

DelphiDabbler HTML Resource Compiler

User Guide

Command Line

The command line syntax of the application is:

```
HTMLRes in-file-list [-mmanifest-file] [-oout-file] [switches]
```

where:

- *in-file-list* is a sequence of zero or more input file names that are to be stored in the resource file. Relative file names are relative to the folder from where *HTMLRes* was started. From v1.1 the *-r* switch may be used to change this behaviour.
- *manifest-file* is a file containing a list of file names to be included in the resource file. Each file name is stored on a separate line in the manifest file. Blank lines are allowed and lines beginning with *#* are treated as comments and are ignored. Comments cannot be placed on the same line as a file name. Here's an example file:

```
# Files required for demo
index.html
page2.html
arrow.gif
style.css
```

Spaces are not allowed between the *-m* switch and the manifest file name. The file names specified in the manifest are, by default, relative to the folder from where *HTMLRes* was executed. From v1.1 you can change this so that the file names are relative to the manifest file's folder by using the *-r* switch (see below).

- *out-file* is the name of the output resource file. This switch is optional. If the *-o* switch is not provided then the default name *out.res* is used. Spaces are not allowed between the *-o* switch and the output file name. From v1.2 the *-u* switch controls how existing output files are handled (see below).
- *switches* is a sequence of zero or more of:
 - q Quite mode. Does not display any normal output. Any error messages are displayed.
 - Q Silent mode. No normal or error messages are displayed.
 - p or -P Causes the program to pause on completion and prompts the user to press return before closing the program. The prompt is always displayed, regardless of whether the *-q* or *-Q* switches have been used.

- r or -R** Causes all relative file names to be taken as relative to the manifest file's directory. If this switch is not specified file names are relative to the directory where *HTMLRes* was executed. The switch applies to the preceding **-m** manifest file command. If there is no **-m** switch then this switch is ignored. **-r** applies to all relative file names, i.e. input file, output file and all files listed in a manifest file.
- u: param or -U: param** Specifies the action to be taken if an output file already exists. **param** indicates the action to take:
- u: fail**
The program fails with an error if the output file does not exist. The output file is not changed.
 - u: overwritefile**
The output file is overwritten (default).
 - u: insertres**
Inserts HTML resources into the existing file. An error is reported if the output file is not a valid resource file or if any resource name already exists in the file. The output file is not modified if an error occurs.
 - u: overwriteres**
Inserts HTML resources into the existing file, overwriting existing resources if there is a name clash. It is an error if the output file is not a valid resource file. In the event of an error the output file is not modified.
- H, -h or -?** Displays a help screen. The **-q** and **-Q** switches are ignored.

At least one input file must be provided, either directly on the command line or in the manifest file unless the **-H, -h or -?** switches are used. See below for restrictions placed on file names.

Error Codes

The program returns 0 on success or an explanatory error code on failure. The error codes are:

Error code	Explanation
1	Command line error.
2	Manifest file expected but doesn't exist.
3	One or more source files don't exist.
4	Duplicate file name.
5	Output file exists (the -u: fail switch has been used).
6	Output file exists but is not a valid resource files (the -u: insertres or -u: overwriteres switch has been used).
7	Resource name already exists in output file (the -u: insertres or -u: overwriteres switch has been used).
255	Unexpected / unknown error.

File Name Restrictions

Since resource names cannot be duplicated in a resource file, and since base file names (without the path) are used as resource names, it follows that duplicate base file names cannot be specified either on the command line or in a manifest file. For example you can't specify both `C:\Foo\MyFile.htm` and `C:\Bar\MyFile.htm`.

If you are intending use the `res://` protocol to access the HTML resources from Internet Explorer or an embedded Web Browser control there are some restrictions on the resource names, and hence the files names, that you can use. Certain file names that begin with a digit or a punctuation character are not recognised and not loaded. Examples of names to avoid are:

- `42.html`
- `%3.css`
- `4-a.jpg`

Compatibility

Not all features are available in all versions of *HTMLRes*. Specifically:

- The `-r` switch is available in v1.1 and later.
- The `-u` switch is available in v1.2 and later.

Example

An example *Delphi* project is supplied with the program. The demo files are installed in the Demo sub-folder of the main program installation folder.

The demo project simply creates a resource only DLL that stores some HTML, CSS and GIF files that can be displayed in Internet Explorer.

The demo files are:

File Name	Description
<code>index.html</code>	The main demo web page.
<code>page2.html</code>	A subsidiary web page.
<code>style.css</code>	A style sheet for the web pages.
<code>arrow.gif</code>	A GIF displayed in <code>index.html</code> .
<code>Demo.hmfst</code>	A manifest file that lists the demo files to be built by <i>HTMLRes</i> .
<code>HTMllib.dpr</code>	The <i>Delphi</i> source code for the resource-only DLL.
<code>DemoReadMem.txt</code>	ReadMe file for demo code. Please read before using the demo code.

Before building the demo project using *Delphi* you must first start a command window (DOS box) and navigate to the demo directory. Now run *HTMLRes* as follows:

```
HTMLRes -mDemo.hmfst -oHTML.res
```

This compiles the files specified in `Demo.hmfst` into the resource file `HTML.res`.

The project file, `HTMLLib.dpr`, contains a resource statement that includes `HTML.res`. The project can now be compiled with *Delphi* to create the required resource DLL.

If the demo DLL is stored in the `C:\Program Files\DelphiDabbler\HTMLRes\Demo` folder we can display the main page in Internet Explorer by entering the following URL in the address bar:

```
res://C:\Program Files\DelphiDabbler\HTMLRes\Demo\HTMLLib.dll/index.html
```

The HTML file, its style sheet and the graphic are all stored in the DLL's resources along with a second page that can be displayed by clicking the link on the index page.

This user guide is copyright © Peter D Johnson, 2007-2008, www.delphidabbler.com.

It is licensed under a [Creative Commons License](http://creativecommons.org/licenses/by-sa/4.0/)

