dilate

keys:	**[from]**	*<number>*	source picture
	[to]	*<number>*	output picture
	times	*<number>*	force repeat count for dilation process
	neighbours	*<number>*	only dilate points with specified number of clear neighbours
	with	*<number>*	picture containing user-supplied map functions
options:	**separately**		dilate objects whilst preserving separation (via 4-connected skeleton background)

Use **dilate** to process a binary picture, by adding a one-pixel border to objects made up of non-zero pixels. **dilate** can also be used to clean up an object by filling holes and points in an object without affecting its real edge. Note that zero value pixels are referred to as *clear* or *background* pixels and non-zero value pixels as *set* or *object* pixels.

Examples

```
dilate display
```

This command adds a one-pixel wide border to all objects (by setting clear pixels that have a set neighbour).

```
dilate 50 to 51 times 3
```

This command adds a three-pixel border to all objects (that is, it repeats the **dilate** command three times).

```
dilate neighbours 4
```

This command adds pixels at sites that have at least four set neighbours and is used to fill points and line holes in objects without affecting their 'real' edge.

```
dilate neighbours 8
```

This command fills isolated point holes only.

```
dilate separately
```

This command dilates objects but preserves the 4-connectivity of the background, so that objects and object points that are originally separated remain so.

dilate

```
stack 3, 6 to 7, dilate 50 with 7
```

This command applies the maps in pictures 3 to 6 in turn, to dilate objects in picture 50.

Description

The **dilate** command is part of the *morphology* group of commands that perform morphological (shape) operations on binary pictures. Other morphology commands include **analyse**, **erode** and **median**. You use **dilate** on binary pictures, that is, pictures that have only two classes of value:

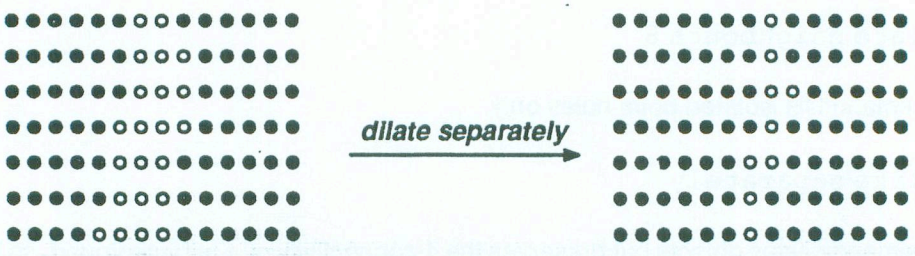
- zero for background pixels
- non-zero for pixels belonging to the shape (object) of interest

Note that as Semper does not actually store pixels in single bit form, **dilate** distinguishes between zero (*clear*) and non-zero (*set*) pixels instead. You can translate a picture into binary form using the **calculate** command, for example, **calculate :51>20** or **calculate :51<thr**.

By default, **dilate** adds a one-pixel wide border around pixels that are set. You can use the **times** key to specify the number of times the **dilate** command is repeated and so determine the width of the border. If you set **times** to zero, **dilate** repeats itself until the picture stabilizes (or until you *abandon* the operation). This is faster than putting **dilate** in a **for** loop, as the command is able to omit picture rows that have already stabilized in previous passes.

By default, **dilate** sets pixels with at least one set neighbour. You can change this minimum number using the **neighbours** key and perform more and more selective dilation as you increase it towards the maximum of 8. The default mode of **dilate** is in fact **dilate neighbours 1**. The command **dilate neighbours** is exactly equivalent to **erode neighbours**, with the object and background interchanged. You may find it helpful to look at the **erode neighbours** examples given in this manual.

The **dilate separately** command dilates objects but preserves the 4-connectivity of background, so that objects and object points that are initially separated remain so. (Refer to *Appendix G, Pixel Connectivity* for an explanation of 4 and 8 connectivity). The diagram given below illustrates how the command **dilate separately** works.



dilate

The **dilate separately** command dilates objects in sequences of four passes, in which pixels are removed, without affecting the 4-connectivity of the background, from the top, the bottom, the left and the right in turn. By default, the sequence is repeated until the picture stabilizes, but you can limit the number of sequences if you wish, using the **times** key.

Notes

see also: **analyse, calculate, erode, median**

Defaults and Ranges

keys/options	defaults	range
[from]	current picture, held in the variable *select*	valid picture number
[to]	source picture	valid picture number
times	1	zero or positive integer
neighbours	1 pixel	integer in range 1 to 8
with	*none*	valid picture number