

utils / coreutils Public[Code](#) [Issues 603](#) [Pull requests 245](#) [Discussions](#) [Actions](#) [Projects](#)[New issue](#)

rm -rf ./ and variants silently delete current directory contents #9749

[Open](#)[#11005](#)

Labels

[U - rm](#)[good first issue](#)[reported-canonical](#)

sylvestre opened on Dec 20, 2025

[Contributor](#)

Component

rm -rf

Description

An issue with `rm -rf` : when deleting with `rm -rf .`, consistent with GNU `rm`, it refuses to delete the current directory, reports an error directly, and does not delete any contents.

```
> rm -rf .  
rm: refusing to remove '.' or '..' directory: skipping '.'
```



However, when using `rm -rf ./` or `rm -rf .///`, it reports an error as if the deletion failed (it should be consistent with `rm -rf .`, refusing directly without deleting anything), but in reality the contents of the current directory have already been deleted:

```
rm: cannot remove './': Invalid input
```



This is because the `clean_trailing_slashes` function collapses `.///` into `"./"`:

```
fn clean_trailing_slashes(path: &Path) -> &Path {  
    ...  
    // Checks if element at the end is a '/'
```



```

    if path_bytes[idx] == dir_separator {
        for i in (1..path_bytes.len()).rev() {
            if path_bytes[i - 1] != dir_separator {
                idx = i;
                break;
            }
        }
        #[cfg(unix)]
        return Path::new(OsStr::from_bytes(&path_bytes[0..=idx]));
        ...
    }
}
path
}

```

However, the `path_is_current_or_parent_directory` function only matches ".", "..", etc. It does not match equivalent paths like "./" or "../".

```

/// Checks if the path is referring to current or parent directory, if it is referring
fn path_is_current_or_parent_directory(path: &Path) -> bool {
    let path_str = os_str_as_bytes(path.as_os_str());
    let dir_separator = MAIN_SEPARATOR as u8;
    if let OK(path_bytes) = path_str {
        return path_bytes == ([b'.'])
            || path_bytes == ([b'..', b'..'])
            || path_bytes.ends_with(&[dir_separator, b'..'])
            || path_bytes.ends_with(&[dir_separator, b'..', b'..'])
            || path_bytes.ends_with(&[dir_separator, b'..', dir_separator])
            || path_bytes.ends_with(&[dir_separator, b'..', b'..', dir_separator]);
    }
    false
}

```

Test / Reproduction Steps

```
# Provide commands and setup to reproduce
```

Impact

This results in the contents of the current directory being recursively deleted.



sylvestre added **U - rm** **good first issue** on Dec 20, 2025



jacek-kurlit on Dec 20, 2025

Contributor ...

I will take this



jacek-kurlit on Dec 21, 2025

Contributor ...

[#9765](#)



FidelSch mentioned this [on Dec 25, 2025](#)

[🔗 Rm error priority #9827](#)



syvestre added **reported-canonical** [on Jan 17](#)



naoNao89 added a commit that references this issue [on Feb 18](#)

rm: fix path parsing for dot/dotdot protection ([uutils#9749](#)) ...

3dbc086



naoNao89 linked a pull request that will close this issue [on Feb 18](#)

[🔗 rm: fix path parsing for dot/dotdot protection #11005](#)



naoNao89 added 4 commits that reference this issue [on Feb 18](#)

rm: fix path parsing for dot/dotdot protection ([uutils#9749](#)) ...

27615a2

rm: fix path parsing for dot/dotdot protection ([uutils#9749](#)) ...

048da76

rm: fix path parsing for dot/dotdot protection ([uutils#9749](#)) ...

5c3b675

rm: fix dot/dotdot path protection ([uutils#9749](#)) ...

9f962fe



syvestre added a commit that references this issue [last week](#)

rm: fix path parsing for dot/dotdot protection ...

[🔗](#) cb45f75

[Sign up for free](#) to join this conversation on GitHub. Already have an account? [Sign in to comment](#)

Metadata

Assignees

No one assigned

Labels

U - rm

good first issue

reported-canonical

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

 **rm: fix path parsing for dot/dotdot protection**

utils/coreutils

Participants

