

The l3graphics package

Graphics inclusion support

The L^AT_EX Project*

Released 2024-01-04

1 l3graphics documentation

1.1 Graphics keys

Inclusion of graphic files requires a range of low-level data be passed to the backend. This is set up using a small number of key–value settings, which are stored in the `graphics` tree.

<code>decodearray</code>	Array to decode color in bitmap graphic: when non-empty, this should be in the form of one, two or three pairs of real numbers in the range $[0, 1]$, separated by spaces.
--------------------------	---

<code>draft</code>	Switch to enable draft mode: graphics are read but not included when this is true.
--------------------	--

<code>interpolate</code>	Switch which indicates whether interpolation should be applied to bitmap graphic files.
--------------------------	---

<code>page</code>	The page to extract from a multi-page graphic file: used for <code>.pdf</code> files which may contain multiple pages.
-------------------	--

<code>pdf-attr</code>	Additional PDF-focussed attributes: available to allow control of extended <code>.pdf</code> structures beyond those needed for graphic inclusion. Due to backend restrictions, this key is only functional with direct PDF mode (pdfT _E X and LuaT _E X).
-----------------------	---

<code>pagebox</code>	The nature of the page box setting used to determine the bounding box of material: used for <code>.pdf</code> files which feature multiple page box specifications. A choice from <code>art</code> , <code>bleed</code> , <code>crop</code> , <code>media</code> , <code>trim</code> . The standard setting is <code>crop</code> .
----------------------	--

<code>type</code>	The type of graphic file being included: if this key is not set, the <i>type</i> is determined from the file extension.
-------------------	---

*E-mail: latex-team@latex-project.org

1.2 Including graphics

<code>\graphics_include:nn</code>	<code>\graphics_include:nn {<keys>} {<file>}</code>
<code>\graphics_include:nV</code>	

Horizontal-mode command which includes the $\langle file \rangle$ as an graphic at the current location. The file $\langle type \rangle$ may be given as one of the $\langle keys \rangle$, or will otherwise be determined from file extension. The $\langle keys \rangle$ is used to pass settings as detailed above.

<code>\l_graphics_ext_type_prop</code>
--

Defines mapping between file extensions and file types; where there is no entry for an extension, the type is assumed to be the extension with the leading `.` removed. Entries should be made in lower case, and the key should be an extension including the leading `.`, for example

```
\prop_put:Nnn \l_graphics_ext_type_prop { .ps } { eps }
```

<code>\l_graphics_search_path_seq</code>
--

Each entry is the path to a directory which should be searched when seeking an graphic file. Each path can be relative or absolute, and should not include the trailing slash. The entries are not expanded when used so may contain active characters but should not feature any variable content. Spaces need not be quoted.

1.3 Utility functions

<code>\graphics_get_full_name:nN</code>	<code>\graphics_get_full_name:nN {<file>} <tl var></code>
<code>\graphics_get_full_name:nNTF</code>	<code>\graphics_get_full_name:nNTF {<file>} <tl var> {<true code>} {<false code>}</code>

Searches for $\langle file \rangle$ first as given and then using the extensions listed in `\l_graphics_search_ext_seq`. The search path used will be the entries of `\l_graphics_search_path_seq`. If found, the full file name including any path and extension will be returned in the $\langle tl var \rangle$. In the non-branching version, the $\langle tl var \rangle$ will be set to `\q_no_value` in the case that the graphics is not found.

<code>\l_graphics_search_ext_seq</code>

Extensions to use for graphic searching when the given $\langle file \rangle$ name is not found by `\graphics_get_full_name:nN`.

<code>\graphics_get_pagecount:nN</code>

`\graphics_get_pagecount:nn {<file>} <tl var>`

Reads the graphics $\langle file \rangle$ and extracts the number of pages, which are stored in the $\langle tl var \rangle$.

1.4 Showing and logging included graphics

```
\graphics_show_list: \graphics_show_list:
\graphics_log_list:  \graphics_log_list:
```

These functions list all graphic files loaded by in a similar manner to `\file_show_list:` and `\file_log_list:`. While `\graphics_show_list:` displays the list in the terminal, `\graphics_log_list:` outputs it to the log file only. In both cases, only graphics loaded by `\l3graphics` are listed.

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

D		<code>\l_graphics_search_ext_seq</code> <i>2</i>
<code>decodearray</code> <i>1</i>		<code>\l_graphics_search_path_seq</code> <i>2</i>
<code>draft</code> <i>1</i>		<code>\graphics_show_list:</code> <i>3</i>
F		I
file commands:		<code>interpolate</code> <i>1</i>
<code>\file_log_list:</code> <i>3</i>		P
<code>\file_show_list:</code> <i>3</i>		<code>page</code> <i>1</i>
G		<code>pagebox</code> <i>1</i>
graphics commands:		<code>pdf-attr</code> <i>1</i>
<code>\l_graphics_ext_type_prop</code> <i>2</i>		Q
<code>\graphics_get_full_name:nN</code> <i>2</i>		quark commands:
<code>\graphics_get_full_name:nNTF</code> <i>2</i>		<code>\q_no_value</code> <i>2</i>
<code>\graphics_get_pagecount:nN</code> <i>2</i>		T
<code>\graphics_get_pagecount:nn</code> <i>2</i>		<code>type</code> <i>1</i>
<code>\graphics_include:nn</code> <i>2</i>		
<code>\graphics_log_list:</code> <i>3</i>		