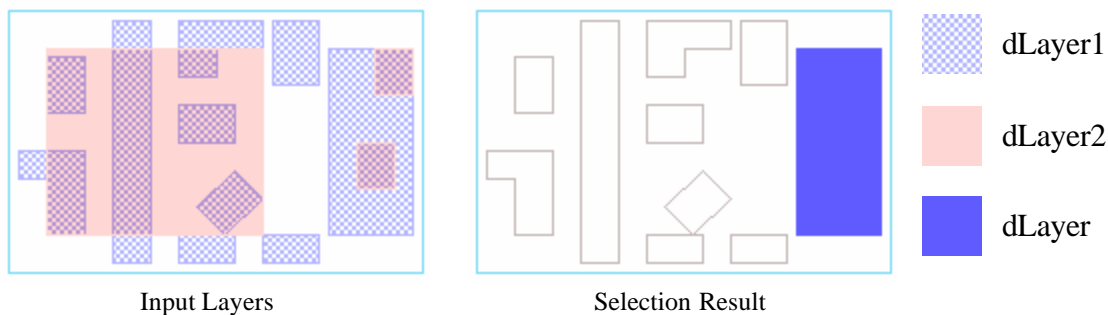
### 定義

dLayer2 のどれかが完全に含まれ、かつ一部が重なっているものが無い dLayer1 の形状 (shape) が選択されます。つまり dLayer2 のポリゴンの一つまたは複数 dLayer1 のポリゴンの一つに完全に含まれていても、それ以外に重なっているだけのポリゴンもある場合には、その dLayer1 のポリゴンは排除されてしまいます。

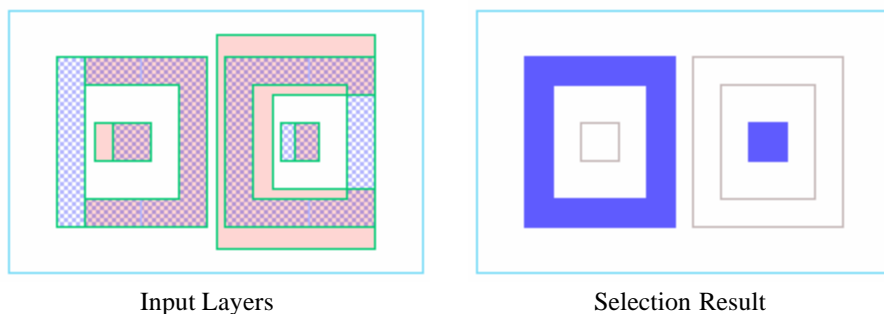
### 演算例

```
dLayer := DiaSelShape dLayer1;dLayer2;"INCLUDE"
```

#### 例1 : Obvious



#### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 Inside

Inside は dLayer2 の形状 (shape) のどれかが完全に含まれている dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelShape dLayer1;dLayer2;"INSIDE [INVERT]"
```

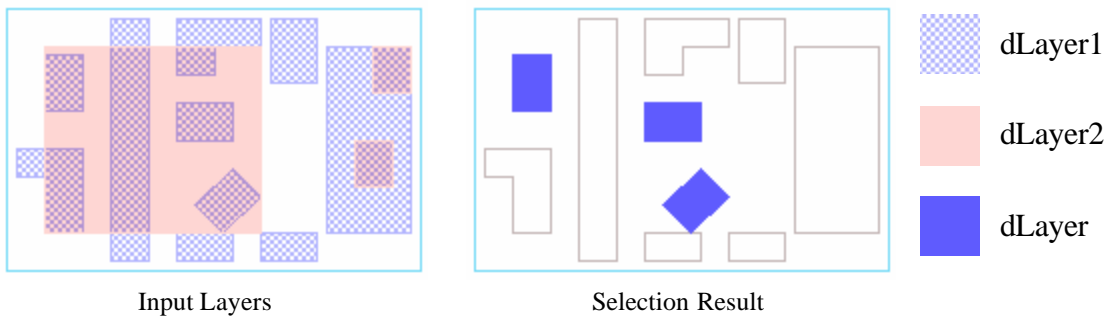
### 定義

前項の Include では他に重なっているものが無いのが条件でしたが、この Inside は dLayer2 のポリゴンのどれかが dLayer1 のポリゴンのどれかに完全に含まれてさえいれば選択されます。

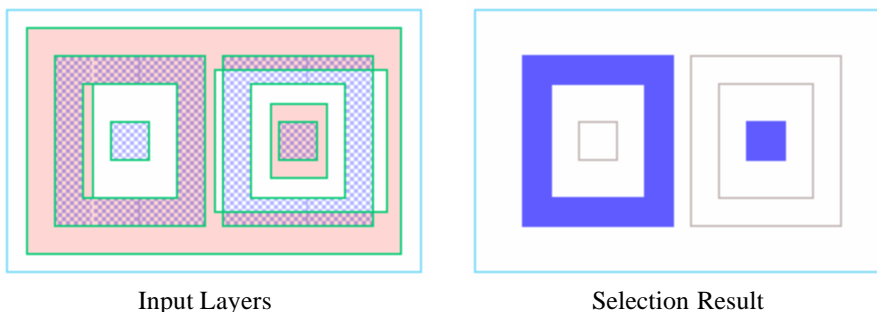
### 演算例

```
dLayer := DLaSelShape dLayer1;dLayer2;"INSIDE"
```

#### 例1 : Obvious



#### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 **Inside Only**

InsideOnly は dLayer2 の形状 (shape) のどれかが完全に含まれ、かつ断片同士が一切接していない dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DlaSelShape dLayer1;dLayer2;"INSIDEONLY [INVERT]"
```

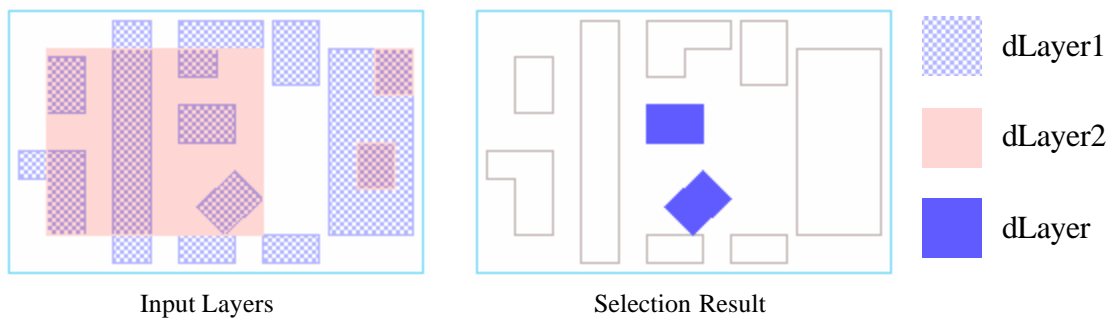
### 定義

前項の Inside から断片同士が接しているポリゴンを除いた形状 (shape) が選択されます。但し断片と頂点角が接している場合は排除されず、そのまま選択されます。

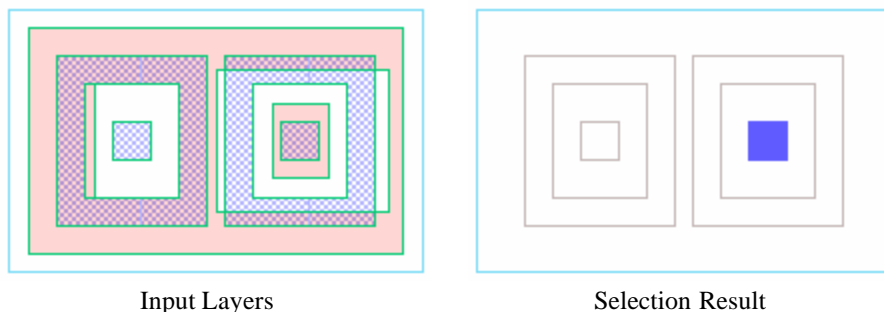
### 演算例

```
dLayer := DlaSelShape dLayer1;dLayer2;"INSIDEONLY "
```

#### 例1 : Obvious



#### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 Outside

Outside は dLayer2 の形状 (shape) のどれにも全く重なっていない dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelShape dLayer1;dLayer2;"OUTSIDE [INVERT]"
```

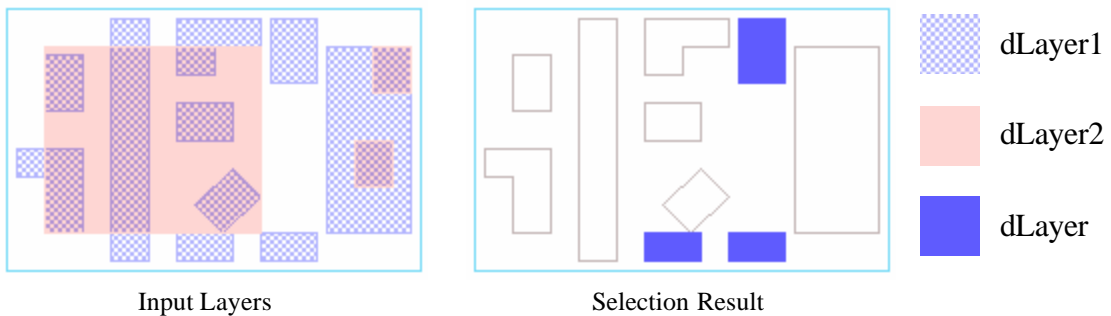
### 定義

dLayer2 のポリゴンのいずれにも全く重なっていない形状 (shape) が選択されます。断片や頂点角のみが重なっていても、ポリゴン内部が重なっていないものは全て排除されます。

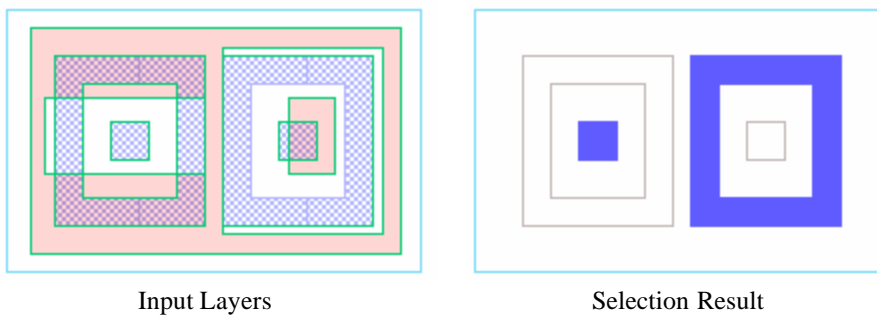
### 演算例

```
dLayer := DLaSelShape dLayer1;dLayer2;"OUTSIDE"
```

#### 例1 : Obvious



#### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 Overlap

Overlap は dLayer2 の形状 (shape) のどれかと一部でも重なっている dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelShape dLayer1;dLayer2;"[Val1 CO1]OVERLAP[CO2 Val2] [INVERT]"
```

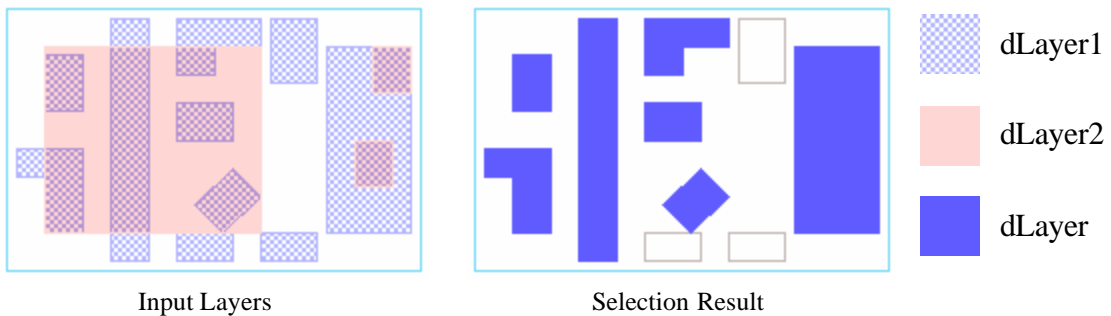
### 定義

dLayer2 のポリゴンのいずれかとほんの一部でも重なっている形状 (shape) が選択されます。但し断片や頂点角のみが重なっていても、ポリゴン内部同士が重なっていないものは全て排除されます。

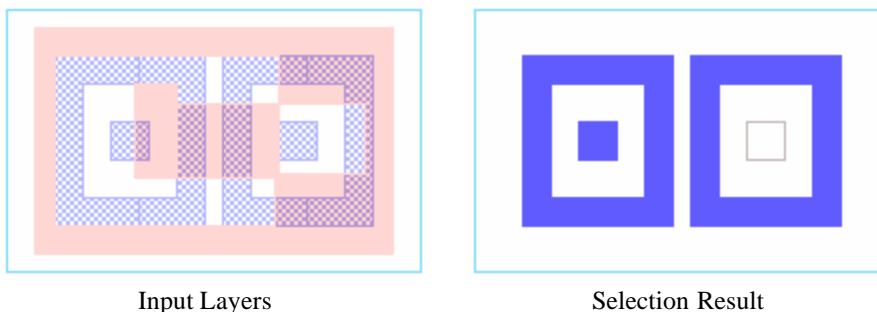
### 演算例

```
dLayer := DLaSelShape dLayer1;dLayer2;"OVERLAP"
```

#### 例1 : Obvious



#### 例2 : Non Obvious



## 関連位置による形状 (shape) 選択演算

### 形状選択演算 Straddle

Straddle は dLayer2 の形状 (shape) のどれかと一部が重なっている dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DiaSelShape dLayer1;dLayer2;"[Val1 CO1]STRADDLE[CO2 Val2] [INVERT]"
```

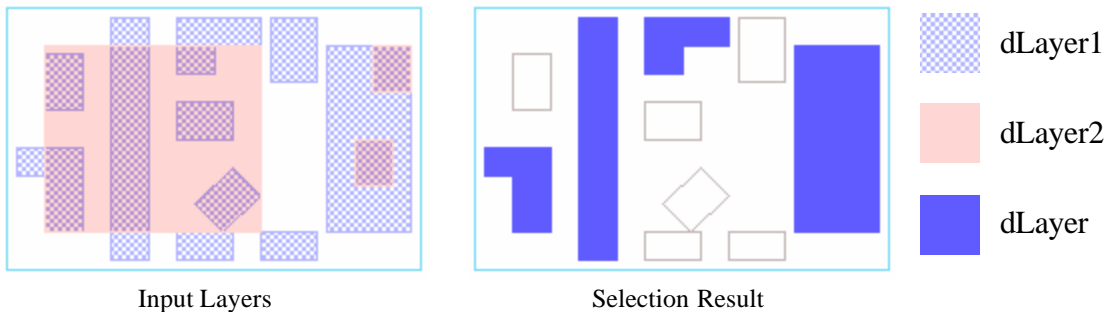
### 定義

dLayer2 のポリゴンのいずれかと一部が重なっている形状 (shape) が選択されます。つまりポリゴン全体が重なってしまっているものは全て排除されます。あくまで一部が重なっているもののみです。また断片や頂点角だけが重なっているものも同様に全て排除されます。

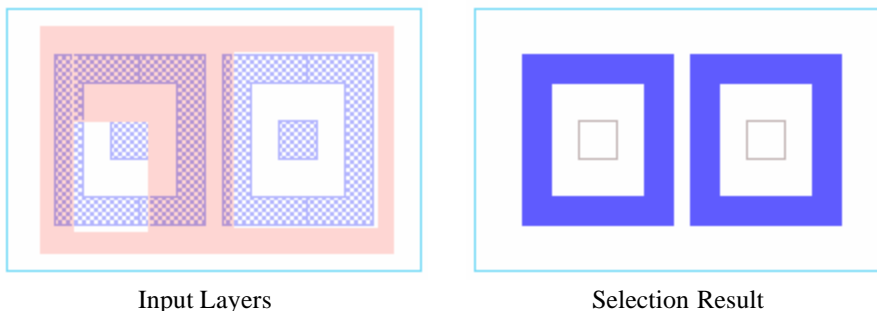
### 演算例

```
dLayer := DiaSelShape dLayer1;dLayer2;"STRADDLE"
```

#### 例1 : Obvious



#### 例2 : Non Obvious





## 関連位置による形状 (shape) 選択演算

### 形状選択演算 Touching

Touching は dLayer2 の形状 (shape) の内側、外側問わず、断片同士が接している dLayer1 の形状 (shape) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelShape dLayer1;dLayer2;"[Val1 CO1]TOUCHING[CO2 Val2] [INVERT]"
```

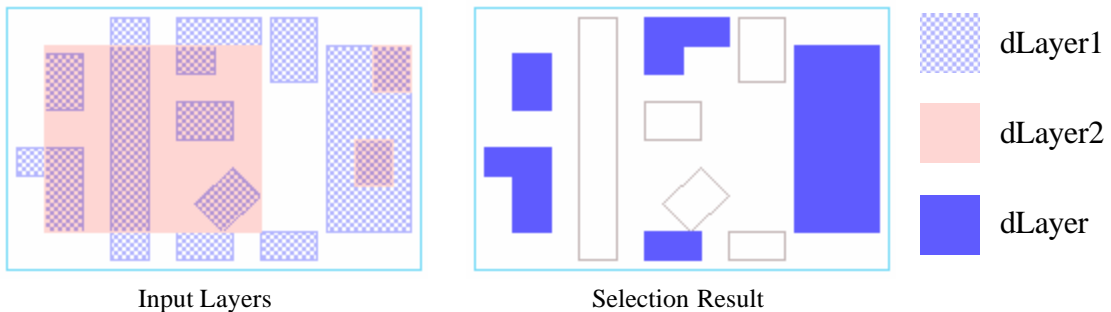
### 定義

dLayer2 のポリゴンのどこかの断片が、dLayer1 のポリゴンのどこかの断片と接している形状 (shape) が選択されます。ちょうど Butting と Coincident を合わせたもので、dLayer1、dLayer2 とともに断片の方向 (内側・外側) は問いません。

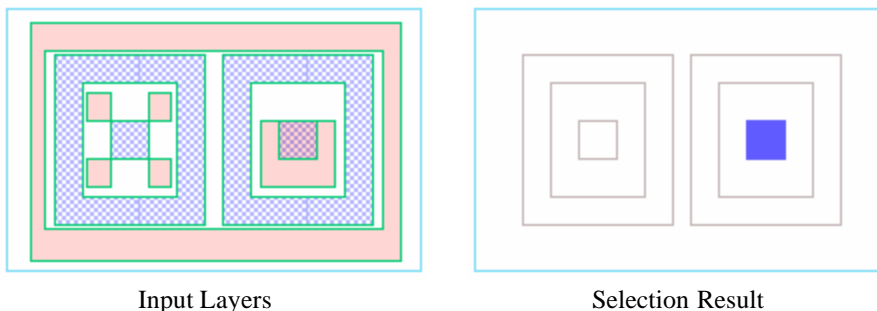
### 演算例

```
dLayer := DLaSelShape dLayer1;dLayer2;"TOUCHING"
```

#### 例1 : Obvious



#### 例2 : Non Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 特定のプロパティによる形状 (shape) 選択演算

dw-2000 でサポートしている derived レイヤーからの形状 (shape) 選択に使用されるプロパティ指定オプションは以下の通りです：

- ・ Acute Angles
- ・ 45 Degree Angles
- ・ Edge Length
- ・ Number of Edges
- ・ Number of Holes
- ・ Octagonal Angles
- ・ Orthogonal Angles
- ・ Perimeter
- ・ Rectangles Only

コマンド入力では前節の「関連位置による形状選択演算」同様、上記それぞれのオプションに対応したコマンドスイッチを選択演算子である「DlaSelShape」コマンドの後に入力します。但し前節では dLayer2 の要素と dLayer1 の要素との配置位置関係からそれぞれのオプションに該当する形状を選択していったのに対し、今回は dLayer1 の要素の特徴からのみ選択していくため演算に使用する derived レイヤーは dLayer1 の一つだけとなります：

```
dLayer := DlaSelShape dLayer1; "[Val1 CO1]SEL_OP[CO2 Val2] [ALL] [NOHOLE] [INVERT]"
```

#### コマンドパラメータ

dLayer	演算後の結果となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
DlaSelShape	選択演算のメインコマンドです。
dLayer1	形状 (shape) 選択される要素属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。

dLayer1 が接続 (connected) derived layer である場合、そのネット番号は結果となるレイヤー (dLayer) で使用される形状 (shape) にそのまま受け継がれます。詳しくは 99 ページ「ネット機能を使用する」の項をご参照下さい。

## 特定のプロパティによる形状 (shape) 選択演算

Val1 及び Val2	抽出する形状 (shape) の数量を条件付けする為のパラメータです。
CO1 及び CO2	Val1 及び Val2 で定義された数値につく等式、不等式です。つまりこのパラメータは $>$ $\cdot >=$ $\cdot =$ $\cdot <$ $\cdot <=$ のいずれかとなります。
SEL_OP	選択演算の種類を決定するコマンドスイッチです。
INVERT	指定した選択演算と逆の結果を返します。

### オプションパラメータ

ALL	このオプションパラメータを設定した場合、dLayer1 から選択された形状の全ての断片が指定した長さや角度を満たさない限り抽出を行いません。デフォルト(= ALL 設定なし)では断片の一つでも条件を満たせば抽出されることとなります。
NOHOLE	アウトラインポリゴン(ブレイクアウトを持たない)のみを指定するためのオプションパラメータです。このオプションを設定するとインナーポリゴン(ポリゴン内部に空白がありブレイクアウトを持つポリゴン)は全て排除されます(但し全てのコマンドオプションに併用できる訳はございませんのでご注意ください)。

本章では前項に掲載したコマンドスイッチ全てに対して二種類 (例 1 :Obvious、例 2 :Non Obvious) の演算例を紹介しています。また紹介する全ての例の基本形となるセルは以下のフォルダに収録してあります :

¥samples¥dla\_examples¥sel\_shapes¥property

それぞれの例に対するスクリプトも同フォルダ下の「scripts」フォルダ内に収録しております。お試しになりたい例にあわせてそれぞれ読み込んでお使いになってみて下さい (例によっては本文と同じになるように演算行を書き換えて頂く必要がありますのでご注意ください)。

また実際の演算を行う前に、それぞれのサンプルライブラリで演算結果として収録されているストラクチャ「Selection Result」のバックアップを取って置いてください。バックアップの取り方は当該ストラクチャを開いた後、Structure:Backup コマンドを実行します。これにより一つの演算が終了し、次の演算に進む際に Structure:Revert を実行することで「Selection Result」ストラクチャをデフォルトの状態に戻すことができます。

## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 **Acute Angles**

Acute は dLayer1 の形状 (shape) から鋭角を持つ形状を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DlaSelShape dLayer1; "ACUTE [ALL] [NOHOLE] [INVERT]"
```

#### 定義

文字通り dLayer1 のポリゴンの中から鋭角 ( $0 < \text{内角} < 90$  の角度を持つ頂点角)を持つポリゴンを全て選択します。

#### 演算例

```
dLayer := DlaSelShape dLayer1; "ACUTE"
```

#### 例1 : Obvious



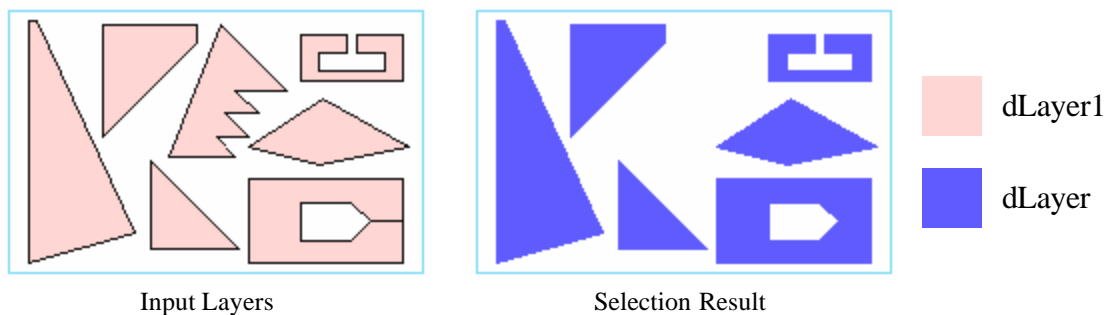
## 特定のプロパティによる形状 (shape) 選択演算

### 演算例 2

例 2 (Non Obvious) では ALL と INVERT、二つのオプションパラメータを使用することにより全ての頂点角が鋭角 (= ALL) のポリゴン以外 (= INVERT) の形状を全て選択するようにしています :

```
dLayer := DiaSelShape dLayer1;"ACUTE ALL INVERT"
```

### 例2 : Non Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 45 Degree Angles

DEG\_45 は dLayer1 の形状 (shape) から 45 度の頂点角を持つ形状を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DlaSelShape dLayer1; "DEG_45 [ALL] [NOHOLE] [INVERT]"
```

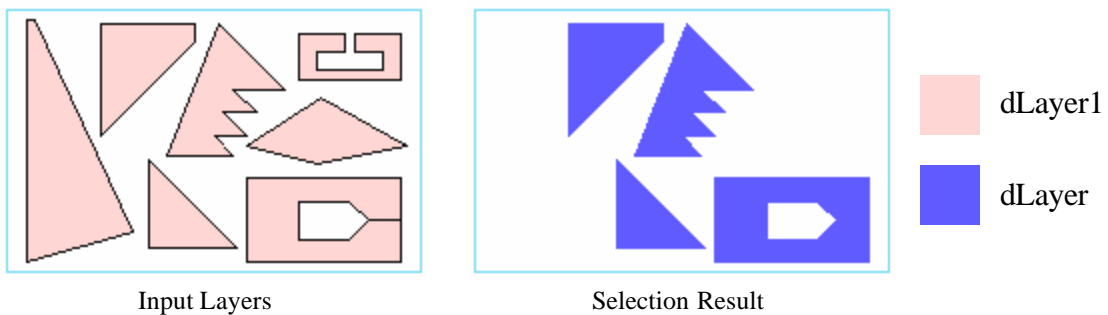
#### 定義

文字通り、dLayer1 のポリゴンの中から 45 度の頂点角を持つポリゴンを全て選択します。デフォルト (ALL 及び INVERT オプションパラメータ無し) では幾つもある角の中から一つでも 45 度の頂点角があれば選択されることとなります。

#### 演算例

```
dLayer := DlaSelShape dLayer1; "DEG_45"
```

##### 例1 : Obvious



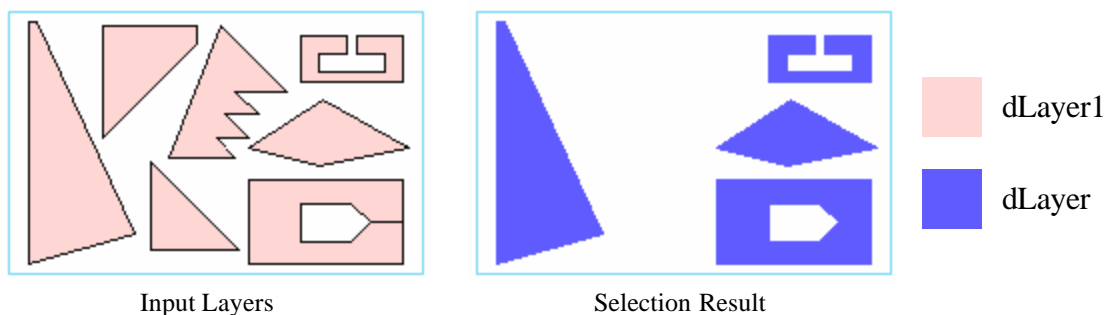
## 特定のプロパティによる形状 (shape) 選択演算

### 演算例 2

例 2 (Non Obvious) では NOHOLE と INVERT、二つのオプションパラメータを使用することにより 45 度の頂点角を持たない形状 (= INVERT) 及び 45 度の頂点角を持っているがインナーポリゴンのもの (= NOHOLE + INVERT) を全て選択するようにしています :

```
dLayer := DLaSelShape dLayer1; "DEG_45 NOHOLE INVERT"
```

### 例2 : Non Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 **Edge Length**

Length は dLayer1 の形状 (shape) から Val1 CO1、CO2 Val2 のパラメータで指定した範囲の長さの断片を持つ形状を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelShape dLayer1; "[Val1 CO1]LENGTH[CO2 Val2] [ALL] [NOHOLE] [INVERT]"
```

### 定義

dLayer1 のポリゴンの中から Val1 CO1、CO2 Val2 のパラメータで指定した範囲の長さの断片を持つポリゴンを全て選択します。Val1 CO1、CO2 Val2 はどちらか一方の指定だけでも構いません。下記演算例では 9.0 以上 9.5 未満の長さの断片を持つ形状のみ選択するよう指定しています。

### 演算例

```
dLayer := DLaSelShape dLayer1; "9.0<=LENGTH<9.5"
```

#### 例1 : Obvious





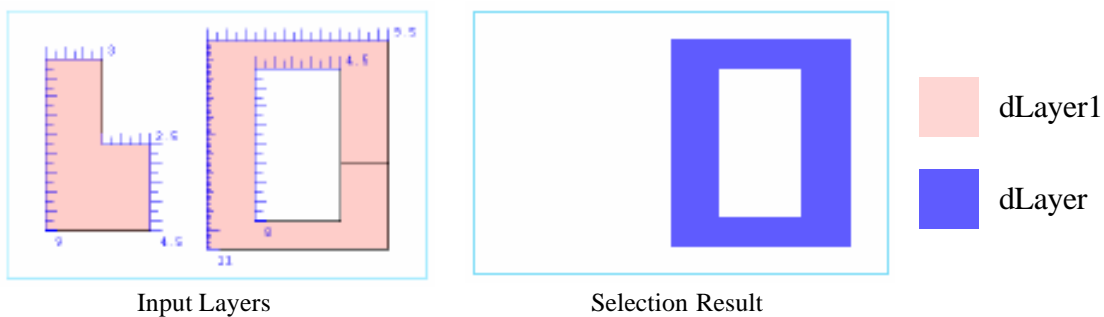
## 特定のプロパティによる形状 (shape) 選択演算

### 演算例 2

例 2 (Non Obvious) では ALL、NOHOLE、INVERT の三つのオプションパラメータを併用することにより、全ての断片の長さが 11 未満、かつブレイクアウトの無いアウトラインポリゴン上で無いポリゴンのみを全て選択するようにしています：

```
dLayer := DiaSelShape dLayer1; "11>LENGTH ALL NOHOLE INVERT"
```

### 例2 : Non Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 Number of Edges

NB\_Edges は dLayer1 の形状 (shape) から Val1 CO1、CO2 Val2 のパラメータで指定した範囲の頂点角数を持つ形状を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DlaSelShape dLayer1;"[Val1 CO1]NB_EDGES[CO2 Val2] [NOHOLE] [INVERT]"
```

### 定義

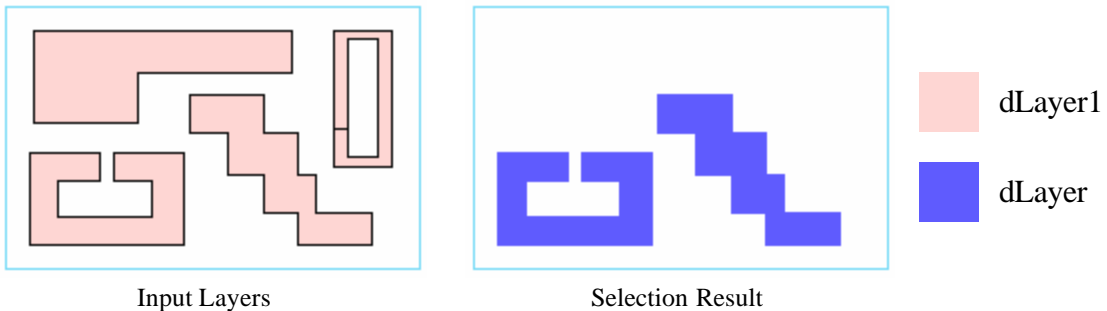
dLayer1 のポリゴンの中から指定した範囲の頂点角数を持つポリゴンを全て選択します。このコマンドスイッチは単独では使用できず、必ず Val1 CO1、CO2 Val2 の両方、またはどちらかのパラメータの併記が必須となりますのでご注意ください。

因みに下記演算例では頂点角が 8 つを超える (9 個以上) ポリゴンのみを選択するよう指定しています。

### 演算例

```
dLayer := DlaSelShape dLayer1;"8<NB_EDGES"
```

#### 例1 : Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 演算例 2

例 2 (Non Obvious) では NOHOLE 及び INVERT、二つのオプションパラメータを併用することにより 頂点角数が 4 個を超え (= 5 個以上) 16 個以下、かつブレークアウトの無いアウトラインポリゴン、で無いポリゴンのみを全て選択するようにしています :

```
dLayer := DLaSelShape dLayer1; "16 >= NB_EDGES > 4 NOHOLE INVERT"
```

### 例 2 : Non Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 Number of Holes

NB\_Holes は dLayer1 の形状 (shape) から Val1 CO1、CO2 Val2 のパラメータで指定した範囲の孔数 (ブレークアウト数) を持つ形状を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DiaSelShape dLayer1;"[Val1 CO1]NB_HOLES[CO2 Val2] [NOHOLE] [INVERT]"
```

### 定義

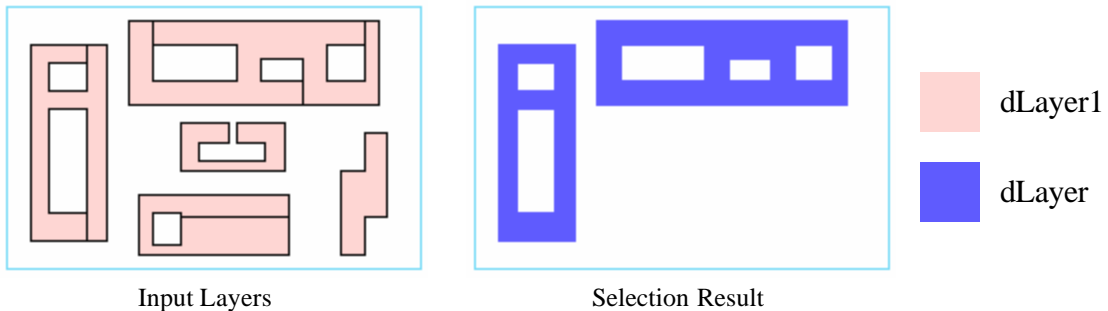
dLayer1 のポリゴンの中から指定した範囲の孔数 (ブレークアウト数) を持つポリゴンを全て選択します。このコマンドスイッチは単独では使用できず、必ず Val1 CO1、CO2 Val2 の両方、またはどちらかのパラメータの併記が必須となりますのでご注意ください。

因みに下記演算例では孔数が 2 個以上のポリゴンのみを選択するよう指定しています。

### 演算例

```
dLayer := DiaSelShape dLayer1;"NB_HOLES>=2"
```

#### 例1 : Obvious

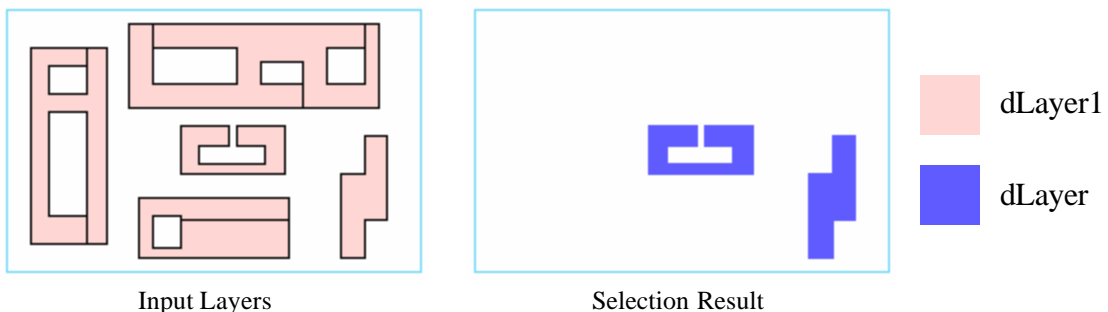


#### 演算例 2

例 2 (Non Obvious) では INVERT オプションパラメータを併用することにより孔数 1 個以上、で無いポリゴンのみを全て選択するようにしています：

```
dLayer := DiaSelShape dLayer1;"1<=NB_HOLES INVERT"
```

#### 例2 : Non Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 Octagonal Angles

OCTA は dLayer1 の形状 (shape) から座標軸と45度単位の角度をなす断片を持つ形状を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DLaSelShape dLayer1; "OCTA [ALL] [NOHOLE] [INVERT]"
```

#### 定義

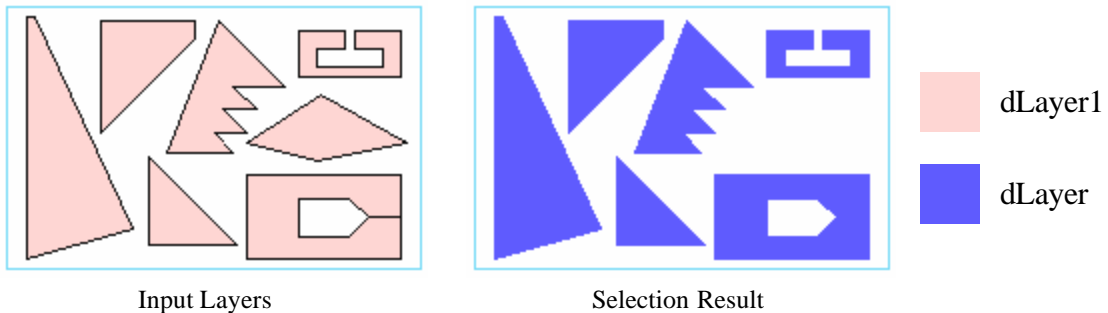
dLayer1 のポリゴンから座標軸と45度単位の角度をなす断片を持つポリゴンを全て選択します。つまり座標軸と0度、45度、90度、135度、180度、225度、270度、315度、360度のいずれかの角度の断片を持つポリゴンは全て選択されます。

因みに下記演算例では座標軸と45度単位の角度をなす断片が一つでもあるポリゴンを選択するよう指定しています。

#### 演算例

```
dLayer := DLaSelShape dLayer1; "OCTA"
```

例1 : Obvious



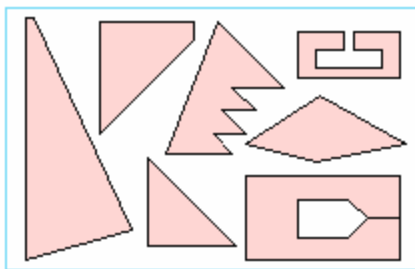
## 特定のプロパティによる形状 (shape) 選択演算

### 演算例 2

例 2 (Non Obvious) では NOHOLE と INVERT、二つのオプションパラメータを使用することにより 座標軸と 45 度単位の角度の断片を持ち、かつブレイクアウトを持たないポリゴン」で無い全てのポリゴンを選択するようにしています：

```
dLayer := DiaSelShape dLayer1; "OCTA NOHOLE INVERT"
```

### 例 2 : Non Obvious



Input Layers



Selection Result



## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 **Orthogonal Angles**

ORTHO は dLayer1 の形状 (shape) から座標軸と水平または垂直の断片を持つ形状を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DLaSelShape dLayer1; "ORTHO [ALL] [NOHOLE] [INVERT]"
```

#### 定義

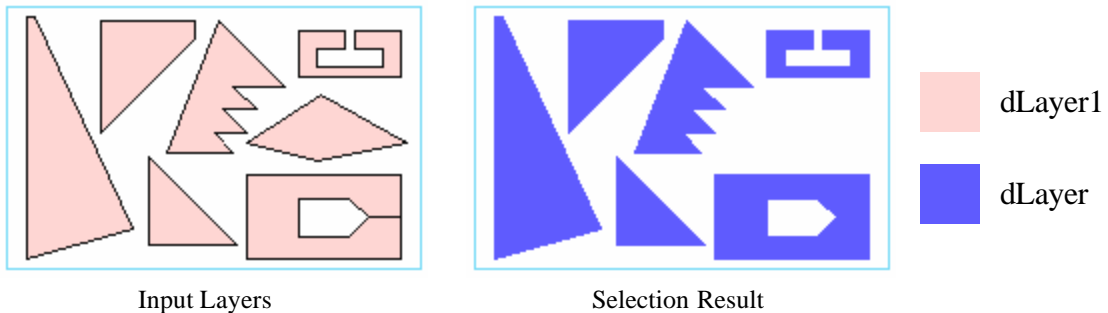
dLayer1 のポリゴンから座標軸と水平または垂直の断片を持つポリゴンを全て選択します。つまり、座標軸と0度、90度、180度、270度、360度のいずれかの角度をなす断片を持つポリゴンは全て選択されます。

因みに下記演算例では座標軸と水平または垂直をなす断片が一つでもあるポリゴンを選択するよう指定しています。

#### 演算例

```
dLayer := DLaSelShape dLayer1; "ORTHO"
```

例1 : Obvious



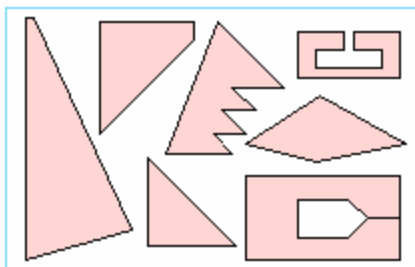
## 特定のプロパティによる形状 (shape) 選択演算

### 演算例 2

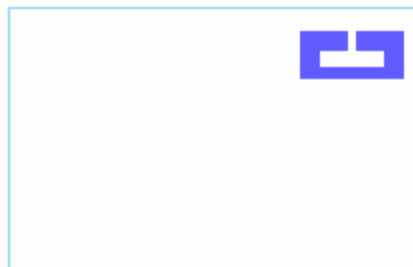
例 2 (Non Obvious) では ALL オプションパラメータを使用することにより、全ての断片が座標軸に対して水平・垂直をなしている全てのポリゴンを選択するようにしています:

```
dLayer := DiaSelShape dLayer1; "ORTHO ALL"
```

### 例2 : Non Obvious



Input Layers



Selection Result





## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 Perimeter

Perimeter は dLayer1 の形状 (shape) から Val1 CO1、CO2 Val2 のパラメータで指定した範囲の長さの周囲長を持つ形状を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelShape dLayer1; "[Val1 CO1]PERIMETER[CO2 Val2] [NOHOLE] [INVERT]"
```

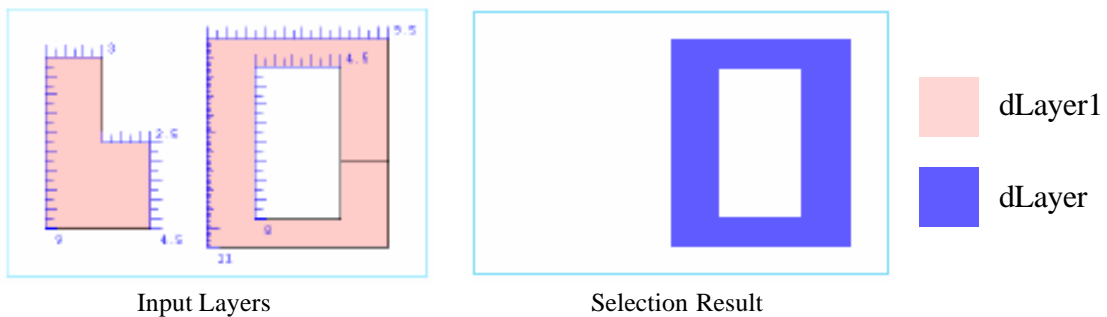
### 定義

dLayer1 のポリゴンの中から Val1 CO1、CO2 Val2 のパラメータで指定した範囲の長さの周囲長を持つポリゴンを全て選択します。Val1 CO1、CO2 Val2 はどちらか一方の指定だけでも構いません。下記演算例では 30 以上 67 未満の長さの断片を持つ形状のみ選択するよう指定しています。

### 演算例

```
dLayer := DLaSelShape dLayer1; "67>PERIMETER>=30"
```

#### 例1 : Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 演算例 2

例 2 (Non Obvious) では NOHOLE 及び INVERT、二つのオプションパラメータを併用することにより 周囲長が 35、かつブレークアウトの無いアウトラインポリゴンで無いポリゴンのみを全て選択するようにしています :

```
dLayer := DLaSelShape dLayer1; "PERIMETER=35 NOHOLE INVERT"
```

### 例 2 : Non Obvious



## 特定のプロパティによる形状 (shape) 選択演算

### 形状選択演算 Rectangles

Rectangle は dLayer1 の形状 (shape) から長方形のアウトラインポリゴンのみを選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DLaSelShape dLayer1; "RECTANGLE [INVERT]"
```

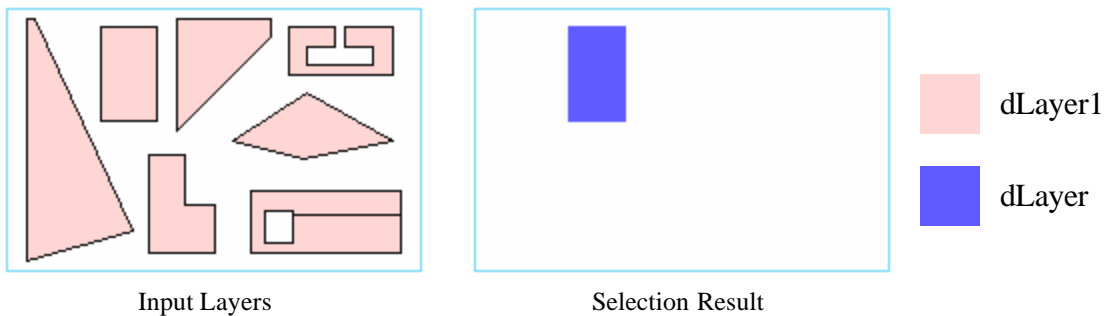
#### 定義

dLayer1 のポリゴンの中から長方形のアウトラインポリゴンであるポリゴンを全て選択します。長方形の形をしていてもブレイクアウトのあるインナーポリゴンは排除されます。

#### 演算例

```
dLayer := DLaSelShape dLayer1; "RECTANGLE"
```

##### 例1 : Obvious

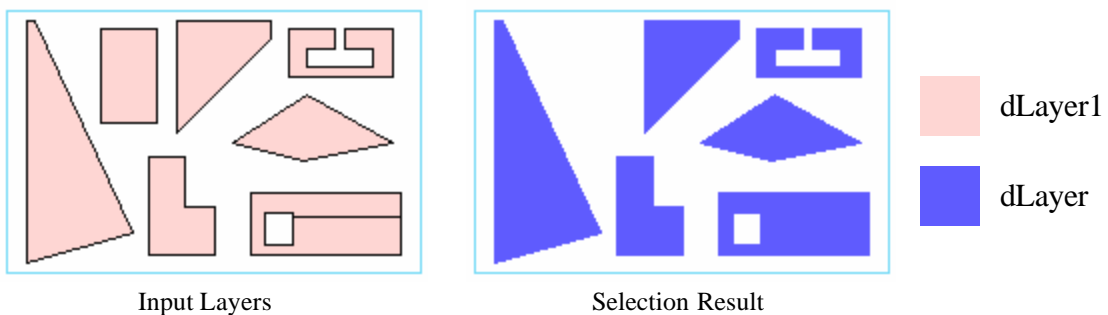


##### 演算例 2

例 2 (Non Obvious) では INVERT オプションパラメータを併用することにより「長方形かつアウトラインポリゴン」で無いポリゴンのみを全て選択するようにしています：

```
dLayer := DLaSelShape dLayer1; "RECTANGLE INVERT"
```

##### 例2 : Non Obvious



## 関連位置によるエッジ選択演算

---

### 関連位置によるエッジ選択演算

dw-2000 でサポートしている derived レイヤーからのエッジ選択に使用される関連位置オプションは以下の通りです：

- ・ Butting
- ・ Coincident
- ・ Touching

コマンド入力では上記それぞれのオプションに対応したコマンドスイッチを選択演算子である「DlaSelEdge」コマンドの後に入力します：

```
dLayer := DlaSelEdge dLayer1;dLayer2;"SEL_OP [INVERT]"
```

### コマンドパラメータ

dLayer	演算後の結果となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
DlaSelEdge	選択演算のメインコマンドです。
dLayer1	エッジ選択されるエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。
dLayer2	dLayer1 の抽出エッジの位置に関連して使用されるエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。
SEL_OP	選択演算の種類を決定するコマンドスイッチです。
INVERT	指定した選択演算と逆の結果を返します。

## 関連位置によるエッジ選択演算

---

## 関連位置によるエッジ選択演算

---

### エッジ選択演算 Butting

Butting は dLayer2 の外側の断片と接している断片を持つ dLayer1 のエッジを選択するコマンドスイッチ (SEL\_OP) です (Butting の定義については 36 ページ参照)。

### コマンド形式

```
dLayer := DiaSelEdge dLayer1;dLayer2;"BUTTING [INVERT]"
```

### 定義

dLayer2 を形成しているポリゴンの外側の断片と dLayer2 の外部で接している断片を持つエッジのみが選択されます。次項に記してある例 2 :Non Obvious の左側の形状のように、一見 dLayer2 の内側の断片に見えるような場合でも、ポリゴン内部の断片では無いので「外側」の断片となります ( 7 ページ参照 )。

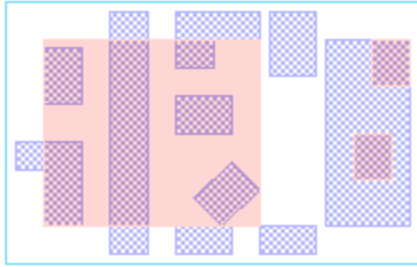
dLayer2 が断片だけの derived レイヤーである場合、その断片の内側・外側の方向はそのレイヤーが生成される前の元となったポリゴンの形状から引き継がれます。

## 関連位置によるエッジ選択演算

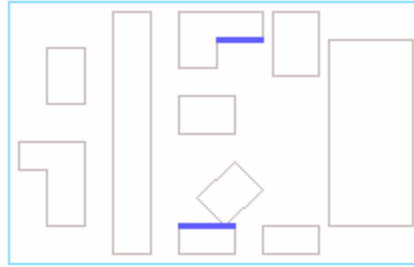
### 演算例

dLayer := DIaSelEdge dLayer1;dLayer2;"BUTTING"

#### 例1 : Obvious

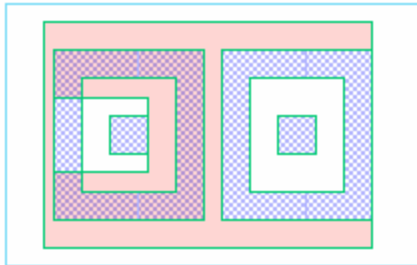


Input Layers

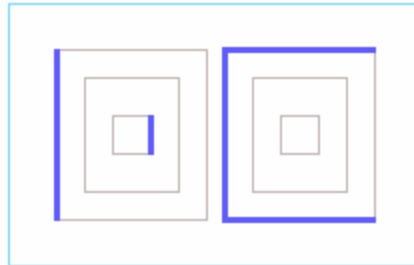


Selection Result

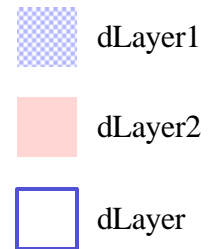
#### 例2 : Non Obvious



Input Layers



Selection Result



## 関連位置によるエッジ選択演算

---

### **エッジ選択演算 Coincident**

Coincident は dLayer2 の断片と接している断片を持ち、かつポリゴンエリアが重なっている dLayer1 のエッジを選択するコマンドスイッチ (SEL\_OP )です (Coincident の定義については 40 ページ参照)。

### **コマンド形式**

```
dLayer := DiaSelEdge dLayer1;dLayer2;"COINCIDENT [INVERT]"
```

### **定義**

dLayer2 の断片と接している断片を持ち、かつポリゴンエリアが重なっているエッジが選択されます。つまり内側同士の断片が重なり合っているエッジのみが選択されるということです。

dLayer2 が断片だけの derived レイヤーである場合、その断片の内側・外側の方向はそのレイヤーが生成される前の元となったポリゴンの形状から引き継がれます。

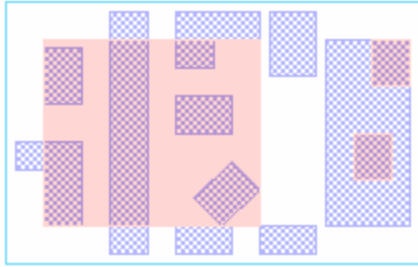


## 関連位置によるエッジ選択演算

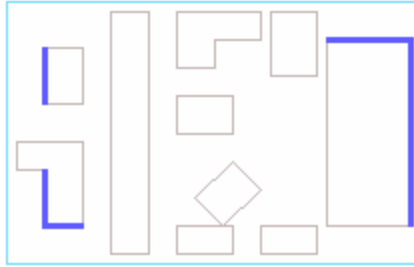
### 演算例

dLayer := DIaSelEdge dLayer1;dLayer2;"COINCIDENT"

#### 例1 : Obvious

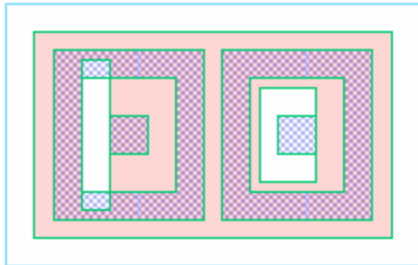


Input Layers

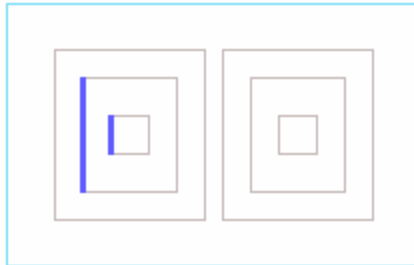


Selection Result

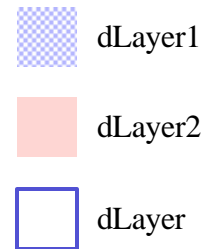
#### 例2 : Non Obvious



Input Layers



Selection Result



## 関連位置によるエッジ選択演算

### エッジ選択演算 Touching

Touching は dLayer2 の断片と接している断片を持つ dLayer1 のエッジを選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DiaSelEdge dLayer1;dLayer2;"TOUCHING [INVERT]"
```

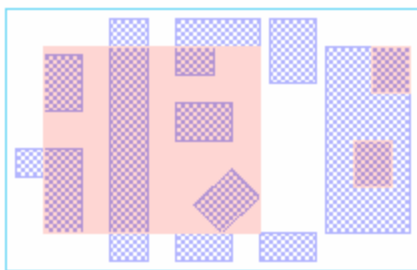
#### 定義

dLayer2 の断片と接しているエッジが選択されます。Butting や Coincident と異なり、接してさえいればその断片の方向は関係ありません。

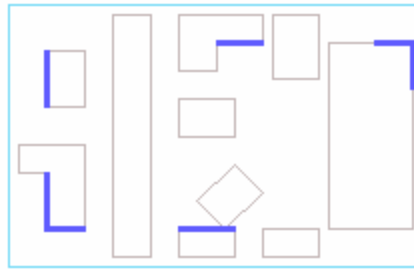
#### 演算例

```
dLayer := DiaSelEdge dLayer1;dLayer2;"TOUCHING"
```

##### 例1 : Obvious

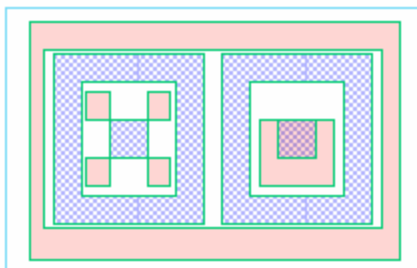


Input Layers

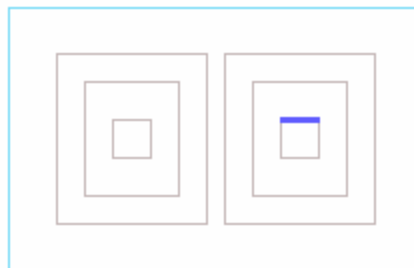


Selection Result

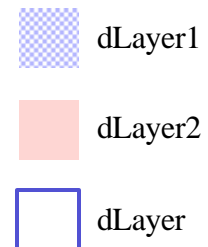
##### 例2 : Non Obvious



Input Layers



Selection Result



## 特定のプロパティによるエッジ選択演算

### 特定のプロパティによるエッジ選択演算

dw-2000 でサポートしている、derived レイヤーからのエッジ選択に使用されるプロパティ指定オプションは以下の通りです：

- Acute
- Bend Radius
- 45 Degrees
- Length
- Number of Edges
- Octagonal
- Orthogonal
- Relative Angle
- Total Length

コマンド入力では前節の「関連位置によるエッジ選択演算」同様、上記それぞれのオプションに対応したコマンドスイッチを選択演算子である「DlaSelEdge」コマンドの後に入力します。但し前節では dLayer2 のエレメントと dLayer1 のエレメントとの配置位置関係からそれぞれのオプションに該当するエッジを選択していったのに対し、今回は dLayer1 のエレメントの特徴からのみ選択していくため演算に使用する derived レイヤーは dLayer1 の一つだけとなります：

```
dLayer := DlaSelEdge dLayer1; "[Val1 CO1]SEL_OP[CO2 Val2] [INVERT]"
```

#### コマンドパラメータ

dLayer	演算後の結果となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
DlaSelEdge	選択演算のメインコマンドです。
dLayer1	エッジ選択されるエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。
Val1 及び Val2	抽出するエッジの数量を条件付けする為のパラメータです。
CO1 及び CO2	Val1 及び Val2 で定義された数値につく等式、不等式です。つまりこのパラメータは > ・ >= ・ = ・ < ・ <= のいずれかとなります。
SEL_OP	選択演算の種類を決定するコマンドスイッチです。
INVERT	指定した選択演算と逆の結果を返します。

## 特定のプロパティによるエッジ選択演算

---

本章では前項に掲載したコマンドスイッチ全てに対して二種類 (例 1 : Obvious、例 2 : Non Obvious) の演算例を紹介しています。また紹介する全ての例の基本形となるセルは以下のフォルダに収録してあります：

`¥samples¥dla_examples¥sel_edges¥property`

それぞれの例に対するスクリプトも同フォルダ下の「scripts」フォルダ内に収録しております。お試しいたいたい例にあわせてそれぞれ読み込み、演算行を書き換え、再度読み込んで (File:Save and Load... ) お使いになってみて下さい

またエッジ選択演算は形状選択演算のように物理的なエレメントとして表示することはできません。従って結果となる derived レイヤーは物理的なレイヤーとしてではなくあくまでも他の演算や XDRC 機能 (要 XDRC モジュール) への入力 derived レイヤーとしてしか使用することはできません。つまりデータベースへの保存は行われないため、同一マクロ内で行われる演算への GPE 変数 (derived レイヤー) としてしか使用することができないということです (XDRC 機能に使用される場合はエラー情報として保持しておくことが可能です)。

因みに次項以降のようにエッジが抽出されるか視覚的に記載しておりますが、本例では XDRC モジュールのコマンドを用いてエラー情報として便宜的に表示しております。従って上記に掲載したフォルダよりサンプルをお試しになった際でも、演算結果となるストラクチャにはエラー情報として表示されているため、DRC:Clear All Errors コマンドを用いて現在の演算によるエラー情報をクリアしてから次の演算例に移るようにして下さい。

DRC メニューに関する詳細は「日本語版メニュー解説マニュアル」の DRC の項を、XDRC モジュールでの断片の確認に関する詳細は「Working with the XDRC」マニュアルの「Viewing Segment Derived Layers」の項をそれぞれご参照下さい。

## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 Acute Angle

Acute は dLayer1 で相互に接合している断片同士が 90 度未満の角をなしているエッジを選択するコマンドスイッチ (SEL\_OP) です。

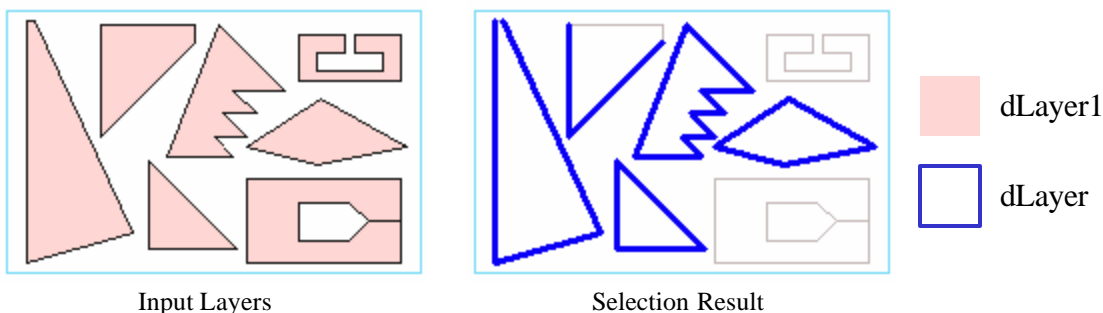
#### コマンド形式

```
dLayer := DiaSelEdge dLayer1;"ACUTE [INVERT]"
```

#### 演算例

```
dLayer := DiaSelEdge dLayer1;"ACUTE"
```

##### 例1 : Obvious

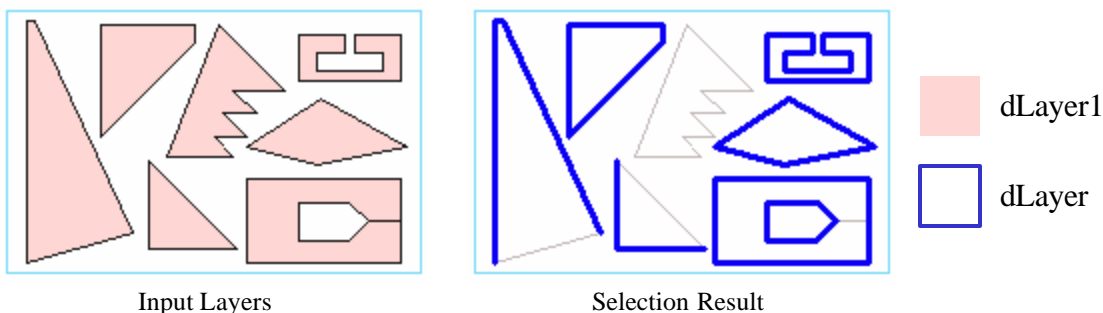


##### 演算例 2

例 2 (Non Obvious) では INVERT オプションパラメータを使用することにより、相互に接合している断片が 90 度以上をなしているエッジが全て選択されます：

```
dLayer := DiaSelEdge dLayer1;"ACUTE INVERT"
```

##### 例2 : Non Obvious



## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 Bend Radius

BendRadii は dLayer1 にある曲線半径 (Bend Radius) を指定してエッジを選択するコマンドスイッチ (SEL\_OP) です。

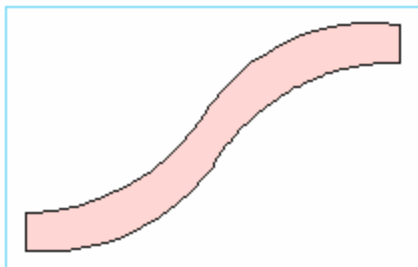
この曲線半径 (Bend Radius) とは、二つの平行でないエッジが最終的に交差するポイントとその二つの平行でないエッジの反対側の終端、それら三つのポイントを通る円の半径のことをここでは指します。また半径の大きさは Val1 CO1、CO2 Val2 それぞれのパラメータの両方、またはいずれか一方を使用して指定します。

### コマンド形式

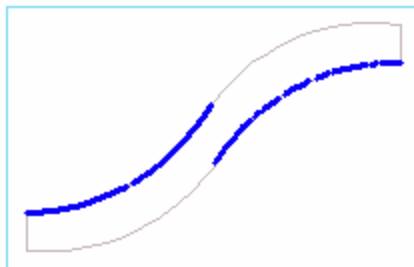
```
dLayer := DiaSelEdge dLayer1; "[Val1 CO1] BENDRADI [CO2 Val2] [INVERT]"
```

### 演算例

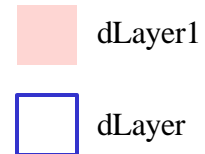
```
dLayer := DiaSelEdge dLayer1; "BENDRADI > 15"
```



Input Layers



Selection Result



## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 45 Degree Angle

DEG\_45 は dLayer1 で座標軸と 45 度の角をなしているエッジを選択するコマンドスイッチ (SEL\_OP) です。

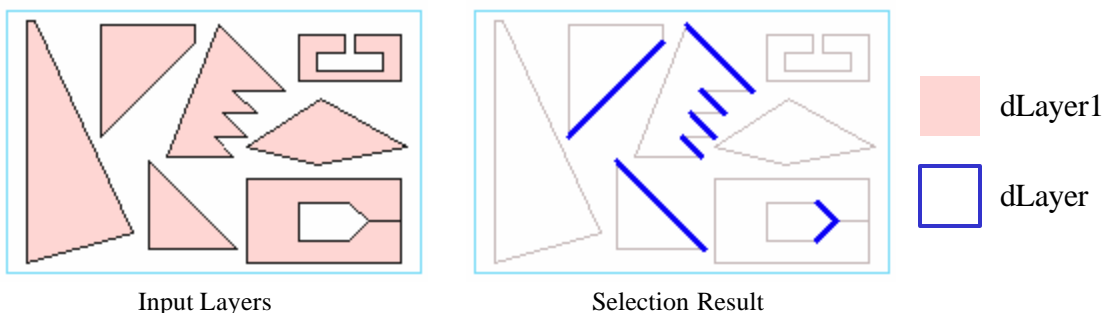
#### コマンド形式

```
dLayer := DiaSelEdge dLayer1;"DEG_45 [INVERT]"
```

#### 演算例

```
dLayer := DiaSelEdge dLayer1;"DEG_45"
```

##### 例1 : Obvious

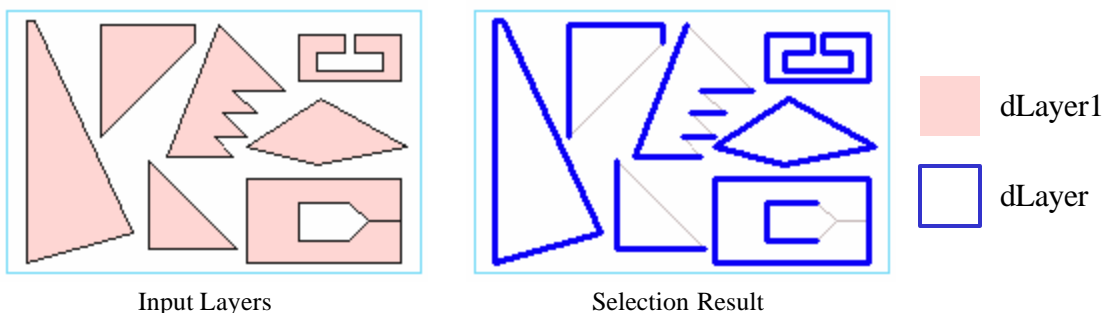


##### 演算例 2

例 2 (Non Obvious) では INVERT オプションパラメータを使用することにより、座標軸と 45 度の角をなしていないエッジが全て選択されます :

```
dLayer := DiaSelEdge dLayer1;"DEG_45 INVERT"
```

##### 例2 : Non Obvious



## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 Length

Length は dLayer1 から Val1 CO1、CO2 Val2 のパラメータで指定した範囲の長さの断片を持つエッジを選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

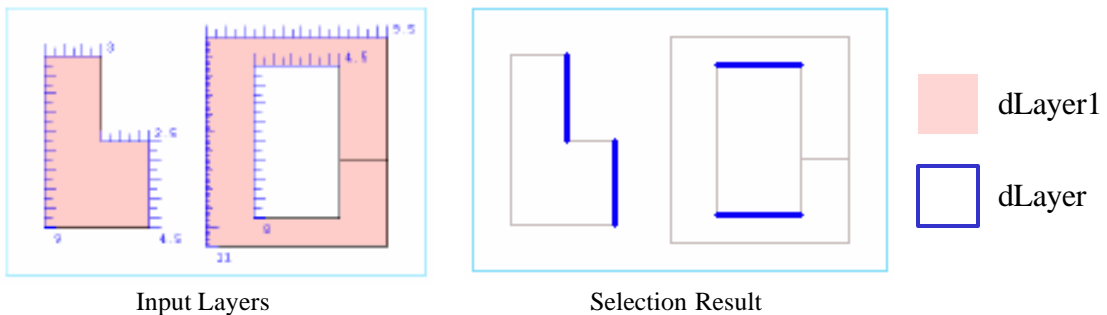
```
dLayer := DLaSelEdge dLayer1; "[Val1 CO1]LENGTH[CO2 Val2] [INVERT]"
```

### 演算例

下記演算例 (Obvious) では断片の長さが 4.5 のエッジのみを選択しています：

```
dLayer := DLaSelEdge dLayer1; "LENGTH=4.5"
```

#### 例1 : Obvious

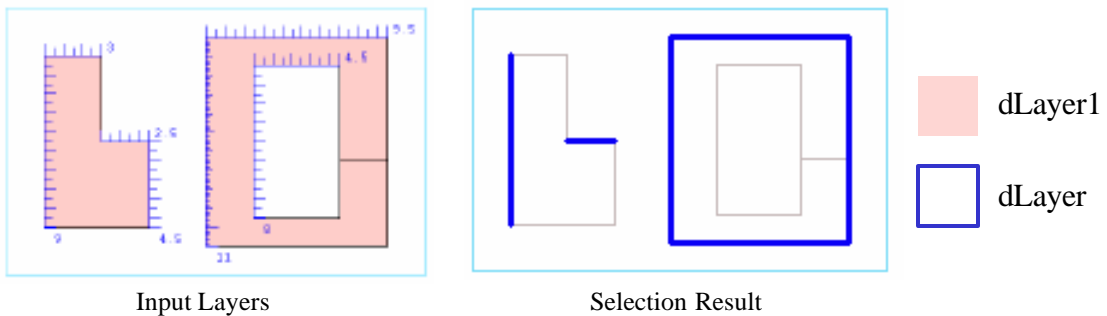


### 演算例 2

例 2 (Non Obvious) では INVERT オプションパラメータを使用することにより「断片の長さが 3 以上 9 未満」で無いエッジが全て選択されます：

```
dLayer := DLaSelEdge dLayer1; "3<=LENGTH<9 INVERT"
```

#### 例2 : Non Obvious





## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 Number of Edges

NB\_Edges は dLayer1 から Val1 CO1、CO2 Val2 のパラメータで指定した範囲の断片数を持つポリゴンのエッジを全て選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

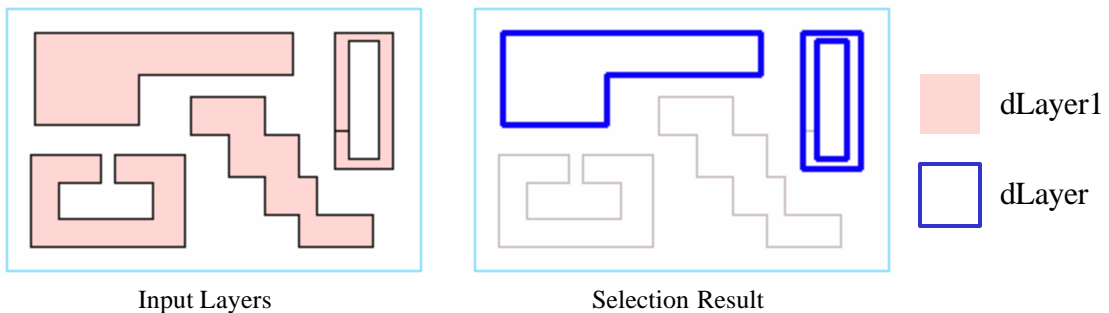
```
dLayer := DlaSelEdge dLayer1;"[Val1 CO1]NB_EDGES[CO2 Val2] [INVERT]"
```

#### 演算例

下記演算例 (Obvious) では断片の数が 8 個以下のポリゴンのエッジを全て選択しています：

```
dLayer := DlaSelEdge dLayer1;"NB_EDGES<=8"
```

##### 例1 : Obvious

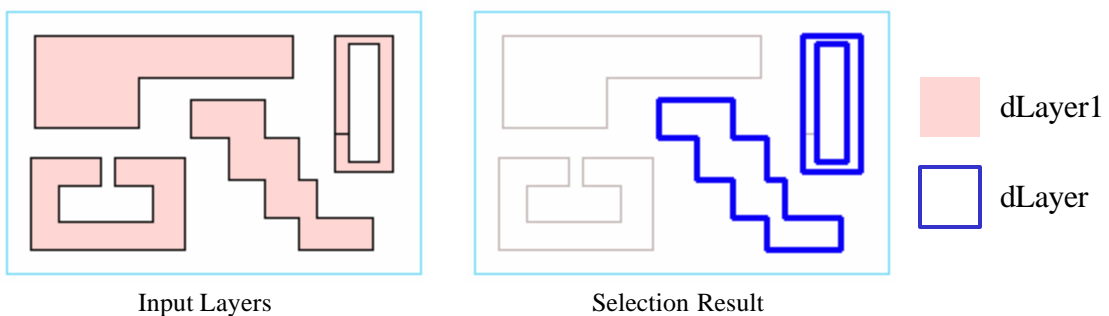


#### 演算例 2

例 2 (Non Obvious) では INVERT オプションパラメータを使用することにより、断片の数が 4 個を超え (5 個以上) 16 個未満で無いポリゴンのエッジが全て選択されます：

```
dLayer := DlaSelEdge dLayer1;"4<NB_EDGES<16 INVERT"
```

##### 例2 : Non Obvious



## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 Octagonal Angles

OCTA は dLayer1 から座標軸と 45 度単位の角度をなすエッジを選択するコマンドスイッチ (SEL\_OP) です。

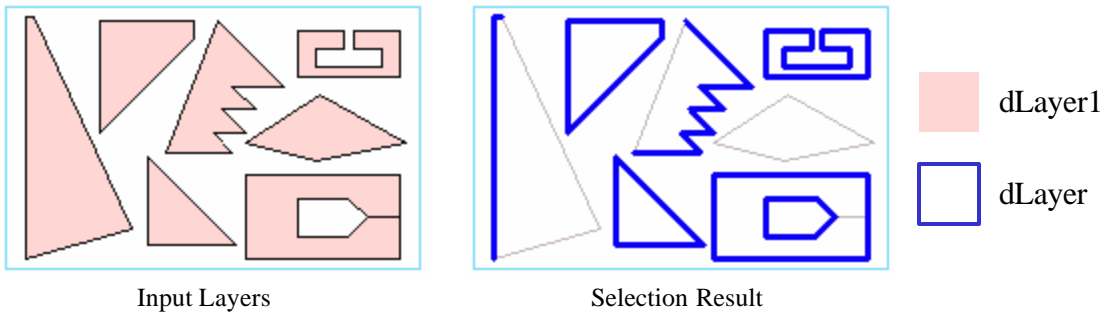
#### コマンド形式

```
dLayer := DiaSelEdge dLayer1; "OCTA [INVERT]"
```

#### 演算例

```
dLayer := DiaSelEdge dLayer1; "OCTA"
```

##### 例1 : Obvious

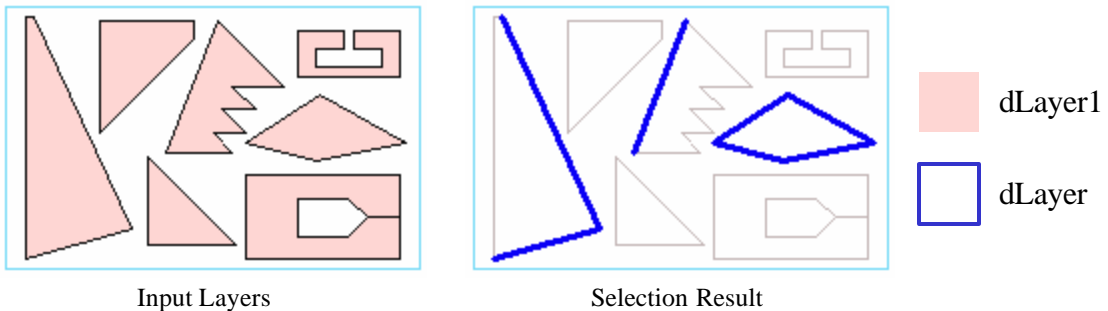


##### 演算例 2

例2 (Non Obvious) では INVERT オプションパラメータを使用することにより、座標軸と45度単位の角度をなしていないエッジが全て選択されます：

```
dLayer := DiaSelEdge dLayer1; "OCTA INVERT"
```

##### 例2 : Non Obvious



## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 Orthogonal Angles

ORTHO はdLayer1 から座標軸と水平または垂直のエッジを選択するコマンドスイッチ (SEL\_OP )です。

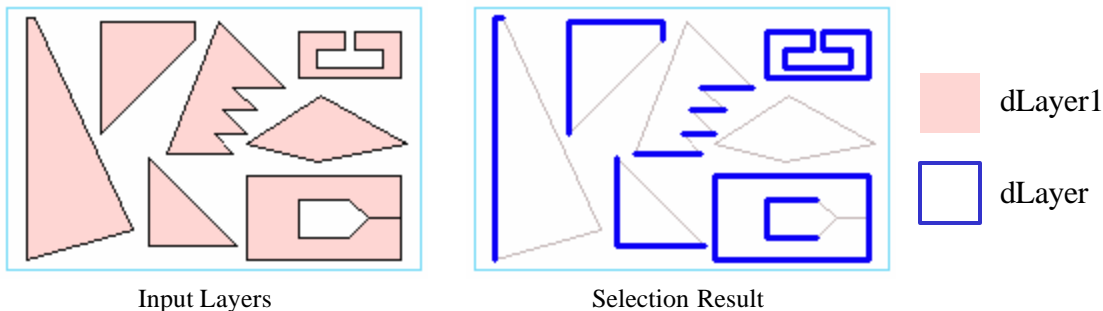
#### コマンド形式

```
dLayer := DlaSelEdge dLayer1;"ORTHO [INVERT]"
```

#### 演算例

```
dLayer := DlaSelEdge dLayer1;"ORTHO"
```

##### 例1 : Obvious

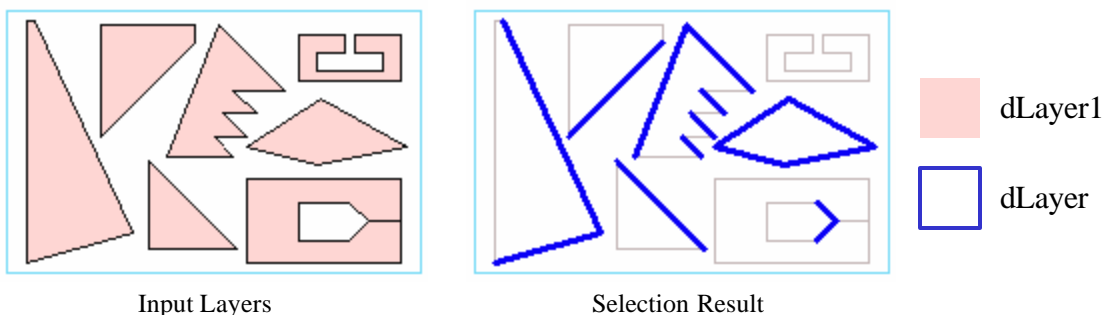


#### 演算例 2

例 2 (Non Obvious) では INVERT オプションパラメータを使用することにより、座標軸と水平または垂直をなしていないエッジが全て選択されます：

```
dLayer := DlaSelEdge dLayer1;"ORTHO INVERT"
```

##### 例2 : Non Obvious



## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 Relative Angle

REL\_Angle は dLayer1 の接合している断片同士が Val1 CO1, CO2 Val2 パラメータで指定した角度をなしているエッジを選択するコマンドスイッチ (SEL\_OP) です。

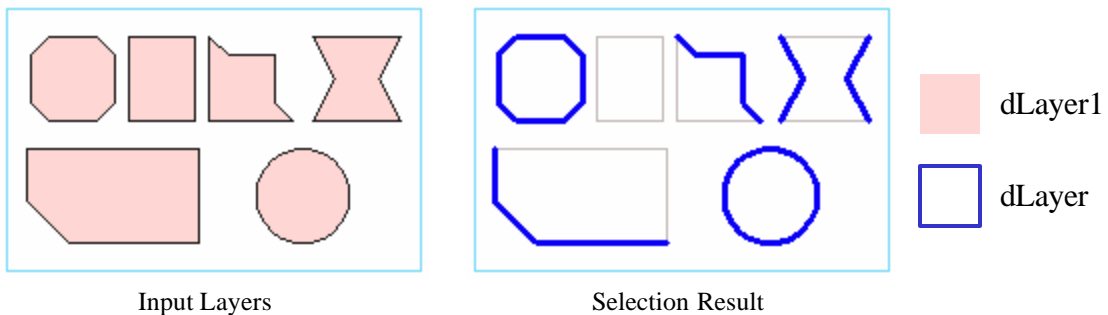
### コマンド形式

```
dLayer := DiaSelEdge dLayer1;" [Val1 CO1] REL_ANGLE [CO2 Val2] [INVERT]"
```

### 演算例

下記演算例では90度を超える角 (90度の角は含まれない)を形成している、相互に接合しているエッジを全て選択しています：

```
dLayer := DiaSelEdge dLayer1;"REL_ANGLE > 90"
```



## 特定のプロパティによるエッジ選択演算

### エッジ選択演算 Total Length

TOT\_Length は dLayer1 からポリゴンの接合する断片の長さの合計が Val1 CO1、CO2 Val2 のパラメータで指定した範囲である時、その条件を満たし接合している全てのエッジを選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

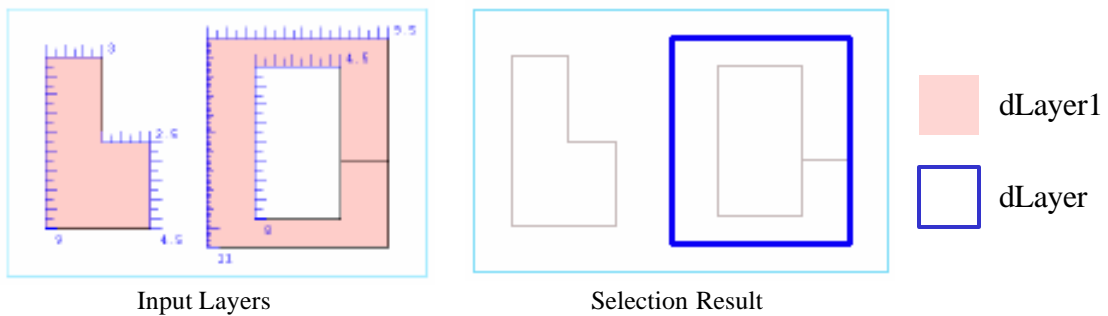
```
dLayer := DiaSelEdge dLayer1;"[Val1 CO1]TOT_LENGTH[CO2 Val2] [INVERT]"
```

### 演算例

下記演算例 (Obvious) ではポリゴンの接合する断片の長さの合計が 29 を超え 70 未満であるエッジのみを選択しています：

```
dLayer := DiaSelEdge dLayer1;" 70>TOT_LENGTH>29"
```

#### 例1 : Obvious

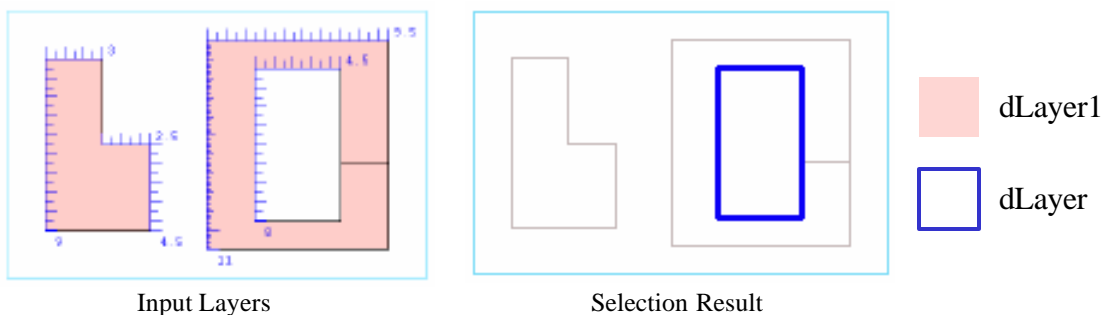


### 演算例 2

例 2 (Non Obvious) では INVERT オプションパラメータを使用することにより、ポリゴンの断片の長さの合計が 25 を超え 45 未満、で無いエッジが全て選択されます：

```
dLayer := DiaSelEdge dLayer1;" 25<TOT_LENGTH< 45 INVERT"
```

#### 例2 : Non Obvious



## 関連位置による断片選択演算

---

### 関連位置による断片選択演算

dw-2000 でサポートしている、derived レイヤーからの断片 (Segment) 選択に使用される関連位置オプションは以下の通りです：

- Avoid
- Butting
- Butting or Overlap
- Coincident
- Inside
- Inside Only
- Outside
- Touching

コマンド入力では上記それぞれのオプションに対応したコマンドスイッチを選択演算子である「DlaSelSeg」コマンドの後に入力します：

```
dLayer := DlaSelSeg dLayer1;dLayer2;" SEL_OP [INVERT]"
```

### コマンドパラメータ

dLayer	演算後の結果となる derived レイヤーの GPE 変数名です。ここでは便宜上「dLayer」という名称を使用しておりますが、実際の構文ではユーザー定義の任意の文字列となります。
DlaSelSeg	選択演算のメインコマンドです。
dLayer1	断片 (Segment) 選択されるエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。
dLayer2	dLayer1 の抽出断片 (Segment) の位置に関連して使用されるエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名です。
SEL_OP	選択演算の種類を決定するコマンドスイッチです。
INVERT	指定した選択演算と逆の結果を返します。

## 関連位置による断片選択演算

---

本章では前項に掲載したコマンドスイッチ全てに対して二種類 (例 1 :Obvious、例 2 :Non Obvious) の演算例を紹介しています。また紹介する全ての例の基本形となるセルは以下のフォルダに収録してあります :

`¥samples¥dla_examples¥sel_segments/position/obvious` (例 1 :Obvious 用)

`¥samples¥dla_examples¥sel_segments/position/non_obvious` (例 2 :Non Obvious 用)

それぞれの例に対するスクリプトも同フォルダ下の「scripts」フォルダ内に収録しております。お試しいなりたい例にあわせてそれぞれ読み込み、演算行を書き換え、再度読み込んで (File:Save and Load...) お使いになってみて下さい

また断片選択演算はエッジ選択演算同様、形状選択演算のように物理的なエレメントとして表示することはできません。従って結果となる derived レイヤーは物理的なレイヤーとしてではなくあくまでも他の演算や XDRC 機能 (要 XDRC モジュール) への入力 derived レイヤーとしてしか使用することはできません。つまりデータベースへの保存は行われなため、同一マクロ内で行われる演算への GPE 変数 (derived レイヤー) としてしか使用することができないということです (XDRC 機能に使用される場合はエラー情報として保持しておくことが可能です)。

因みに次項以降のように断片が抽出されるか視覚的に記載しておりますが、本例では XDRC モジュールのコマンドを用いてエラー情報として便宜的に表示しております。従って上記に掲載したフォルダよりサンプルをお試しになった際でも、演算結果となるストラクチャにはエラー情報として表示されているため、DRC:Clear All Errors コマンドを用いて現在の演算によるエラー情報をクリアしてから次の演算例に移るようにして下さい。

DRC メニューに関する詳細は「日本語版メニュー解説マニュアル」の DRC の項を、XDRC モジュールでの断片の確認に関する詳細は「Working with the XDRC」マニュアルの「Viewing Segment Derived Layers」の項をそれぞれご参照下さい。

## 関連位置による断片選択演算

### 断片選択演算 Avoid

Avoid は dLayer2 から完全に外部に位置する dLayer1 の断片 (segment) を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DiaSelSeg dLayer1;dLayer2;"AVOID [INVERT]"
```

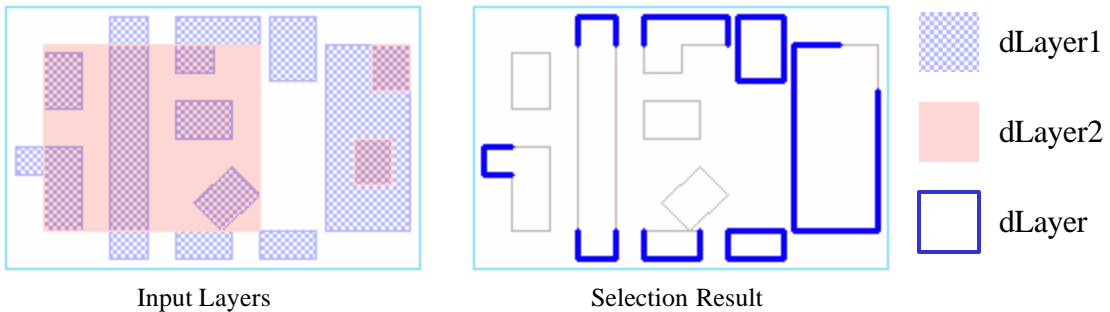
### 定義

dLayer2 から完全に孤立した断片のみが選択されます。内部にあったり断片同士重なり合っている場合は全て排除されます。但し頂点角のみが接触している場合は排除されず選択されることとなります。

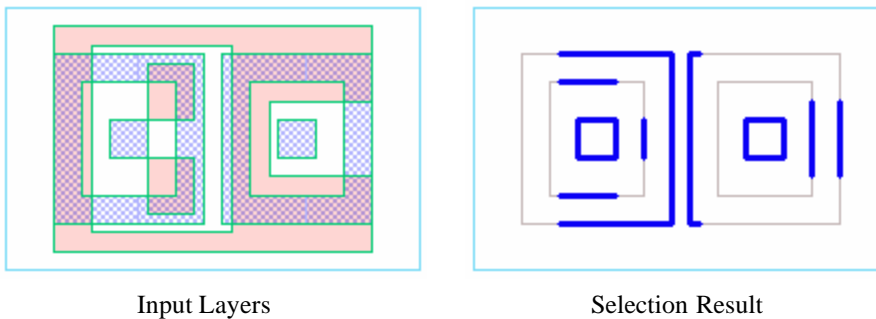
### 演算例

```
dLayer := DiaSelSeg dLayer1;dLayer2;"AVOID"
```

#### 例1 : Obvious



#### 例2 : Non Obvious





## 関連位置による断片選択演算

---

### 断片選択演算 **Butting**

Butting は dLayer2 の形状 (shape) の外側の断片と、dLayer1 の形状 (shape) の外側の断片が相互に接している断片を選択するコマンドスイッチ (SEL\_OP) です。

( Butting の定義に関しては 36 ページ参照 )

### コマンド形式

```
dLayer := DiaSelSeg dLayer1;dLayer2;"BUTTING [INVERT]"
```

### 定義

dLayer2 を形成しているポリゴンの外側の断片と、dLayer1 を形成しているポリゴンの外側の断片が dLayer2 の外部で相互に接している断片のみが選択されます。

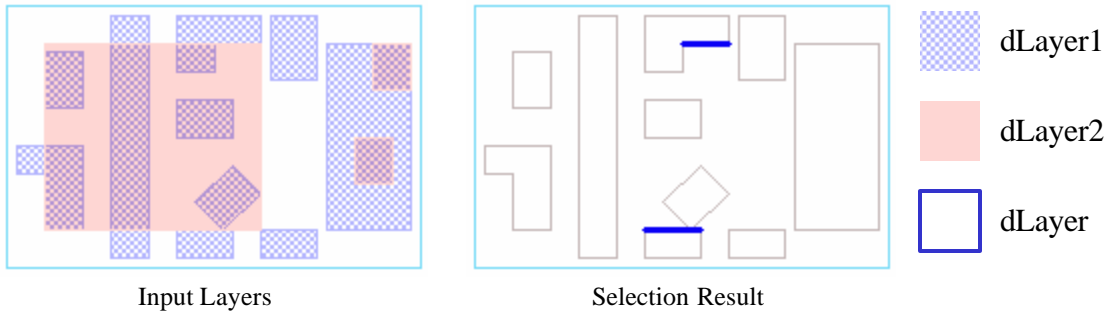
(断片の方向については 7 ページ参照)。

## 関連位置による断片選択演算

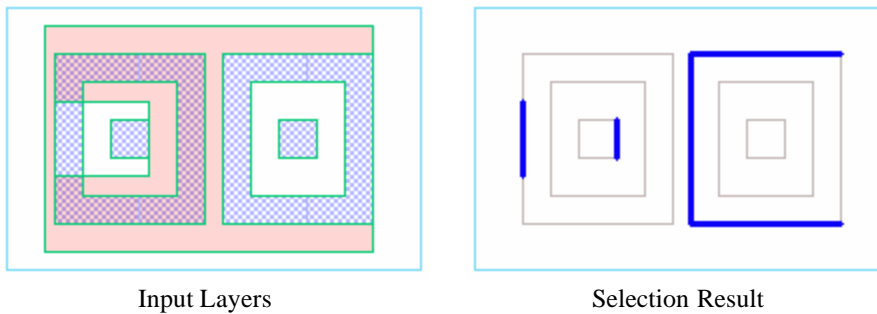
### 演算例

dLayer := DIaSelSeg dLayer1;dLayer2;"BUTTING"

#### 例1 : Obvious



#### 例2 : Non Obvious



## 関連位置による断片選択演算

### 断片選択演算 **Butting or Overlap**

ButtOrOver は dLayer2 の形状 (shape )の外側の断片と接している断片を持つか、dLayer2 のどこかと一部でも重なっている dLayer1 の断片を選択するコマンドスイッチ (SEL\_OP )です。

#### コマンド形式

```
dLayer := DIaSelSeg dLayer1;dLayer2;" BUTTOROVER [INVERT]"
```

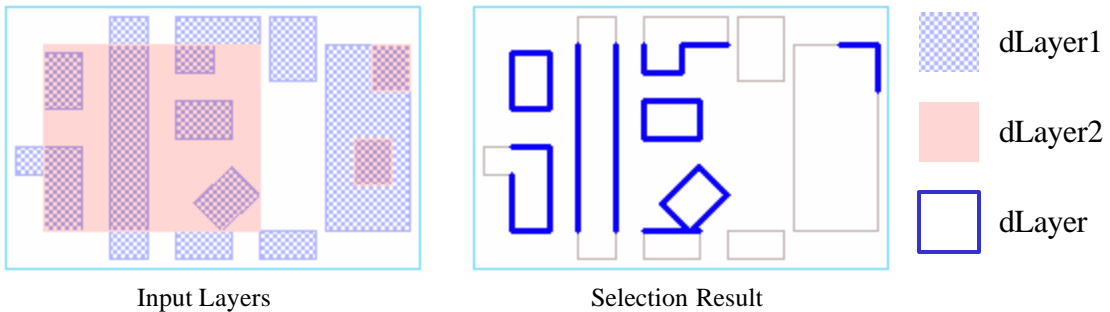
#### 定義

dLayer2 の外側の断片と接している断片を持つか、dLayer2 のどこかと一部でも重なっている断片のみが選択されます。但し、頂点角のみが接している形状については排除されます。

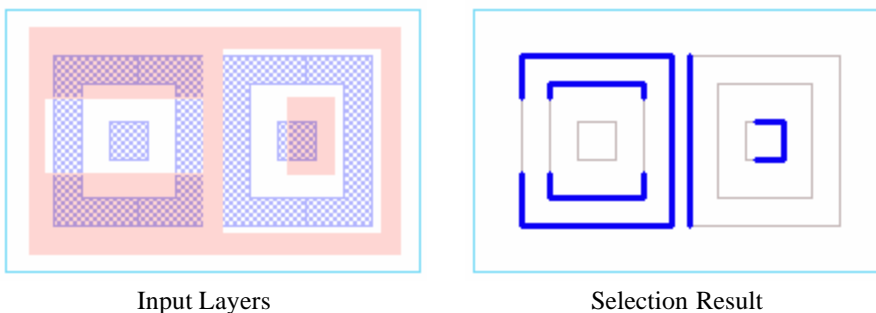
#### 演算例

```
dLayer := DIaSelSeg dLayer1;dLayer2;"BUTTOROVER"
```

##### 例1 : Obvious



##### 例2 : Non Obvious



## 関連位置による断片選択演算

---

### 断片選択演算 **Coincident**

Coincident は dLayer2 の形状 (shape) とポリゴンエリアが重なっている dLayer1 の形状 (shape) から、dLayer2 の断片と接している dLayer1 の断片を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DLaSelSeg dLayer1;dLayer2;"COINCIDENT [INVERT]"
```

### 定義

dLayer2 のポリゴンと重なり、かつ重なっている断片のみが選択されます。つまり内側同士の断片が重なり合っている断片のみが選択されるということです。

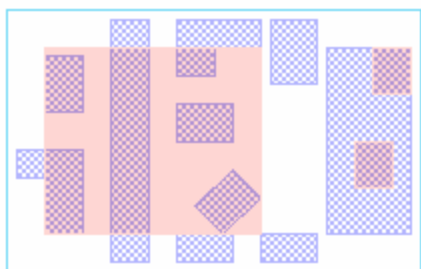
dLayer2 が断片だけの derived レイヤーである場合、その断片の内側・外側の方向はそのレイヤーが生成される前の元となったポリゴンの形状から引き継がれます。

## 関連位置による断片選択演算

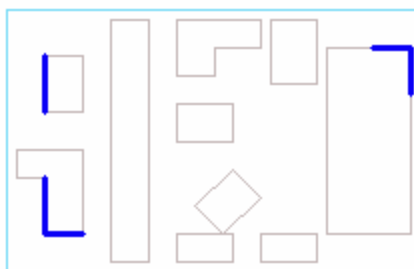
### 演算例

dLayer := DIaSelSeg dLayer1;dLayer2;"COINCIDENT"

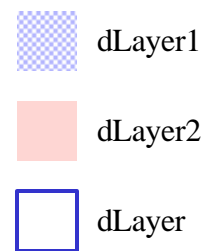
#### 例1 : Obvious



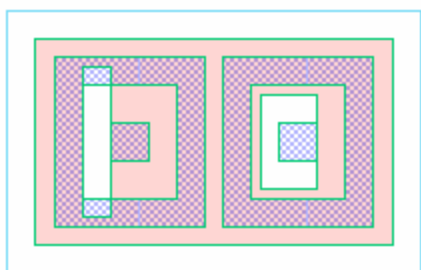
Input Layers



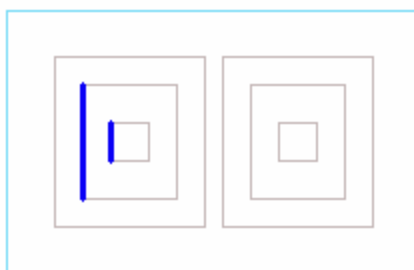
Selection Result



#### 例2 : Non Obvious



Input Layers



Selection Result

## 関連位置による断片選択演算

### 断片選択演算 Inside

Inside は dLayer2 の形状 (shape) のどれかが完全に含まれている dLayer1 の断片を選択するコマンドスイッチ (SEL\_OP) です。

### コマンド形式

```
dLayer := DiaSelSeg dLayer1;dLayer2;"INSIDE [INVERT]"
```

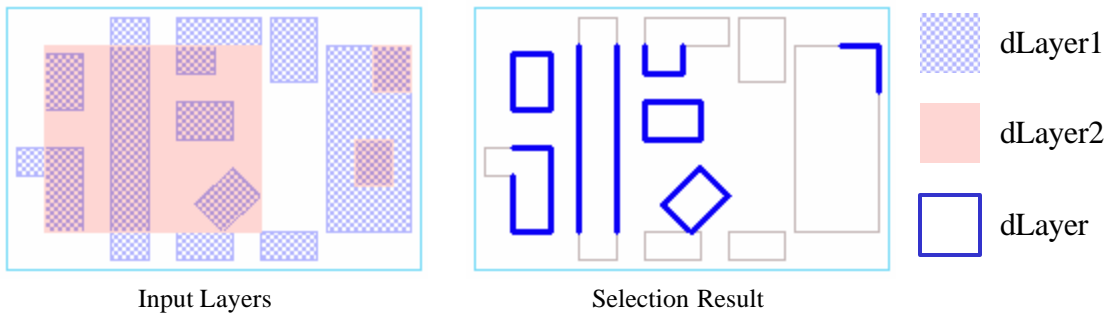
### 定義

dLayer2 のポリゴンのどれかが dLayer1 のポリゴンのどれかに完全に含まれてさえいれば選択されます。但し結果となる断片 (dLayer) は dLayer2 に含まれている部分だけであって、dLayer1 のポリゴンの断片全体と言うわけではありませんのでご注意ください。また断片同士や頂点角だけが重なっている場合は選択されません。

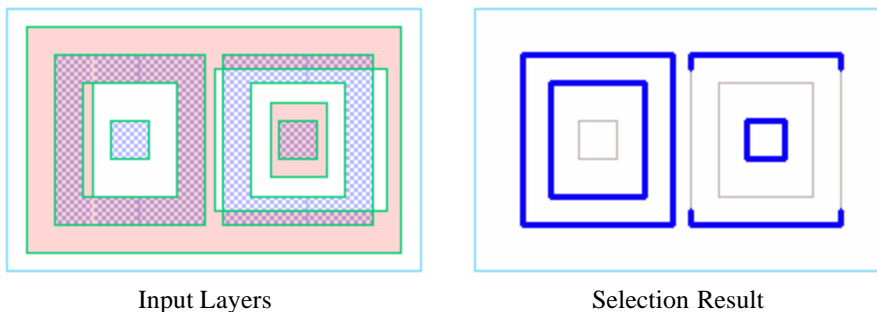
### 演算例

```
dLayer := DiaSelSeg dLayer1;dLayer2;"INSIDE"
```

#### 例1 : Obvious



#### 例2 : Non Obvious



## 関連位置による断片選択演算

### 断片選択演算 **Inside Only**

InsideOnly は dLayer2 の形状 (shape) のどれかが完全に含まれ、かつ断片同士が一切接していない dLayer1 の断片を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DLaSelSeg dLayer1;dLayer2;"INSIDEONLY [INVERT]"
```

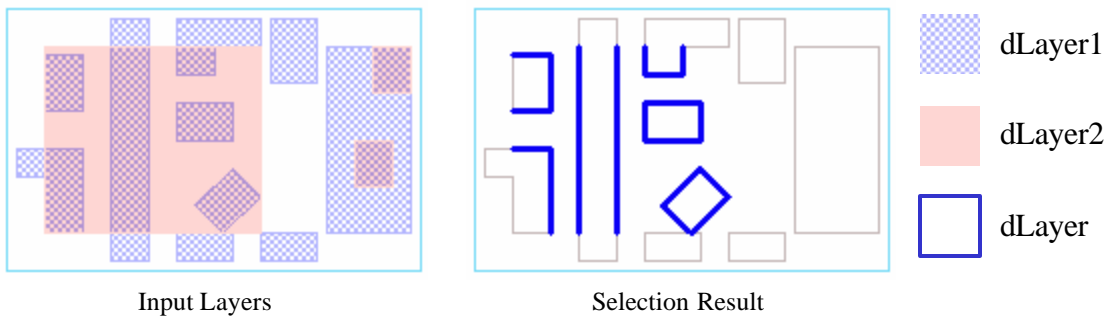
#### 定義

前項の Inside から断片同士が接しているものを除いた断片が選択されます。但し断片と頂点角だけが接してる場合は排除されず、そのまま選択されます。

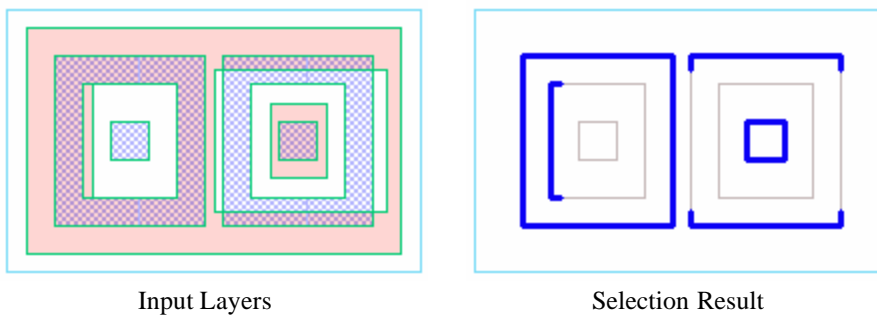
#### 演算例

```
dLayer := DLaSelSeg dLayer1;dLayer2;"INSIDEONLY"
```

##### 例1 : Obvious



##### 例2 : Non Obvious



## 関連位置による断片選択演算

### 断片選択演算 Outside

Outside は dLayer2 の形状 (shape) のどれにも全く重なっていない dLayer1 の断片を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DIaSelSeg dLayer1;dLayer2;"OUTSIDE [INVERT]"
```

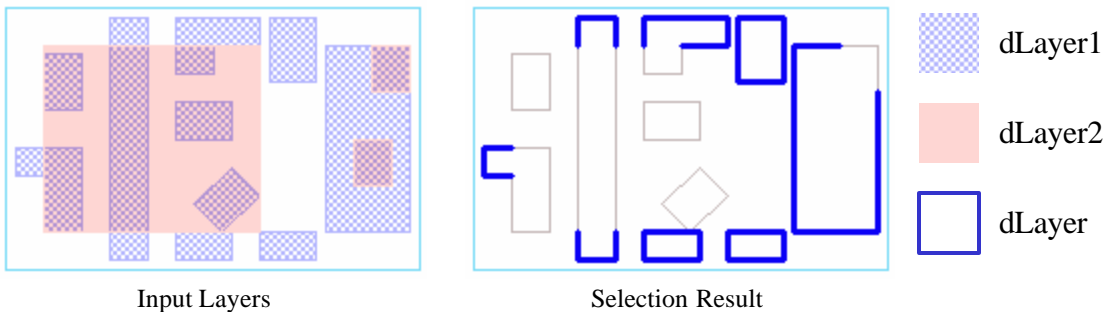
#### 定義

dLayer1 のポリゴンエリアで、dLayer2 のポリゴンのいずれにも全く重なっていないポリゴンエリアの断片が選択されます。頂点角や断片同士が重なっていても、ポリゴンエリアが重なっていない場合には選択されますのでご注意ください。

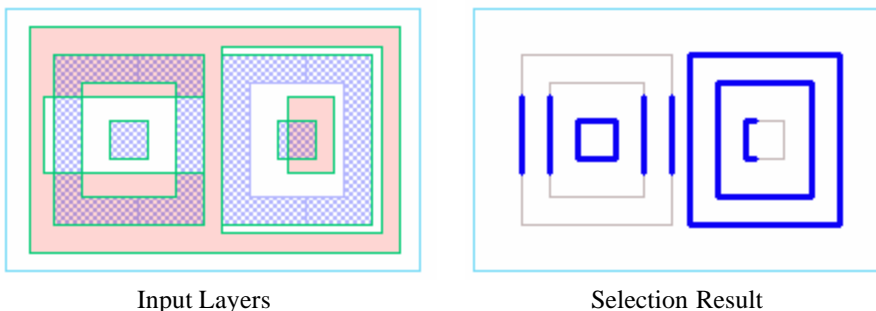
#### 演算例

```
dLayer := DIaSelSeg dLayer1;dLayer2;"OUTSIDE"
```

##### 例1 : Obvious



##### 例2 : Non Obvious





## 関連位置による断片選択演算

### 断片選択演算 Touching

Touching は dLayer2 の形状 (shape) の内側、外側問わず、断片同士が接している dLayer1 の断片を選択するコマンドスイッチ (SEL\_OP) です。

#### コマンド形式

```
dLayer := DLaSelSeg dLayer1;dLayer2;"TOUCHING [INVERT]"
```

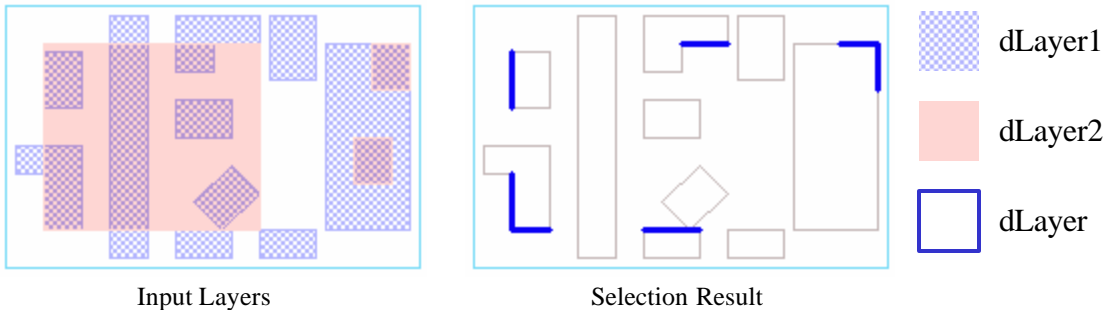
#### 定義

dLayer2 のポリゴンのどこかの断片が、dLayer1 のポリゴンのどこかの断片と接している部分が選択されます。ちょうど Butting と Coincident を合わせたもので、dLayer1、dLayer2 とともに断片の方向 (内側・外側) は問いません。

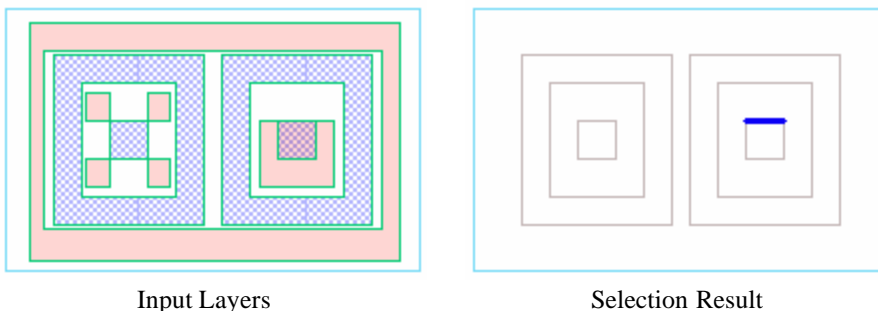
#### 演算例

```
dLayer := DLaSelSeg dLayer1;dLayer2;"TOUCHING"
```

##### 例1 : Obvious



##### 例2 : Non Obvious



# Using Net Functions

---

**ネット機能を使用するにはオプションのXDRCモジュールまたはHLVSモジュールが必要となります。**

本章では複数のレイヤーからなる形状 (shape) が持つ電気的な特性を表示する、ネット機能 (またはネット番号機能) について解説していきます。このネット機能は他の derived レイヤー機能同様、他の機能 (グループ演算及び選択演算) と統合して実行していくことが可能です。

ネットは異なるインプットレイヤーからなる形状 (shape) が相互に重なり合う時に、及びコモンエリアが接続レイヤー (または電氣的接続を表すレイヤー) からの形状 (shape) に重なることにより生成されます。

またネットは、電氣的に接続されているレイヤーから非接続レイヤーへ伝搬させることも可能です (スタンピング)。

## 本章での解説項目一覧 :

- 電氣的に接続しているレイヤー (Electrically Connecting Layers )
  - 順次接続レイヤー (Sequentially Connecting Layers )
  - 非順次接続レイヤー (Non-Sequentially Connecting Layers )
- レイヤーへネット番号を割り当てる
- ネットに名称を割り当てる
- ネットプロパティリストの作成
  - プロパティリストから計算を実行する
- ネットを使用している形状 (shape) を選択する

本章では上記に挙げた項目を次項以降一つずつ解説致しておりますが、その際に紹介している例の基本形となるセルは以下のフォルダに収録しております :

¥samples¥dla\_examples¥nets

それぞれの例に対するスクリプトも同フォルダ下の「scripts」フォルダ内に収録しております。お試しになりたい例にあわせてそれぞれ読み込み、実行してみてください。

## 電氣的に接続しているレイヤー

### 電氣的に接続しているレイヤー

#### Electrically Connecting Layers

コモンエリアを持つレイヤーから形成される形状 (shape) と言っても、それが自動的に共通の電気特性を持つと言うわけではありません。重なっている形状 (shape) 間での接続を確立するためには、接続レイヤーからの形状でそれらコモンエリアが重なり合わなくてはなりません。

本項では順次接続及び非順次接続の方法について解説して行きます。

#### 順次接続レイヤー (Sequentially Connecting Layers)

この方法は順次 (シーケンシャル) と呼ばれるもので、コマンド内で定義された順に接続していく方法です。この場合接続される条件があり、接続レイヤーと完全に重なり合っている形状 (接続レイヤーを完全に含む) が二つまで優先的に同じネット番号が割り当てられます。但し完全に重なり合っている形状 (エレメント属性名または derived レイヤー) が 3 つ以上ある場合には、**DlaConnect コマンドから先に定義された形状が優先**されることとなります。完全に重なり合っている形状が二つ定義された時点で、それ以降定義された形状は全て別のネット番号となってしまいます。

因みに一部しか重なり合っていない形状でも、完全に重なり合っている形状が二つ以上定義される前に定義されれば、それはその完全に重なり合っている形状と同じネット番号が割り当てられます。

レイヤーを順次接続 (または接続レイヤーを形成) する為に使用するコマンド形式は以下の通りです :

```
DlaConnect Layer1 ; Layer2..... ; LayerN ; LayerC
```

#### コマンドパラメータ

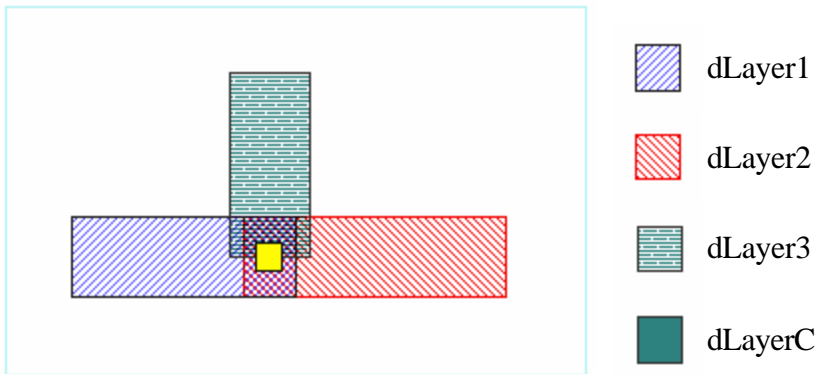
DlaConnect	順次接続に使用されるメインコマンドです。
Layer1...LayerN	接続される形状 (shape) となる derived レイヤーの GPE 変数名、またはエレメント属性 (Criterion) 名です。
LayerC	接続レイヤーです。電氣的な接続を表す形状を含むエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名となります。

次項以降で紹介しているスクリプト実行後に返されるネットプロパティリストは CLI (GPE:CLI) に表示されます。このリストの [ ] で区切られた範囲の表示される数値は、一番目がネット番号、二番目が入力レイヤーに関係した形状の周囲長となります。ネットプロパティリストに関しては 108 ページ「ネットプロパティリストの作成」の項も合わせてご参照下さい。

## 電氣的に接続しているレイヤー

### 演算例 1

```
DlaConnect dLayer1 ; dLayer2 ; dLayer3 ; dLayerC
```



### 演算結果

```
connect_seq1
dLayer1: <<Net property list (0x12480c0) [1, 23]>>
dLayer2: <<Net property list (0x124854c) [1, 26]>>
dLayer3: <<Net property list (0x1248558) [2, 20]>>
```

### 接続結果

dLayer1 の形状 : ネット番号 1  
 dLayer2 の形状 : ネット番号 1  
 dLayer3 の形状 : ネット番号 2

演算結果からdLayer3の形状はdLayer1及びdLayer2の形状と同じネットでは無いことが分かります。なぜならdLayer1の形状と接続レイヤーであるdLayerCの形状との重なり合っているエリアがdLayer2の形状と完全に重なり合っており、順次接続であることからdLayer1の形状とdLayer2の形状との接続が優先されるからです。

定義の順序を変えた場合 :

上記演算例を元に、例えば DlaConnect コマンド以下の dLayer1 ~ 3 の定義順を

```
DlaConnect dLayer1 ; dLayer3 ; dLayer2 ; dLayerC
```

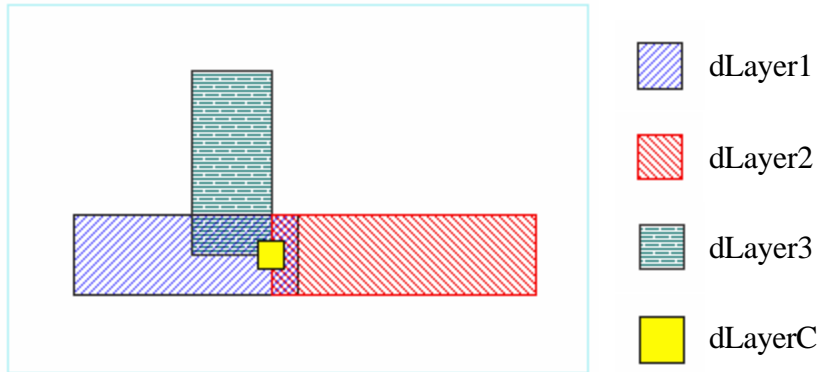
と変えて実行してみると、全ての形状にネット番号 1 が割り当てられます。つまり接続レイヤーを完全に含んだ形状が二つ出てくるまでは、一部しか重なっていない形状である dLayer3 にも一番目のネット番号が割り当てられるということになります。

では dLayer3 を接続レイヤーを完全に含むように変形し演算例 1 のスクリプトを実行したらどうなるでしょうか？結果は接続レイヤーを完全に含んだ形状である dLayer1・dLayer2 が既に先に定義されているため、dLayer3のみネット番号 2 番となります。

## 電氣的に接続しているレイヤー

### 演算例 2

```
DlaConnect dLayer1 ; dLayer2 ; dLayer3 ; dLayerC
```



### 演算結果

```
connect_seq2
dLayer1: <<Net property list (0x1248660) [1, 23]>>
dLayer2: <<Net property list (0x12480b4) [1, 26]>>
dLayer3: <<Net property list (0x124809c) [1, 20]>>
```

### 接続結果

dLayer1 の形状 : ネット番号 1  
 dLayer2 の形状 : ネット番号 1  
 dLayer3 の形状 : ネット番号 1

演算結果から dLayer1、dLayer2 及び dLayer3、全ての形状が同じネットであることが分かります。なぜなら：

- ・ dLayer1 と dLayer2 の形状が重なり合っているエリアが dLayerC の形状と重なっている。
- ・ dLayer1 と dLayer3 の形状が重なり合っているエリアが dLayerC の形状と重なっており、かつ dLayer2 の形状がこのコモンエリアと重なり合っていない。それ故 dLayer1 と dLayer3 との接続が確立されている。

## 電氣的に接続しているレイヤー

### 非順次接続レイヤー (Non-Sequentially Connecting Layers)

非順次 (ノンシーケンシャル) では順次の場合と異なり、定義されたレイヤーの順序は関係なく、入力レイヤー同士が重なり、その重なり合ったエリア (コモンエリア) が接続レイヤーの形状と重なり合っさえば、それらのレイヤーは接続されたものとされます。

レイヤーを非順次接続 (または接続レイヤーを形成) する為に使用するコマンド形式は以下の通りです:

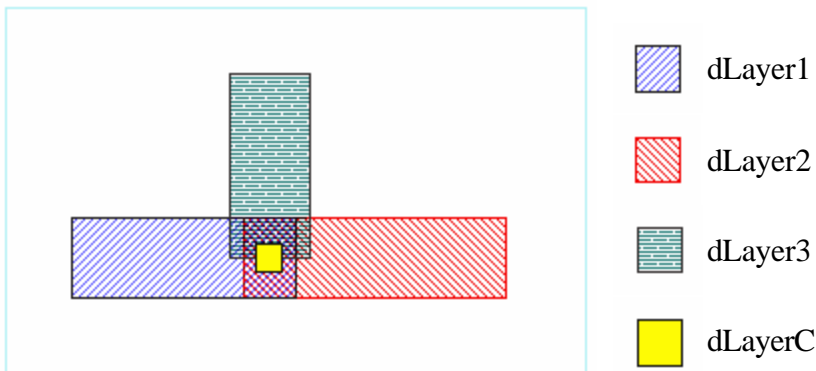
```
DlaConnectNonSeg Layer1 ; Layer2..... ; LayerN ; LayerC
```

#### コマンドパラメータ

DlaConnectNonseg	非順次接続に使用されるメインコマンドです。
Layer1...LayerN	接続される形状 (shape) となる derived レイヤーの GPE 変数名、またはエレメント属性 (Criterion) 名です。
LayerC	接続レイヤーです。電氣的な接続を表す形状を含むエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名となります。

#### 演算例

```
DlaConnectNonSeq dLayer1 ; dLayer2 ; dLayer3 ; dLayerC
```



#### 演算結果

```
connect_non_seq
dLayer1: <<Net property list (0x1248294) [1, 23]>>
dLayer2: <<Net property list (0x12481d4) [1, 26]>>
dLayer3: <<Net property list (0x124de94) [1, 20]>>
```

## 電氣的に接続しているレイヤー

---

### 接続結果

dLayer1 の形状 : ネット番号 1

dLayer2 の形状 : ネット番号 1

dLayer3 の形状 : ネット番号 1

定義順序、及び完全に重なっているか、一部だけ重なっているかに関係なく、dLayer1、dLayer2 及び dLayer3、全ての形状が同じネット番号となります。

## 電氣的に接続しているレイヤー

### レイヤーにネット番号を割り当てる

ネット番号は接続レイヤーから非接続レイヤーへと伝えることができます。通常非接続レイヤーが接続レイヤーと重なり合うと、接続レイヤーのネット番号が割り当てられることとなりますが、もしその非接続レイヤーが他の接続レイヤーと重なり合っており、既に別のネット番号を持っている場合には、ネット番号の競合と言う問題が生じてしまいます。このような場合非接続レイヤーの形状に、競合したネット番号を持っています、と言うフラグを持たせることができます。

レイヤーにネット番号を割り当てるコマンド形式は以下の通りです：

```
DlaAssignNet LayerAssign ; LayerNet
```

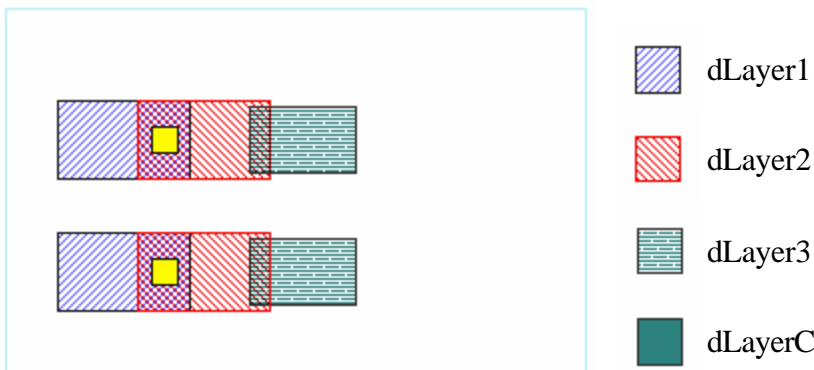
### コマンドパラメータ

DlaAssignNet	ネット番号割り当てに使用されるメインコマンドです。
LayerAssign	ネット番号を割り当てる derived レイヤーの GPE 変数名です。
LayerNet	LayerAssign に割り当てるネット番号を既に持っている接続レイヤーのエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名となります。

以下で紹介しているスクリプト実行後に返されるネットプロパティリストは CLI (GPE:CLI)に表示されます。このリストの「」で区切られた範囲の表示される数値は、一番目がネット番号、二番目が入力レイヤーに関係した形状の周囲長となります。ネットプロパティリストに関しては 108 ページ「ネットプロパティリストの作成」の項も合わせてご参照下さい。

### 演算例 1

```
DlaAssignNet dLayer3 ; dLayer2
```





## 電氣的に接続しているレイヤー

### 演算結果

```
assign1
dLayer1: <<Net property list (0x1248294) [1, 16][2, 16]>>
dLayer2: <<Net property list (0x124dc0c) [1, 16][2, 16]>>
dLayer3: <<Net property list (0x124e008) [1, 13][2, 13]>>
```

### 接続結果

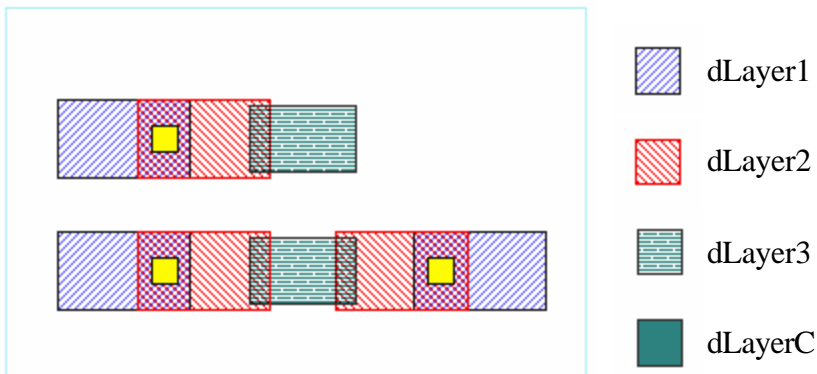
dLayer1 の形状 1 及び 2 : ネット番号 1 及び 2

dLayer2 の形状 1 及び 2 : ネット番号 1 及び 2

dLayer3 の形状 1 及び 2 : ネット番号 1 及び 2

### 演算例 2

```
DiaAssignNet dLayer3 ; dLayer2
```



### 演算結果

```
assign2
dLayer1: <<Net property list (0x12482ac) [1, 16][2, 16][3, 16]>>
dLayer2: <<Net property list (0x1248444) [1, 16][2, 16][3, 16]>>
dLayer3: <<Net property list (0x1248438) [1, 13][-2, 13]>>
```

### 接続結果

dLayer1 の形状 1、2 及び 3 : ネット番号 1、2 及び 3

dLayer2 の形状 1、2 及び 3 : ネット番号 1、2 及び 3

dLayer3 の形状 1 及び 2 : ネット番号 1 及び -2 (2 番と 3 番が競合)

## 電氣的に接続しているレイヤー

---

### ネットに名称を割り当てる

接続レイヤーの形状に重なり合っているのがテキスト要素を含む形状である場合、そのテキストをネットに名称として割り当てることができます。またそのようにして名称を持たせたネットの名称を他のネットに伝えることも可能です。因みにネット番号同様、名称も競合する場合がありますが、このような場合ネット番号の時と同様にフラグを持たせることができます。

ネットに名称を割り当てるコマンド形式は以下の通りです：

```
DlaAssignNetName LayerNet ; LayerLabel
```

### コマンドパラメータ

DlaAssignNetName	ネットに名称を割り当てるのに使用されるメインコマンドです。
LayerNet	名称を割り当てるネット番号を既に持っている接続レイヤーの要素属性名、またはユーザー定義された derived レイヤーの GPE 変数名となります。
LayerLabel	ネット名に使用されているテキスト要素を含む形状の要素属性名、またはユーザー定義された derived レイヤーの GPE 変数名となります。

## 電氣的に接続しているレイヤー

### ネットプロパティリストの作成

ネットに含まれる形状の具体的なプロパティリストを作成することができます。

プロパティリスト作成のコマンド形式は以下の通りです：

```
PropList := DlaGetNetProperty Layer ; "PROPERTY" ; ["OPTION"]
```

#### コマンドパラメータ

PropList	プロパティリストの内容が返される GPE 変数です。ここでは便宜上 PropList としておりますが、実際の構文ではユーザー定義の任意の変数または「=」以下の構文をそのまま他の構文に代入することも可能です。
DlaGetNetProperty Layer	ネットプロパティリスト作成に使用するメインコマンドです。 ネットレイヤーのエレメント属性名、またはユーザー定義された derived レイヤーの GPE 変数名となります。
PROPERTY	返される値の種類です。以下の四種類があります： <ul style="list-style-type: none"> <li>• AREA Layer の面積を返します。</li> <li>• PERIMETER Layer の周囲長を返します。</li> <li>• LENGTH 断片 (segment) derived レイヤーのみに適用できるコマンドで、Layer の接続断片の合計の長さを返します。</li> <li>• COUNT_SHAPE Layer に含まれる形状 (shape) の数を返します。</li> </ul>
OPTION	ネット上に複数の形状がある時に利用するオプションで、AVG (平均)、SUM (合計)、MIN (最小)、MAX (最大)の四種類から選択できます。COUNT_SHAPE 以外の PROPERTY に併用することができます。因みにデフォルト(null - 指定なし)では SUM が適用されます。

#### 返される値について

二種類の値が返されます。一番目にそのネットが属するネット番号を、そして二番目に PROPERTY で指定した種類の値が返されます。

## 電氣的に接続しているレイヤー

---

### プロパティリストから計算を実行する

プロパティリストに返された値をそのまま計算に使用し、新たなプロパティリストを作成することができます。計算の方法は至ってシンプルで、プロパティリストの計算式をそのまま (またはプロパティリストに使用されている GPE 変数) に演算子と値を併記するだけです。

プロパティリストの計算は以下のいずれかの並びとなります：

`NewPropList := PropList OPERATOR Value`

または：

`NewPropList := Value OPERATOR PropList`

または：

`NewPropList := PropList OPERATOR PropList`

### コマンドパラメータ

NewPropList	計算の結果、新たに作成されるプロパティリストです。
PropList	計算に使用するプロパティリストの計算式、またはプロパティリストに使用されている GPE 変数 (作成済みのプロパティリスト) です。
Value	計算に使用する任意の数値です。
OPERATOR	計算に使用する演算子です。演算子には + (加算)、- (減算)、* (乗算)、/ (除算) のいずれかが入ります。

### 返される値について

前項で記載した値同様、ネット番号と計算された値の二つの値が返されます。

## 電氣的に接続しているレイヤー

### ネットを使用している形状 (shape) を選択する

DlaSelShape 機能とネット機能を融合したようなもので、ネット同士の選択演算を行うことができます。

ネットの選択演算のコマンド形式は以下の通りです：

```
dLayer := DlaSelShapeByNet dLayer1;dLayer2;"[Val1 CO1]SEL_OP[CO2 Val2] [INVERT]"
```

#### コマンドパラメータ

dLayer	選択演算の結果となる derived レイヤーです。
DlaSelShapeByNet	ネット選択演算のメインコマンドです。
dLayer1	ユーザー定義されたネット選択される接続 derived レイヤーの GPE 変数名です。
dLayer2	dLayer1 の抽出形状 (shape) の位置に関連して使用されるユーザー定義された derived レイヤーの GPE 変数名です。
SEL_OP	関連位置による選択演算のコマンドスイッチです。詳しくは 32 ページからの「関連位置による形状 (shape) 選択演算」の項をご参照下さい。
Val1 及び Val2	抽出する形状 (shape) の数量を条件付けする為のパラメータです。
CO1 及び CO2	Val1 及び Val2 で定義された数値につく等式、不等式です。つまりこのパラメータは > ・ >= ・ = ・ < ・ <= のいずれかとなります。
	<b>この数値定義オプションは全ての演算に使用されるわけではございません。また使用できる演算でも必ずしも定義しなければいけないということもありません。あくまで条件が必要な時のみ使用します。</b>
INVERT	指定した選択演算と逆の結果を返します。

## 電氣的に接続しているレイヤー

### 演算例

```
dLayer := DiaSelShapeByNet dLayer3 ; dLayer2 ; "COINCIDENT"
```

本例では dLayer2 の形状 (shape) の断片と接している断片を持ち、かつポリゴンエリアが重なっている (内側同士の断片が重なり合っている) dLayer3 の形状 (shape) が選択されています。



Input

Result



dLayer1



dLayer2



dLayer3



dLayerC



dLayer



*dw-2000 Derived Layers*  
Copyright by  
**2004 MEDIX INTERNATIONAL CORP.**  
*May 2004, Printed in Japan*