

inlinegraphicx

Includegraphics, with an
automatic inline positioning.

Version 0.20c – 15/09/2026

Cédric Pierquet

c pierquet – at – outlook . fr

<https://github.com/cpierquet/latex-packages/tree/main/inlinegraphicx>

Inline insertion,  with automatic adjustments.

Inline insertion,  with automatic adjustments.

Inline insertion,  with *dummybox* if not found.

Inline insertion,  with automatic adjustments but without depth.

Inline  proof with small resizing.

Inline insertion, () with automatic adjustments and choice of optimal height.

Contents

1	Introduction	3
1.1	Loading, useful packages	3
1.2	Usage	3
2	The main macro	3
2.1	Arguments	3
2.2	Examples	4
3	The safeincludegraphics wrapper	5
3.1	Purpose	5
3.2	Examples	5
4	History	6
5	The code	6

1 Introduction

1.1 Loading, useful packages

In order to load `inlinegraphicx`, simply use:

```
\usepackage{inlinegraphicx}
```

The only loaded package is `graphicx`, the rest of the code is in $\text{\LaTeX}$ 3.

1.2 Usage

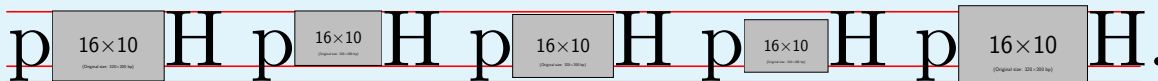
The purpose of this package is to provide command, based on `\includegraphics`, to insert graphic elements *inline*, with automatic positioning and scaling.

Before giving up, the package also tries appending several common extensions to the file name (`.pdf`, `.png`, `.jpg`, `.jpeg`, `.eps`). If the file is still not found, a placeholder image is inserted instead, a small generic icon drawn automatically.

The code determines (total)height and, if necessary, depth of letters in the current font, in order to position the image correctly :

- (total)height is given by (total)height of `qH` in the current font ;
- depth is given by depth of `qH` in the current font.

In order to adjust manually size/positioning, `[keys]` are available.



2 The main macro

2.1 Arguments

The macro for inline insertion is `\inlinegraphics`.

```
\inlinegraphics(*)[scale=...,strut=...]<includegraphics options>{image}
```

The starred version remove depth positioning, wheres normal version include depth.
Available keys are:

- `scale` : re-scaling of compute height (default `1`) ;
- `strut` : characters for height/depth (default `qH`).

2.2 Examples

Inline insertion, `\inlinegraphics{example-image-16x10}` with automatic adjustments.

Inline insertion, `\inlinegraphics[scale=0.75]{example-image-16x10}` with automatic adjustments and scaling.

Inline insertion, `\inlinegraphics*{example-image-16x10}` with automatic adjustments and without depth.

Inline insertion,  with automatic adjustments.

Inline insertion,  with automatic adjustments and scaling.

Inline insertion,  with automatic adjustments and without depth.

```
\scalebox{3}[3]{Inline (\inlinegraphics[strut={()}]{example-image-16x10})
q\inlinegraphics{example-image-16x10}H insertion}.
```

Inline () qH insertion.

```
\sffamily{\Huge Inline (\inlinegraphics[strut={()}]{example-image-16x10})
q\inlinegraphics{example-image-16x9}H insertion}.
```

Inline () qH insertion.

```
\ttfamily{\LARGE Inline (\inlinegraphics[strut={(\acute{E})}]{example-image-16x10}\acute{E})
q\inlinegraphics{example-image-16x10}H and q\inlinegraphics{notfoundfile}H insertion}.
```

Inline () $\acute{E}$ qH and qH insertion.

```
\ttfamily{\large Inline insertion,
(\inlinegraphics[strut={()}]<angle=10>{example-image-16x10}) with option given to
includegraphics.}
```

Inline insertion, () with option given to includegraphics.

3 The safeincludegraphics wrapper

3.1 Purpose

`\safeincludegraphics` is a lightweight, fail-safe wrapper around `\includegraphics`. Unlike `\inlinegraphics`, it performs no inline vertical alignment and parses no key of its own: whatever is given as `[options]` is passed verbatim to `\includegraphics`.

```
\safeincludegraphics[options]{file}
```

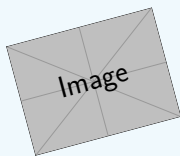
Before inserting the image, the same existence test as `\inlinegraphics` is performed. If the file is still not found, a warning is issued and the `example-image.pdf` placeholder is inserted instead (shipped with the `mwe` package, and available on virtually any L^AT_EX installation, without the need to load `mwe` itself). The placeholder is scaled exactly like a regular image, with the same `[options]`.

3.2 Examples

```
\safeincludegraphics[width=2cm]{example-image-a}
```



```
\safeincludegraphics[width=2cm,angle=15]{thisfiledoesnotexist}
```



4 History

0.20c: Macro for params
0.20b: Dummy graphic if file not found
0.20a: Bugfix with gset (tks to D. Bitouze)
0.2.0: LaTeX3 version of code
0.1.1: Pass option to includegraphics within the macro
0.1.0: Initial version

5 The code

```
% Author      : C. Pierquet
% licence     : Released under the LaTeX Project Public License v1.3c or later, see http://www.latex-project.org/lppl.txt

\NeedsTeXFormat{LaTeX2e}
\ProvidesExplPackage{inlinegraphicx}{2026-09-15}{0.20c}{Insert inline graphicx with LaTeX3}

%====HISTORY
% v 0.20c  Macro for params
% v 0.20b  Placeholder if file not found
% v 0.20a  Bugfix with gset + l3 improvements
% v 0.2.0  LaTeX3 conversion
% v 0.1.1  Alt options for includegraphics
% v 0.1.0  Initial version

%====PACKAGE
\RequirePackage { graphicx }

%====MESSAGES
\msg_new:nnn { inlinegraphicx } { file-not-found }
{
  Image-`#1'-not-found;-inserting-a-dummy-image-instead.
}

%====VARIABLES (toutes \l_ = locales au groupe de la macro)
\dim_new:N \l_toheight_inlinegraphicx_dim
\dim_new:N \l_depth_inlinegraphicx_dim
\dim_new:N \l_raisebox_inlinegraphicx_dim
\dim_new:N \l_height_inlinegraphicx_dim
\dim_new:N \l_dummyinner_inlinegraphicx_dim
\dim_new:N \l_dummydiag_inlinegraphicx_dim
\dim_new:N \l_dummyht_inlinegraphicx_dim
\dim_new:N \l_dummydp_inlinegraphicx_dim
\dim_new:N \l_dummyraise_inlinegraphicx_dim
\fp_new:N \l_scale_inlinegraphicx_fp
\fp_new:N \l_invscale_inlinegraphicx_fp
\tl_new:N \l_strut_inlinegraphicx_tl
\tl_new:N \l_dummy_inlinegraphicx_tl
\tl_new:N \l_resolved_inlinegraphicx_tl
\clist_new:N \l_extensions_inlinegraphicx_clist
\clist_new:N \l_searchpath_inlinegraphicx_clist
\bool_new:N \l_inlinegraphicx_exists_bool

%====KEYS
\keys_define:nn { inlinegraphics }
{
  scale .fp_set:N      = \l_scale_inlinegraphicx_fp,
  scale .initial:n     = { 1.0 },
  scale .default:n     = { 1.0 },
  strut .tl_set:N      = \l_strut_inlinegraphicx_tl,
  strut .initial:n     = { qH },
  strut .default:n     = { qH },
  dummy .tl_set:N      = \l_dummy_inlinegraphicx_tl,
  dummy .initial:n     = { },
  extensions .clist_set:N = \l_extensions_inlinegraphicx_clist,
  extensions .initial:n = { .pdf, .png, .jpg, .jpeg, .eps },
  searchpath .clist_set:N = \l_searchpath_inlinegraphicx_clist,
  searchpath .initial:n = { },
}

%====MACRO CONFIG GLOBAL
\NewDocumentCommand \inlinegraphicssetup { m }
{ \keys_set:nn { inlinegraphics } { #1 } }

%====FONCTIONS INTERNES
\cs_new_protected:Npn \inlinegraphicx_check_exist:n #1
{
  \bool_set_false:N \l_inlinegraphicx_exists_bool
  \tl_set:Nn \l_resolved_inlinegraphicx_tl { #1 }
  % 1) le nom tel quel
  \file_if_exist:nT { #1 }
  {
    \bool_set_true:N \l_inlinegraphicx_exists_bool
    \tl_set:Nn \l_resolved_inlinegraphicx_tl { #1 }
  }
  % 2) le nom + chaque extension
  \bool_if:NF \l_inlinegraphicx_exists_bool
```

```

{
  \clist_map_inline:Nn \l_extensions_inlinegraphicx_clist
  {
    \file_if_exist:nT { #1 ##1 }
    {
      \bool_set_true:N \l_inlinegraphicx_exists_bool
      \tl_set:Nn \l_resolved_inlinegraphicx_tl { #1 ##1 }
      \clist_map_break:
    }
  }
}
% 3) chaque dossier de recherche + nom (+ extensions)
\bool_if:NF \l_inlinegraphicx_exists_bool
{
  \clist_map_inline:Nn \l_searchpath_inlinegraphicx_clist
  {
    \file_if_exist:nT { ##1 #1 }
    {
      \bool_set_true:N \l_inlinegraphicx_exists_bool
      \tl_set:Nn \l_resolved_inlinegraphicx_tl { ##1 #1 }
      \clist_map_break:
    }
    \bool_if:NF \l_inlinegraphicx_exists_bool
    {
      \clist_map_inline:Nn \l_extensions_inlinegraphicx_clist
      {
        \file_if_exist:nT { ##1 #1 ###1 }
        {
          \bool_set_true:N \l_inlinegraphicx_exists_bool
          \tl_set:Nn \l_resolved_inlinegraphicx_tl { ##1 #1 ###1 }
          \clist_map_break:
        }
      }
    }
    \bool_if:NT \l_inlinegraphicx_exists_bool { \clist_map_break: }
  }
}
\bool_if:NF \l_inlinegraphicx_exists_bool
{ \msg_warning:nnn { inlinegraphicx } { file-not-found } { #1 } }
}

\cs_new_protected:Npn \inlinegraphicx_dummybox:
{
  \dim_set:Nn \l_dummyinner_inlinegraphicx_dim
  { \l_height_inlinegraphicx_dim - 1.2pt }
  \dim_set:Nn \l_dummydiag_inlinegraphicx_dim
  { \fp_eval:n { 1.41421356 * \l_dummyinner_inlinegraphicx_dim } pt }
  \fboxsep 0pt \fboxrule 0.6pt
  \hbox_set:Nn \l_tmpb_box
  {
    \fbox
    {
      \makebox [ \dim_use:N \l_dummyinner_inlinegraphicx_dim ] [ c ]
      {
        \makebox [ 0pt ] [ c ]
        {
          \rotatebox
          [ origin = c ]
          { 45 }
          {
            \rule
            { \dim_use:N \l_dummydiag_inlinegraphicx_dim }
            { 0.5pt }
          }
        }
        \makebox [ 0pt ] [ c ]
        {
          \rotatebox
          [ origin = c ]
          { -45 }
          {
            \rule
            { \dim_use:N \l_dummydiag_inlinegraphicx_dim }
            { 0.5pt }
          }
        }
      }
    }
  }
}
% --- on mesure la boîte obtenue puis on la recentre verticalement
% sur la hauteur cible, et on force ses dimensions finales.
\dim_set:Nn \l_dummyht_inlinegraphicx_dim { \box_ht:N \l_tmpb_box }
\dim_set:Nn \l_dummydp_inlinegraphicx_dim { \box_dp:N \l_tmpb_box }
\dim_set:Nn \l_dummyraise_inlinegraphicx_dim
{
  0.5 \l_height_inlinegraphicx_dim
  -
  0.5 \l_dummyht_inlinegraphicx_dim
  +
  0.5 \l_dummydp_inlinegraphicx_dim
}
\hbox_set:Nn \l_tmpb_box
{
  \raisebox
  { \dim_use:N \l_dummyraise_inlinegraphicx_dim }
  { \box_use:N \l_tmpb_box }
}
\box_set_ht:Nn \l_tmpb_box { \l_height_inlinegraphicx_dim }
\box_set_dp:Nn \l_tmpb_box { 0pt }
\box_use:N \l_tmpb_box

```

```

}

\cs_new_protected:Npn \inlinegraphicx_insert:nn #1 #2
{
  \bool_if:NTF \l_inlinegraphicx_exists_bool
  {
    \includegraphics
    [ height = { \dim_use:N \l_height_inlinegraphicx_dim }, #1 ]
    { \l_resolved_inlinegraphicx_tl }
  }
  {
    \tl_if_empty:NTF \l_dummy_inlinegraphicx_tl
    { \inlinegraphicx_dummybox: }
    {
      \includegraphics
      [ height = { \dim_use:N \l_height_inlinegraphicx_dim }, #1 ]
      { \l_dummy_inlinegraphicx_tl }
    }
  }
}

%====MAIN MACRO
\NewDocumentCommand \inlinegraphics { s O { } D < > { } m }
{
  \group_begin:

  \keys_set:nn { inlinegraphics }
  {
    scale = { 1.0 },
    strut = { qH },
    dummy = { },
  }
  \keys_set:nn { inlinegraphics } { #2 }

  \inlinegraphicx_check_exist:n { #4 }

  % ---
  \bool_if:NTF #1
  {
    % --- Mode étoilé : hauteur = hauteur de la capitale du strut
    \hbox_set:Nn \l_tmpa_box { \tl_use:N \l_strut_inlinegraphicx_tl }
    \dim_set:Nn \l_totheight_inlinegraphicx_dim { \box_ht:N \l_tmpa_box }
    \dim_set:Nn \l_height_inlinegraphicx_dim
    {
      \fp_eval:n
      {
        \l_scale_inlinegraphicx_fp * \l_totheight_inlinegraphicx_dim
      }
      pt
    }
    \inlinegraphicx_insert:nn { #3 } { #4 }
  }
  {
    % --- Mode normal : hauteur totale (ht+dp) du strut avec alignement vertical
    \hbox_set:Nn \l_tmpa_box { \tl_use:N \l_strut_inlinegraphicx_tl }
    \dim_set:Nn \l_totheight_inlinegraphicx_dim
    { \box_dp:N \l_tmpa_box + \box_ht:N \l_tmpa_box }
    \dim_set:Nn \l_depth_inlinegraphicx_dim
    { \box_dp:N \l_tmpa_box }
    \dim_set:Nn \l_height_inlinegraphicx_dim
    {
      \fp_eval:n
      {
        \l_scale_inlinegraphicx_fp * \l_totheight_inlinegraphicx_dim
      }
      pt
    }
    \fp_set:Nn \l_invscale_inlinegraphicx_fp
    { 0.5 * ( 1.0 - \l_scale_inlinegraphicx_fp ) }
    \dim_set:Nn \l_raisebox_inlinegraphicx_dim
    {
      \l_depth_inlinegraphicx_dim
      -
      \fp_eval:n
      {
        \l_invscale_inlinegraphicx_fp * \l_totheight_inlinegraphicx_dim
      }
      pt
    }

    \raisebox { - \dim_use:N \l_raisebox_inlinegraphicx_dim }
    { \inlinegraphicx_insert:nn { #3 } { #4 } }
  }
  \group_end:
}

%====SIMPLE WRAPPER
\NewDocumentCommand \safeincludegraphics { O { } m }
{
  \group_begin:
  \inlinegraphicx_check_exist:n { #2 }
  \bool_if:NTF \l_inlinegraphicx_exists_bool
  { \includegraphics [ #1 ] { \l_resolved_inlinegraphicx_tl } }
  { \includegraphics [ #1 ] { example-image.pdf } }
  \group_end:
}

\endinput

```