

Contributing Code to Evergreen Is Easier Than You Think

Lebbeous Fogle-Weekley
Equinox Software, Inc.

Contributing Happens Many Ways

Many things you can do help code (and your name) get into Evergreen.

Good bug reports (and feature requests) draw better development work faster

Patch testing confirms that development work works

Actual code for the OPAC and other parts of Evergreen make prominent, visible differences.

Bug Reporting

Evergreen uses Launchpad for bug reports. <http://launchpad.net/evergreen>

“Bug reports” include feature requests and gentle suggestions that things could work differently.

Don't be shy.

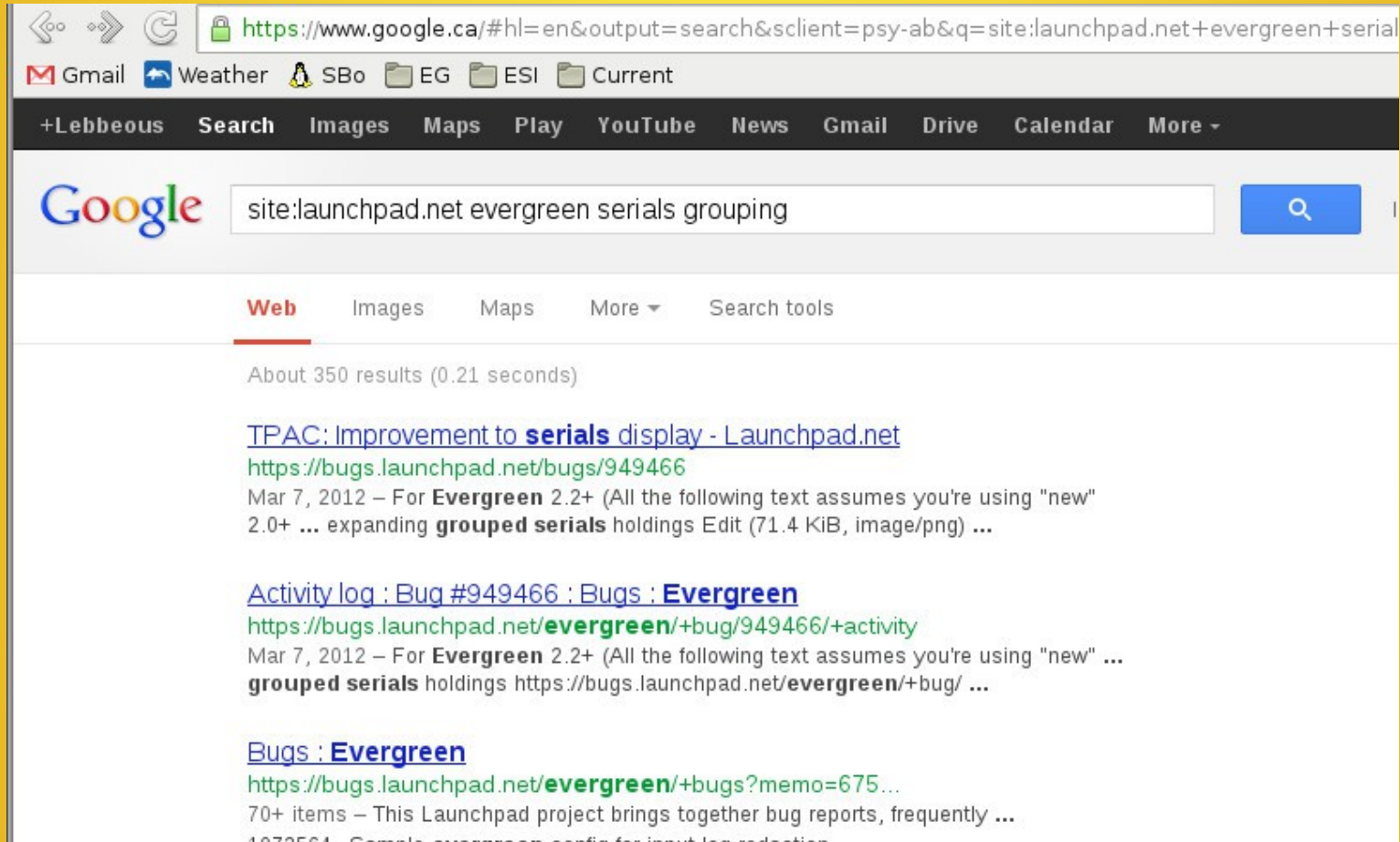
Research for a Bug Report

Search Launchpad and mailing lists for previous reports or conversations.

Ask in *#evergreen* on IRC.

Don't worry about it too much.

How to Search Launchpad



The screenshot shows a Google search interface. The search bar contains the text "site:launchpad.net evergreen serials grouping". Below the search bar, the "Web" tab is selected, and the search results are displayed. The first result is titled "TPAC: Improvement to **serials** display - Launchpad.net" with a URL of <https://bugs.launchpad.net/bugs/949466>. The second result is titled "Activity log : Bug #949466 : Bugs : **Evergreen**" with a URL of <https://bugs.launchpad.net/evergreen/+bug/949466/+activity>. The third result is titled "Bugs : **Evergreen**" with a URL of <https://bugs.launchpad.net/evergreen/+bugs?memo=675...>. The search results are dated Mar 7, 2012.

Google search results for "site:launchpad.net evergreen serials grouping".

Web Images Maps More Search tools

About 350 results (0.21 seconds)

[TPAC: Improvement to **serials** display - Launchpad.net](#)
<https://bugs.launchpad.net/bugs/949466>
Mar 7, 2012 – For **Evergreen** 2.2+ (All the following text assumes you're using "new" 2.0+ ... expanding **grouped serials** holdings Edit (71.4 KiB, image/png) ...

[Activity log : Bug #949466 : Bugs : **Evergreen**](#)
<https://bugs.launchpad.net/evergreen/+bug/949466/+activity>
Mar 7, 2012 – For **Evergreen** 2.2+ (All the following text assumes you're using "new" ... **grouped serials** holdings <https://bugs.launchpad.net/evergreen/+bug/> ...

[Bugs : **Evergreen**](#)
<https://bugs.launchpad.net/evergreen/+bugs?memo=675...>
70+ items – This Launchpad project brings together bug reports, frequently ...
1073564 - Sample **evergreen** config for input log reduction

Developers Don't Mean to Be Jerks

We sometimes mark bugs as duplicates of other bugs, or we tell you that your issue is a result of local configuration or something, and not applicable to Evergreen in general.

This is just information management; we don't mean it personally.

Moreover, sometimes we're wrong.

Tips For Good Bug Reporting

I asked Mike Peters of Emerald Data Networks for some tips from his experience.

My favorite of his tips? Hit developers when they're looking. Bug-fixing activity surges at conferences, real life gatherings, and online meetings.

Add Data to Someone Else's Bug

Sometimes you'll find that you have an interest in a bug someone else reported.

If you can reproduce their symptoms on your system, add data to the bug report to help it get attention. Data makes a richer “me too.”

More Tips for Good Bug Reporting

These, and the last slide, are still thanks to Mike Peters.

Cultivate relationships with sites on the same or nearly the same Evergreen version

Don't be afraid to report issues big or small.

Research and add detail as much as you can.

Tag your bugs in Launchpad with likely keywords to make it easier for other to find.

Guide to Form in Bug Reporting

Just report it! :-)

http://evergreen-ils.org/dokuwiki/doku.php?id=dev:bug_wrangler:faq

Or Google: *evergreen bug wrangler faq*

Ok, there are actual guidelines, and this document is carefully written and less verbose than some on the wiki, so it's a good one to read.

But don't let perfect be the enemy of good when reporting bugs.

Patch Testing

When somebody tackles the problem described in a bug report by producing some changes to the Evergreen code, let's just call their product a patch (and not worry about other terms like “branch” and “commit” for now).

Patches Need Your Testing

Depending on the developer and the situation, a bug report may get a well-tested patch or one that hasn't been tried at all.

Patch testing doesn't have to be the exclusive domain of developers. You can help push this train.

People Skills and IT Skills

You don't have to be a system administrator to test a patch if you have access to a test server where you can get somebody else to load the patch and appropriate data for you.

Jason Stephenson of the Merrimack Valley Library Consortium has made user testing possible for dozens if not hundreds of patches already in this way.

We Can't Ask Jason to Do It All

Installing an Evergreen server, not to mention providing controlled access and loading patches to be tested, takes continual effort.

I've talked with others, like Kathy Lussier of MassLNC, who feel that patch testing servers, like the public demo servers, but different, would be a huge boon to the community.

Not All Patches Call for User Tests, But Many Do

Bug reporters can help by listing explicit steps to reproduce their issue right in the Launchpad bug report.

When the same steps are played back to confirm that a patch fixes a problem, you can call them a *test plan*.

Test plans greatly expedite the whole testing and sign-off process and help confirm correctness.

How Do We Capture Value from User Tests?

We need to have a discussion, perhaps in an online developers' meeting (all can attend) about how to count user tests where our system today depends more on developer sign-offs.

By making it easier for user-testers to usher patches into Evergreen proper, we can incentivize more testing.

Let's not forget about tests done at sites working with a vendor to solve a particular problem. Let's document that testing more publicly and put it to use.

Patch Meetings

So for a little while we had these meetings where developers would identify fixable bugs and harvest some low-hanging fruit or scratch longstanding itches.

What happened to these meetings? We should bring them back at a sustainable interval and broaden the participant pool to include others from the community who can test on servers with some help from system administrators.

Another topic for a developers' meeting.

As with Bug Reporting, Be Noisy

Bug reports can languish at any stage in their life cycle if people who might push it to the next stage get occupied by something else.

As with bug reports, it's okay to (in a professional, friendly way) pester the folks whose attention you need to get acknowledgment of your testing effort.

Code Code

Let's talk about contributing actual code directly now.

There's one place in particular where it's relatively easy to get a foothold in the code. Some knowledge of HTML and CSS can get you far.

The OPAC

```
lebbeous@dev-vm1: ~/git/EG-EQ
<input type='hidden' name='part' value='>
  [% END %]
    [% END %]
  </td>
</tr>
[% END %]
</table>

<p>
  [%- org_select_id = 'pickup_lib'; -%]
  <label for="[% org_select_id %]">[% l('Pickup location:') %]</label>
  [% PROCESS "opac/parts/org_selector.tt2";
    INCLUDE build_org_selector name='pickup_lib' value=ctx.default_pickup_lib id=org_select_id can_have_vols_only=1 %]
  </p>
  <p>
    [% l('Notify when hold is ready for pickup?') %]
    <blockquote>
      <input type="checkbox" id="email_notify" name="email_notify" value="t"
        [% IF !ctx.user.email %]disabled="true"[% ELSIF ctx.default_email_notify %]checked="checked"[% END %]>
      <label for="email_notify">[% l('Yes, by Email') %]</label><br>
    </blockquote>
    [% l('Email Address:') %] <span name="email_address">[% ctx.user.email %]</span>
  </blockquote>
  [%- IF allow_phone_notifications == 'true' %]
  <input type="checkbox" id="phone_notify_checkbox" name="phone_notify_checkbox"
    [% IF ctx.default_phone_notify %]checked="checked"[% END %]>
    <label for="phone_notify_checkbox">[% l('Yes, by Phone') %]</label><br>
  </blockquote>

```

90,1 72%

HTML and CSS

Melissa Lefebvre of Bibliomation points out that CSS in particular is useful in smoothing out layout problems.

She also tells me that efforts to clean up the HTML, i.e. using `<div>` tags instead of tables and applying classes instead of inline styles, really pay off when sites want to do OPAC template customizations.

Javascript in Staff Client Interfaces

Ben Shum, also of Bibliomation, believes that the Javascript in many staff client interfaces, including legacy HTML-based interfaces for configuration, may be the next most approachable area after the OPAC.

Ben also salutes the sharing of customizations that can lead to ideas for general improvements to the main code base.

Ben feels strongly that new developers, e.g. through the GSOC program, should be encouraged. Possibly those with serious library knowledge should be matched with newcomers with programming ability to mutual benefit.

The Technical Side: Git, diff, patch and whatnot

I assumed people have changes in mind that they would make to Evergreen, and that they would want to know how to get their changes published with a minimum of fuss with developers' tools.

Yesterday at the developer hackfest I learned I was wrong, at least for some of you.

Again, Don't Be Shy!

Paraphrasing Melissa, don't assume that no one else wants to do what you did and that your code won't end up in a release. They do and it will.