Developer's update — Evergreen Conference 2012

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Overview

- What we've done, by the numbers
- Google Summer of Code
- RM & RMaints
- Evergreen 2.3
- Path to Evergreen 2.2.0
- Evergreen 2.1.x
- What's in 2.2.0
- The future

2 Development statistics redux

No cool visualization this time!

3 Code contributors

Over the past year, we have seen:

- 2209 commits from a total of **29** different authors (8 active core committers)
- 9 contributors outside of the core committer group with 5 or more commits:
  - *Jason Stephenson* - 48
  - *Michael Peters* - 26
  - *Scott Prater* - 20
  - *Joseph Lewis* - 19
  - *James Fournie* - 16
  - *Robin Isard* - 12
  - *Liam Whalen* - 6
  - *Ben Shum* - 6
  - *Steven Callender* - 5
- One female contributor - *Sarah Chodrow* (More, please!)

4 Communication

- Developer mailing list
  - 970 messages
- Internet relay chat (IRC) channel
  - 76,476 lines and other stats
    - **tsbere** and **dbs** in a neck-and-neck race with 13,474 and 12,062 lines, respectively
    - 26 people averaged more than one line per day
- Developer IRC meetings
  - 19 meetings held

5 Releases

- **2.0 series**
  - *April 2011* - 2.0.5
  - *May 2011* - 2.0.6
  - *June 2011* - 2.0.7
  - *August 2011* - 2.0.8, 2.0.9
  - *October 2011* - 2.0.10, 2.0.10a
- **2.1 series**
  - *October 2011* - 2.1.0, 2.1.0a
  - *November 2011* - 2.1.1
- **2.2 series**
  - *November 2011* - 2.2 alpha1
  - *March 2012* - 2.2 alpha2, 2.2 alpha3
  - *April 2012* - 2.2 beta1, 2.2 beta2

6 Google Summer of Code students

- Daniel Rizea
- Joseph Lewis
- Pranjal Prabhash
- Swenyu Duan

7 Daniel Rizea

- **Project**: Create Android client(s) for Evergreen
- **Mentor**: Dan Wells
- **Location**: Bucharest, Romania
- **Project Information**
  - Create patron app
    - Perform basic searches
    - Access "My Account" features
    - Potential enhancements: offline list browsing, barcode scanning, calendar integrations
  - Create staff app
    - Possible uses: inventory, weeding, roving circ

8 Joseph Lewis

- **Project**: Upgrade Dojo to bring Evergreen to the modern web
- **Mentor**: Tom Berezansky
- **Blog**: http://evergreengsoc.blogspot.com/

Joseph participated in last year's GSoC, re-writing the org unit settings editor to give us categories, filtering, and history.

This year he will be working on getting us to a later version of Dojo, hopefully resulting in better browser support and fewer Dojo-related bugs. Not to mention being able to take advantage of features in later Dojo versions.

Most of his work is likely going to be inside of the staff client's configuration interfaces, but there are a lot of old conventions that need replacing. He has his work cut out for him.

9 Pranjal Prabhash

- **Project**: Create a PHP client code for OpenSRF and Evergreen
- **Mentor**: Lebbeous Fogle-Weekley

Pranjal is pursuing his master's degree in Mathematics at Birla Institute of Technology and Science (BITS), Pilani, India. He's got experience in C and C++, Java, and PHP, and he's beginning to sink his teeth into OpenSRF.

Pranjal is finishing off his final exams right now, but he's excited to start making more noise in the community ASAP!

Bob Wicksall: are you in the room? Lebbeous would like to talk to you about the work he knows you've already done on PHP client code.

10 Swenyu Duan

- **Project**: Optimize Evergreen: Convert PL/Perl-based PostgreSQL stored procedures to PL/SQL or PL/C
- **Mentor**: Mike Rylander

Welcome, Swenyu Duan from Peking University! (Not actually here. . .)

- **IRC:** stormdsy
- **email:** dsy88 at sina dot com

Problem: Evergreen makes use of many PL/Perl Postgres stored procedures such as these normalization routines. Many of these are very stable, and would benefit from being reimplemented in a faster language — PL/SQL if particularly simple or query-heavy, or PL/C if more complicated or data-processing intensive. Your mission is to identify and address performance bottlenecks. (This is an excellent opportunity to coordinate your efforts with another student working on stress and regression testing!)

11 Releases

- One basic question: what will we get?
- Second basic question: when, already?

We have a better handle on what… when, needs work.

Time-based releases

12 Time-based releases

The goal

1. Two major release a year, six months apart.

March and September?

1. Bugfix releases on a periodic schedule

Every month?

13 Release Manager

Responsible for making sure quality releases get made on time.

Not single-handedly

14 RM Responsibilities

- Propose release schedule
- Be final, neutral technical arbiter of what does or does not make it into a release
- Help coordinate the efforts of coders and testers to make sure that patches are looked at on a timely basis
- Be the designated tie-breaker

AND

- *Regularly* communicate the status of the release to the rest of the community

15 Term

- Chosen among the active committers; currently by developer consensus.
- Term is until the release of X.Y.0
- After the release, time to choose the next RM

16 Release Maintainer

- Does most of what the release manager does, but for bugfix releases.
- There *can* be more than one
- Chosen among the active committers and developers

17 Evergreen 2.3

Introducing Release Manager Bill Erickson

18 Path to Evergreen Evergreen 2.2.0

Introducing Release Maintainer Lebbeous Fogle-Weekley

- Bug triage over next few days
- Freezing master for a bit to focus on 2.2 release
  - That's right: no pushing to master
- Release candidate in 3-4 weeks
- Release by June 8th
- Where the Release Manager will prepare updates for the broader community on development status, Lebbeous has volunteered to apply the digital shoe leather and present the prepared updates, making sure that communication is happening between the groups within the Evergreen Community

19 Evergreen 2.1

Introducing Release Maintainer Dan Scott

20 What's in 2.2.0

Lots!

21 OPAC

1. Template Toolkit OPAC
2. Auto suggest in catalog search
3. Copy location groups
4. Custom org unit trees
5. Configuring search limits

22 Circulation

1. Fine accrual on closed dates
2. User activity types
3. Target copies for holds at closed libraries
4. SMS text messaging
5. Telephone notices
6. Opt-in circ and hold history
7. Barcode autocompletion
8. Alter hold pickup lib from shelf avoids transit
9. Grace period auto extension and backdate awareness

23 Cataloging

1. Authority control sets
2. Batch importing MARC records
3. Prevent bibliographic records from having attached copies
4. Restrict Z39.50 sources by permission group
5. Switch copy location name and library short name in copy editor

24 Reports

1. New views for reporting sources

25 Acquisitions

1. Improved navigation of acquisitions providers and funds
2. Vandelay integration into acquisitions

26 Serials

1. Serials note editing
2. Numerous bugfixes

27 Staff client

1. Customizable toolbar
2. Double clicking in the staff client
3. Recent staff searches
4. Open new tab button
5. Hide fields in copy editor
6. Secondary sort columns

28 System

1. Authentication proxy

29 The future

We know that there will be lots of cool new features in 2.3... here are some ideas and works in progress:

- Inventory
- Acquisitions multiple POs.
- Enhanced EDI. orders
- Recommendations engine
- Improved reports module
- Dedupping stub records nightly to prevent duplication for non centralized ordering.
- Address check against postal database.

30 The future includes you

But those are just some ideas...

but what do *you* want?

Thanks!