1. How to Submit a Perl Documentation Patch

Walt Mankowski

Myxa Corp.

`waltman@netaxs.com`

June 14, 2001

---

2. Finding Perl Documentation That Needs Updating

- perlbug (http://bugs.perl.org)
- USENET (comp.lang.perl.misc)
- perlmonks (http://www.perlmonks.com)
- Perl beginner's mailing list (beginners-subscribe@perl.org)
- #perl

- Coworkers
- etc.

---

3. Is the Patch Necessary?

Probably accepted

- Fix typos
- Correct grammar, punctuation, and code
- Add or improve examples

Discuss on p5p

- Documenting previously undefined features
    - rand(0)

---

4. Example

Someone on comp.lang.perl.misc was trying to find a random number between 1 and 10. He posted the following code:

```
        $rand_num = int(rand * 10);
```

This looks like it might be right but actually has two subtle bugs in it.

---

5. Bugs in User’s Code

Here’s the user’s code again:

```
        $rand_num = int(rand * 10);
```

Bug #1 -- Precedence Problem

Perl will treat this code as

```
        $rand_num = int(rand (* 10));
```

Which becomes

```
        $rand_num = int(rand 0);
```

which is the same as

```
        $rand_num = int(rand 1);
```

which always returns 0.

Bug #2 -- Wrong Scale

He could have used either

```
        $rand_num = int(10 * rand);
```

or more simply

```
        $rand_num = int(rand 10);
```

but both of those have the problem that they return random numbers between 0 and 9 instead of between 1 and 10.

---

6. Some thoughts on this bug

- Finding random numbers is a fairly common occurrence, especially for beginners.
- It’s surprisingly easy to get very wrong.
- It’s easy to make subtle bugs that beginners are unlikely to catch.
- Lots of people have poor math skills.

---

7. Current Documentation for rand()

Here’s what the perl 5.6.1 documentation says about rand():

```
        rand EXPR
        rand    Returns a random fractional number greater than or
                equal to "0" and less than the value of EXPR.
                (EXPR should be positive.)  If EXPR is omitted,
                the value "1" is used.  Automatically calls
                "srand" unless "srand" has already been called.
                See also "srand".

                (Note: If your rand function consistently returns
                numbers that are too large or too small, then your
                version of Perl was probably compiled with the
                wrong number of RANDBITS.)
```

Note that it doesn't say anything at all about how to generate random integers, just random floating point numbers.

8. Updated Documentation for rand()

I'd like to add an example showing how to find random integers. I'd also like to make it clearer that the range starts at 0 and doesn't include EXPR. This is what I'd like the updated documentation to say:

```
rand EXPR
rand    Returns a random fractional number greater than or
        equal to "0" and less than the value of EXPR.
        (EXPR should be positive.)  If EXPR is omitted,
        the value "1" is used.  Automatically calls
        "srand" unless "srand" has already been called.
        See also "srand".

        Apply "int()" to the value returned by "rand()" if
        you want random integers instead of random frac-
        tional numbers.  For example,

            int(rand(10))

        returns a random integer between "0" and "9",
        inclusive.

        (Note: If your rand function consistently returns
        numbers that are too large or too small, then your
        version of Perl was probably compiled with the
        wrong number of RANDBITS.)
```

9. Updating the Documentation

- Make updates to the development version of Perl.

  ```
  rsync -avz rsync://ftp.linux.activestate.com/perl-current/ .
  ```

- Documentation standards (or lack thereof).
- Documentation is done in POD (`perldoc perlpod`), and lives in the `pod/` directory of the perl source tree.
- Save a copy of the original pod file before making any changes.
- Make changes.
- Update test cases if necessary.

10. Current Documentation for rand() in POD

```
=item rand EXPR

=item rand

Returns a random fractional number greater than or equal to C<0> and less
than the value of EXPR.  (EXPR should be positive.)  If EXPR is
omitted, the value C<1> is used.  Automatically calls C<srand> unless
C<srand> has already been called.  See also C<srand>.

(Note: If your rand function consistently returns numbers that are too
large or too small, then your version of Perl was probably compiled
with the wrong number of RANDBITS.)
```

---

11. Updated POD Documentation for rand()

```
=item rand EXPR

=item rand

Returns a random fractional number greater than or equal to C<0> and less
than the value of EXPR.  (EXPR should be positive.)  If EXPR is
omitted, the value C<1> is used.  Automatically calls C<srand> unless
C<srand> has already been called.  See also C<srand>.

Apply C<int()> to the value returned by C<rand()> if you want random
integers instead of random fractional numbers.  For example,

    int(rand(10))

returns a random integer between C<0> and C<9>, inclusive.

(Note: If your rand function consistently returns numbers that are too
large or too small, then your version of Perl was probably compiled
with the wrong number of RANDBITS.)
```

---

12. Preparing the Patch

- Patches are just inverse diffs
- There are two types of diffs:
  - context diffs (`diff -c`)
  - unified diffs (`diff -u`)
- Either type accepted, but unified diffs are preferred as they’re more human-readable
- Apply patch from root of perl source tree

13. Patching perlfunc’s Documentation for rand()

```
$ cd pod

$ cp -va perlfunc.pod perlfunc.pod.orig

$ emacs perlfunc.pod

$ cd ..

$ diff -u pod/perlfunc.pod.orig pod/perlfunc.pod

--- pod/perlfunc.pod.orig   Thu Apr  5 18:54:57 2001
+++ pod/perlfunc.pod        Sun Apr 29 21:32:35 2001
@@ -3509,6 +3509,13 @@
  omitted, the value C<1> is used.  Automatically calls C<srand> unless
  C<srand> has already been called.  See also C<srand>.

 +Apply C<int()> to the value returned by C<rand()> if you want random
 +integers instead of random fractional numbers.  For example,
 +
 +    int(rand(10))
 +
 +returns a random integer between C<0> and C<9>, inclusive.
 +
  (Note: If your rand function consistently returns numbers that are too
  large or too small, then your version of Perl was probably compiled
  with the wrong number of RANDBITS.)

$
```

14. Testing the Patch

- Apply patch locally before submitting
- View with `perldoc` to check formatting
- If updating test cases, run tests

15. Testing Patch to perlfunc.pod

```
$ diff -u pod/perlfunc.pod.orig pod/perlfunc.pod >mypatch

$ cd pod

$ mv perlfunc.pod perlfund.pod.new
$ cp perlfunc.pod.orig perlfunc.pod

$ cd ..

$ patch -p0 <mypatch
patching file ‘perlfunc.pod’

$ diff perlfunc.pod perlfunc.pod.new
```

16. Submitting the Patch

- Patches should be sent to p5p (perl5-porters@perl.org)
- Email should contain:
  - A clear, descriptive subject
  - An explanation of what you’ve changed and why you’ve changed it
  - The patch itself
- Avoid mailers which
  - Word wrap the patch
  - MIME-encode the patch
  - See Porting/patching.pod for workarounds

17. Submitting Patch to perlfunc.pod

```
Date: Sun, 29 Apr 2001 21:53:48 -0400
From: Walt Mankowski <waltman@netaxs.com>
To: perl5-porters@perl.org
Subject: [DOC PATCH bleadperl] Document generation of random integers

A beginner was asking how to do this on clpm yesterday.  Adding this
patch:

* Documents how to do it.

* Saves the beginner the trouble of hunting through perlfunc to figure
  out how to truncate floating point numbers.

* Shows the range which is returned (which might not be intuitive).

Walt

--- pod/perlfunc.pod.orig       Thu Apr  5 18:54:57 2001
```

```
+++ pod/perlfunc.pod    Sun Apr 29 21:19:58 2001
@@ -3509,6 +3509,13 @@
 omitted, the value C<1> is used.  Automatically calls C<srand> unless
 C<srand> has already been called.  See also C<srand>.

+Apply C<int()> to the value returned by C<rand()> if you want random
+integers instead of random fractional numbers.  For example,
+
+    int(rand(10))
+
+returns a random integer between C<0> and C<9>, inclusive.
+
 (Note: If your rand function consistently returns numbers that are too
 large or too small, then your version of Perl was probably compiled
 with the wrong number of RANDBITS.)
```

18. Followup

- What happens if the patch gets accepted?
  - The pumpking (currently Jarkko Hietaniemi) will announce on p5p that he’s accepted the patch.
  - The change should appear fairly soon in the development tree, and in the next release version of Perl.
- What happens if the patch gets rejected?
  - Accept the wisdom of the gurus.
  - It’s not the end of the world.
  - If you’re still convinced you’re right, try repleading your case sometime in the future.
- The pumpking is always right

19. Followup to Patch Submission

```
Date: Mon, 30 Apr 2001 06:56:54 -0500
From: Jarkko Hietaniemi <jhi@iki.fi>
To: perl5-porters@perl.org
Subject: Re: [DOC PATCH bleadperl] Document generation of random integers

Thanks, applied.
```

20. Related Topics

- Patching CPAN modules
  - Submit patch to maintainer instead of p5p
- Patching core modules and perl source
  - Same procedure as described here.

---

21. More Information

- perldoc perlhack
- perldoc perlpod
- perldoc Porting/patching.pod

22. Thank You

- Any questions?
