Arvados - Bug #14539

[SDKs] [arv-mount] Use "." placeholder to persist empty directories

11/28/2018 04:16 PM - Tom Clegg

Status:	Resolved	**Start date:**	12/18/2018
Priority:	Normal	**Due date:**	
Assigned To:	Lucas Di Pentima	**% Done:**	100%
Category:	SDKs	**Estimated time:**	0.00 hour
Target version:	2019-02-13 Sprint		

Description

When saving a collection, encode an empty directory like this:

```
./dir/emptysubdir d41d8cd98f00b204e9800998ecf8427e+0 0:0:\056
```

Subtasks:

Task # 14541: Review 14539-pysdk-empty-dirl	**Resolved**

Related issues:

Related to Arvados - Bug #14806: [crunch1] unescape filenames when parsing ma...	**Resolved**	**02/05/2019**

Associated revisions

Revision 4620478d - 01/17/2019 04:43 PM - Lucas Di Pentima

Merge branch '14539-pysdk-empty-dir'
Closes #14539

Arvados-DCO-1.1-Signed-off-by: Lucas Di Pentima <ldipentima@veritasgenetics.com>

History

#1 - 11/28/2018 04:26 PM - Tom Clegg

- Target version changed from Arvados Future Sprints to 2018-12-12 Sprint

#2 - 11/28/2018 04:26 PM - Tom Morris

- Target version changed from 2018-12-12 Sprint to 2018-11-28 Sprint

#3 - 11/28/2018 04:26 PM - Tom Morris

- Target version changed from 2018-11-28 Sprint to 2018-12-12 Sprint

#5 - 11/28/2018 04:40 PM - Lucas Di Pentima

- Assigned To set to Lucas Di Pentima

#6 - 12/12/2018 04:31 PM - Lucas Di Pentima

- Target version changed from 2018-12-12 Sprint to 2018-12-21 Sprint

#7 - 12/17/2018 03:43 PM - Lucas Di Pentima

- Status changed from New to In Progress

#8 - 12/17/2018 11:42 PM - Lucas Di Pentima

Updates at 56959ea8492ec4f08aa90c89a8f41e5c278e3c41 - branch 14539-pysdk-empty-dir
Test run: https://ci.curoverse.com/job/developer-run-tests/1013/

- Persist empty subdirectories by adding an empty file named \056 (".") to the manifest.
- Don't allow explicitly use that name on file or directory names.
- Test fixes & additions.

#9 - 12/18/2018 12:55 AM - Lucas Di Pentima

Update 41e5e2b86 fixes fuse tests: https://ci.curoverse.com/job/developer-run-tests/1015/

#10 - 12/18/2018 04:57 PM - Tom Clegg

The "save empty dir" part LGTM but I would like the test to include an empty directory whose name needs escaping, like "/foo bar/baz waz".

This part doesn't seem right. (Matching a specific encoding like \056 is a clue encoding/decoding is not being done properly.)

```
diff --git a/sdk/python/arvados/collection.py b/sdk/python/arvados/collection.py
index 627f0346d..c2517c618 100644
--- a/sdk/python/arvados/collection.py
+++ b/sdk/python/arvados/collection.py
@@ -600,6 +600,9 @@ class RichCollectionBase(CollectionBase):

         pathcomponents = path.split("/", 1)
         if pathcomponents[0]:
+            # Don't allow naming files/dirs \\056
+            if pathcomponents[0] == "\\056":
+                raise IOError(errno.EINVAL, "Invalid name", pathcomponents[0])
             item = self._items.get(pathcomponents[0])
             if len(pathcomponents) == 1:
                 if item is None:
```

Creating files and directories named '\056' should work fine. The manifest would look like this:

```
./\134056 d41d8cd98f00b204e9800998ecf8427e+0 0:0:\134056
```

The specific case of '\056' might be rare enough not to care about, but this seems to be revealing that something isn't escaping/unescaping backslashes properly. Indeed, a file named \040 seems to be written as \040 and then read back as " ", and this causes arv-mount to see its own updates as conflicts:

```
tomclegg@shell.4xphq:~/keepw/home/test$ touch '\040'
tomclegg@shell.4xphq:~/keepw/home/test$ ls -l
total 1
-rwxrwxrwx 1 tomclegg tomclegg 0 Dec 18 16:35 \040
-rwxrwxrwx 1 tomclegg tomclegg 0 Dec 18 16:35  ~20181218-163603~conflict~
tomclegg@shell.4xphq:~/keepw/home/test$ touch '\040'
tomclegg@shell.4xphq:~/keepw/home/test$ touch '\040'
tomclegg@shell.4xphq:~/keepw/home/test$ ls -l
total 2
-rwxrwxrwx 1 tomclegg tomclegg 0 Dec 18 16:35 \040
-rwxrwxrwx 1 tomclegg tomclegg 0 Dec 18 16:35  ~20181218-163603~conflict~
-rwxrwxrwx 1 tomclegg tomclegg 0 Dec 18 16:35  ~20181218-163644~conflict~
-rwxrwxrwx 1 tomclegg tomclegg 0 Dec 18 16:35  ~20181218-163656~conflict~
```

We should fix that instead of disallowing files/dirs named \056.

#11 - 12/21/2018 10:10 PM - Lucas Di Pentima

Ok, so after lots of testing and hair pulling, I think I got it to work. Squashed the many test commits by using rebase -i:

Update at 13e7ad8135a0bafc3d1d225ff7e4c62de4f3b43f
Test run: https://ci.curoverse.com/job/developer-run-tests/1020/

- Moved the escaping to _normalize_stream(), doing it on find_or_create() was not right.
- Removed code prohibiting to use '\056' as a file/dir name.
- Updated tests.

#12 - 12/21/2018 10:59 PM - Lucas Di Pentima

Failed -remainer tests re-run: https://ci.curoverse.com/job/developer-run-tests-remainder/1052/

#13 - 01/02/2019 04:18 PM - Lucas Di Pentima

- Target version changed from 2018-12-21 Sprint to 2019-01-16 Sprint

#14 - 01/14/2019 03:46 PM - Tom Clegg

LGTM, thanks.

A couple of follow-ups, though:

This comment in FuseRmTest is now obsolete:

```
         # Can't have empty directories :-( so manifest will be empty.
```

Along with space and backslash, are there other chars that are legal in filenames but would be misinterpreted in a manifest? I suspect newline needs to be escaped, and we should probably do tab too (even if tab works unescaped, it could be distracting while troubleshooting).

#15 - 01/15/2019 05:30 PM - Lucas Di Pentima

Update at 735143cbf
Test run: https://ci.curoverse.com/job/developer-run-tests/1032/

Added escaping for \n and \t on file and stream names. Just in case I tested the use of : on file names and it seemed to me that it doesn't need escaping, no issues observed at the PySDK level.

#16 - 01/15/2019 07:01 PM - Tom Clegg

escape() should *always* escape backslash as \134, even when it isn't part of a valid escape sequence. This implementation encodes filename r'\400' as r'0:0:\400', which contradicts Keep manifest format.

Although the Python SDK can read unescaped colons in filenames, escaping them seems like a good idea. It is explicitly allowed by Keep manifest format.

By the same logic, rather than blacklisting chars, we should probably escape ":", space, and all non-printable chars (something like [:\000-]). That would take care of \r, nul, etc.

#17 - 01/16/2019 04:12 PM - Lucas Di Pentima

- Target version changed from 2019-01-16 Sprint to 2019-01-30 Sprint

#18 - 01/16/2019 09:38 PM - Lucas Di Pentima

Updates at 8b2eb7d1f
Test run: https://ci.curoverse.com/job/developer-run-tests/1033/

Fixes literal backslash escaping, enhances other special chars escaping & updates test.

#19 - 01/16/2019 09:48 PM - Tom Clegg

LGTM, thanks.

I might be getting hung up on this tangential issue, but I'll ask anyway: was there any particular reason to escape only [\t\n\r:] but not other control chars like [:\000-\040]?

#20 - 01/17/2019 12:11 AM - Lucas Di Pentima

Updates at a974fc22e
Test run: https://ci.curoverse.com/job/developer-run-tests/1034/

Sorry Tom, I've now updated escape() to cover all the special chars between \000 and \040, and also ':'. Updated test too.

#21 - 01/17/2019 09:51 AM - Lucas Di Pentima

Removed useless comments and simplified the regex on c5e7cbef5
Test run: https://ci.curoverse.com/job/developer-run-tests/1035/

#22 - 01/17/2019 04:40 PM - Tom Clegg

LGTM, thanks!

#23 - 01/17/2019 05:02 PM - Lucas Di Pentima

- Status changed from In Progress to Resolved

Applied in changeset arvados|4620478d694697eff07e501187d784c6c98ccfa9.

#24 - 02/04/2019 08:36 PM - Peter Amstutz

- Status changed from Resolved to Feedback

- Target version changed from 2019-01-30 Sprint to 2019-02-13 Sprint

#25 - 02/04/2019 08:39 PM - Peter Amstutz

crunch-job (crunchv1) was broken by these changes because manifests that previously looked like

```
0:525929984:sha256:6289c13e51744248b713b5d124c6148a5084544093b23996103b430fb2af7c7a.tar
```

Now have ":" quoted as \072

```
0:525929984:sha256\0726289c13e51744248b713b5d124c6148a5084544093b23996103b430fb2af7c7a.tar
```

As a result, it wouldn't match crunch-jobs's regular expression for picking out the image id.

Here's a quick and dirty crunch-job fix:

14539-fix-crunch-job @ ade82492825d6b78e2da35822e80f79a03e6ea67

#26 - 02/04/2019 08:46 PM - Tom Clegg

I get trying to do the absolute minimum, but maybe generic unescaping is easy enough that it doesn't need a special-case fix?

```
$filename =~ s/\\([0-3][0-7][0-7])/chr(oct($1))/ge;
```

#27 - 02/04/2019 08:57 PM - Tom Clegg

- Related to Bug #14806: [crunch1] unescape filenames when parsing manifests in crunch-job added

#28 - 02/04/2019 08:58 PM - Tom Clegg

- Status changed from Feedback to Resolved

#29 - 02/05/2019 09:21 PM - Tom Morris

- Release set to 21

#30 - 02/05/2019 09:40 PM - Tom Morris

- Release deleted (21)

#31 - 03/01/2019 06:30 PM - Tom Morris

- Release set to 15