

## Doomsday Engine - Bug #1038

### [Unix] Locating resource files (wad/pk3)

2012-05-11 00:33 - alepulver

<b>Status:</b> Closed	<b>Start date:</b> 2012-05-11
<b>Priority:</b> Low	<b>% Done:</b> 80%
<b>Assignee:</b>	
<b>Category:</b> Defect	
<b>Target version:</b> 1.9.7	
<b>Description</b> I'm using Doomsday 1.9.8 build 495 on Ubuntu 12.04 64-bits, but the problem also happens with build 490 and 1.9.7 stable.  After selecting DOOM.WAD or DOOM2.WAD from Snowberry, the program doesn't recognize them at startup. I've ruled out the possibilities of case-sensitive issues or spaces in paths. Curiously, if I add a game without a WAD file (nonexistent path) in Snowberry, then the other WADs are recognized and I can play.  Using "doomsday -file doom/doom.wad" doesn't work either. I have attached the output of "doomsday -v -v doom/doom.wad". If you need more information I'll try to provide it.  <b>Labels:</b> Startup	
<b>Related issues:</b> Related to Feature #1647: Replace FS1 with FS2-based resource management <b>Progressed</b> <b>2013-10-31</b>	

#### History

##### #1 - 2012-05-09 04:18 - alepulver

Output of "doomsday -v -v doom/doom.wad"

##### Attachments:

- [http://sourceforge.net/p/deng/bugs/\\_discuss/thread/30e1def2/f709/attachment/log.txt](http://sourceforge.net/p/deng/bugs/_discuss/thread/30e1def2/f709/attachment/log.txt)

##### #2 - 2012-05-11 00:33 - skyjake

Note that the correct way to specify the IWAD is:

```
doomsday -iwad /full/path/to/doom/doom.wad
```

You can also specify a directory:

```
doomsday -iwad /full/path/to/doom/
```

If a relative path is specified, it is assumed to be relative to ~/.doomsday, not pwd.

##### #3 - 2012-05-18 07:40 - alepulver

Thanks for the reply.

About the relative path, if you consider it a feature instead of bug, then I think it should be documented in the manpage at least. Or even better, print a message for each -iwad argument like "searching %s for IWADs" so it would be obvious to users.

There is another issue that I consider a bug: paths with spaces not working. Should I fill another bug report for this one?

##### #4 - 2012-05-18 15:35 - danij

Paths with spaces work fine, you must enclose them in parenthesis so that the engine knows its one argument rather than several (the space character is an argument separator).

For example:

```
doomsday -iwad "path/to/some/folder with a space/doom.wad"
```

I'm pretty sure this and the Doomsday install relative paths are already documented in the manpage (I could be wrong, I'm a Windows user).

##### #5 - 2012-05-18 16:19 - alepulver

```
strace -f doomsday -iwad "$PWD/blah blah/" | grep blah
```

**Attachments:**

- [http://sourceforge.net/p/deng/bugs/\\_discuss/thread/30e1def2/94aa/attachment/sp.txt](http://sourceforge.net/p/deng/bugs/_discuss/thread/30e1def2/94aa/attachment/sp.txt)

**#6 - 2012-05-18 17:06 - alepulver**

```
strace -f doomsday -iwad "$PWD/blah/" | grep blah
```

**Attachments:**

- [http://sourceforge.net/p/deng/bugs/\\_discuss/thread/30e1def2/5681/attachment/nosp.txt](http://sourceforge.net/p/deng/bugs/_discuss/thread/30e1def2/5681/attachment/nosp.txt)

**#7 - 2012-05-18 17:07 - alepulver**

About documentation of relative paths, the command (that searches for the strings "absolute" and "relative"):

```
$ man doomsday | grep -i "absolute|relative"
```

Does not match anything.

I know about escaping space characters from the shell, and it's not working for me. As a proof I have attached two system call traces. One for "doomsday -iwad "\$PWD/blah/" (see attached nosp.txt) and other for "doomsday -iwad "\$PWD/blah blah/" (see attached sp.txt). Both directories "blah" and "blah blah" exist and contain DOOM.WAD, but only in "blah" it's detected as you can see in the logs. You can also see at the top of both logs that the command line arguments aren't split, so it should work.

BTW I think having to add a trailing "/" to directories for -iwad to work is not intuitive. I've done so in the previous commands, but it wasn't documented either. In my opinion it should detect if the path is a directory or not, and act appropriately.

**#8 - 2012-05-18 17:11 - danij**

As I said, you need to escape paths for Doomsday also. It's been a while since I last used a Unix-based platform from the command line but have you tried:

```
doomsday -iwad ""$PWD/blah blah/""
```

By the looks of your output files the issue is that the path is not being passed to Doomsday wrapped with the necessary parentheses, resulting in the following instead:

```
/home/ale/.doomsday/runtime/blah
```

**#9 - 2012-05-18 17:57 - alepulver**

This means that `argv2 = "/home/ale/Downloads/blah blah/"`:

```
execve("/usr/bin/doomsday", ["doomsday", "-iwad", "/home/ale/Downloads/blah blah/"], [/* 66 vars */]) = 0
```

There should be no need to add "" around when it's already clearly delimited by the normal NULL end-of-string character, at least you can put many paths under the same -iwad argument (which in case to be true, is not documented too).

Actually it works with:

```
$ doomsday -iwad "\"$PWD/blah blah/\""
```

(which I guess it's what you meant for adding real " characters to argv<sup>2</sup>)

But it's the only Linux program I know of that requires real quotes for this (i.e. it's not considered a normal practice).

The same goes for requiring that trailing "/". For example if you have a directory "blah", the command "cp DOOM.WAD blah" copies DOOM.wad into blah/, and all users expect this kind of behavior when specifying directories. The only exception I know of is rsync, and they have a big note in the manpage.

This is what happened to me, and I think other Linux users which use the console will experience:

```
$ doomsday -iwad DOOM.WAD
```

```
(nothing)
```

```
$ doomsday -iwad .
```

```
(nothing)
```

```
$ doomsday -iwad $PWD
```

```
(nothing)
```

```
$ doomsday -iwad $PWD/
```

```
(now it loads, in case there weren't spaces)
```

```
$ doomsday -iwad "$PWD/"
```

(at this point, supposing there were spaces, I don't think most users would guess to double quote it, as it's not necessary for a process to separate an argument and no other programs require it)

I understand you use Windows, so if you don't believe me please try it for yourselves on Linux or ask other users to load a custom IWAD from the command line (both directly and from a folder) to hear their opinions about it.

**#10 - 2012-05-18 18:26 - danij**

I guess that the reason you've had so much trouble with this is that you haven't yet tried to specify more than two arguments on the command line?

Using your example, `argv2` is supposed to be `"/home/ale/Downloads/blah blah/"`. Not supplying parentheses results in `argv[2: /home/ale/Downloads/blah , 3: blah/ ]` i.e., three arguments rather than two.

How should Doomsday know to repair your arguments by combining the last two?

#### #11 - 2012-05-18 19:21 - alepulver

I guess that the reason you've had so much trouble with this is that you haven't yet tried to specify more than two arguments on the command line?

I'm glad to see your reply is not unreasonable. It was just had a misunderstanding.

Could you please tell me where in my logs did you see the `argv[2: /home/ale/Downloads/blah , 3: blah/ ]` behavior?

As far as I can see, `sp.txt` (which has spaces) says:  
`execve("/usr/bin/doomsday", ["doomsday", "-iwad", "/home/ale/Downloads/blah blah/"], [/* 66 vars */]) = 0`

And `nosp.txt`, which uses `"$PWD/blah/"` (which has no spaces) says:  
`execve("/usr/bin/doomsday", ["doomsday", "-iwad", "/home/ale/Downloads/blah/"], [/* 66 vars */]) = 0`  
Which is correct, since this is the log that looks for a folder name without spaces.

I think it was stupid from my part to use the same prefix for both folder names ("blah" and "blah blah"). It should have been clear if I used something like "nospaces" and "with spaces". But remember that they are two different folders.

How should Doomsday know to repair your arguments by combining the last two?

As I've shown, the arguments don't need to be repaired as they are together. The bug is in the program. Please review what I've said before. Sorry for using two unrelated names in which one is prefix of the other.

#### #12 - 2012-05-18 19:31 - danij

You'll need to bear with me not being a Unix user, I'm interpreting as I go. I see what the issue is here, the shell is processing `"$PWD/blah blah/"` and supplying the result, i.e. `"/home/ale/Downloads/blah blah/"` (without quotes) as argument 2.

You are indeed correct, the problem you are having is due to Doomsday not considering the behavior of the Unix shell. This will be fixed.

#### #13 - 2012-05-18 20:07 - alepulver

No problem. Sorry if I sounded harsh before. I appreciate your efforts put into this project.

I believe fixing this will also make it work from the GUI. In the original description I said "ruled out the possibilities of spaces in paths", but I was wrong. My Windows disk mounted at `/media` has folders with spaces in the name, and that probably caused the problem with Snowberry I mentioned. After selecting a path without spaces it works. But I could'n make it work by enclosing the path in double quotes from the GUI, in a similar way I did at the command-line.

If you want, I could try writing a patch and attaching it for review.

BTW the same should happen in Mac OS X, as it uses bash (or a similar one) I think. I'm actually surprised this works in Windows, as I thought it had a shell derived from Bourne (but well, Windows it not what I use for development).

#### #14 - 2012-05-19 01:26 - skyjake

For historical reasons, Doomsday's command line parsing is done differently on Windows compared to Unix. Qt is able to help us with the parsing, but I haven't yet had a chance to address the situation. I can take a look at this now, though. The double-quoting discussed below is not something we should be doing.

I agree that the `-iwad` issue with not detecting directories is an annoyance that needs to be fixed. I've added it to our todo list. (A patch for this would be nice. :))

As a Unix user I can attest that the way relative paths is interpreted is unconventional and lacks proper documentation. Personally I would prefer changing it to be relative to the `pwd`. Initially Doomsday was a Windows-only GUI application, which still strongly influences our command line behavior. The situation will improve over time.

#### #15 - 2012-05-19 06:11 - skyjake

I wrote a proposal to address this issue:

**#16 - 2012-05-19 08:49 - alepulver**

Thanks for adding it to the roadmap. I would like to see a short term solution also, so I fixed string separation with the following code.

Unfortunately, now it finds the IWAD well but ignores the jdoom.pk3 file. I don't really understand why that happens (as the command is "doomsday -iwad /my\ path/" or similar). I'll take a closer look these days.

Are there updated build and installation instructions? Maybe I did something wrong. I just used the qt-make command in debian/rules, "make" and "make install".

```
static char* buildCommandLineString(int argc, char** argv) {
char* cmdLine;
int i, length = 0;

// Assemble a command line string.
for(i = 0; i < argc; ++i)
{
length += strlen(argv[i]) + 3;
}

// Allocate a large enough string.
cmdLine = M_Calloc(length);

length = 0;
for(i = 0; i < argc; ++i)
{
cmdLine[length++] = '"';
strcpy(cmdLine + length, argv[i]);
length = strlen(argv[i]);
cmdLine[length+] = '"';
if(i < argc - 1)
cmdLine[length++] = ' ';
}
return cmdLine;
}
```

**#17 - 2012-05-19 21:32 - skyjake**

Friday's build 511 introduces the new command line parsing discussed in the proposal. Spaces should now be interpreted correctly, and relative paths for various command line options are relative to the pwd, not the Doomsday runtime directory.

Let us know how the new code works for you.

**#18 - 2012-05-24 04:33 - alepulver**

Compilation error from git

**Attachments:**

- [http://sourceforge.net/p/deng/bugs/\\_discuss/thread/30e1def2/bb3c/attachment/buildlog.txt](http://sourceforge.net/p/deng/bugs/_discuss/thread/30e1def2/bb3c/attachment/buildlog.txt)

**#19 - 2012-05-25 03:47 - alepulver**

In the webpage the last build is 507. I've tried to compile from git but got an error (see the attached buildlog.txt).

**#20 - 2012-05-25 03:49 - skyjake**

Build 511 will be available in about 6 hours.

It looks you're linking against the wrong libdeng2.so, perhaps it's trying to use one installed in a system-wide library directory?

**#21 - 2012-05-25 04:03 - alepulver**

After running "make clean", and removing "-j4" (for parallel building) it compiled.

But when executing, it does the same as I previously reported after patching the Unix path routine. When I type "load doom1" for example, it says the IWAD is found but jdoom.pk3 not (and it's at /usr/share/doomsday/data/jdoom/).

strace filtered by jdoom attached as trace.txt.

**#22 - 2012-05-25 04:25 - alepulver**

strace -f doomsday -iwad <path...> | grep -i 'jdoom\|wad'

## Attachments:

- [http://sourceforge.net/p/deng/bugs/\\_discuss/thread/30e1def2/1ed7/attachment/trace.txt](http://sourceforge.net/p/deng/bugs/_discuss/thread/30e1def2/1ed7/attachment/trace.txt)

### #23 - 2012-05-25 04:26 - skyjake

Could you try the 511 build from the code.iki.fi repository when it's available?

I find it strange that it's trying to access "/usr/share/doomsday/Data/jdoom" (capital D on Data), when no one should be doing that in the code.

When it's compiling, what is the value defined for DENG\_BASE\_DIR in the call to gcc? That's what is used for locating the .pk3s under /usr/share/doomsday/data/.

### #24 - 2012-05-25 06:28 - alepulver

In engine/Makefile, DEFINES contains -DDENG\_BASE\_DIR="/usr/share/doomsday/^\\"

BTW the ^ is not what you want (here it's fine because there are no spaces in /usr/share/doomsday, and I agree no sane person would put a space in a system path, but just to let you know).

The escape (\) stops the quote (") from actually grouping consecutive arguments: prog \"one two\" gets argv<sup>1</sup> as |one| and argv<sup>2</sup> as |two|. On the other hand: prog "one two" gets argv<sup>1</sup> as |one two|.

The "Data" probably comes from somewhere else. Actually "Data" is the last directory of the specified path, however I don't think these two could get mixed.

I'll try the 511 build tomorrow.

### #25 - 2012-05-25 06:51 - alepulver

About my comment on -DDENG\_BASE\_DIR="/usr/share/doomsday/^\\" , there was something missing: as you do want a " in the command, the correct way to do it is:

```
-DDENG_BASE_DIR="\\"/usr/share/doomsday/^\\""
```

Which (if you look below) is the same way I used to add the quote to the doomsday command before this patch. That way the compiler will see |-DDENG\_BASE\_DIR="/usr/share/doomsday/"|, correctly interpreting the define as a string and not an illegal group of tokens (in case of -Dprintf=myprintf it would be fine without \\", the same as -Dprintf="myprintf").

I hope it clarifies quoting in the Unix shell.

### #26 - 2012-05-25 07:20 - skyjake

Thanks for the proper quoting, I applied it to engine.pro.

### #27 - 2012-05-25 09:09 - alepulver

OK.

Build 511 behaves the same. I'll try to track this problem tomorrow.

### #28 - 2012-05-26 04:35 - alepulver

I think I've identified most of the problems:

1)

Adding file path |/home/ale/temp/deng/builddir/blah one|  
resulting path is |Packages:/home/ale/temp/deng/builddir/|

It always removes the last path component, ignoring if it's a directory or not. So it has the effect of finding any IWADs under the same directory for files, and failing for directories (at least you append a "/something" just to let it remove it).

2)

resulting path is |\$(App.DataPath)/|  
resulting path is |\$(Game.DataPath)/|  
resulting path is |\$(App.DefsPath)/|  
resulting path is |\$(Game.DefsPath)/|  
resulting path is |\$(Game.DefsPath)/\$(Game.IdentityKey)/|  
resulting path is |\$(App.DataPath)/graphics/|  
resulting path is |\$(Game.DataPath)/models/|  
[...]

I think these should be real paths and the substitution failed somewhere. That's why it didn't load jdoom.pk3.

---

In general, I think the code tries to do too many things to the paths, when just leaving them alone should work (in case you don't change current directory). I guess that's for Windows portability (because for Mac OS X I believe it's the same as Unix).

For example, in engine/portable/src/dd\_main.c you shouldn't try to replace ~/ for /home/user: the shell does it (you can try it with "echo", which just prints arguments and has no other code inside). And if you write "doomsday file" the shell will ignore spaces when running doomsday (so you don't need to skip blanks, and if you do the program will probably fail for file names starting with spaces). That's why command-line parsing has been traditionally done by a very simple function (getopt).

I'll review in detail the code to make sure everything works fine (and trying not to break other things), but it may take a few days.

#### #29 - 2012-05-26 06:14 - alepulver

I've attached doomsday.diff, which removes code looking for "~/" in Unix, and fixes IWAD paths to files. Previously if the path was a directory, the last component was stripped away as if it was a file.

Now the only remaining problem is reading more IWADs than specified (see FIXME note at libdeng2/src/core/commandline.cpp). For example, if you use "-iwad path/to/DOOM.WAD" it may load DOOM2.WAD if it's in the same directory, because the part "/DOOM.WAD" gets removed (i.e. -iwad only supports paths).

You could either fix it or allow directories only. In the case of supporting both, I think it would be cleaner to use two separate options (like -iwad and -iwaddir).

About missing jdoom.pk3, the problem is not the expansion of "\${App...}" as I thought before; it's more tricky. If the last component of the -iwad path is "/Data" or "/Data/SOME.WAD", it tries to load ddBasePath/Data instead of ddBasePath/data. It doesn't happen with a path that ends in "/data/..." (lowercase 'd') or anything else.

I'll try to fix this one these days. If you have any clues, let me know.

#### #30 - 2012-05-28 00:50 - alepulver

Patch to fix -iwad dir, and ignore ~/ (exp by shell)

#### Attachments:

- [http://sourceforge.net/p/deng/bugs/\\_discuss/thread/30e1def2/c85e/attachment/doomsday.diff](http://sourceforge.net/p/deng/bugs/_discuss/thread/30e1def2/c85e/attachment/doomsday.diff)

#### #31 - 2012-05-28 00:51 - skyjake

Sorry, but we can't remove the manual ~ expansion. If Doomsday is launched via an IDE (e.g. Qt Creator) or when the options are read from a response file, we have to expand the tildes manually. If the shell has already expanded the tilde, Doomsday doesn't really have to worry about it as it just gets the expanded full path from the get-go.

I agree that it is a little unexpected and/or strange, but -iwad is working as expected. It expects to receive a directory as the argument, not a file. (It would be more aptly called -iwaddir.) What Doomsday then does is that it deduces a directory from the given filename (stripping the last part), and looks for all IWAD files in the resultant directory and all its subdirectories. You will have to then use the -game option to specify which game to actually load (see man page), and Doomsday will automatically pick the right IWAD file for you.

At present there is no way to just say "pick this IWAD and use only it, nothing else".

Re: jdoom.pk3 not found

This sounds like a legitimate bug somewhere. Regardless of what is given to -iwad, it should know to look for the standard .pk3s in the /usr/share/doomsday/data/(game) directory. Only the IWAD files should be affected by the -iwad option.

#### #32 - 2012-05-28 04:20 - alepulver

Well, you do have to worry if you're opening a file that starts with ~ (you'll expand it and fail). Not really a big problem of course, but shows that it's not the right way to do things. I believe it's strange that an IDE will pass ~ unexpanded anyways...

First you said: "It expects to receive a directory as the argument, not a file"

And later: "What Doomsday then does is that it deduces a directory from the given filename (stripping the last part)"

I guess you meant: it expects a file name, not a directory, but considers the file's parent directory. I know about the game option, what I was saying is that the behavior should be consistent: either you accepts files or directories or both, instead of treating nonexistent files/dirs as its parent directory and existing directories as their parent too. You may ignore everything that's not a file, otherwise it can get confusing. The patch does that (you can ignore directories instead of adding "/something").

What I mean is (supposing I have doom/DOOM2.WAD):

```
$ doomsday -iwad doom/lalalalala  
(game found!)
```

It's really NOT expected.

Please let me know when you find a solution to the Data/data issue.

#### #33 - 2012-05-28 04:41 - skyjake

Re: ~ expansion

I'm not sure I follow, what is wrong with manually expanding it? Why does it fail? (example?)

Re: -iwad expecting a directory

It expects to receive the path of a directory *ending in a slash*. If it sees something that is not ending in a slash, it makes the *incorrect* assumption that it is okay to strip the last portion of the path to make a valid path. This should be fixed by checking what is the actual type of the possibly existing file/directory.

I completely agree that it should be fixed. At least it should give you a warning when it strips the (bogus or valid) filename part of the specified path to get to a directory.

#### #34 - 2012-05-28 06:13 - alepulver

Re: ~ expansion

Suppose I am under /some/directory and I have /DOOM.WAD there (that's the file DOOM.WAD under a directory named "~", so the path is /some/directory/~//DOOM.WAD).

\$ doomsday -iwad /DOOM.WAD

(will look for /home/user/DOOM.WAD instead of /some/directory//DOOM.WAD, because it can't tell the difference between both)

It's a contrived example, I agree, but something similar happened before when it tried to parse quotes or other characters in the file name. For example a file starting with '}' won't be recognized too. In general, I think it's better to have a parameter to process the argument in a special way (like -resource path/to/file).

It's fine for me if this stays the same, as I don't use those characters as file/dir names.

Re: -iwad expecting a directory

The "ending in a slash" (even if I think it's not nice) was working before you changed the code to QDir/etc, because the method absolutePath will remove the trailing "/" and the last path component will be stripped away always later.

Could you look into my patch? Except the removal of ~, that's exactly what I have done. See the makeAbsolutePath method. It works with existing files, nonexistent files/dirs and directories. It even avoids changing the current directory for path expansion, as it's not required.

#### #35 - 2012-05-28 06:33 - skyjake

Good point, I hadn't considered that absolutePath() will strip the extra slash.

Your proposed changes to CommandLine::makeAbsolutePath() look sensible; I'll apply them sometime this week. Thanks for the patch.

#### #36 - 2012-05-28 09:35 - alepulver

OK. You might also want to change the part:

```
else if (!QFile::exists(dir.path()))
dir.setPath("");
to something like:
else if (!QFile::exists(dir.path()))
[print error "file/dir not found %s"]
[exit here?]
```

I wasn't sure about returning "". It could be better to quit there, or print a message rather than just ignoring the arguments like that.

#### #37 - 2012-05-28 17:04 - danij

What is the status of this, have all the issues raised in this report been addressed?

#### #38 - 2012-09-19 12:06 - skyjake

As far as I can see, the only unclear issue is the following:

Re: jdoom.pk3 not found

This sounds like a legitimate bug somewhere. Regardless of what is given to -iwad, it should know to look for the standard .pk3s in the /usr/share/doomsday/data/(game) directory. Only the IWAD files should be affected by the -iwad option.

I can double-check that a regular installation in Ubuntu works as expected for me. This sounds like an isolated incident, though.

**Edit:** Tried build 627 in Ubuntu 12.04 64-bit. No problems launching via Snowberry.

#### #39 - 2012-09-19 14:21 - skyjake

- priority: 6 --> 3

#### #40 - 2012-09-19 14:54 - skyjake

The remaining issue here is quite likely related to [\[#1088\]](#).

#### #41 - 2014-06-02 08:19 - skyjake

- Tags set to *FileSystem, Resources*
- Category set to *Defect*
- Status changed from *New* to *In Progress*
- % Done changed from *0* to *80*

**#42 - 2014-06-02 08:20 - skyjake**

This should be fixed by [040906a](#); needs wider testing for verification, though. (Snowberry has not been modified.)

**#43 - 2014-11-23 20:54 - skyjake**

- Assignee deleted (*skyjake*)

**#44 - 2016-03-27 09:00 - skyjake**

- Status changed from *In Progress* to *Closed*