

## Arvados - Bug #15836

### Escape / convert forward slashes in collection names accessed via WebDAV

11/18/2019 09:00 PM - Tom Morris

<b>Status:</b>	Resolved	<b>Start date:</b>	01/08/2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Tom Clegg	<b>% Done:</b>	100%
<b>Category:</b>	Keep	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	2020-01-29 Sprint		
<b>Description</b>			
We have a customer report that the Mountain Goat WebDAV client doesn't display collections which have slashes (/) in their names. That's not terribly surprising considering it's use as a path delimiter, but it's not great to not be able to access the collection at all. We should consider a configurable option to replace the slash with another character such as underscore (_) or hyphen (-).			
<b>Subtasks:</b>			
Task # 15974: Review 15836-slash-in-collection-name			<b>Resolved</b>

#### Associated revisions

##### Revision 4237a24f - 01/16/2020 09:07 PM - Tom Clegg

Merge branch '15836-slash-in-collection-name'

fixes #15836

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tom@tomclegg.ca>

#### History

##### #1 - 11/18/2019 09:28 PM - Tom Clegg

The WebDAV server doesn't show collection/project names that have slashes (nor ones named "." or "..").

permittedName() -- [source: sdk/go/arvados/fs\\_base.go#L619](https://source.sdk/go/arvados/fs_base.go#L619)

I'm not sure this can be fixed with an escape char: either the escape char is an invalid filename char and clients can't send/receive it, or it is and clients get confused when we treat it as an escape char. Exporting it as something horrible like "<solidus>" would at least make it easier to figure out what the problem is, even if it doesn't make web/filesystem clients behave smoothly.

Perhaps the best course is to disallow these names on the API side so users don't fall into the trap at all.

##### #2 - 12/16/2019 03:59 PM - Ward Vandewege

Desired outcome is to replace unallowed characters with something else (e.g. \_) in webdav, just like we do for FUSE. That way they will be listed in the client. We should still allow creation of collection with those characters, because they are valid (and in use!).

##### #3 - 12/31/2019 08:50 PM - Peter Amstutz

- Target version changed from To Be Groomed to 2020-01-15 Sprint

##### #4 - 01/02/2020 04:25 PM - Tom Clegg

- Assigned To set to Tom Clegg

##### #5 - 01/02/2020 06:13 PM - Tom Clegg

I think we should allow but discourage this, since it seems impossible to implement it without surprising/confusing corner cases. Perhaps a config variable "string to replace slashes with in readonly webdav listings" -- where "" means don't list them at all.

If the replacement string is "\_":

- When looking up "/home/foo\_\_bar/baz", we look up "foo\_\_bar" by name, rather than populating the entire "/home" path. If looking up "foo\_\_bar" and that doesn't exist, we'll need to check for collections/projects named "foo\_/bar" or "foo//bar" as well, before giving up.
- If a new "foo\_bar" collection/project is created in a project where "foo/bar" already exists, URLs that used to reach the old "foo/bar" will now reach the new "foo\_bar".
- In future when "mkdir /home/foo\_bar" is possible to do via webdav, and "foo/bar" already exists, we'll need to decide whether that should create a new collection that masks the old one, or return "directory already exists" (probably the latter)

##### #6 - 01/02/2020 07:03 PM - Peter Amstutz

- Category set to Keep

#### #7 - 01/02/2020 08:31 PM - Lucas Di Pentima

I've tried using Mountain Duck client version 3.3.3 and the replacement string '%2F' on a collection name is displayed as is and without issues.

#### #8 - 01/07/2020 04:06 PM - Tom Clegg

15836-slash-in-collection-name @ [9d79488376e90532512733748eed1aa78af1c125](https://ci.arvados.org/view/Developer/job/developer-run-tests/1704/) --  
<https://ci.arvados.org/view/Developer/job/developer-run-tests/1704/>

#### #9 - 01/07/2020 04:07 PM - Tom Clegg

- Status changed from New to In Progress

#### #10 - 01/07/2020 08:16 PM - Tom Clegg

15836-slash-in-collection-name @ [3c848757efb9b1697962cb51068c87658565908b](https://ci.arvados.org/view/Developer/job/developer-run-tests/1708/) --  
<https://ci.arvados.org/view/Developer/job/developer-run-tests/1708/>

#### #11 - 01/15/2020 03:50 PM - Peter Amstutz

- Target version changed from 2020-01-15 Sprint to 2020-01-29 Sprint

#### #12 - 01/16/2020 04:51 PM - Peter Amstutz

From the config.yml comment

1. Possible values include "%2f" and "{slash}"

Should be "example" values, otherwise could be read as those are the only two possible values.

If the default empty value is used, names containing "/" cannot be used when creating or renaming a collection or project.

Passive voice. Should be 'creating or renaming a collection with a name containing "/" will be rejected by the API server'

```
for tryURL, expectRegexp := range map[string]string{
base:      `(?ms).*href="." + nameShownEscaped + `"/S+` + nameShown + `.*`,
base + nameShown + "/": `(?ms).*href="./filename"S+filename.*`,
} {
```

Should the path "base + nameShownEscaped" be tested as well? Or is the URL encoding happening in url.Parse() or http.Request() ?

Could you merge/rebase master again so I can do some manual testing?

#### #13 - 01/16/2020 05:13 PM - Tom Clegg

Peter Amstutz wrote:

From the config.yml comment

1. Possible values include "%2f" and "{slash}"

Should be "example" values, otherwise could be read as those are the only two possible values.

Reworded.

```
# WebDAV. Example values are "%2f" and "{slash}". Names that
# contain the substitution string itself may result in confusing
# behavior, so a value like "_" is not recommended.
```

If the default empty value is used, names containing "/" cannot be used when creating or renaming a collection or project.

Passive voice. Should be 'creating or renaming a collection with a name containing "/" will be rejected by the API server'

Reworded.

```
# If the default empty value is used, the server will reject
# requests to create or rename a collection when the new name
# contains "/".
```

```
for tryURL, expectRegexp := range map[string]string{
base:      `(?ms).*href="\.\/ + nameShownEscaped + `/\S+` + nameShown + `.*`,
base + nameShown + "/" : `(?ms).*href="\.\/filename"\S+filename.*`,
} {
```

Should the path "base + nameShownEscaped" be tested as well? Or is the URL encoding happening in url.Parse() or http.Request() ?

Right, nameShownEscaped is more correct here -- an unescaped "{" is not allowed in an URL path segment, even though it happens to be accepted by Go.

Could you merge/rebase master again so I can do some manual testing?

Sure, merged master.

15836-slash-in-collection-name @ [4164f38ef0f900962e3c365745c59d5c20a63c66](#)

**#14 - 01/16/2020 07:25 PM - Tom Clegg**

Added an entry to the upgrade notes.

15836-slash-in-collection-name @ [d766bbe3770d56f8ef391999ec51c42cb751b319](#)

**#15 - 01/16/2020 09:06 PM - Peter Amstutz**

Tom Clegg wrote:

Added an entry to the upgrade notes.

15836-slash-in-collection-name @ [d766bbe3770d56f8ef391999ec51c42cb751b319](#)

This LGTM

**#16 - 01/16/2020 09:35 PM - Anonymous**

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

Applied in changeset [arvados|4237a24fcc5e0ff8cf6429b04844ca5f8e5b48c5](#).

**#17 - 01/22/2020 02:10 PM - Peter Amstutz**

- Release set to 22