

Resource Hacker™

... a freeware resource compiler & decompiler for Windows® applications

Version 5.1.8 (360)

Last updated: 20 November 2020

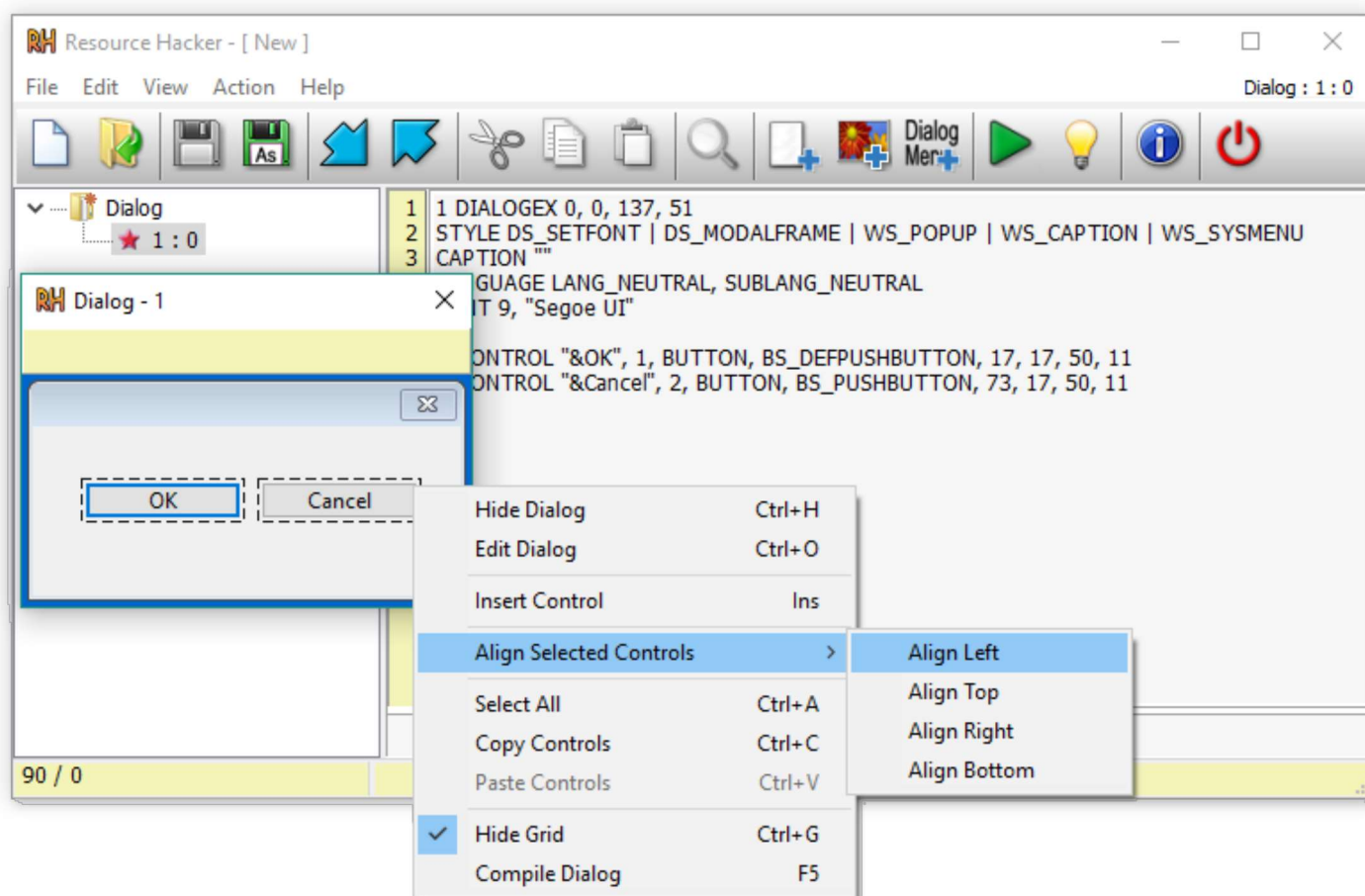
Copyright © 1999-2020 Angus Johnson

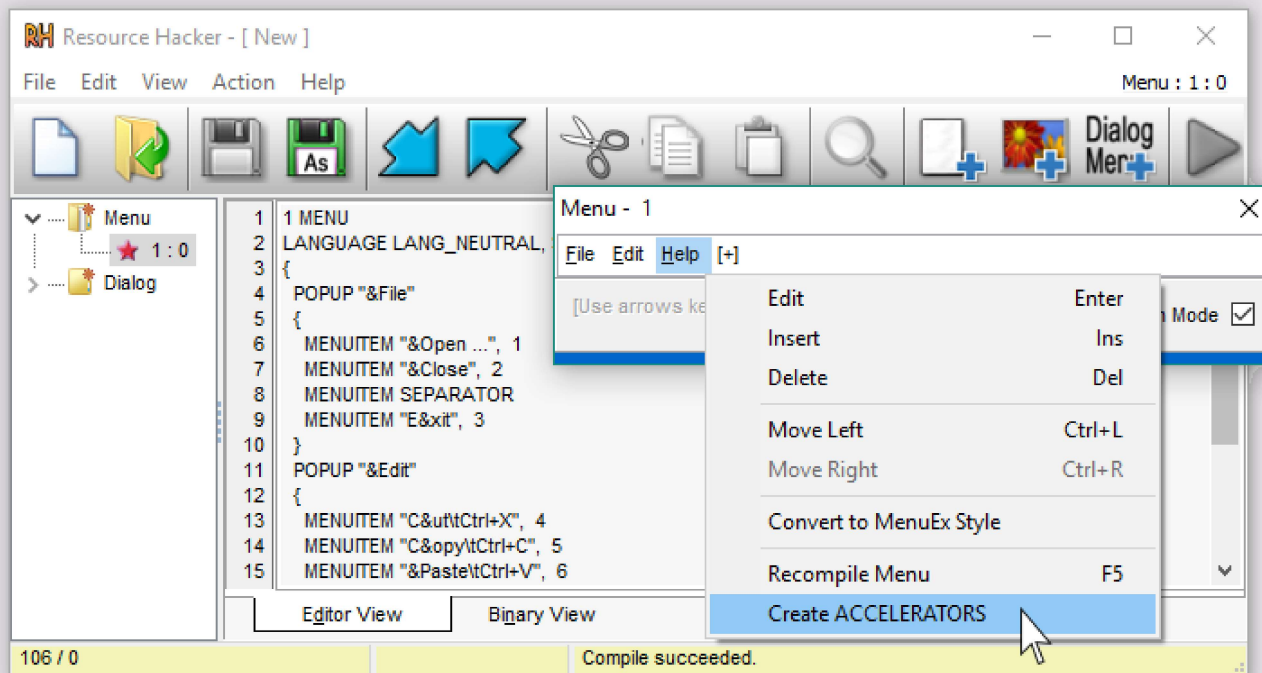
Freeware - no nags, no ads and fully functional.

[Download](#)

Overview:

Resource Hacker™ is a resource editor for 32bit and 64bit Windows® applications. It's both a **resource compiler** (for *.rc files), and a **decompiler** - enabling viewing and editing of resources in executables (*.exe; *.dll; *.scr; etc) and compiled resource libraries (*.res, *.mui). While Resource Hacker™ is primarily a GUI application, it also provides many options for compiling and decompiling resources from the command-line.





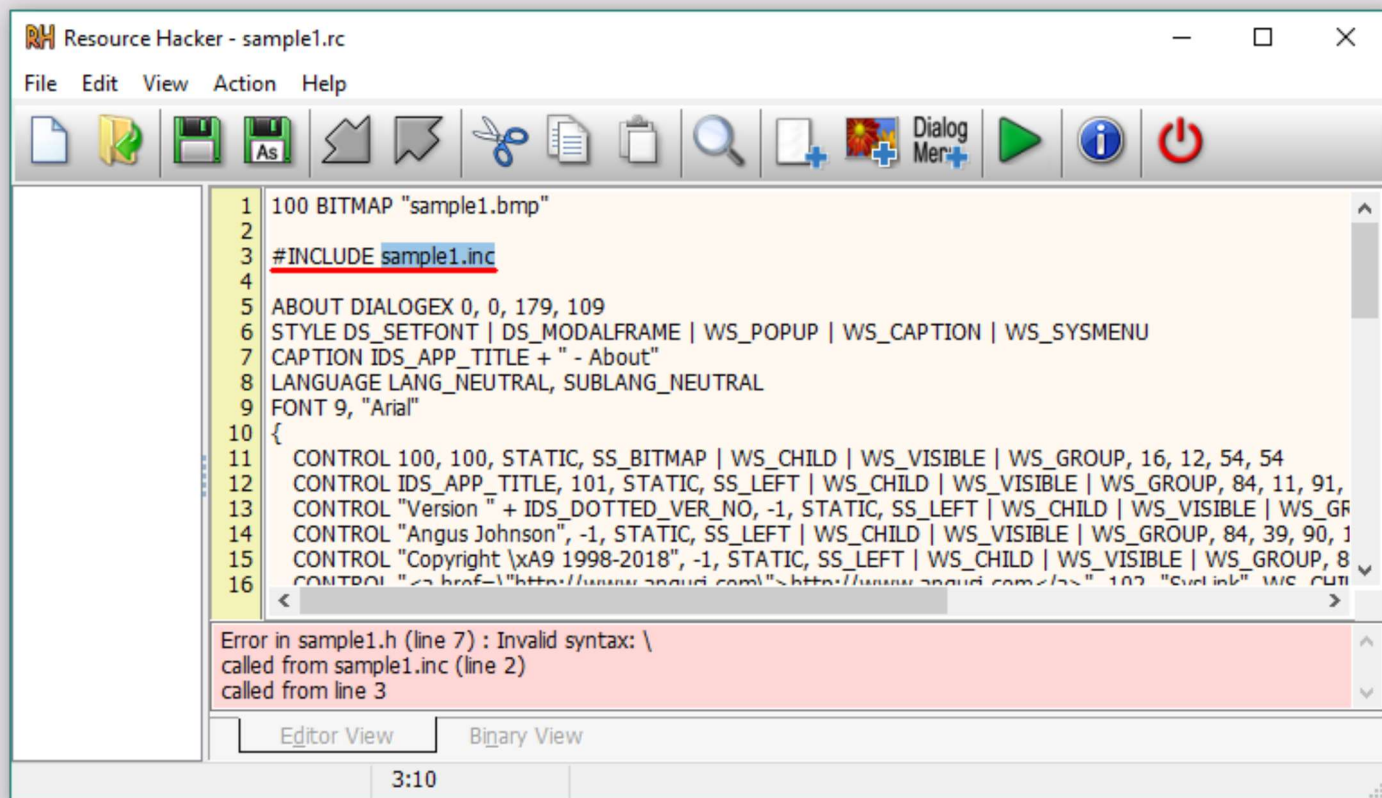
Compiling:

Compiling can be initiated either by opening an existing resource script file, or by creating one from scratch using Resource Hacker's editor.

A complete list of Resource-Definition Statements can be found [here](#).

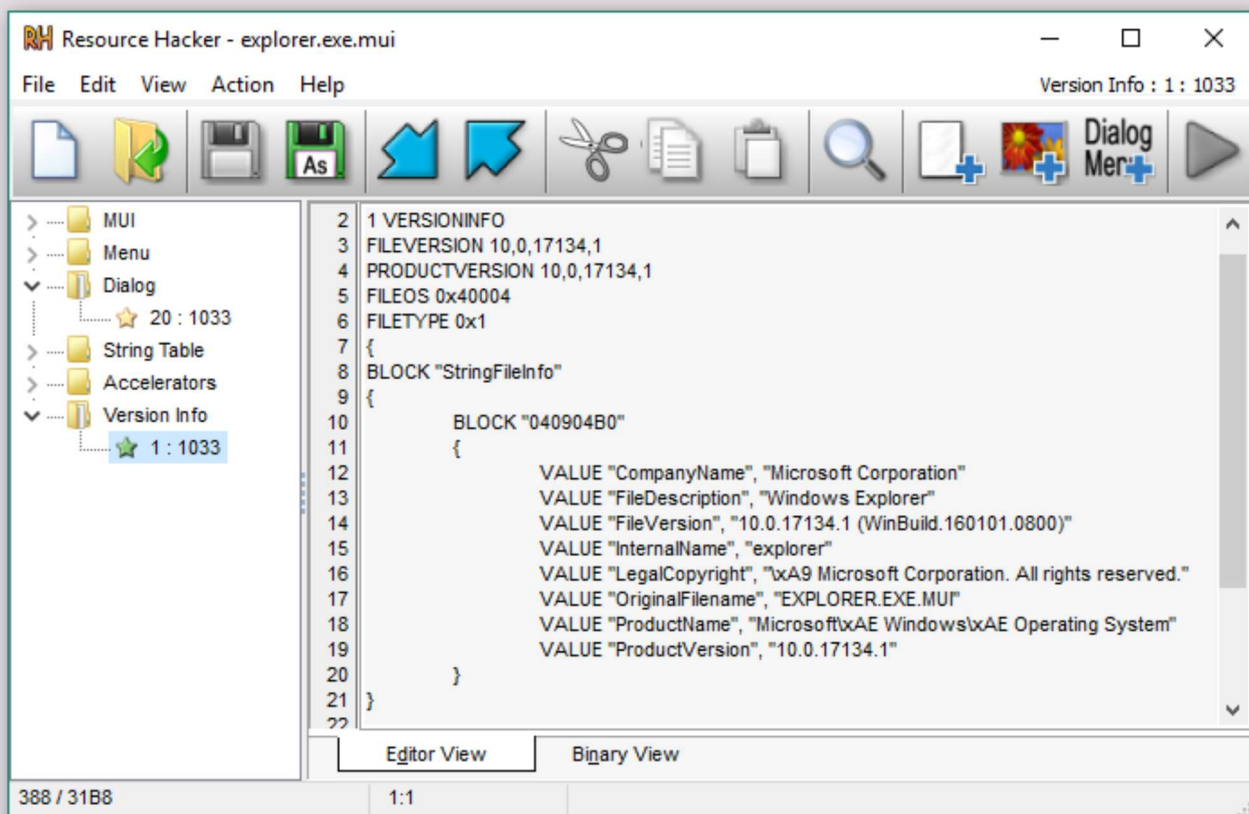
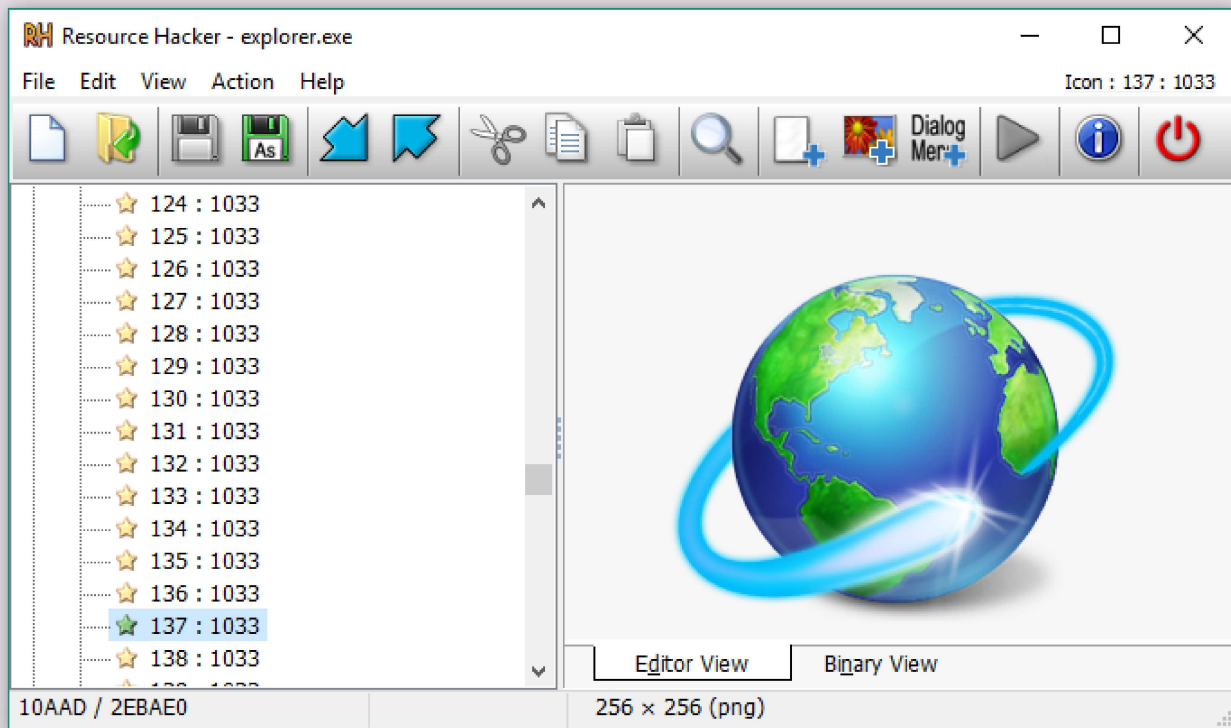
Additional features of Resource Hacker's compiler include: The `#INCLUDE` directive (to access definition statements in header files etc) can be nested to multiple levels, as can the `#IF`, and `#IFDEF` directives. `#DEFINE`, `#UNDEF`, `#IF`, `#ELIF`, `#ELSE`, `#IFDEF`, `#IFNDEF`, `#INCLUDE`, and `#PRAGMA` directives are all supported. Strings, between double-quote (") characters, may contain typical 'C' style backslashed 'escaped' characters — `\t`, `\n`, `\\`, `\"`, `\x`, `\u` and `\377` (octal). A double-quote within a string must be 'escaped' using either a preceding backslash or with another double-quote. Script comments are preceded either by double forward-slashes (//) or by a semi-colon (;). Filenames with relative paths are allowed. Filenames that contain spaces must be enclosed within double-quote characters.

Compiler error messages are reported, even errors nested within `INCLUDE` statements ...

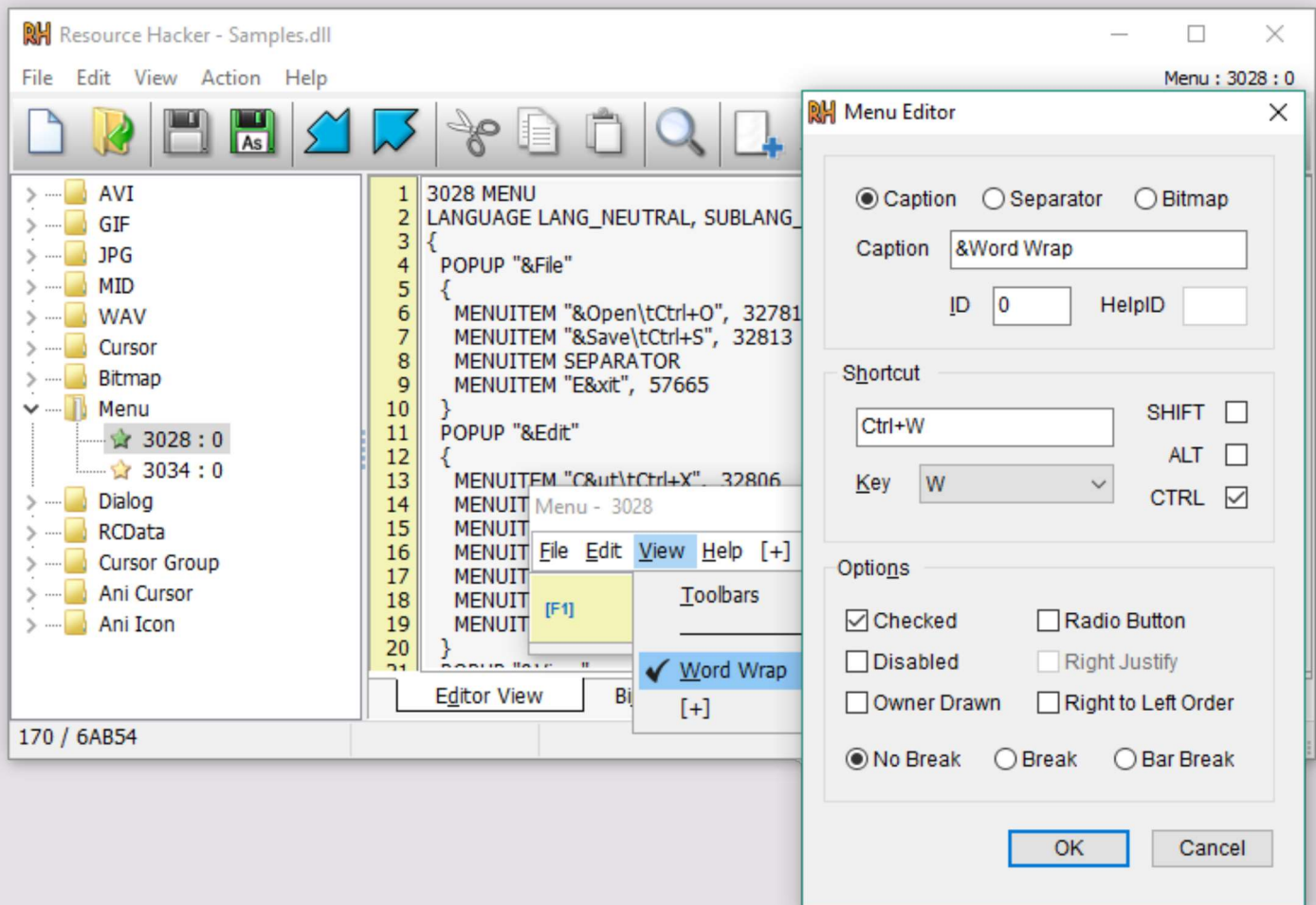


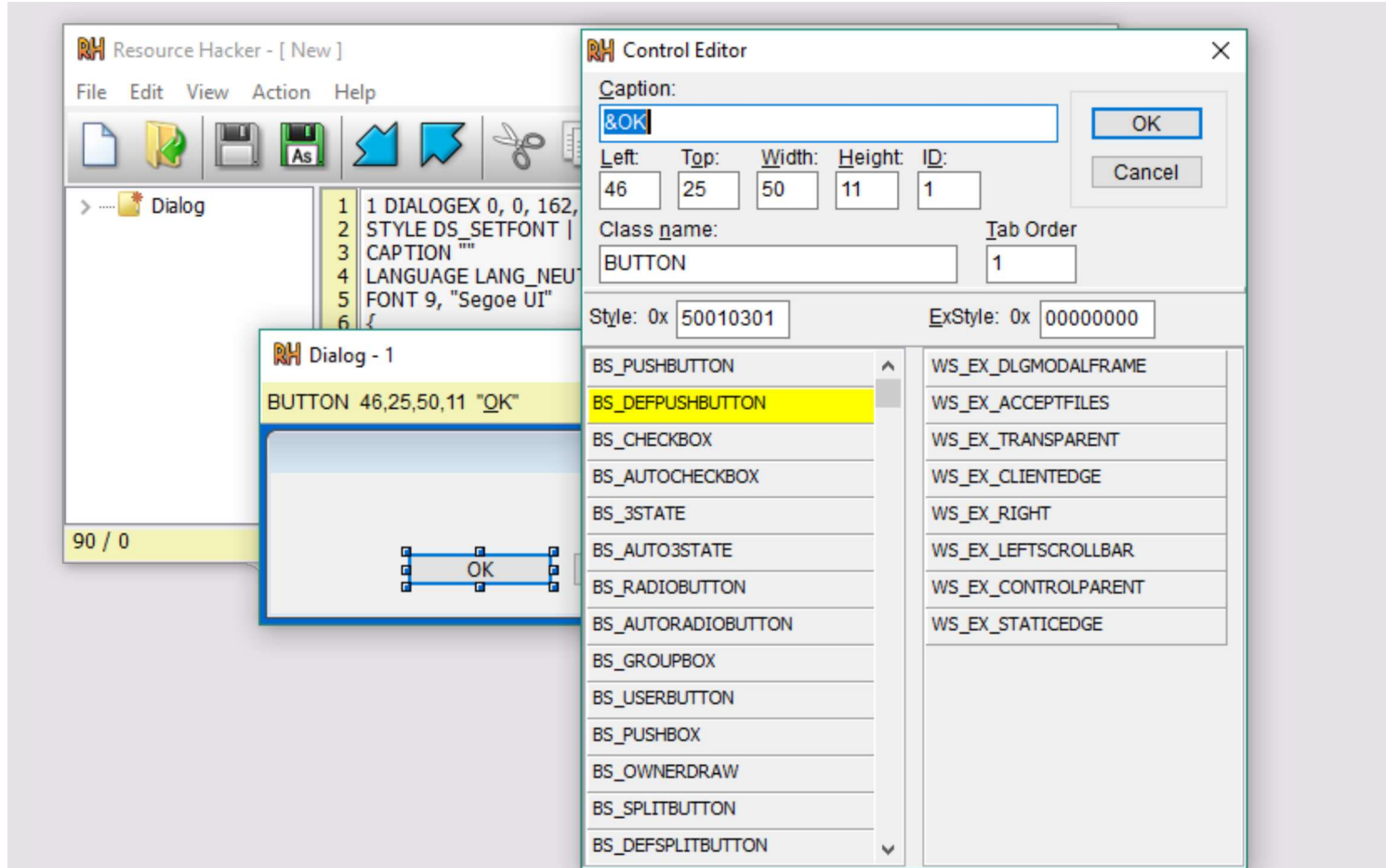
Viewing and Editing Resources:

Once a resource file has been opened, its resources will generally be displayed as either an image (or group of images) or as decompiled text. Binary resources, usually images, can't be edited directly with Resource Hacker, but they can still be very easily exported and imported once they've been modified by an external image editor. (I see no benefit in duplicating what third-party image editors do so well.)

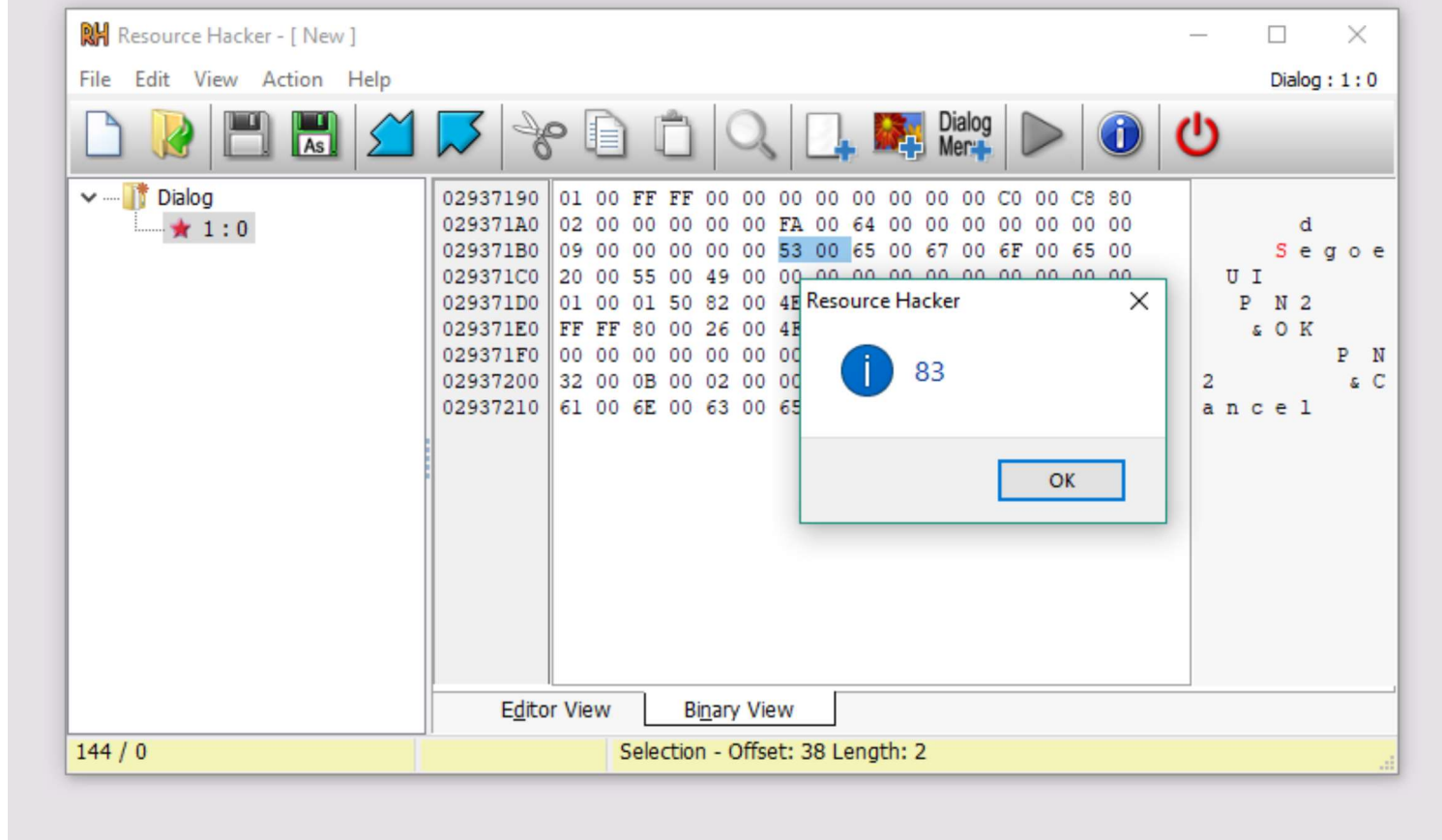


Menu and Dialog resource types have their own WYSIWYG designers:

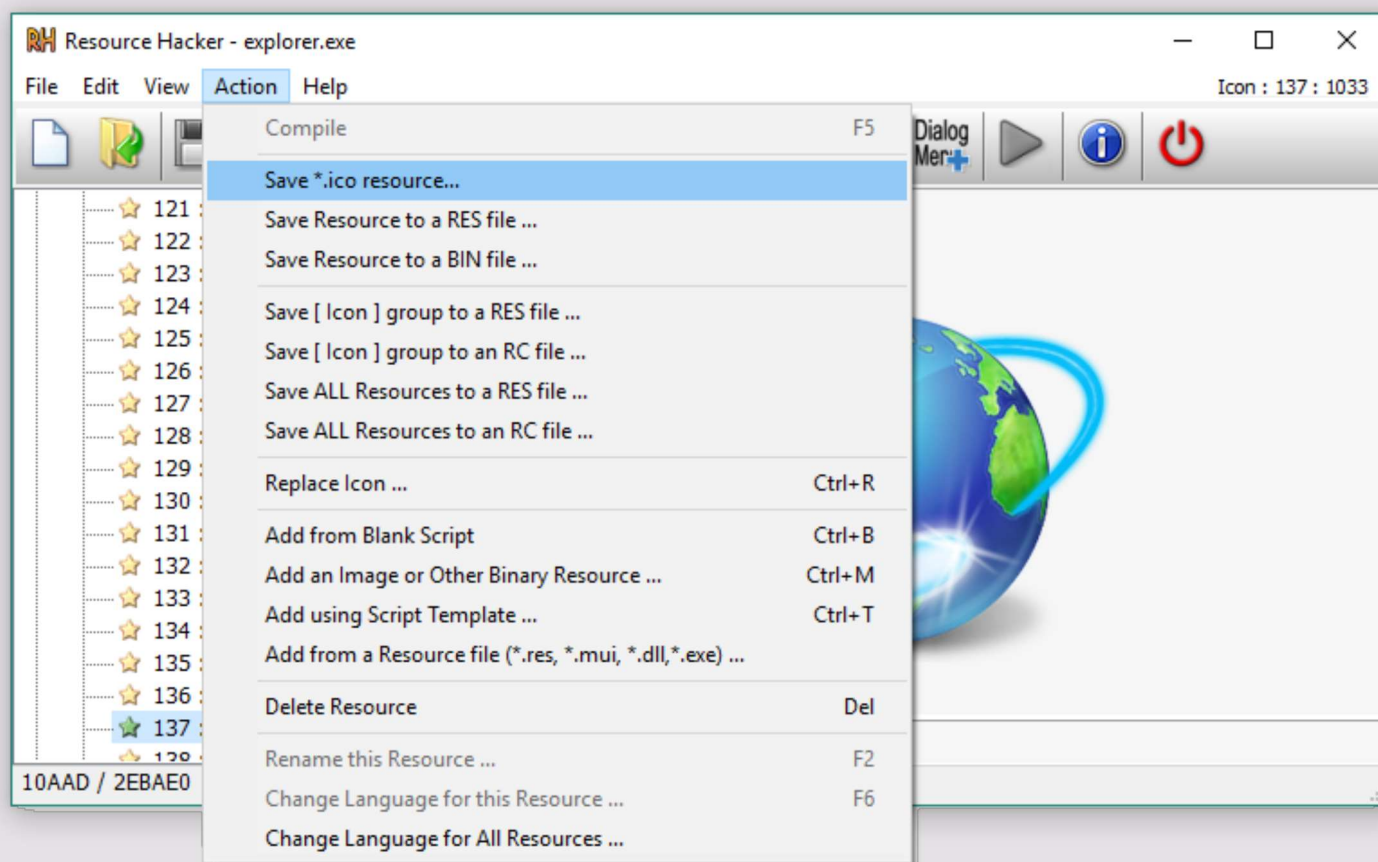




Binary resources that have unknown formats will be displayed as read-only binary text. (Any resource can also be viewed in this fashion if desired.)



Other Actions:



Command Line Syntax:

Just about all the functionality of Resource Hacker™ can be accessed from the command line without having to open the Resource Hacker™ GUI.

Command line instructions and Resource Hacker™ scripts can remove the drudgery entailed with repeating Resource Hacker™ tasks.

Command-line instructions are a combination of switch statements followed by switch parameters as explained in the following table: **Command line statements:**

Switch	Parameter
-open	filename - the name of the file that is to be modified. It should be a Windows PE file (*.exe, *.dll etc) or a compiled or uncompiled resource file (*.res or *.rc)
-save	usually a filename for the new or modified file, but can also be a folder when extracting multiple resources
-resource	filename - contains a resource being added to the opened file.
-action	action to be performed on the opened file <ul style="list-style-type: none"> • add - add a resource, but fails if it already exists • addoverwrite - add a resource, and overwriting if it already exists • addskip - add a resource, but skipping if it already exists • compile - compiles a resource script file (*.rc) to a binary resource file (*.res) • delete - delete a resource • extract - extract a resource • modify - modify a resource

- **changelanguage(langID)** - changes the language of ALL resources
- mask resource mask - Type,Name,Language
commas are mandatory but each of Type, Name & Language are optional
- log Filename or **CONSOLE** or **NUL**
CONSOLE can be abbreviated to **CON**
Logs the details of the operation performed
If this switch is omitted, the log will be written to *resourcehacker.log*
- script filename - contains a multi-command script, NOT a resource script for more info: -help script
- help options - command-line or script (always logged to CONSOLE) other switches are ignored.

Notes:

- Switch identifiers (except -script) may be abbreviated down to a single char (eg -res or -r).
- Switch instructions do not have to be in any particular order.
- File names that contain spaces must be enclosed within double quotes.

Batch file Examples (using rh.exe instead of ResourceHacker.exe in places for brevity)

reshack_help.bat:

```
ResourceHacker.exe -help
@pause :: to see the console output before the CMD window closes.
```

reshack_compile_res_script.bat:

```
rh.exe -open .\in\resources.rc -save .\out\resources.res -action compile -log NUL
```

reshack_add_icon_to_old.bat:

```
rh.exe -open old.exe -save new.exe -action addskip -res my.ico -mask ICONGROUP,MAINICON,
```

reshack_extract_all_source_icons_to_icons_folder.bat:

```
rh.exe -open source.exe -save .\icons -action extract -mask ICONGROUP,, -log CON
@pause
```

reshack_extract_all_dialogs.bat:

```
rh.exe -open source.exe -save .\dialogs.rc -action extract -mask DIALOG,, -log rh.log
```

reshack_execute_my_script.bat:

```
ResourceHacker.exe -script myscript.txt
```

Resource Hacker™ Scripts:

Resource Hacker™ Scripts are executed from the command-line using the following syntax:

```
ResourceHacker.exe -script ScriptFile
```

Resource Hacker™ Scripts are text files with the following format:

```
//comments are preceded by double slashes
[FILENAMES]
Open=
Save=
Log=
[COMMANDS]
//one or more of the following commands ...
-add            SourceFile, ResourceMask
-addskip       SourceFile, ResourceMask
-addoverwrite SourceFile, ResourceMask
-addoverwrite SourceFile, ResourceMask
-modify        SourceFile, ResourceMask
```



```
-extract      TargetFile or TargetFolder, ResourceMask
-delete      ResourceMask
-changelanguage(langID)
```

Note: Filenames that include spaces must be enclosed within double quotes.

Resource Hacker™ Script examples:

rh_script_myprog_rus.txt -

```
//This script deletes all Language Neutral (0)
//string-table, menu and dialog resource items
//in MyProg.exe before replacing them
//with Russian (1049) items...
[FILENAMES]
Exe=      MyProg.exe
SaveAs=   MyProg_Rus.exe
Log=      MyProg_Rus.log
[COMMANDS]
-delete   MENU,,0
-delete   DIALOG,,0
-delete   STRINGTABLE,,0
-add      MyProg_Rus.res, MENU,,1049
-add      MyProg_Rus.res, DIALOG,,1049
-add      MyProg_Rus.res, STRINGTABLE,,1049
```

rh_script_myprog_upd_images.txt -

```
//This script updates 2 bitmaps and an
//icon in MyProg.exe ...
[FILENAMES]
Exe=      MyProg.exe
SaveAs=   MyProg_Updated.exe
[COMMANDS]
-addoverwrite Bitmap128.bmp, BITMAP,128,
-addoverwrite Bitmap129.bmp, BITMAP,129,0
-addoverwrite MainIcon.ico, ICONGROUP,MAINICON,0
```

rh_script_myprog_upd_all.txt -

```
//This script replaces all resources
//in MyProg.exe with all the resources
//in MyProgNew.res
[FILENAMES]
Exe=      MyProg.exe
SaveAs=   MyProg_Updated.exe
[COMMANDS]
-delete   ,,, //delete all resources before...
-add      MyProgNew.res ,,, //adding all the new resources
```

"Packed" or "Compressed" Executables:

Some executable files are "packed" or "compressed" using compression algorithms. Not only does this reduce file size, it also makes viewing and modifying resources marginally more difficult. I suspect that this resource 'hiding' is (or was) a common objective in this process. Anyhow, in deference to these authors, I've chosen not to unpack files with Resource Hacker. As a side note, it seems that "packed" executables have become quite uncommon over the last 20 years, and software authors are exposing more rather than less information in executable resources. I suspect that earlier concerns about the loss of intellectual property with reverse engineering have been allayed.

Licence to Use - Terms and Conditions:

This Resource Hacker™ software is released as freeware provided that you agree to the following terms and conditions:

- This software is not to be distributed via any website domain or any other media without the prior written approval of the copyright owner.
- This software is not to be used in any way to illegally modify software.

DISCLAIMER: A user of this Resource Hacker™ software acknowledges that he or she is receiving this software on an "as is" basis and the user is not relying on the accuracy or functionality of the software for any purpose. The user further acknowledges that any use of this software will be at the user's own risk and the copyright owner accepts no responsibility whatsoever arising from the use or application of the software.

The above licence terms constitute "copyright management information" within the meaning of Section 1202 of Title 17 of the United States Code and must not be altered or removed from the licensed works. Their alteration or removal from the licensed works, and the distribution of licensed works without all the above licence terms in an unaltered way, may contravene Section 1202 and give rise civil and/or criminal consequences.

Download version 5.1.8:

EXE install (4MB)

ZIP install (3MB)

Changes in 5.1.8 (360):

Bugfix: Some PNG files were no longer being displayed.

Changes in 5.1.8:

Added: Zooming/scrolling views of image resources

Update: Enabled scrolling for very tall dialogs

Bugfix: Very large binary resources (>>100Mb) could crash the hex editor

Bugfix: Occasional bug decompiling dialogs styles (WS_TABSTOP)

Bugfix: RC file missing when extracting a binary resource type from commandline