# Scalable In-Line EPS Graphics in `groff`

The macro **`dospark`** is defined as follows:

```
.de dospark
.psbb \\$1
.nr ht0 \\n[ury]-\\n[lly]
.nr wd0 \\n[urx]-\\n[llx]
.nr deswd (\\n[.ps]/\\n[ht0])*\\n[wd0]
.if \\$2&(\\$2>0) .nr deswd (u; \\$2p)
.nr desht \\n[.ps]
.if \\$3 .nr desht (u; \\$3p)
.nr xht 0
.if (\\n[desht]>\\n[.ps]) .nr xht \\n[desht]-\\n[.ps]
\X'ps: import \\$1 \
 \\n[llx] \\n[lly] \\n[urx] \\n[ury] \
 \\n[deswd] \\n[desht]'\
\h'\\n[deswd]u'\x'-\\n[xht]u'
..
```

**Usage:**

```
.dospark EPSfilename [deswid [desht]]
```

with two optional arguments:

**`deswid`** is the desired width of the graphic in **points** (default)
**`desht`** is the desired height of the graphic in **points** (default)

If given simply as numbers, the size-units are taken to be **points**. A **`groff`** scaling factor (e.g. "**`m`**") can be appended to either.

If **`desht`** is missing, then the height of the graphic will be the current point size (**`1m`** in **`groff`** units).

If **`deswid`** is ***also*** missing, then the width of the graphic will be such as to preserve the aspect-ratio of the original.

If **`desht`** is present (to set a custom height), and it is desired to preserve the aspect ratio, then use **`0`** for **`deswid`** (if **`desht`** is present then **`deswid`** must also be present; otherwise **`desht`** will be interpreted as **`deswid`**).

When **`desht`** is present, and greater than the current point size, extra vertical space is automatically added above the current line to make room for the graphic. The graphic's baseline (bottom of the Bounding Box) will always be aligned with the baseline of the current text line.

**Some Examples of Usage**

The following illustrations use an EPS file

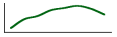
```
spark.average.fund.size.eps
```

to illustrate variants of the usage of the macro **`dospark`**.

***Example A, default usage with no arguments***

```
Using 2002\^\[en]\^2009, the numbers tell
us that managers of university endowments,
pension funds, and other institutional
investors are bailing out of venture pools
and consequently the average venture fund
is quickly getting smaller
.dospark spark.average.fund.size.eps
a trend even more dramatic when one
confines one's view to just the size of
new funds raised.
```

**The above input produces the following output:**

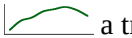
Using 2002–2009, the numbers tell us that managers of university endowments, pension funds, and other institutional investors are bailing out of venture pools and consequently the average venture fund is quickly getting smaller  a trend even more dramatic when one confines one's view to just the size of new funds raised.

***Example B, changing the width only***

To make the graphic less wide (e.g. 3 **`m`**), change the **`.dospark`** line to:

```
.dospark spark.average.fund.size.eps 3m
```

**which gives rise to:**

Using 2002–2009, the numbers tell us that managers of university endowments, pension funds, and other institutional investors are bailing out of venture pools and consequently the average venture fund is quickly getting smaller  a trend even more dramatic when one confines one's view to just the size of new funds raised.

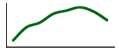
**Comment:** It is useful to be able to use a size unit such as **`m`**, since such units are interpreted with reference to the current point size (e.g. here, **`1m`** is the same size as $N$ points, i.e. $N/72$ inches, where $N$ is the current point size). This enables the proportion between graphic size and current text size to be preserved across font size changes.

***Example C, changing the height only***

Sometimes, in order (e.g.) to make changes in a quantity being graphed stand out better, it may be desirable to increase its height without changing the width. For illustration. we increase the height by 50%, without changing the width. Change the **.dospark** line to:

```
.dospark spark.average.fund.size.eps 0 1.5m
```

**which gives rise to:**

Using 2002–2009, the numbers tell us that managers of university endowments, pension funds, and other institutional investors are bailing out of venture pools and consequently the average venture fund is quickly getting smaller  a trend even more dramatic when one confines one's view to just the size of new funds raised.

**Comment:** Compare with *Example B*; note the extra vertical space.

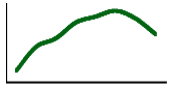
Finally, to illustrate the preservation of proportions across changes in point size, an example in which both the change in width in ***Example B*** and the change in height in ***Example A*** are made in the context of an increase of ambient point size from 11 points (the defauilt for this document) to 20 points.

***Example D, changing the width, height and point size***

```
.lp
.vs 24p
\s[20]Using 2002\^\[en]\^2009, the numbers tell
us that managers of university endowments,
pension funds, and other institutional
investors are bailing out of venture pools
and consequently the average venture fund
is quickly getting smaller
.dospark spark.average.fund.size.eps 3m 1.5m
a trend even more dramatic when one
confines one's view to just the size of
new funds raised.\s0
```

**Comment:** Note the change in vertical line spacing (**.vs 24p**), the change in point size for the paragraph (**\s[20]**), and the two arguments to **.dospark** (**3m** as in ***Example B***, **1.5m** as in ***Example C***).

**The above input produces the following:**

Using 2002–2009, the numbers tell us that managers of university endowments, pension funds, and other institutional investors are bailing out of venture pools and consequently the average venture fund is quickly getting smaller  a trend even more dramatic when one confines one's view to just the size of new funds raised.

An explanation of how the macro **.dospark** works is given on the following page.

**Explanation of the macro dospark**

```
.de dospark
.psbb \\$1
```

**.psbb** *filename* looks inside an EPS file *filename* for the line
**%%BoundingBox: llx lly urx ury**
and sets groff number registers **\n[llx]**, **\n[lly]**, **\n[urx]**, **\n[ury]** to the corresponding values.

```
.nr ht0 \\n[ury]-\\n[lly]
.nr wd0 \\n[urx]-\\n[llx]
```
These set groff registers **\n[ht0]**, **\n[wd0]** to the original height and width of the graphic in points, as defined in the file.

```
.nr deswd (\\n[.ps]/\\n[ht0])*\\n[wd0]
```
Sets the default desired width **\n[deswd]** of the graphic to **\n[wd0]**, preserving the aspect ratio.

```
.if \\$2&(\\$2>0) .nr deswd (u; \\$2p)
```
If the first optional argument **deswid** is present, and greater than zero, then it is converted to device units (**u**) and **\n[deswd]** is set to this value.

```
.nr desht \\n[.ps]
```
Sets the default desired height **\n[desht]** of the graphic to **\n[.ps]** (the current point size, in device units)

```
.if \\$3 .nr desht (u; \\$3p)
```
If the second optional argument **desht** is present, it is converted to device units (**u**) and **\n[desht]** is set to this value.

```
.nr xht 0
```
The default extra line-spacing (above the line) is set to zero.

```
.if (\\n[desht]>\\n[.ps]) .nr xht \\n[desht]-\\n[.ps]
```
If the desired height **\n[desht]** is greater than the current point size **\n[.ps]** (in device units), then the required extra height **\n[xht]** is set to their difference.

```
\X'ps: import \\$1 \
 \\n[llx] \\n[lly] \\n[urx] \\n[ury] \
 \\n[deswd] \\n[desht]'\
\h'\\n[deswd]u'\x'-\\n[xht]u'
..
```

The above shows a single command string split across four lines. In the order of the splitting above, its parts have the following effects.

**1:** Import the EPS file
**2:** using the BoundingBox coordinates extracted from the EPS file
**3:** stipulating that the printed graphic shall be scaled so as to have the desired width **\n[deswd]** and the desired height **\n[desht]**
**4:** and finally:
- **a)** move horizontally (**\h'\\n[deswd]u'**) the width of the graphic (after the command **\X'ps: ... '**, the output position ('currentpoint') is where it was before the command)
- **b)** add extra line spacing **\n[xht]** ***above*** ("**-**") the line (**\x'-\\n[xht]u'**)