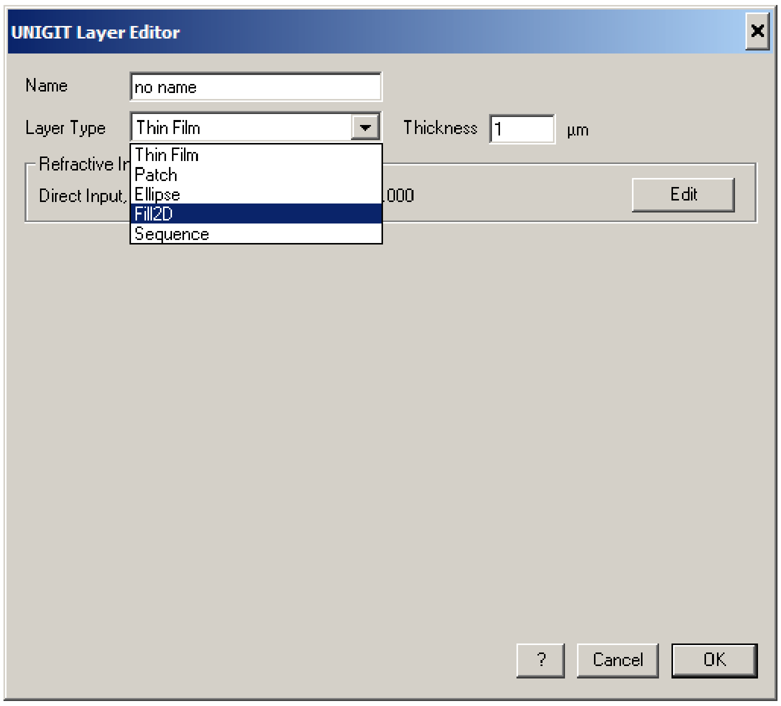
2D Arbitrary Filling in Unigit

Unigit version 1.02.02 offers a new feature for arbitrary pixel filling of a 2d layer. The following gives a short description how to apply this new layer type. In the stack editor select the layer type Fill2D as shown below.


UNIGIT Layer Editor

Name: no name

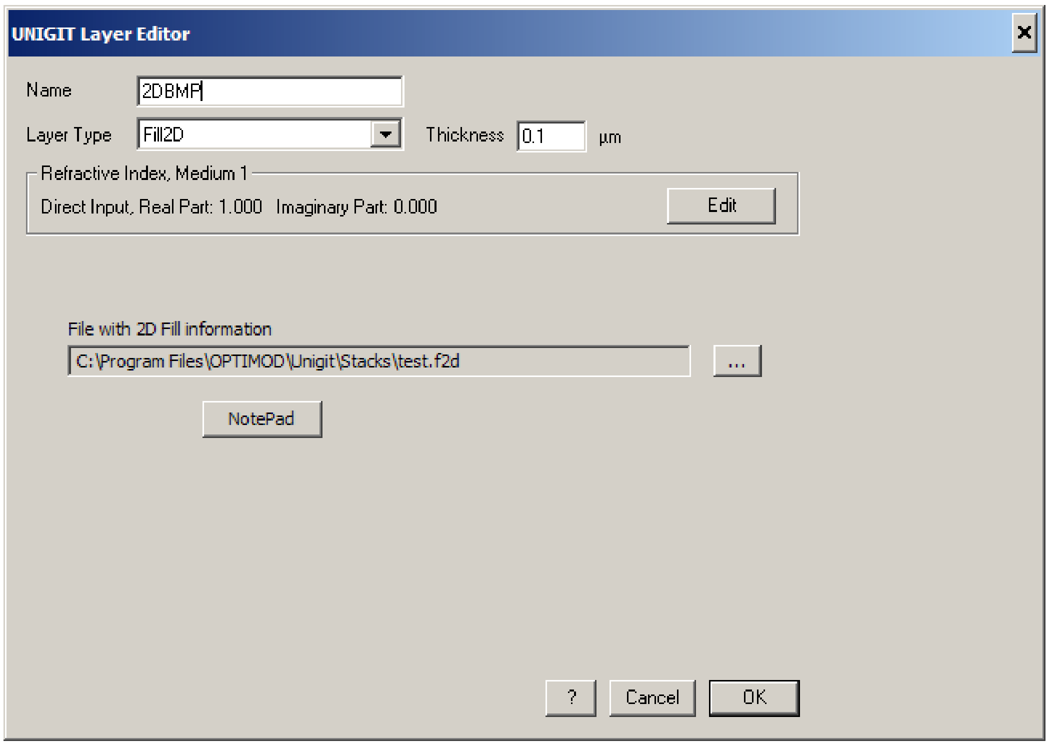
Layer Type: Thin Film

Thickness: 1 µm

Refractive In…

Direct Input, … .000

Then, the layer editor opens with the following window:


UNIGIT Layer Editor

Name: 2DBMP

Layer Type: Fill2D

Thickness: 0.1 µm

Refractive Index, Medium 1

Direct Input, Real Part: 1.000 Imaginary Part: 0.000

File with 2D Fill information: C:\Program Files\OPTIMOD\Unigit\Stacks\test.f2d

Here, you need to specify the layer thickness and you can enter a name for the layer similar to other layer types. Moreover, you need to specify a material for the embedding medium and a file (with the extension .f2d) which contains the bitmap description of the layer.

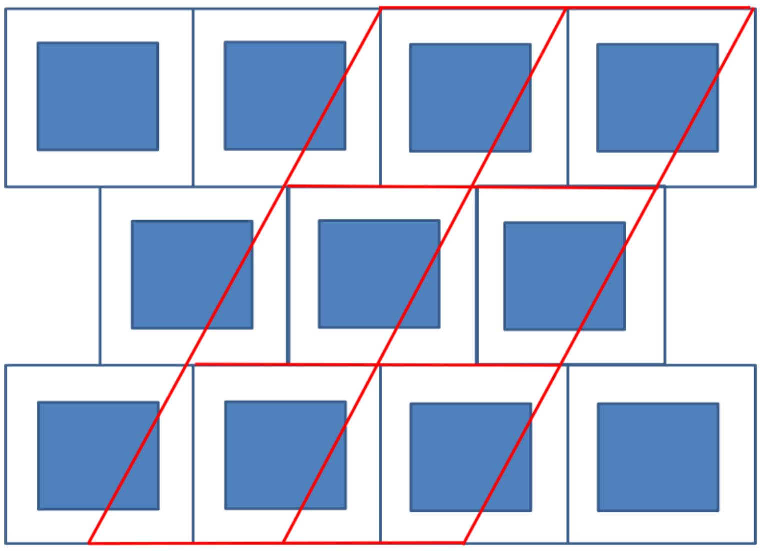
The file has to be specified as follows:

```
2               /number of materials beside the embedding material

0               /specification ID for material 1 (0 = direct input)

1.5 0.0         /n&k for material 1

1               /specification ID for material 2 (1 = n&k file)

'C:\Program Files\OPTIMOD\Unigit\NKData\SILICON.NK'          /pathname of the n&k file for material 2

6   97          /Fourier discretization power (e.g., discretization is 2^6) & number of lines to follow

528   0         /pixel number & material ID, e.g., 528 pixels with material 0 (embedding material)

32   1          /pixel number & material ID, e.g., 32 pixels with material 1

32   2          /pixel number & material ID, e.g., 32 pixels with material 2

……..

32   1          /pixel number & material ID, e.g., 32 pixels with material 1

528   0         /pixel number & material ID, e.g., 528 pixels with material 0 (embedding material)
```

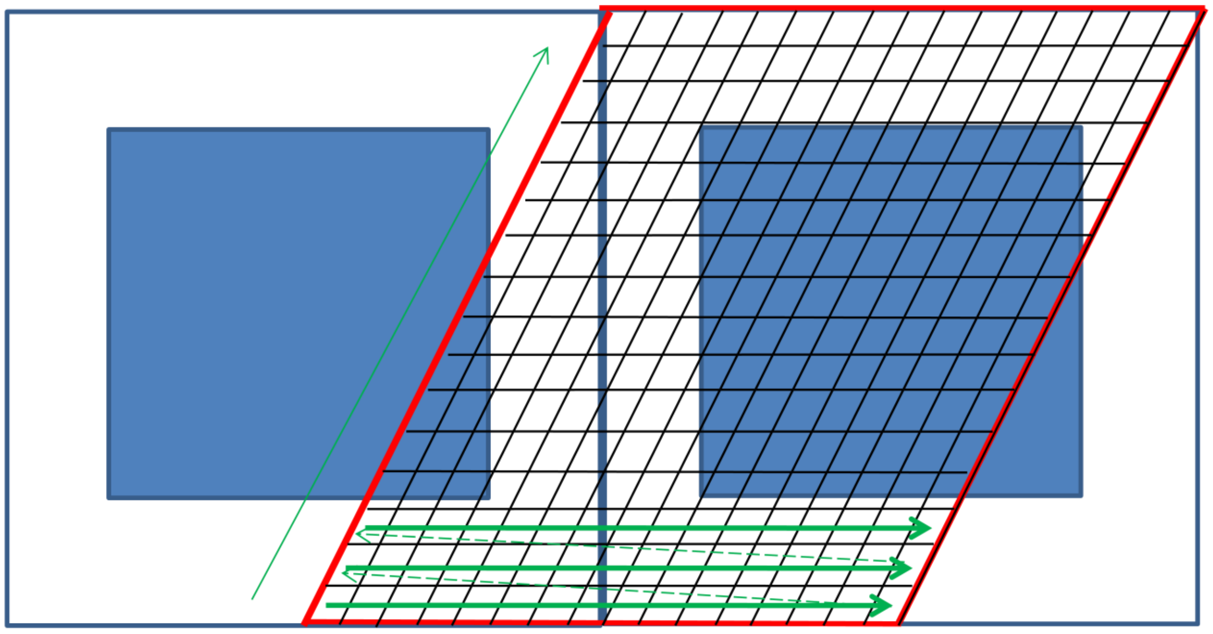
Note, the total number of pixel description lines in this example must be 97 and the total number of added pixel numbers (i.e., 528 + 32 + 32 + … + 32 + 528) must be equal to (2^6)^2 = 4096

A graphical example for a non-orthogonal cell is shown below.

First, the elementary cell has to be defined as shown.



As a next step, one has to do the discretization in unit cell coordinates. The linear pixel number has to be a power of 2 (64, 128, 256 etc.). The pixel number in both axes has to be the same (no matter what the pitch is).



The counting direction is indicated by the green arrows, i.e., scanning row by row with blanking like behaviour (jump from the end of one row to the begin of the next row) as shown. As long as pixels have the same material (defined by largest area percentage) just add, when material is changing -> save the value and restart counting (across scan blanking).

This would result in the following .f2d information for the graphical example assuming material 0 is white and material 1 is blue:

48 0 (3 lines with 16 pixel each)

3   1 (3 pixels blue)

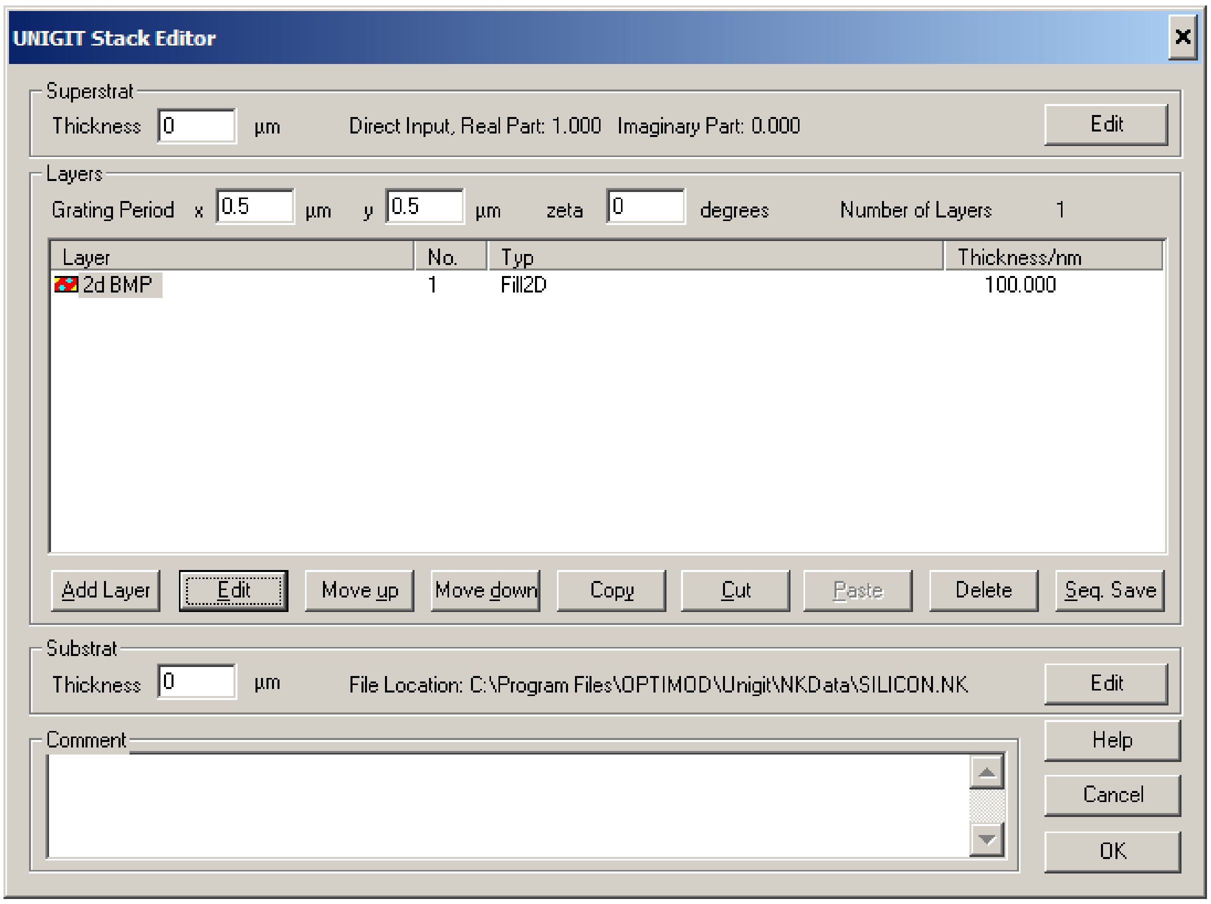
6   0 (6 pixels white)

10 1 (7 pixels in row #4 right hand and 3 pixels in row #5 left hand, counting across the blanking)

5   0 (5 pixels white)

etc.

Finally, the layer shows up in the stack editor as shown below:


UNIGIT Stack Editor

Superstrat
Thickness 0 µm   Direct Input, Real Part: 1.000 Imaginary Part: 0.000

Layers
Grating Period x 0.5 µm y 0.5 µm zeta 0 degrees   Number of Layers 1

Layer	No.	Typ	Thickness/nm
2d BMP	1	Fill2D	100.000

Substrat
Thickness 0 µm   File Location: C:\Program Files\OPTIMOD\Unigit\NKData\SILICON.NK

Comment

The installation of version 1.02.02 comes with 3 stack files that describe all the same geometry (a rectangular area with n = 1.5 embedded in vacuum with n = 1) but with different means:

- test_2dfill1.txt using the ellipse descriptor
- test_2dfill2.txt using the new bitmap descriptor
- test_2dfill3.txt using the patch descriptor

Running these files should give very similar results.