This got reported to me by a Debian user running Tails Installer 4.4.18 with French locales:

```
2017-08-12 14:08:26,435 [creator.py:747 (mount_device)] DEBUG: Mounting /org/freedesktop/UDisks2/block_devices/sdb1
2017-08-12 14:08:26,591 [creator.py:774 (mount_device)] DEBUG: Mounted /dev/sdb1 to /media/user/Tails
2017-08-12 14:08:28,594 [creator.py:440 (delete_liveos)] INFO: Suppression du système d'exploitation existant
2017-08-12 14:08:28,598 [creator.py:444 (delete_liveos)] DEBUG: Considering /media/user/Tails/EFI
```

It looks like we assume ASCII encoding for the exception string representation, either in this string %-formatting operation or in gui.py where we catch TailsInstallerError and display its message.

For the record, in Tails Install 4.4.18 the corresponding code is:

```python
self.log.debug("Considering " + path)
if os.path.isfile(path):
    […]
elif os.path.isdir(path):
    try:
        _set_liberal_perms_recursive(path)
    except OSError, e:
        self.log.debug(_("Unable to chmod %(file)s: %(message)s") % {'file': path, 'message': str(e)})
    try:
        shutil.rmtree(path)
    except OSError, e:
        raise TailsInstallerError(_("Unable to remove directory from" " previous LiveOS: %(message)s") % {'message': str(e)})
```

... and line 467 is the last one I'm quoting above.
I see we have lots of other instances of very similar pieces of code. In most cases, this bug will not break Tails Installer functionality, because the affected code is about reporting a critical error that already happened. But it will prevent users from sending us useful bug reports (by preventing the actual error from being part of the bug report) and will create poor UX (by crashing instead of displaying an explanatory error message).

**Associated revisions**

- **Revision 0e172164 - 10/20/2018 10:32 AM - intrigeri**
  Enable the bugfix-15166-installer-crash APT overlay (refs: #15166).

- **Revision 0ce2f62a - 10/20/2018 05:40 PM - intrigeri**
  Merge branch 'bugfix/15166-installer-crash' into stable (refs: #15166)

**History**

- **#1 - 01/14/2018 09:00 AM - intrigeri**
  kurono, would you be interested in having a look?

- **#2 - 08/16/2018 12:47 PM - Anonymous**
  Ping?

- **#3 - 09/28/2018 01:57 PM - kurono**
  - Assignee set to kurono

- **#4 - 10/19/2018 08:57 PM - kurono**
  - Status changed from Confirmed to In Progress
  - Assignee deleted (kurono)
  - Target version set to Tails_3.11
  - QA Check set to Ready for QA
  - Feature Branch set to kurono/bug/15166-installer-crashes-no-ascii

I was able to reproduce this with a virtual machine and selecting the french language. Then I manually raised an exception with

```python
raise OSError(u'Suppression du système d\'exploitation live existant')
```

I introduced utf-8 encoding in the exceptions, and it seems to fix the problem.
It passed all the usb tests.

- **#5 - 10/20/2018 09:10 AM - intrigeri**
  - Assignee set to intrigeri
  - Target version changed from Tails_3.11 to Tails_3.10.1

This bug seems pretty bad so I'll try to review'n'merge this in time for 3.10. Not sure that'll work out but let's try!

- **#6 - 10/20/2018 10:00 AM - intrigeri**
  I don't intend to block on the following concerns for this review'n'merge: the proposed code makes sense to me and I trust you that it does fix at least some instances of this bug.
Now, even if I merge this in a hurry for 3.10, I'll probably leave this ticket "In Progress" because I'm not sure the branch fixes all instances of the bug and I'm not convinced by the way the initially chosen approach to fix it.

I'm a bit confused wrt. what our logger should receive, i.e. a Python string or encoded bytes. E.g. I see self.log.debug(cmd_decoded) which suggests a Python string is preferred. But your branch changes many (but not all) calls to log.[warning,error,debug,etc.] so they're now passed encoded bytes, which seems consistent with the fact that our _ claims to return a "UTF-8 encoded bytestring". If our logger needs an encoded bytestring, fine, but then shouldn't self.live.log.exception(unicode(e)) be encoded to utf-8 too? Same for self.log.debug(cmd_decoded) and — I suspect — a bunch of other similar calls.

More generally, I'm not sure that manually adding "turn into a Unicode string and then encode to utf-8" everywhere is the best approach. I bet we'll always forget it occasionally. So I think we should automate this, probably by giving it a more clever stream handler or formatter in _setup_logger, that will use something like the _to_unicode and unicode_to_utf8 functions we already have, in order to support both all kinds of inputs.

What do you think?

#7 - 10/20/2018 10:42 AM - intrigeri
- % Done changed from 0 to 20
- Feature Branch changed from kurono/bug/15166-installer-crashes-no-ascii to bugfix/15166-installer-crash, kurono/liveusb-creator:bug/15166-installer-crashes-no-ascii

Released 5.0.12 with these changes, uploaded, enabled the corresponding APT overlay on a branch, asked Jenkins to test this.

#8 - 10/20/2018 05:41 PM - intrigeri
- Assignee changed from intrigeri to kurono
- Target version changed from Tails_3.10.1 to Tails_3.11
- QA Check deleted (Ready for QA)
- Feature Branch deleted (bugfix/15166-installer-crash, kurono/liveusb-creator:bug/15166-installer-crashes-no-ascii)

Merged, I'm giving this ticket back to you for the next steps, as agreed on XMPP.

#9 - 11/11/2018 05:13 PM - kurono
- Assignee changed from kurono to intrigeri
- % Done changed from 20 to 30
- QA Check set to Ready for QA
- Feature Branch set to kurono/bug/15166-installer-crashes-no-ascii

intrigeri wrote:

I don't intend to block on the following concerns for this review'n'merge: the proposed code makes sense to me and I trust you that it does fix at least some instances of this bug.

Now, even if I merge this in a hurry for 3.10, I'll probably leave this ticket "In Progress" because I'm not sure the branch fixes all instances of the bug and I'm not convinced by the way the initially chosen approach to fix it.

I'm a bit confused wrt. what our logger should receive, i.e. a Python string or encoded bytes. E.g. I see self.log.debug(cmd_decoded) which suggests a Python string is preferred. But your branch changes many (but not all) calls to log.[warning,error,debug,etc.] so they're now passed encoded bytes, which seems consistent with the fact that our _ claims to return a "UTF-8 encoded bytestring". If our logger needs an encoded bytestring, fine, but then shouldn't self.live.log.exception(unicode(e)) be encoded to utf-8 too? Same for self.log.debug(cmd_decoded) and — I suspect — a bunch of other similar calls.
More generally, I'm not sure that manually adding "turn into a Unicode string and then encode to utf-8" everywhere is the best approach. I bet we'll always forget it occasionally. So I think we should automate this, probably by giving it a more clever stream handler or formatter in _setup_logger, that will use something like the _to_unicode and unicode_to_utf8 functions we already have, in order to support both all kinds of inputs.

What do you think?

Yes, as said before I agree it was not the best solution. It seems like the problem is not really with the logger, but with the encoding of the exception message, which can't be unicode at least one create a custom Exception that manages that. I have done it, and and have reverted the former changes, no the exception are managed as str(e) again. It passed all the tests cases.

#10 - 11/12/2018 03:49 PM - intrigeri

It seems like the problem is not really with the logger, but with the encoding of the exception message, which can't be unicode at least one create a custom Exception that manages that.

I agree we have a problem with inconsistent decoded string vs. bytestring in exception handling and I'm glad you're on it! Your solution looks good to me. I'll test it in non-English locales in various situations where we raise an exception.

Wrt. the logger, I'm still a bit confused: could you please test & confirm that it works as expected regardless of whether we pass it a decoded string or a bytestring, in cases where it matters?

#11 - 11/16/2018 01:29 PM - intrigeri

- Assignee changed from intrigeri to kurono
- QA Check changed from Ready for QA to Info Needed

#12 - 12/08/2018 12:47 PM - kurono

intrigeri wrote:

It seems like the problem is not really with the logger, but with the encoding of the exception message, which can't be unicode at least one create a custom Exception that manages that.

I agree we have a problem with inconsistent decoded string vs. bytestring in exception handling and I'm glad you're on it! Your solution looks good to me. I'll test it in non-English locales in various situations where we raise an exception.

Wrt. the logger, I'm still a bit confused: could you please test & confirm that it works as expected regardless of whether we pass it a decoded
string or a bytestring, in cases where it matters?

Ok, I think I understand now, our code works as intended with decoded strings and encoding applied to decoded unicode bytestring for instance the following works:

```python
self.log.info(_(u"Suppression du système d'exploitation live existant"))
self.log.info(_(u"Suppression du système d'exploitation live existant").encode("utf-8"))
```

but the following does not:

```python
s = "Suppression du système d'exploitation live existant"
b = bytes(s, 'utf-8')
self.log.info(b)
```

and the normal string with unicode characters does not work:

```python
self.log.info("Suppression du système d'exploitation live existant")
```

I will take a look of all the instances when we call the logs to see if we are not doing it right.

---

#13 - 12/16/2018 01:56 PM - CyrilBrulebois
- Target version changed from Tails_3.11 to Tails_3.12

#14 - 01/21/2019 05:40 PM - kurono
and the normal string with unicode characters does not work:

[...]

Funny, I can't reproduce this again. So, I would say, as far as we don't use something like:

```python
s = "Suppression du syste\'me d'\'exploitation live existant"
b = bytes(s, 'utf-8')
self.log.info(b)
```

Our code is safe to manage log messages with Unicode characters.

---

**#15 - 01/25/2019 08:18 AM - intrigeri**

- Target version changed from Tails_3.12 to Tails_3.13

Thanks!

We're in the code freeze for 3.12. Technically bugfixes can be considered during that time but I'll focus on severe regressions first, so I doubt I'll have time to review this in time for 3.12.

---

**#16 - 02/22/2019 08:47 AM - intrigeri**

- Assignee changed from intrigeri to kurono
- % Done changed from 30 to 50
- QA Check changed from Ready for QA to Info Needed

Code review passes. Can you please send this to Jenkins so I can easily see test results? You'll need to export your Tails Installer changes to config/chroot_local-patches/ in tails.git.

---

**#17 - 02/23/2019 12:01 PM - kurono**

- Assignee changed from kurono to intrigeri
- QA Check changed from Info Needed to Ready for QA

intrigeri wrote:
Code review passes. Can you please send this to Jenkins so I can easily see test results? You'll need to export your Tails Installer changes to config/chroot_local-patches/ in tails.git.

Done. Please check Jenkins.

#18 - 02/24/2019 06:22 AM - intrigeri

Done. Please check Jenkins.

Done, test suite passes \o/

I'll merge, release and upload a new Tails Installer.

(FTR I'd rather the branch submitter to check Jenkins themselves before submitting, as per https://tails.boum.org/contribute/merge_policy/#submit.)

#19 - 02/24/2019 06:32 AM - intrigeri

I'm a tiny bit worried that we have no reproducer here, so it's hard to tell for sure whether the proposed changes will fix real-world issues without introducing regressions. But kurono's analysis seems sound so I'll cross fingers and go ahead :)

#20 - 02/24/2019 06:36 AM - intrigeri
- Status changed from In Progress to 11
- Assignee deleted (intrigeri)
- % Done changed from 50 to 100
- QA Check changed from Ready for QA to Pass

Released 5.0.15 and uploaded. Congrats!

#21 - 03/20/2019 02:28 PM - CyrilBrulebois
- Status changed from 11 to Resolved