FESTNET: A VOLUNTEER DATABASE

Tales of human factors from the field. Literally.

Sean Woods
15th Annual Tcl/Tk Developer's Conference,
Manassas, VA 2008

TABLE OF CONTENTS

FESTNET: A VOLUNTEER DATABASE

Tales of human factors from the field. Literally.

Sean Woods
15th Annual Tcl/Tk Developer's Conference,
Manassas, VA 2008

Abstract

The Philadelphia Folk Festival operates a campground, and every year it's the Camping Committee that sets it up and keeps it running. The Committee is run by over 400 volunteers, and coordinating them effectively requires a great deal of record keeping. For 20 years, the committee has been using a computerized database of some form or another. For the last 10 years Sean Woods has been operating that Database in a Tcl based application. This paper will describe many of the challenges Sean has encountered, lessons learned, and ways in which Tcl in particular has aided him on his quest to keep the data flowing, the volunteers in the field, and him from out in front of a computer for the entire festival.

What Is FESTNET?

FESTNET is a database used by the campground at the Philadelphia Folk Festival to track its volunteers. While it has taken many forms of the past 20 years, it has always had 5 major functions:

1. Track registrations of volunteers.
2. Maintain a mailing list of volunteers to solicit for next year's folk festival.
3. Schedule volunteers to work.
4. Flag those that miss work.
5. Track those who we do not wish to invite back next year.

It it's present form, FESTNET is a Tclhttpd application that is interfaced through the web. It uses elements of my Tao object framework integrated with a 10 year old codebase written specifically for Tclhttpd. The present version uses Sqlite 3.x as a database backend. The many shapes and forms the system has taken over the years will be discussed in a History section later in this paper. I'd rather get the areas of general interest introduced first, and leave the "you would to have been there" for appendices.

The FESTNET User Interface

Before I delve into the schema and the nuts and bolts of the system, I want to take a minute to describe how the system looks to the users. The interface is designed to be relatively simple, because most of its users will operate the system only once per year. And being volunteers, they really don't want to have to jump through hoops, remember numbers, or otherwise need to know anything more than some personal information and where to point their browser to sign up.

The UI has 3 different modes: Basic, Data Entry, and Admin.

- Almost all of the volunteers access the system with the Basic interface. This allows them to log on, register for the upcoming folk festival, update their contact information, and see who else is coming.
- The Data Entry interface allows office staff to access the records of other volunteers, enter scheduling information, register volunteers, generate rosters, enter evaluations, and access some limited reporting features.
- The Administrative Interface is open only to supervisors and the database administrators. The interface gives access to generate mailing lists, perform certain high level organizational functions, as well as maintain the database.

Logging In

Based on my own experience with websites, I realized early on the require a password for this website was counterproductive. People log on once per year, and there is no guarantee that the email they gave last year would even work if we were to implement at password reset feature. So the balance between security and access was to use each volunteer's date of birth as their password. While we have a volunteer ID number we would prefer people log in with (complete with a lookup feature built into the log in page) we also allow people to log in using their last name as the username. A pairing of a last name and date of birth is sufficiently unique that we have not had a problem to date. Granted, all we would need is a set of twins to muck up the works, but that's why we also have the login with one's userid feature too.

Perfect from a security standpoint? Oh hell no. Perfect from a "I don't get calls and emails from frustrated volunteers telling me they can't log in?" Yes. And given that we are only trading in publicly available information, there really isn't a whole lot that someone could glean if they already knew a person's DOB and name.

The major fly in the ointment is that I require people to enter their DOB in ISO format: YYYY-MM-DD. Despite having examples on the login form, every year I do get complaints from people about how confusing that is. After 10 years, I just roll my eyes.

Registration

When a volunteer logs onto the system, they receive either a notice that they are registered for the upcoming folk festival, or a notice that they are not, with a link to follow to kick off the process. Registration is simply filling out a form that says "Yes I'm coming" or "No I'm not" with a field recording how many kids under 12 years old they will be bringing, and a space for any comment they might have. That's it. The rest of the basic interface is simply forms that allow the volunteer to update their contact info.

For many of our volunteers this process is either too time consuming, or too confusing, so they email/call/smoke signal someone they know that is a supervisor or clerk. (And an email address is listed right on the login page.) Because the website is up 24/7 throughout the year, registration can happen any time from anywhere. Well, anywhere with a web connection.