Terrence Monroe Brannon	
bauhaus@metaperl.com	920 S. New Hampshire Ave
http://www.metaperl.com	Los Angeles, CA 90006
(818) 359-0893	

Skills

Perl

6 years of corporate-level experience. Software publications on CPAN.org (author id TBONE). Print publications on perl.com and The Perl Journal. Emphasizes using Perl as a community language and makes efforts to stay in close contact with all daily news sources on the language. Demonstrated strength in every area in which Perl sees wide usage, namely:

Object-Oriented Programming - the use of inheritance and polymorphism to simplify complex system development.
Database-Driven Applications - procedural Perl programs manipulating business objects modeled in database schemas.
Data Munging - the reading, transformation and writing of data with XML and HTML being the current ubiquitous cases.
Web Programming - Customized, dynamic website generation via mod_perl/CGI.

SQL/Databases

Competent. Work experience (schema creation, SQL and stored procedures) with Sybase, Informix, MySQL, Postgresql and Oracle. Both print publications were on Perl database usage.

Other / Resume URLS

Competent in XML/HTML with a focus on dynamic generation of these via Perl. Fluent in C. 16 years of experience with various Unix types. Competent in Haskell.

Sole developer of http://www.GimbleRUs.com, a MySQL-driven gaming website. Perl model classes were done in Class::DBI, view classes in HTML::Seamstress and the controller was CGI::Prototype. Also used Data::FormValidator for form validation. Get source code for the entire website by typing `darcs get http://www.gimblerus.com`

Publications

T.M. Brannon, "Beyond Hardcoded Database Applications with DBIx::Recordset" The Perl Journal. Issue 20. http://www.tpj.com/
T.M. Brannon "DBIx::Recordset - a DBI Extension for Application-Level Database Use", Perl.com, Feb. 27, 2001. http://www.perl.com/pub/2001/02/dbix.html

Poirazi, P, Brannon, T & Mel, B.W. (2003) Arithmetic of Subthreshold Synaptic Summation in a Model CA1 Pyramidal Cell. Neuron, 37, 977-987.
Poirazi, P, Brannon, T & Mel, B.W. (2003) Pyramidal Neuron as 2-Layer Neural Network. Neuron, 37, 989-999

Education

M.S. Computational Neuroscience, University of Southern California, May 1999. Thesis research: Non-linear synaptic integration in cortical pyramidal neurons.
M.S. Computer Science, Lehigh University, December 1993. Thesis title: "Chameleon: An Embedded Language Perspective for Programming in Multiple Computer Languages."
B.A. Computer Science, Morehouse College, May 1987.

Employment

Perl Developer **September 2005 - present**
Vendare Media El Segundo, CA
Tools used: Perl (HTML::Mason, HTML::Tree, Log::Log4Perl), CVS.
Skills used: Web programming, Search service development

Worked on the web-based Vendare search application. This application delivers HTML and XML results of searching. Was solely responsible for upgrading the framework from generating HTML via HTML::Mason to generating HTML via HTML::Seamstress. Also solely responsible for generating XML search results.

Perl Developer **October 2004 - September 2005**
Valueclick Westlake Village, CA
Tools used: Perl (Class::DBI, mod_perl, mod_rewrite,HTML::Mason), MySQL, Perforce.
Skills used: Database processing, UI development

1. Added a new status, "hidden", to advertising campaigns. Made relevant changes in the display and model. The simple model fix

was to filter campaigns as a function of the new status. Another model fix was to dynamically add the new status to the business logic as appropriate. The final model fix was to split a network table and rewrite all inline SQL to acesss the tables using the object-relational layer Class::DBI.

2. Developed Javascript::Select::Chain, a perl API to a Javascript library which allowed for arbitrary levels of pulldown chaining.

3. Developed a generic class for sorting and paginating database query results.

4. Use mod_rewrite to control ad serving logic.

5. Took a manual 3-stage process and made it runnable via one shell call.

Perl Developer **January 2003 - August 2004**
Direct Synergy Westwood, CA
Tools used: Perl (DBI, Mail::Box, Spreadsheet::ParseExcel, Spreadsheet::WriteExcel), MySQL, Sybase, SQL Server 2000
Skills used: Excel processing, Database Processing, Data Munging

This company was in the business of attracting, warehousing, and selling business leads for various target markets. My job was to use or design Perl-based technology on databases, data files, and Excel files related to their business process:

- Wrote a data cleaner, removing records having any of the following: profanity, duplicates, opt-outs, invalid email.

- Took 16 tables representing leads from their various portals and created a set of normalized MySQL tables to centralize the disparate data sources. Developed Perl programs to load the centralized table from data (CSV) files via timestamped staging and integration steps. The transform on data fields to database table columns was symbolic and facilitated by the use of DBIx::Recordset. During the process, I developed a module which allows for succinct specification of MySQL table duplication (see DBIx::Table::Dup on CPAN).

- Provided on-the-fly statistics, such as the number of leads of a certain gender in a certain age range or from a particular zip prefix. From such criteria also developed lead lists filtered on opt-out, redundancy, max count criteria. Insured that subsequent similar requests would not lead to repeat leads.

- Took an Excel file and re-formatted all phone numbers into a consistent format using a hash of Perl regular expressions.

Perl-Interwoven Developer **September 2002 - December 2002**
City of New York, Department of IT Manhattan, NY
Tools used: Perl (XML::TreeBuilder, Spreadsheet::ParseExcel, Spreadsheet::WriteExcel),
Skills used: XML processing, Data Munging, Text parsing (regexps)

Worked on the City of New York's 3-1-1 Call Center Application. This application did customer service via Siebel 7 using content from Interwoven Teamsite. My job was to populate and integrity-check the TeamSite repository. To do so, entailed writing several Perl programs:

xls-to-html: Take content developed in Excel spreadsheets and convert

it to HTML. In the process, sanity check the Excel spreadsheet for unprintable characters and empty rows.

html-to-dcrs: Take the HTML and convert it to two target formats, one for digestion by Interwoven teamsite and another for use by Siebel 7. Creating the Interwoven content entailed the copying of presentation and data capture templates, making directories and creating a large number of small XML files called ``data content records''. To aid the creation of Siebel integration objects, the plain text was converted to a searchable XML database. The database had 3 levels:agency, division, and service.

agency-def-maint: This program did two things. It could create a TeamSite repository for a new agency based on an XML file. It could also ``synchronize'' a repository with an XML file, removing items from the repository which did not exist in the XML file.

rename-dcr: Changing the name of a data content record has ramifications in other parts of the TeamSite repository as well as in the XML database. This script did a ``smart rename'', by taking care of all dependencies implicit in the rename of a DCR as a function of the DCR's type.

Senior Application Engineer **February 2001 - October 2001**
Oracle, Applications Integration Group Redwood Shores, CA
Tools used: Perl (DBI, Net::FTPServer, LWP), Oracle DBMS, PL/SQL, HTML
Skills used: mod_perl/CGI, Database-Driven Application, SQL

- Worked in the Applications Integration group. This group is responsible for releasing software patches and complete distributions via a database-driven mod_perl web site. Though the term "project" is used throughout this job description, the actual work varied in size from quick bug fixes to extensive solo development efforts.

Projects completed

- Project 8: Authored perltest.pl, which generated and formatted the questions for the Oracle India Perl Competency test. Also responsible for more than 50% of the actual questions used.
- Project 7: Sole developer of ARUFTPD, an virtual file-system FTP server. This server was derived from the CPAN Net::FTPServer to provide navigation and retrieval of files and directories on a remote file system with an FTP front end. Authentication and authorization were handled via SQL queries on an Oracle database.
- Project 6: Used libwww-perl to write a "web spider" to simulate

browser navigation across multiple screens (including login and authentication) in order to allow command-line based download of patch files. My initial implementation visited several pages in succession, parsing each page for its action to decide on the next page to request. The redesigned version subclassed the existing download class with a batch derivation and the resulting LWP script made one page call and the derived class called all the screens by calling the Perl methods to generate each screen. In my eyes, my initial approach was better, but higher management preferred the latter. Wrote a regression test suite to ensure usage under normal and exceptional cases.

- Project 5: Developed a mod_perl handler, which allowed for partial-content (resumable) downloads.
- project 4: Wrote a PL/SQL program which scanned all Oracle product families for those with upload responsibilities and added these responsibilities to the appropriate administrators.
- Project 3: Used subqueries to update the comment field of all patches related to a particular patchset.
- Project 2: When our OraDB API was re-implemented using DBI instead of Oraperl, I wrote an evaluation of the code conversion. I also designed and implemented an API for calling stored functions and procedures that would add the ability to bind OUT parameters without breaking existing calls which only used IN parameters.
- Project 1: Developed a new dynamic HTML page and the corresponding SQL/PLSQL which would allow previously obsoleted patches to have their reasons for obsolescence edited and committed to the database. Program business logic was conditional to the mode of screen processing (obsolete versus re-obsolete). Diagrammed the process flow with Visio prior to making system changes.

Perl Programmer **May 2000 – January 2001**
Instinet/Reuters Manhattan, NY.
Tools used: Perl (Net::FTP, Config::INIFiles)
Skills used: Object-oriented programming, Data munging, parsing

Developed INETCROSSFTP, an FTP system with the following phases: drag and drop GUI interface, file conversion, FTP file upload/download/monitoring, and PGP-based file encryption. Behavior reuse, composition and configuration as a function of individual client requests, static configuration file and dynamically supplied command line options was achieved via a 3-level object-oriented hierarchy of iterable behaviors. The base class was HostCycle. All subclasses cycled across FTP servers performing their until() method until it returned true at which time they executed their then() method.

Identification of common Net::FTP usages led to my development of Net::FTP::Common for CPAN.

Also developed a smaller script which was responsible for monitoring FTP repositories for incoming trade data and transferring it to a different machine via FTP.

Developed scripts and modules to parse and generate fixed-width text files for various EDI (electronic data interchange) feeds. Improved my CPAN module Parse::FixedLength. Extended Text::FixedLength with Text::FixedLength::Extra by creating a simpler API and adding support for leading zero and floating point formatting.

Lead Developer **January 1999 – May 2000**
Angryman.com Queens, NY.
Tools used: Perl (DBI, CGI, DBIx::Recordset, Cache::Cache), MySQL, HTML
Skills used: mod_perl/CGI, Database-Driven Application, Object-Oriented Programming

Developed a persistent distributed framework in which Vote objects were committed to a File::Cache and then resurrected and committed to database via slave objects.

Used Date::Manip to develop Date::Horoscope, a module that determines Zodiac sign based on birth date (These days Date::Range might be more appropriate). Also used Date::Manip to determine a persons age bracket based on birthdate.

Used libwww-perl to convert slow-loading dynamically generated pages to static cached equivalents. Ultimately developed CGI::Cache (now maintained by David Coppit).

Used HTML::Embperl to develop a multi-screen "Create-a-Poll wizard" Used DBIx::Recordset to commit the generated poll to database.

Used Date::Manip to realistically populate a MySQL customer database with lease initiation and shipment dates. (lease-processing/random-init.pl).

Developed a CGI interface for SQL temporal queries of a database.

Wrote DBI scripts to randomly allocate client leads to salesmen based on various time criteria (ie, 100 oldest leads, all leads on a certain date, etc).

Developed a set of Perl, CGI, and Javascript scripts as well as a MySQL database to collate and analyze flat file user execution logs from 300 web servers. The phases of the project were:

1. Designed a multi-table MySQL database in which to store log entries.
2. Wrote a Perl/DBI script to parse Apache logs and store them in the created database.
3. Wrote a Perl/DBI script to query the database. Queries made of use table joins. Complex queries were handled by creating temporary tables.

4. Developed a web interface to the Perl/CGI query script.