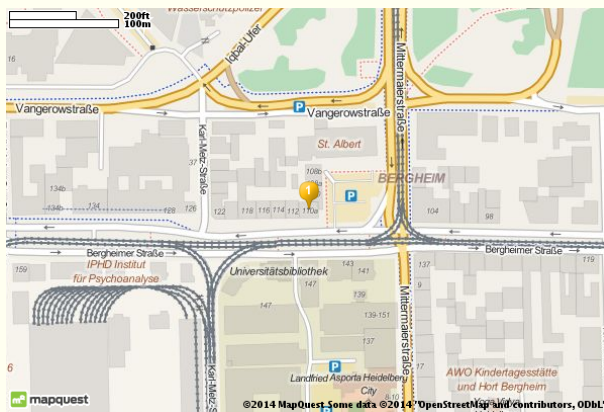


# getmap.sty

## v1.2

### Downloading maps from Open-StreetMap or Google Maps



## 2014/07/15

Package author:  
**Josef Kleber**

<b>1 Options</b>	4
1.1 mode . . . . .	4
1.2 key . . . . .	4
1.3 scale . . . . .	4
1.4 zoom . . . . .	4
1.5 xsize . . . . .	4
1.6 ysize . . . . .	5
1.7 file . . . . .	5
1.8 imagetype . . . . .	5
1.9 type . . . . .	5
1.10 color . . . . .	5
1.11 number . . . . .	5
1.12 inputencoding . . . . .	5
1.13 overwrite . . . . .	6
<b>2 Command(s)</b>	6
2.1 \getmap . . . . .	6
<b>3 Examples</b>	7
<b>4 Implementation</b>	9
<b>5 References</b>	14
<b>6 Change History</b>	15
<b>7 Index</b>	16

### **Abstract**

The goal of this package is the simplest possible provision of map images (OpenStreetMap and Google Maps are supported). In the simplest case, the specification of an address is sufficient. The package loads the map using the `\write18` feature, which you must activate to use this package. The image will be downloaded by an external Lua script. You can use this script, as well as the attached bash script also from the command line.

### **Acknowledgment**

I want to thank Norbert Preining, who did most of the recoding of `osmimage` (Bash  $\rightarrow$  Lua). Moreover many thanks to Taco Hoekwater, Reinhard Kotucha and Heiko Oberdiek for their valuable contributions.

## 1 Options

The following options can be used as package options with global scope, as well as options for the `\getmap` command with local scope!

### 1.1 mode (osm|gm)

This option sets the mode, that is the source of the images. OpenStreetMap or Google Maps!

### 1.2 key (Fmjtd|luur20u22d,75=o5-9aylh6)

In osm mode, the download script requires a key in order to use the service of MapQuest. By default, it uses a key, which is registered for getmap. But you can register and use your own key with this option. The default key is stored in `getmap.cfg`. You can copy this file to your local T<sub>E</sub>X tree and store your own key there<sup>1</sup>! This file will be found after running `texhash`!

### 1.3 scale (3385 (osm), 1 (gm))

osm: This option allows you to specify a display scale for the map image in the range of 1692 – 221871572. You will not necessarily see a difference between 5000 and 5500. A scale value of 3385 corresponds to a zoom level of 17.

gm: For the free version of Google Maps the image size is limited to 640x640. You can set scale to a value of 2, to get exactly the same map in doubled size in pixels.

### 1.4 zoom

This option allows you to specify a zoom level in the range of 1 – 18 ( osm - this option overwrites a possibly given scale) or 0 – 21 (gm).

### 1.5 xsize (600)

This option specifies the width of the map in pixels. If you only want to slightly increase or decrease the map extract, you should adjust the size of the map. You still have full control over the size of the map in the document with the options of `\includegraphics`. (max: 3840 (osm) or 640 (gm))

changed default value  
to 600 in version 1.2!

<sup>1</sup>Mapquest will deliver an url-encoded key, which must be decoded to ASCII, e.g. by [Url decode](#)

**1.6 ysize (400)**

This option specifies the height of the map in pixels. max: 3840 (osm) or 640 (gm))

**1.7 file (getmap)**

This option allows you to specify the name of the image (without extension).

changed default value  
to getmap in version  
1.2!

**1.8 imagetype (png)**

This option allows you to specify the type of the image.

osm: png|jpeg|jpg|gif

gm: png|png8|png32|gif|jpg (progressive)|jpg-baseline (flat)

**1.9 type (map (osm)|roadmap (gm))**

This option specifies the type of the map.

osm: map|sat|hyb      It seems as if there would be only a few regions of Mother Earth, for which satellite and hybrid images are available.

gm: roadmap|satellite|hybrid|terrain

**1.10 color (yellow\_1) (osm)|blue (gm))**

This option specifies the color of the marker. Possible colors:

osm <http://open.mapquestapi.com/staticmap/icons.html>

gm: black, brown, green, purple, yellow, blue, gray, orange, red, white or in hex format 0x3399FF

**1.11 number (1)**

This option specifies the number of the marker.

**1.12 inputencoding**

This option specifies the input encoding of your file. The download script requires the strings encoded in utf8. For the safe conversion the input encoding of the file is required. Normally, you don't have to specify an encoding. The

package tries to evaluate the encoding given to inputenc or assumes utf8. Usually that should work.

### 1.13 overwrite (false|true)

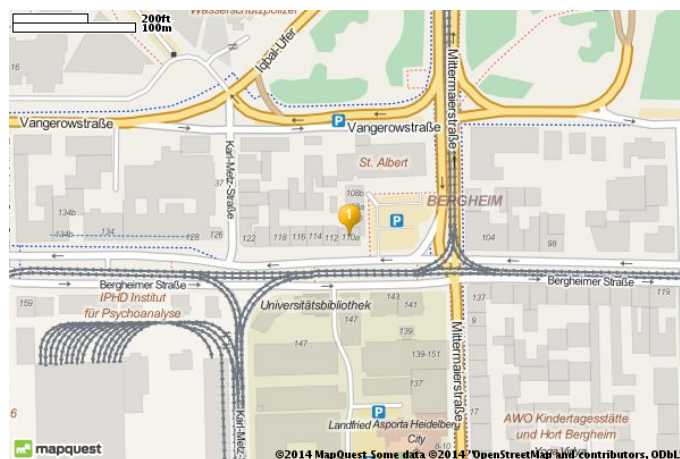
With this option, you can specify whether the image should be downloaded in any case. By default, the option is set to false in order to save bandwidth and compilation time. Nevertheless a check is performed on the existence of the image and the image will be downloaded, if it is not present. In the case of true, the image will be downloaded anyway! BTW, overwrite is equivalent to overwrite=true.

## 2 Command(s)

### 2.1 \getmap

`\getmap[options]{address}` With the `\getmap` command you can download a map, if you enable `\write18` (TeXLive: `-shell-escape`, MiKTeX: `--enable-write18`). This is only necessary if you actually download an image. You can use the options described above to specify the properties of the downloaded image. After executing the command, the image is available in the current working directory!

In the simplest case, you only need an address, a POI or geographic coordinates (latitude, longitude) to download the map. `{address}` must be fully expanded and must not contain macros! By default, the image is saved under the name `getmap.png`! If you need only one map (e.g. the office of Dante e.V.) in your document, it can be as simple as:

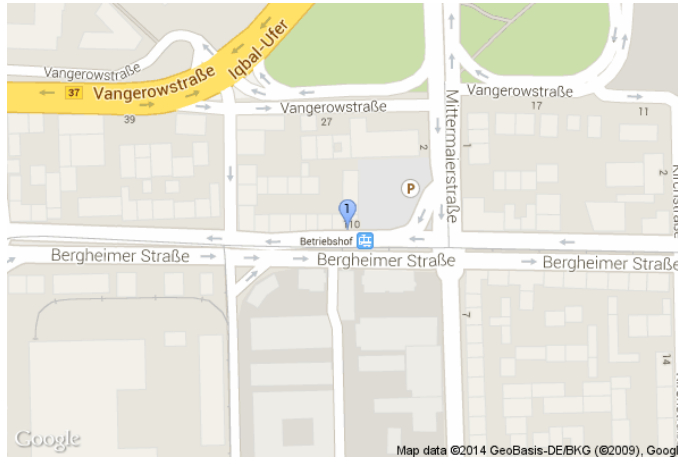


```
1 \getmap{Bergheimer Straße 110A, 69115 Heidelberg, Germany}
2 \includegraphics[width=9cm]{getmap}
```

### 3 Examples



The same map as before from Google Maps:



```
1 \getmap[file=dantegm,mode=gm]{Bergheimer Straße 110A,%
2                               69115 Heidelberg, Germany}
3 \includegraphics[width=9cm]{dantegm}
```

The same map as satellite image:



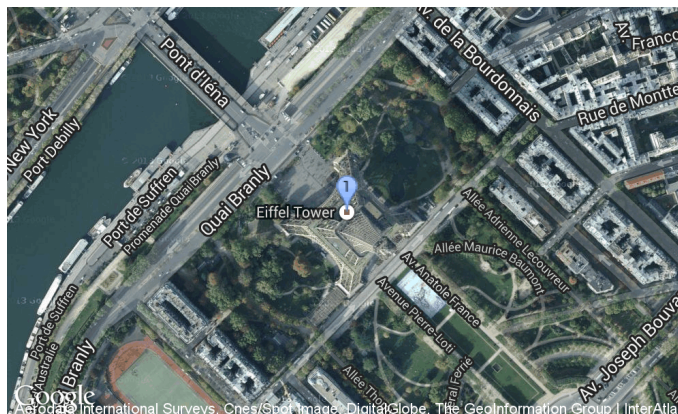
```
1 \getmap[file=dantegmsat,mode=gm,type=satellite]
2       {Bergheimer Straße 110A, 69115 Heidelberg, Germany}
3 \includegraphics[width=9cm]{dantegmsat}
```

L'afrique, mon amour!



```
1 \getmap[file=africa,mode=gm,type=terrain,xsize=500,ysize=500,%
2     scale=2,zoom=3]{0,16}
3 \includegraphics[width=9cm]{africa}
```

L'amour, ...



```
1 \getmap[file=paris,mode=gm,type=hybrid,xsize=500,ysize=300,%
2     scale=2,zoom=16]{Tour Eiffel, Paris}
3 \includegraphics[width=9cm]{paris}
```



## 4 Implementation

```
1 <{*package>
```

First, we provide the L<sup>A</sup>T<sub>E</sub>X package getmap.

```
2 \NeedsTeXFormat{LaTeX2e}%
3 \ProvidesPackage{getmap}[2014/07/15 v1.2 getmap.sty - Josef Kleber (C) 2014]%
```

We need a few packages!

```
4 \RequirePackage{xkeyval}%
5 \RequirePackage{stringenc}%
6 \RequirePackage{ifthen}%
```

We provide a macro `\GM@JK@define@key`, which defines package options with global scope and options for `\getmap` with local scope. It takes four arguments `{<prefix>}`, `{<package>}`, `{<option>}` and `{<default>}`.

```
7 \newcommand*{\GM@JK@define@key}[4]%
8 {%
9   \expandafter\gdef\csname#1@#3\endcsname{#4}%
10  \define@key{#2.sty}{#3}[#4]%
11  {%
12    \expandafter\gdef\csname#1@#3\endcsname{##1}%
13  }%
14  \define@key{#2}{#3}%
15  {%
16    \expandafter\def\csname#1@#3\endcsname{##1}%
17  }%
18}%
```

Now, we can use this macro to define our options.

```
19 \GM@JK@define@key{\GM@JK}{getmap}{mode}{osm}%
20 \GM@JK@define@key{\GM@JK}{getmap}{key}{}%
21 \GM@JK@define@key{\GM@JK}{getmap}{xsize}{600}%
22 \GM@JK@define@key{\GM@JK}{getmap}{ysize}{400}%
23 \GM@JK@define@key{\GM@JK}{getmap}{scale}{3385}%
24 \GM@JK@define@key{\GM@JK}{getmap}{zoom}{}%
25 \GM@JK@define@key{\GM@JK}{getmap}{type}{map}%
26 \GM@JK@define@key{\GM@JK}{getmap}{imagetype}{png}%
27 \GM@JK@define@key{\GM@JK}{getmap}{color}{yellow_1}%
28 \GM@JK@define@key{\GM@JK}{getmap}{number}{1}%
29 \GM@JK@define@key{\GM@JK}{getmap}{file}{getmap}%
30 \GM@JK@define@key{\GM@JK}{getmap}{inputencoding}{}%
31 \GM@JK@define@key{\GM@JK}{getmap}{overwrite}{true}%
```

For options without default value, we define reasonable default values! We overwrite the default for `overwrite`, because we don't want `overwrite` to be true by default, but that `overwrite` is equivalent to `overwrite=true`!

Moreover, we load `getmap.cfg` to set the default key. You can copy this file to your local T<sub>E</sub>X tree and replace the key with your own!

We try to use the input encoding specified for `inputenc` or `utf8` instead.

```

32 \gdef\GM@JK@overwrite{false}%
33 \gdef\GM@JK@key{}%
34 %
35 \IfFileExists{getmap.cfg}%
36 {%
37   \input{getmap.cfg}%
38 }%
39 {%
40   \gdef\GM@JK@key{Fmjtd|luur20u22d,75=o5-9aylh6}%
41 }%
42 %
43 \@ifpackageloaded{inputenc}%
44 {%
45   \gdef\GM@JK@inputencoding{\inputencodingname}%
46 }%
47 {%
48   \gdef\GM@JK@inputencoding{utf8}%
49 }%
50 %

```

Later, we will need a switch, if `\write18` is enabled.

```

51 \newif\ifGM@JK@writexviii\GM@JK@writexviiiifalse%
52 %

```

We execute the package options to define and set the option macros.

```

53 \ExecuteOptionsX{mode,xsize,ysize,scale,zoom,type,imagetype,color,number,file}%
54 %
55 \ProcessOptionsX\relax%
56 %

```

We need to reset some defaults in `gm` mode.

```

57 %
58 \ifthenelse{\equal{\GM@JK@mode}{gm}}{%
59 {%
60   \gdef\GM@JK@scale{1}%
61   \gdef\GM@JK@zoom{17}%
62   \gdef\GM@JK@type{roadmap}%
63   \gdef\GM@JK@color{blue}%
64 }%
65 }%
66 %

```

We check if `\pdf@shellescape` is available to test if `\write18` is enabled.

If false, we assume `\write18` is available and hope for the best.

If true, we set the switch `\GM@JK@writexviii` accordingly!

```

67 %
68 \ltx@ifundefined{pdf@shellescape}%

```

```

69 {%
70   \PackageInfo{getmap}{\pdfshellescape is undefined}%
71   \PackageInfo{getmap}{can not test if \write18 is available}%
72   \GM@JK@writexviii>true%
73 }%
74 {%
75   \PackageInfo{getmap}{\pdfshellescape is available}%
76   \ifnum\pdfshellescape=1\relax%
77     \PackageInfo{getmap}{\write18 enabled}%
78     \GM@JK@writexviii>true%
79   \else%
80     \GM@JK@writexviii>false%
81   \fi%
82 }%
83 %

```

We define a macro that is executed as `\write18` call. First, we test if `\write18` is enabled and issue a package error if not! Otherwise we execute `\write18` depending on the mode

```

84 \newcommand*\GM@JK@shellescape%
85 {%
86   \ifGM@JK@writexviii\relax%
87     \ifthenelse{\equal{\GM@JK@mode}{osm}}%
88     {%
89       \immediate\write18{getmapdl \space-l\space "\GM@JK@location@string"%
90                               \space-m\space osm%
91                               \space-k\space "\GM@JK@key@string"%
92                               \space-x\space \GM@JK@xsize%
93                               \space-y\space \GM@JK@ysize%
94                               \space-z\space "\GM@JK@zoom"%
95                               \space-s\space \GM@JK@scale%
96                               \space-t\space \GM@JK@type%
97                               \space-i\space \GM@JK@imagetype%
98                               \space-c\space "\GM@JK@color"%
99                               \space-n\space \GM@JK@number%
100                              \space-o\space \GM@JK@file}%
101     }%
102     {%
103       \immediate\write18{getmapdl \space-l\space "\GM@JK@location@string"%
104                               \space-m\space gm%
105                               \space-x\space \GM@JK@xsize%
106                               \space-y\space \GM@JK@ysize%
107                               \space-z\space \GM@JK@zoom%
108                               \space-s\space \GM@JK@scale%
109                               \space-t\space \GM@JK@type%
110                               \space-i\space \GM@JK@imagetype%
111                               \space-c\space "\GM@JK@color"%
112                               \space-n\space \GM@JK@number%
113                              \space-o\space \GM@JK@file}%
114     }%
115   \else%

```

```

116   \PackageError{getmap}{\write18 disabled}%
117   {\write18 disabled\MessageBreak%
118   Use -shell-escape (TeXLive)\MessageBreak%
119   or\space\space--enable-write18 (MiKTeX)}%
120   \fi%
121 }%

```

`\getmap` Here, we define the user command to download the map.

```
\getmap[<options>]{<address>}
```

```

122 \newcommand*\getmap[2][]%
123 {%

```

We start a group to keep the setting of options local. Then we test, if we are in `gm` mode to reset some defaults! Finally, we set the local options again to override defaults if necessary!

```

124 \begingroup%
125   \setkeys{getmap}{#1}%
126   \ifthenelse{\equal{\GM@JK@mode}{gm}}{%
127     {%
128       \def\GM@JK@scale{1}%
129       \def\GM@JK@zoom{17}%
130       \def\GM@JK@type{roadmap}%
131       \def\GM@JK@color{blue}%
132     }%
133   }%
134   \setkeys{getmap}{#1}%
135   \PackageInfo{getmap}{using \GM@JK@inputencoding\space encoding}%
136   \def\GM@JK@location{#2}%

```

`texlua` expects its arguments encoded in utf8!

```

137   \StringEncodingConvert%
138     {\GM@JK@location@string}%
139     {\detokenize\expandafter{\GM@JK@location}}%
140     {\GM@JK@inputencoding}{utf-8}%
141   \StringEncodingSuccessFailure%
142   {%
143     %success
144   }%
145   {% failure
146     \errmessage{Converting to UTF-8 failed}%
147   }%
148   \StringEncodingConvert%
149     {\GM@JK@key@string}%
150     {\detokenize\expandafter{\GM@JK@key}}%
151     {\GM@JK@inputencoding}{utf-8}%
152   \StringEncodingSuccessFailure%
153   {%
154     %success
155   }%

```

```

156     {% failure
157       \errmessage{Converting to UTF-8 failed}%
158     }%

```

We check, if overwrite is true and download the map. If not, we check if the image is already in the working directory. If not, we download the image!

```

159     \ifthenelse{\equal{\GM@JK@overwrite}{true}}{%
160     {%
161       \GM@JK@shellescape%
162     }%
163     {%
164       \IfFileExists{\GM@JK@file.\GM@JK@imagetype}%
165       {%
166         \PackageInfo{getmap}{overwrite=false; (\GM@JK@file.\GM@JK@imagetype)%
167           using existing file!}%
168       }%
169       {%
170         \PackageInfo{getmap}{overwrite=false; (\GM@JK@file.\GM@JK@imagetype)%
171           file does not exist! downloading ...}%
172         \GM@JK@shellescape%
173       }%
174     }%
175   \endgroup%
176 }%

177 </package>

```

## 5 References

- [1] Google, Inc. Static Maps API V2 Developer Guide, 2014.  
<https://developers.google.com/maps/documentation/staticmaps/>.
- [2] MapQuest, Inc. Introducing the Data Manager API Web Service, 2014.  
<http://developer.mapquest.com>.
- [3] MapQuest, Inc. MapQuest Open Platform Web Services, 2014.  
<http://open.mapquestapi.com/>.
- [4] MapQuest, Inc. Static Map Service: Standard Icons, 2014.  
<http://open.mapquestapi.com/staticmap/icons.html>.
- [5] MapQuest, Inc. Zoom To Scale Mapping, 2014.  
<http://open.mapquestapi.com/staticmap/zoomToScale.html>.

## 6 Change History

v1.0	
General: CTAN upload .....	9
v1.1	
\getmap: Bugfix: problem	
in URL when using	
\usepackage[utf8]{inputenc}	12
v1.2	
General: added getmap.cfg to store	
	default key (FR by Ulrike Fischer) .....
	9
	added support for Google Maps
	9
	changed default values of xsize
	(600) and file (getmap) .....
	9
	renamed osmimage.lua to
	getmapdl.lua .....
	9

**Symbols**`\ifpackageloaded` ..... 43**D**`\define@key` ..... 10, 14`\detokenize` ..... 139, 150**G**`getmap (Package)` ..... 4, 9`\getmap` ..... [122](#)`\GM@JK@color` .. [63](#), [98](#), [111](#), [131](#)`\GM@JK@define@key` [7](#), [19](#), [20](#), [21](#),  
[22](#), [23](#), [24](#), [25](#), [26](#), [27](#), [28](#),  
[29](#), [30](#), [31](#)`\GM@JK@file` [100](#), [113](#), [164](#), [166](#),  
[170](#)`\GM@JK@imagetype` . [97](#), [110](#), [164](#),  
[166](#), [170](#)`\GM@JK@inputencoding` .. [45](#), [48](#),  
[135](#), [140](#), [151](#)`\GM@JK@key` ..... [33](#), [40](#), [150](#)`\GM@JK@key@string` .... [91](#), [149](#)`\GM@JK@location` ..... [136](#), [139](#)`\GM@JK@location@string` [89](#), [103](#),  
[138](#)`\GM@JK@mode` ..... [58](#), [87](#), [126](#)`\GM@JK@number` ..... [99](#), [112](#)`\GM@JK@overwrite` ..... [32](#), [159](#)`\GM@JK@scale` .. [60](#), [95](#), [108](#), [128](#)`\GM@JK@shellescape` [84](#), [161](#), [172](#)`\GM@JK@type` ... [62](#), [96](#), [109](#), [130](#)`\GM@JK@writexviiifalse` [51](#), [80](#)`\GM@JK@writexviiitrue` . [72](#), [78](#)`\GM@JK@xsize` ..... [92](#), [105](#)`\GM@JK@ysize` ..... [93](#), [106](#)`\GM@JK@zoom` ... [61](#), [94](#), [107](#), [129](#)**I**`\IfFileExists` ..... [35](#), [164](#)`\ifGM@JK@writexviii` ... [51](#), [86](#)`\input` ..... [37](#)`inputenc (Package)` ..... [6](#), [10](#)`\inputencodingname` ..... [45](#)**L**`\ltx@ifUndefined` ..... [68](#)**O**`overwrite (Style option)` .. [9](#), [13](#)**P**`Package``getmap` ..... [4](#), [9](#)`inputenc` ..... [6](#), [10](#)`\pdf@shellescape` ... [70](#), [75](#), [76](#)**S**`\setkeys` ..... [125](#), [134](#)`\StringEncodingConvert` .. [137](#),  
[148](#)`\StringEncodingSuccessFailure`  
..... [141](#), [152](#)`Style option``overwrite` ..... [9](#), [13](#)**W**`\write` [71](#), [77](#), [89](#), [103](#), [116](#), [117](#)