

## DFM Property (partial syntax)

**DFM PROPERTY** *primary\_layer* [*secondary\_layer* ...]  
 [INTERSECTING | {OVERLAP [ABUT ALSO [SINGULAR]]  
 [EXACT] [MULTI] [REGION] [UNMERGED]] | {NODAL [MULTI]]}  
 [PRESERVE] [CORNER] [CONNECTED | NOT CONNECTED]  
 ["*expression*"] ["!"*constraint*] ...

Measurement functions are summed over secondary layer objects unless in the scope of a per-shape function.

## DFM RDB (partial syntax)

**DFM RDB** *layer1 file\_name* [*layer2* ...]  
 [NOPSEUDO] [NOEMPTY] [ALL CELLS] [SAME CELL]  
 [RESULT CELLS *cell\_list*] [CHECKNAME *name*]  
 [COMMENT "*comment\_string*"]

## DFM Function

**DFM FUNCTION** ["*function\_name* '("*type argument* ["*type argument* ..."]'  
 { *expression* |  
 TABLE [LINEAR | SPLINE] '{*input1 result1* [*input2 result2*] }'  
 TVF\_NUMBER\_FUNCTION("*tcl\_proc*","*TVF\_function*")' }  
 ""]

where *type* is one of (partial list):

- NUMBER – A numeric value.
- LAYER – A string representing a layer name.
- STRING – A string representing a property name.

## DFM Copy (partial syntax)

**DFM COPY** *layer1* [*layer2* ...] [REGION [UNMERGED] | EDGE |  
 CENTERLINE | TIE [CENTER] | UNMERGED] [CELL LIST *list*]

## DFM Space

**DFM SPACE** *layer1* [*layer2*] *distance\_constraint*  
 { BY EXT | BY INT | BY ENC | ALL }  
 [HORIZONTAL VERTICAL | HORIZONTAL | VERTICAL]  
 [COUNT *shielding\_constraint*] [BY LAYER *layer*]  
 [MEASURE ALL] [[NOT] CONNECTED] [GRID *step*]

## DFM Stamp

**DFM STAMP** *layer1* BY *netID\_property*

## DFM Defaults (partial syntax)

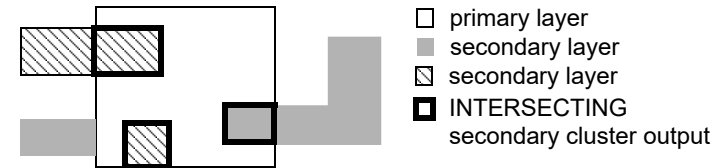
**DFM DEFAULTS** RDB [FILE *filename* | NOFILE]  
 [NOPSEUDO [NULL]] [NOEMPTY [NULL]]  
 [OUTPUT [NULL]] [ALL CELLS [NULL]]  
 [RESULT CELLS {*cell\_list* | NULL}]  
 [CHECKNAME {*check\_name* | NULL}]

## Math Operators

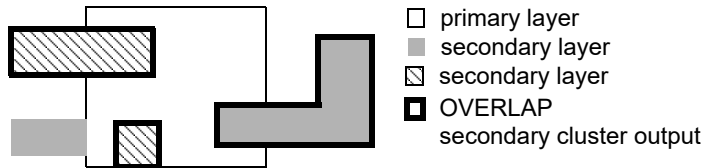
Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Power
!	Returns 1 if its argument is 0, and 0 otherwise.
&&	Both conditions must be true (AND). Valid only inside conditional expressions.
	At least one condition must be true (OR). Valid only inside conditional expressions.
~	Returns 1 (true) if its argument is non-positive; returns 0 (false) otherwise.

## DFM Property Clustering Modes

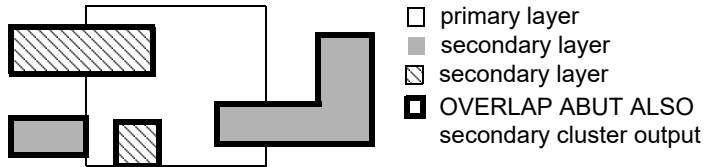
**INTERSECTING** – Default; valid only with polygon layers. Each cluster contains one primary layer polygon plus all secondary layer polygons that overlap it. Overlapping secondary layer polygons are clipped by the primary layer polygon.



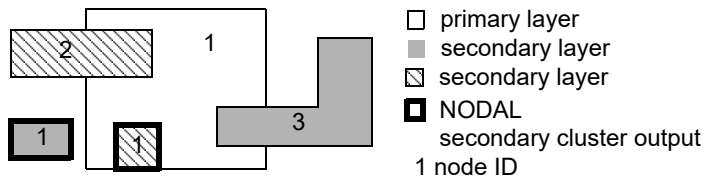
**OVERLAP** – Valid for all layer types. Each cluster contains one layout object on the primary layer plus all layout objects on secondary layers that overlap the primary layer object.



**OVERLAP ABUT ALSO** – Valid for all layer types. Each cluster contains one layout object on the primary layer plus all layout objects on secondary layers that overlap or abut the primary layer object.



**NODAL** – Valid only with polygon or edge layers. Each cluster contains one layout object on the primary layer plus all layout objects on secondary layers that have the same node number as the primary layer object. Spatial relationships between layout objects in a cluster are ignored.



## Property Access and Calculation

Access property values:

- Numeric – PROPERTY(*layer,property\_name*[,*ordinal*])
- Vector numeric – VPROPERTY(*layer,property\_name*[,*ordinal*])
- Net number – NETPROPERTY(*layer,property\_name*[,*ordinal*])
- Vector net number – NETVPROPERTY(*layer,property\_name*[,*ordinal*])
- String – SPROPERTY(*layer,property\_name*[,*ordinal*])
- Vector string – SVPROPERTY(*layer,property\_name*[,*ordinal*])

Per-shape handling of secondary layer objects: SUM(), PROD(), MIN(), MAX(), VECTOR()

Create net number properties:

- NETID(*layer*[,*ordinal*]) – Returns the node number associated with a geometry on layer as property type netID. The ordinal number can be specified if layer is a secondary layer.
- VNETID(NETID(*layer*)) – Creates a vector of netID values.

## Selected Math Functions

- **ABS(x)** – Absolute value of x.
- **CEIL(x)** – Returns x rounded up to the nearest integer  $\geq x$ .
- **COS(x)** – Cosine of x (radians).
- **E()** – Returns the constant value of e.
- **EXP(x)** – Exponential (base e) of x.
- **FLOOR(x)** – Returns x rounded down to the nearest integer  $\leq x$ .
- **FMAX(x,y)** – Returns the larger value of x and y.
- **FMIN(x,y)** – Returns the smaller value of x and y.
- **FMOD(x,y)** – Returns the remainder of x/y.
- **LOG(x)** – Natural logarithm of x.
- **LOG10(x)** – Computes the base-10 logarithm of x.
- **PI()** – Returns the constant value of pi.
- **POW(x,y)** – Computes x to the power of y ( $x^y$ ).
- **REMAINDER(x,y)** – Returns the remainder  $x - (n*y)$ , where n is the nearest integer to the exact value of  $x/y$ .
- **RINT(x)** – Returns x rounded to the nearest integer.
- **ROUND(x,y)** – returns x rounded to the nearest value  $n*y$ , where n is an integer.
- **SIN(x)** – Sine of x (radians).
- **SQRT(x)** – Square root of x.
- **TAN(x)** – Tangent of x (radians).
- **TRUNC(x)** – Truncate (round to nearest integer less than x).

## Comparison Functions with Built-In Tolerance

**DRC\_EQ**, **DRC\_GE**, **DRC\_GT**, **DRC\_LE**, **DRC\_LT**, and **DRC\_NE** perform absolute and relative comparisons with a tolerance of  $10e-8$ . They return a floating point value of 1.0 or 0.0. For example **DRC\_EQ(x, y)** returns 1.0 if x and y are equal within the tolerance.

## Expression Chains

Formed by using multiple expressions with the same property name inside a DFM Property operation. The property value is the result of the first expression that is evaluated successfully.

## Non-Persistent Properties

Properties named "+", "<+name>", or "-" are evaluated and tested against constraints, but are not saved to the output layer.

"+" and "<+name>" – Enables chaining (the first successful evaluation of an expression for a property starting with "+" prevents subsequent expressions for the property from being evaluated and tested against constraints).

"-" – Does not allow chaining (all expressions for property "-" are evaluated and tested against constraints; enables successive filtering of geometries).

## Conditional Expressions

*(condition) ? result\_if\_true : result\_if\_false*

where *condition*, *result\_if\_true*, and *result\_if\_false* are valid DFM expressions. The *condition* expression must include a relational operator (<, <=, ==, >=, > or !=). Note: spaces are required around "?" and ":-".

## Vectors

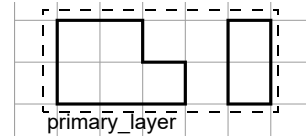
	Vectors of Numbers	Vectors of NetIDs
Create	VECTOR()	VNETID()
Access	VPROPERTY()	NETVPROPERTY()
Concatenate	CONCAT()	CONCAT()
Evaluate	VMIN(), VMAX()	VNETIDMIN(), VNETIDMAX()
Compare	==, !=, <, <=, >, >=	==, !=

## Measurement Functions (with secondary layers)

**AREA(polygon\_layer<sup>1</sup>)** = 7

**COUNT(any\_layer)** = 2

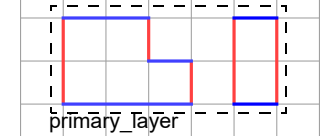
<sup>1</sup> Valid also for error layers.



**PERIM(polygon\_layer)** = 16

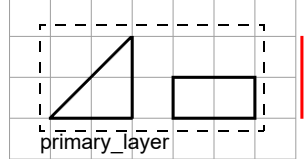
**PERIMX(polygon\_layer)** = 8

**PERIMY(polygon\_layer)** = 8



**PERIMXP(polygon\_layer)** = 8

**PERIMYP(polygon\_layer)** = 6

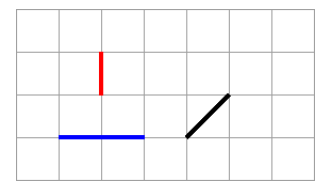


Returns x and y projections of polygon edges.

**LENGTH(edge\_layer)** = 1.414

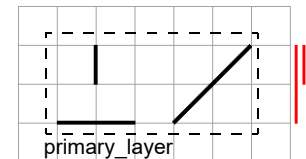
**LENGTHX(edge\_layer)** = 2

**LENGTHY(edge\_layer)** = 1



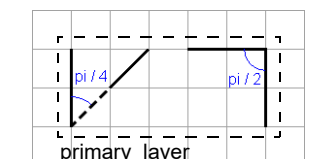
**LENGTHXP(edge\_layer)** = 4

**LENGTHYP(edge\_layer)** = 3



Returns x and y projections of edges.

**ANGLE(error\_layer)** =  $(3/4)*\pi$  = 2.36

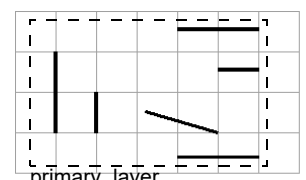


Returns angle between edge pairs in radians. Must be used in a per-shape function.

**EC(error\_layer)** = 3

**ECX(error\_layer)** = 2

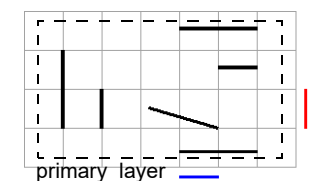
**ECY(error\_layer)** = 1



Edge pair projection length.

**ECXP(error\_layer)** = 2

**ECYP(error\_layer)** = 1

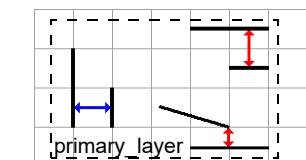


Edge pair EC projected onto an axis..

**EW(error\_layer)** = 2.5

**EWX(error\_layer)** = 1

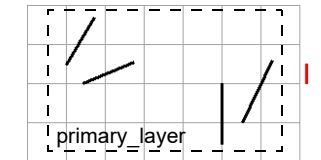
**EWY(error\_layer)** = 1.5



Edge pair minimum separation distance.

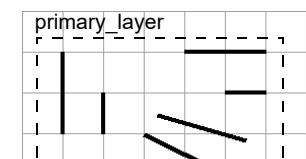
**EWXP(error\_layer)** = 1

**EWYP(error\_layer)** = 0.5



Edge pair projected minimum separation distance.

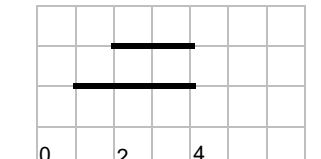
**EWP(error\_layer)** = 2



Edge pair minimum separation distance between parallel edge pairs.

**ECMIN(primary\_error\_layer<sup>1</sup>)** = 2

**ECMAX(primary\_error\_layer<sup>1</sup>)** = 4



Smaller or larger x or y coordinate value of EC measurement. Must be used within a per-shape function for a secondary layer.

<sup>1</sup> Valid also for edge layers.