NAME

glib-compile-resources − GLib resource compiler

SYNOPSIS

glib−compile−resources [OPTION...] {FILE}

DESCRIPTION

glib−compile−resources reads the resource description from *FILE* and the files that it references and creates a binary resource bundle that is suitable for use with the **GResource** API. The resulting bundle is then written out as−is, or as C source for linking into an application.

The XML resource files normally have the filename extension .gresource.xml. For a detailed description of the XML file format, see the **GResource** documentation.

OPTIONS

−h, **−−help**
Print help and exit

−−version
Print program version and exit

−−target=*TARGET*
Store the compiled resources in the file *TARGET*. If not specified a filename based on the *FILE* basename is used.

−−sourcedir=*DIRECTORY*
The files referenced in *FILE* are loaded from this directory. If not specified, the current directory is used.

−−generate
Write the output file in the format selected for by its filename extension:

.c
C source

.h
C header

.gresource
resource bundle

−−generate−source
Instead of a writing the resource bundle in binary form create a C source file that contains the resource bundle. This can then be compiled into an application for easy access.

−−generate−header
Generate a header file for use with C code generated by **−−generate−source**.

−−generate−dependencies
Prints the list of files that the resource bundle references to standard output. This can be used to track dependencies in the build system. For example, the following make rule would mark *test.gresource* as depending on all the files that *test.gresource.xml* includes, so that is is automatically rebuilt if any of them change:

test.gresource: test.gresource.xml $(shell $(GLIB_COMPILE_RESOURCES) −−generate−dependencies test.gresour

Note that this may or may not be portable to non−GNU **make**.

Also see **−−dependency−file**.

−−c−name
Specify the prefix used for the C identifiers in the code generated by **−−generate−source** and

−−generate−header.

−−manual−register
By default code generated by **−−generate−source** uses automatic initialization of the resource. This works on most systems by using the compiler support for constructors. However, some (uncommon) compilers may not support this, you can then specify **−−manual−register**, which will generate custom register and unregister functions that your code can manually call at initialization and uninitialization time.

−−internal
By default code generated by **−−generate−source** declares all initialization functions as **extern**. So they are exported unless this is prevented by a link script or other means. Since libraries usually want to use the functions only internally it can be more useful to declare them as G_GNUC_INTERNAL which is what **−−internal** does.

−−external−data
By default code generated by **−−generate−source** embeds the resource data as a string literal. When **−−external−data** is given, the data is only declared in the generated C file, and the data has to be linked externally.

−−dependency−file=*FILE*
Write dependencies in the same style as gcc −M −MF to the given file. If **FILE** is −, the dependencies are written to the standard output. Unlike **−−generate−dependencies**, this option can be combined with other **−−generate** options to generate dependencies as a side−effect of generating sources.

−−generate−phony−targets
When creating a dependency file with **−−dependency−file** include phony targets in the same style as gcc −MP. This would typically be used with make.

−−compiler=*NAME*
Generate code that is going to target the given compiler *NAME*. The current two compiler modes are "gcc", for all GCC−compatible toolchains; and "msvc", for the Microsoft Visual C Compiler. If this option isn't set, then the default will be taken from the **CC** environment variable.

ENVIRONMENT

XMLLINT
The full path to the **xmllint** executable. This is used to preprocess resources with the xml−stripblanks preprocessing option. If this environment variable is not set, **xmllint** is searched for in the **PATH**.

GDK_PIXBUF_PIXDATA
Deprecated since gdk−pixbuf 2.32, as **GResource** supports embedding modern image formats without conversion.

The full path to the **gdk−pixbuf−pixdata** executable. This is used to preprocess resources with the to−pixdata preprocessing option. If this environment variable is not set, **gdk−pixbuf−pixdata** is searched for in the **PATH**.

JSON_GLIB_FORMAT
The full path to the **json−glib−format** executable. This is used to preprocess resources with the json−stripblanks preprocessing option. If this environment variable is not set, **json−glib−format** is searched for in the **PATH**.