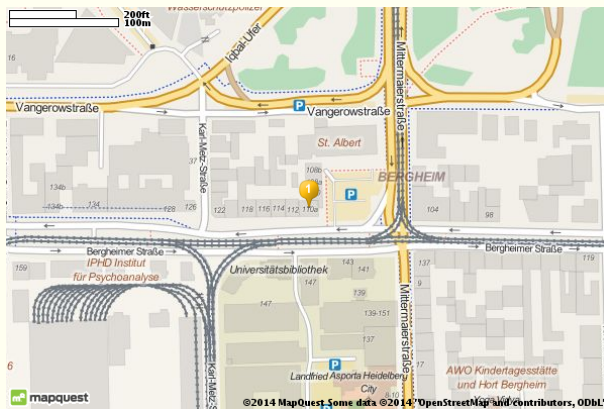


getmap.sty

v1.3

Downloading maps from Open-StreetMap, Google Maps or Google Street View



2014/07/19

Package author:
Josef Kleber

1 Options	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 heading	5
1.13 pitch	6
1.14 fov	6
1.15 inputencoding	6
1.16 overwrite	6
2 Command(s)	6
2.1 \getmap	6
3 Examples	8
4 Implementation	11
5 References	17
6 Change History	18
7 Index	19

Abstract

The goal of this package is the simplest possible provision of map images (OpenStreetMap, Google Maps and Google Street View 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 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|gsv)`

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

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 \TeX tree and store your own key there¹! 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!

¹Mapquest 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 heading (0)

This option specifies the heading (direction) in degrees in the range of 0 – 360 (gsv mode). (0: north, 90: east, ...)

1.13 pitch (0)

This option specifies the pitch (angle) of the camera view in degrees in the range of -90 – 90 (gsv mode).

1.14 fov (90)

This option specifies the field of horizontal view (kind of zoom) in degrees in the range of 0 – 120 (gsv mode).

1.15 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.

1.16 overwrite (false|true)

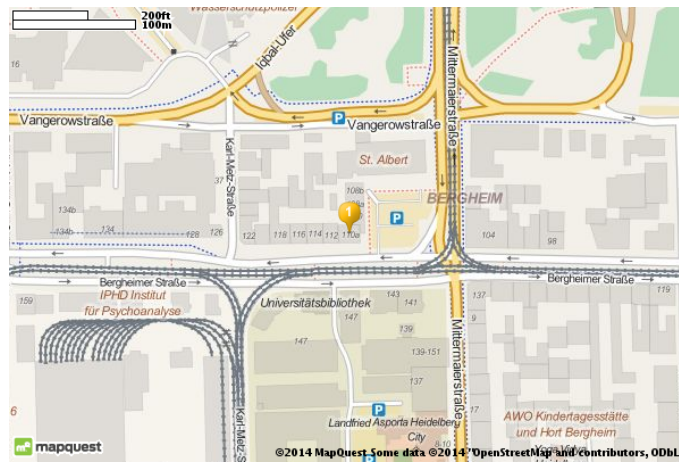
With this option, you can specify whether the image should be downloaded in any case. By default, the option is set to false in order to save bandwidth and compilation time. Nevertheless a check is performed on the existence of the image and the image will be downloaded, if it is not present. In the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



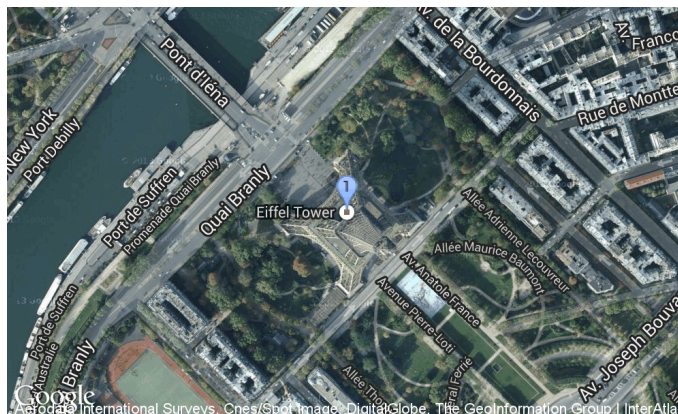
```
1 \getmap[file=dantegmsat,mode=gm,type=satellite]
2       {Bergheimer Straße 110A, 69115 Heidelberg, Germany}
3 \includegraphics[width=9cm]{dantegmsat}
```


L'afrique, mon amour!



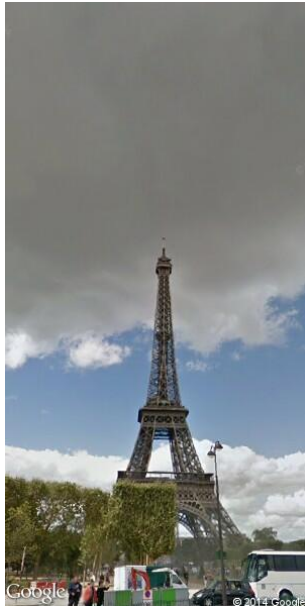
```
1 \getmap[file=africa,mode=gm,type=terrain,xsize=500,ysize=500,%
2     scale=2,zoom=3]{0,16}
3 \includegraphics[width=9cm]{africa}
```

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}
```

Street View now:



```
1 \getmap[file=parisgsv,mode=gsv,heading=320,pitch=30,fov=40,%  
2 xsize=300,ysize=600]{Avenue Piere-Loti, Paris}  
3 \includegraphics[width=4cm]{parisgsv}
```

and from the platform:



```
1 \getmap[file=parisgsvp,mode=gsv,heading=30,pitch=-25,fov=60]%  
2 {Tour Eiffel, Paris}  
3 \includegraphics[width=8cm]{parisgsvp}
```

4 Implementation

```
1 <*package>
```

First, we provide the L^AT_EX package getmap.

```
2 \NeedsTeXFormat{LaTeX2e}%
3 \ProvidesPackage{getmap}[2014/07/19 v1.3 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}{heading}{0}%
30 \GM@JK@define@key{GM@JK}{getmap}{fov}{90}%
31 \GM@JK@define@key{GM@JK}{getmap}{pitch}{0}%
32 \GM@JK@define@key{GM@JK}{getmap}{file}{getmap}%
33 \GM@JK@define@key{GM@JK}{getmap}{inputencoding}{}%
34 \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 \TeX tree and replace the key with your own!

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

```

35 \gdef\GM@JK@overwrite{false}%
36 \gdef\GM@JK@key{}%
37 %
38 \IfFileExists{getmap.cfg}%
39 {%
40   \input{getmap.cfg}%
41 }%
42 {%
43   \gdef\GM@JK@key{Fmjtd|luur20u22d,75=o5-9aylh6}%
44 }%
45 %
46 \@ifpackageloaded{inputenc}%
47 {%
48   \gdef\GM@JK@inputencoding{\inputencodingname}%
49 }%
50 {%
51   \gdef\GM@JK@inputencoding{utf8}%
52 }%
53 %

```

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

```

54 \newif\ifGM@JK@writexviii\GM@JK@writexviiiifalse%
55 %

```

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

```

56 \ExecuteOptionsX{mode,xsize,ysize,scale,zoom,type,imagetype,color,number,file,heading,fov,
57 %
58 \ProcessOptionsX\relax%
59 %

```

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

```

60 %
61 \ifthenelse{\equal{\GM@JK@mode}{gm}}{%
62 {%
63   \gdef\GM@JK@scale{1}%
64   \gdef\GM@JK@zoom{17}%
65   \gdef\GM@JK@type{roadmap}%
66   \gdef\GM@JK@color{blue}%
67 }%
68 }%
69 %

```

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!

```

70 %
71 \ltx@ifundefined{pdf@shellescape}%
72 {%
73   \PackageInfo{getmap}{\pdf@shellescape is undefined}%
74   \PackageInfo{getmap}{can not test if \write18 is available}%
75   \GM@JK@writexviii>true%
76 }%
77 {%
78   \PackageInfo{getmap}{\pdf@shellescape is available}%
79   \ifnum\pdf@shellescape=1\relax%
80     \PackageInfo{getmap}{\write18 enabled}%
81     \GM@JK@writexviii>true%
82   \else%
83     \GM@JK@writexviii>false%
84   \fi%
85 }%
86 %

```

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

```

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

```

```

115                                     \space-i\space \GM@JK@imagetype%
116                                     \space-c\space "\GM@JK@color"%
117                                     \space-n\space \GM@JK@number%
118                                     \space-o\space \GM@JK@file}%
119    }%
120    {%
121      \ifthenelse{\equal{\GM@JK@mode}{gsv}}{%
122        {%
123          \immediate\write18{getmapdl \space-l\space "\GM@JK@location@string"%
124            \space-m\space gsv%
125            \space-x\space \GM@JK@xsize%
126            \space-y\space \GM@JK@ysize%
127            \space-H\space \GM@JK@heading%
128            \space-F\space \GM@JK@fov%
129            \space-T\space \GM@JK@pitch%
130            \space-o\space \GM@JK@file}%
131        }%
132      }%
133      \PackageError{getmap}{invalid mode}{invalid mode! Use osm, gm or gsv!}%
134    }%
135  }%
136 }%
137 \else%
138   \PackageError{getmap}{\write18 disabled}%
139   {\write18 disabled\MessageBreak%
140    Use -shell-escape (TeXLive)\MessageBreak%
141    or\space\space--enable-writel8 (MiKTeX)}%
142 \fi%
143 }%

```

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

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

```

144 \newcommand*\getmap[2][]%
145 {%

```

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!

```

146 \begingroup%
147   \setkeys{getmap}{#1}%
148   \ifthenelse{\equal{\GM@JK@mode}{gm}}{%
149     {%
150       \def\GM@JK@scale{1}%
151       \def\GM@JK@zoom{17}%
152       \def\GM@JK@type{roadmap}%
153       \def\GM@JK@color{blue}%
154     }%
155   }%
156   \setkeys{getmap}{#1}%

```

In gsv mode, we have an implicit `imagetype=jpg`. Therefore, we have to set it to allow the later test on the existence of the image file!

```

157 \ifthenelse{\equal{\GM@JK@mode}{gsv}}%
158 {\def\GM@JK@imagetype{jpg}}{}%
159 \PackageInfo{getmap}{using \GM@JK@inputencoding\space encoding}%
160 \def\GM@JK@location{#2}%

```

texlua expects its arguments encoded in utf8!

```

161 \StringEncodingConvert%
162 {\GM@JK@location@string}%
163 {\detokenize\expandafter{\GM@JK@location}}%
164 {\GM@JK@inputencoding}{utf-8}%
165 \StringEncodingSuccessFailure%
166 {%
167   %success
168 }%
169 {% failure
170   \errmessage{Converting to UTF-8 failed}%
171 }%
172 \StringEncodingConvert%
173 {\GM@JK@key@string}%
174 {\detokenize\expandafter{\GM@JK@key}}%
175 {\GM@JK@inputencoding}{utf-8}%
176 \StringEncodingSuccessFailure%
177 {%
178   %success
179 }%
180 {% failure
181   \errmessage{Converting to UTF-8 failed}%
182 }%

```

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!

```

183 \ifthenelse{\equal{\GM@JK@overwrite}{true}}%
184 {%
185   \GM@JK@shellescape%
186 }%
187 {%
188   \IfFileExists{\GM@JK@file.\GM@JK@imagetype}%
189   {%
190     \PackageInfo{getmap}{overwrite=false; (\GM@JK@file.\GM@JK@imagetype)%
191       using existing file!}%
192   }%
193   {%
194     \PackageInfo{getmap}{overwrite=false; (\GM@JK@file.\GM@JK@imagetype)%
195       file does not exist! downloading ...}%
196     \GM@JK@shellescape%
197   }%
198 }%
199 \endgroup%

```

```
200 }%
```

```
201 </package>
```


5 References

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

6 Change History

v1.0		
General: CTAN upload	11	
v1.1		
\getmap: Bugfix: problem		
in URL when using		
\usepackage[utf8]{inputenc}	14	
v1.2		
General: added getmap.cfg to store		
default key (FR by Ulrike Fis-		
cher)	12	
added support for Google Maps	11	
changed default values of xsize		
(600) and file (getmap)	11	
renamed osmimage.lua to		
getmapdl.lua	11	
v1.3		
General: added support for Google		
Street View	11	

Symbols

\@ifpackageloaded 46

D

\define@key 10, 14

\detokenize 163, 174

G

getmap (Package) 4, 11

\getmap 144

\GM@JK@color . 66, 101, 116, 153

\GM@JK@define@key 7, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28,
29, 30, 31, 32, 33, 34

\GM@JK@file 103, 118, 130, 188,
190, 194

\GM@JK@fov 128

\GM@JK@heading 127

\GM@JK@imagetype 100, 115, 158,
188, 190, 194

\GM@JK@inputencoding .. 48, 51,
159, 164, 175

\GM@JK@key 36, 43, 174

\GM@JK@key@string 94, 173

\GM@JK@location 160, 163

\GM@JK@location@string 92, 108,
123, 162

\GM@JK@mode ... 61, 90, 106, 121,
148, 157

\GM@JK@number 102, 117

\GM@JK@overwrite 35, 183

\GM@JK@pitch 129

\GM@JK@scale .. 63, 98, 113, 150

\GM@JK@shellescape 87, 185, 196

\GM@JK@type ... 65, 99, 114, 152

\GM@JK@writexviiiifalse 54, 83

\GM@JK@writexviiiitruer . 75, 81

\GM@JK@xsize 95, 110, 125

\GM@JK@ysize 96, 111, 126

\GM@JK@zoom ... 64, 97, 112, 151

I

\IfFileExists 38, 188

\ifGM@JK@writexviii ... 54, 89

\input 40

inputenc (Package) 6, 12

\inputencodingname 48

L

\ltx@ifUndefined 71

O

overwrite (Style option) . 11, 12,
15

P

Package

getmap 4, 11

inputenc 6, 12

\pdf@shellescape ... 73, 78, 79

S

\setkeys 147, 156

\StringEncodingConvert .. 161,
172

\StringEncodingSuccessFailure
..... 165, 176

Style option

overwrite 11, 12, 15

W

\write 74, 80, 92, 108, 123, 138,
139