Add a Binary Payload to your Shell Scripts

Feb 19, 2009 By Mitch Frazier (/users/mitch-frazier)

Generally when we think of shell scripts we think of editable text, but it's possible to add binary data to your shell script as well. In this case we're going to talk about adding a binary *payload* to the end of your shell script.

Adding a binary payload to a shell script could, for instance, be used to create a single file shell script that installs your entire software package which could be composed of hundreds of files. You merely append the tar or gzip file of your package as a binary payload to the script file, when the script runs it extracts the payload and does its task with the extracted files.

For this example I assume the appended file is a `tar.gz` file. The payload is appended to the end of an *installation* script preceded by a marker line (PAYLOAD:). The appended data is either uuencoded or just binary data. The script that follows takes a single argument which should be the `tar.gz` to append to the *installation* script. The `installation` script template `install.sh.in` is copied to `install.sh` with the payload appended. This script is named `addpayload.sh` follows:

```
#!/bin/bash

# Check for payload format option (default is uuencode).
uuencode=1
if [[ "$1" == '--binary' ]]; then
        binary=1
        uuencode=0
        shift
fi
if [[ "$1" == '--uuencode' ]]; then
        binary=0
        uuencode=1
        shift
fi

if [[ ! "$1" ]]; then
        echo "Usage: $0 [--binary | --uuencode] PAYLOAD_FILE"
        exit 1
fi


if [[ $binary -ne 0 ]]; then
        # Append binary data.
        sed \
                -e 's/uuencode=./uuencode=0/' \
                -e 's/binary=./binary=1/' \
                        install.sh.in >install.sh
        echo "PAYLOAD:" >> install.sh

        cat $1 >>install.sh
fi
if [[ $uuencode -ne 0 ]]; then
        # Append uuencoded data.
        sed \
                -e 's/uuencode=./uuencode=1/' \
                -e 's/binary=./binary=0/' \
                        install.sh.in >install.sh
        echo "PAYLOAD:" >> install.sh

        cat $1 | uuencode - >>install.sh
fi
```

In addition to appending the payload it also modifies the installer script to tell it whether the payload is binary or uuencoded.

The template script `install.sh.in` is out *installation* script which at this point just untars the payload and nothing else. Actually, it doesn't even untar the payload it just tests it with `tar`'s `-t` option:

```
#!/bin/bash

uuencode=1
binary=0

function untar_payload()
{
        match=$(grep --text --line-number '^PAYLOAD:$' $0 | cut -d ':' -f 1
        payload_start=$((match + 1))
        if [[ $binary -ne 0 ]]; then
                tail -n +$payload_start $0 | tar -tzvf -
        fi
        if [[ $uuencode -ne 0 ]]; then
                tail -n +$payload_start $0 | uudecode | tar -tzvf -
        fi
}

read -p "Install files? " ans
if [[ "${ans:0:1}"  ||  "${ans:0:1}" ]]; then
        untar_payload
        # Do remainder of install steps.
fi

exit 0
```

In the function `untar_payload` the script uses `grep` to search throught itself (`$0`) for the marker and then it extracts the line number from the `grep` output and adds one to it. This line number is then passed to `tail` preceded by a plus sign which causes `tail` to output everything starting at that line number. The data is then fed directly into `tar` for extraction if the payload is binary. If it's uuencoded then it's first fed into `uudecode` before being fed into `tar`.

To create our *installer* let's use a simple payload file that contains three files name a, b, and c. We'll add the payload as an uuencoded block:

```
$ sh addpayload.sh --uuencode abc.tar.gz
$ cat install.sh
#!/bin/bash

... # Installer script lines (see above)
read -p "Install files? " ans
... # More installer script lines (see above)
exit 0

PAYLOAD:
begin 644 -
M'XL(`))%G$D``^W12PJ$0`Q%T2REEI!HK%J/BM`]Z(F?_?O#J8+0&=TS"8'`
M"[Q6_D\WV7V?5AH]=COWBYB9%_4J:Q$UK6J7I`&_R3+-[9B2_+YS_[F]&\8I
JXJ%874#&J_X;^H_0!V2\ZC_3/P````````````````/!D!0OB?_,`*```
`
end
```

At the end of the file you see the `PAYLOAD:` marker and the uuencoded block. If we now run the script we get:

```
$ sh install.sh
Install files? y
-rw-r--r-- mitch/users       0 2009-02-18 11:29 a
-rw-r--r-- mitch/users       0 2009-02-18 11:29 b
-rw-r--r-- mitch/users       0 2009-02-18 11:29 c
```

I won't show you the `--binary` usage but it produces the same result, albeit with a slightly smaller foot print since the payload does not have to be uuencoded.

Mitch Frazier is an Associate Editor for *Linux Journal*.

Comments

Comment viewing options

Threaded list - expanded ▼ | Date - newest first ▼ | 50 comments per page ▼ | Save settings

Select your preferred way to display the comments and click "Save settings" to activate your changes.

Good Article (/content/add-binary-payload-your-shell-scripts#comment-333878)
Submitted by Barun Saha (not verified) on Fri, 02/27/2009 - 00:28.

This is a very good article, Mitch.

bzip as well (/content/add-binary-payload-your-shell-scripts#comment-333839)
Submitted by Tim Parks (not verified) on Thu, 02/26/2009 - 12:18.

In addition to bnary, it seems like the technique would work for compressed data as well (e.g., bzip2).

bzip2 --stdout $1 >>install.sh

Of course the install.sh would have to be changed to account for the compressed information.

Nice article.

cat $1 or cat $2 ? (/content/add-binary-payload-your-shell-scripts#comment-333658)
Submitted by Libu (not verified) on Mon, 02/23/2009 - 04:26.

Shouldn't the two cat's have been "cat $2 " ??

i.e.
cat $2 >>install.sh
.
.
cat $2 | uuencode - >>install.sh

$1 is correct - you must (/content/add-binary-payload-your-shell-scripts#comment-333661)
Submitted by Jon Brett (not verified) on Mon, 02/23/2009 - 06:54.

$1 is correct - you must have skipped-over the "shift" operation when you read the script.

yup, the shift just didn't (/content/add-binary-payload-your-shell-scripts#comment-333706)
Submitted by Anonymous (not verified) on Tue, 02/24/2009 - 05:25.

yup, the shift just didn't register.

thanks.

minor change (/content/add-binary-payload-your-shell-scripts#comment-333592)
Submitted by John McKown (not verified) on Fri, 02/20/2009 - 13:26.

You have two "cat" lines in your script:

cat $1 >>install.sh

and

cat $1 | uuencode - >>install.sh

In the first case, I think it would be safer to do:

cat "$1" >>install.sh

In the second case, I would eliminate the cat entirely:

uuencode "$1" - >>install.sh

The main change is quoting the $1 just in case the file name contains something that the shell would interpret.

--
John McKown

Yes (/content/add-binary-payload-your-shell-scripts#comment-333596)
Submitted by Mitch Frazier (/users/mitch-frazier) on Fri, 02/20/2009 - 16:51.


(/users/mitch-frazier)

Good changes. I don't use uuencode very often and I didn't find the man page very enlightening. That was the first syntax I stumbled upon that did what I wanted.

As far as quoting "$1" goes, you're not gonna put spaces in the file name or something strange like that are you?

Mitch Frazier is an Associate Editor for *Linux Journal*.

Me, personally? No, way. But (/content/add-binary-payload-your-shell-scripts#comment-333664)
Submitted by John McKown (not verified) on Mon, 02/23/2009 - 11:37.

Me, personally? No, way. But I've actually got a vendor product which was ported from Windows which has file and subdirectory names with embedded blanks and dollar signs.

Typo (/content/add-binary-payload-your-shell-scripts#comment-333557)
Submitted by Anonymous (not verified) on Thu, 02/19/2009 - 13:25.

Typo in your install.sh.in on line 10. Missing $ before "binary"

Thanks (/content/add-binary-payload-your-shell-scripts#comment-333565)
Submitted by Mitch Frazier (/users/mitch-frazier) on Thu, 02/19/2009 - 18:21.


(/users/mitch-frazier)

Fixed. Strange, it didn't fail when I tested it.

Mitch Frazier is an Associate Editor for *Linux Journal*.

Cool. Similar to makeself (/content/add-binary-payload-your-shell-scripts#comment-333555)
Submitted by Anonymous on Thu, 02/19/2009 - 12:53.

Cool. Similar to makeself and shar.