

19/09/2006 22:33:00

Terrence Monroe Brannon, Python & Perl Developer
580 American Ave #B110, King of Prussia, PA (818) 359-0893
metaperl+jobs@gmail.com - <http://www.metaperl.com>

QUALIFICATIONS

Interested in blue sky modern coding or refactoring legacy code using these coding ethics:

- Holistic code strengthening as opposed to “bugfixing by band-aids”
- Modular development, documentation and testing of application business logic
- Rigorous validation of subroutine and web input
- Separation of concerns to leverage functional composition
- Open source whenever possible.

SKILLS

Perl, Python, SQL (Postgresql, MS-SQL, MySQL, and Oracle), HTML, XML, Unix, Linux, OS X

PUBLICATIONS

- T.M. Brannon, "Beyond Hardcoded Database Applications with DBIx::Recordset" The Perl Journal. Issue 20.
- 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: "Chameleon: An Embedded Language Perspective for Programming in Multiple Computer Languages."*
- B.A. Computer Science, Morehouse College, May 1991.

EXPERIENCE

Lead Data Analyst UtekCorp.com, Tampa, FL June 2006 – April 2007
Tools used: Python (pycurl, pyparsing, ftputil), Perl, Apache, CGI, MySQL, MS SQL Server
Skills used: data munging, object-oriented programming, web programming

Performed in a data processing role spanning several UTEKcorp.com web properties:

- KnowledgeExpress.com – implemented a configurable suite of object-oriented Python and Perl data processing programs. These programs used heterogeneous access and parsing methods to regularize a jungle of vendor data for commission to our MS-SQL database and uniform front-end search.
- Pharma-Transfer.com – wrote a Python program to email a newsletter of recent relevant search results to each customer.
- TechEx.com – performed database and CGI program maintenance as well as some front-end web design.

Perl Developer VendareMedia.com, El Segundo, CA October 2005 – January 2006
Tools used: Perl (HTML::Mason, HTML::Tree, Log::Log4Perl), CVS.
Skills used: web programming, object-oriented-programming.

Dynamically delivered HTML and XML results of searching to parked domain names. 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 Valueclick.com, Westlake Village, CA October 2004 – October 2005
Tools used: Perl (HTML::Mason, CGI/mod_perl, Class::DBI, MySQL), Perforce.
Skills used: web programming, resumable Perl scripting, object-oriented programming

- 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 access the tables using the object-relational layer Class::DBI.
- Developed the CPAN module Javascript::Select::Chain, a perl API to a Javascript library which allowed for arbitrary levels of pulldown chaining.
- Developed a generic class for sorting and paginating database query results.
- Made a user welcome page load 8 times faster. Did so by profiling the rendering and moving the slow computations and their rendering to another page.

Perl Data Analyst DirectSynergy.com, Westwood, CA January 2003 – June 2003
Tools used: Perl (DBI, Spreadsheet::ParseExcel, Spreadsheet::WriteExcel), MySQL, Sybase
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 which removed records having any of the following: profanity, duplicates, opt-outs, or 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 (DBIx::Table::Dup).
- 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.
- Wrote an email-based wizard which validated and corrected Excel spreadsheets.

Perl Consultant Accenture/City Gov't of New York September 2002 – December 2002
Tools used: Perl (XML::TreeBuilder, Spreadsheet::ParseExcel, Spreadsheet::WriteExcel),
Skills used: XML processing, Data Munging, Text parsing

Worked on the City of New York's 311 Call Center Application. My job was to populate and integrity-check the content repository. To do so, entailed writing several dependency-aware Perl programs to convert Excel spreadsheets to XML.

Senior Perl Developer Oracle, Redwood Shores, CA January 2001 – October 2001

Tools used: Perl (DBI, Net::FTPServer, LWP), Oracle, PL/SQL, HTML

Skills used: web programming, SQL, object-oriented programming

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.

- 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.
- 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.
- 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.
- Developed a mod_perl handler, which allowed for partial-content (resumable) downloads.
- Wrote a PL/SQL program which scanned all Oracle product families for those with upload responsibilities and added these responsibilities to the appropriate administrators.

Perl Consultant

May 1998 – January 2001

New York and Los Angeles

Tools used: Perl (Net::FTP, Config::INIFiles, DBIx::Recordset, Cache::Cache, CGI/mod_perl)

Skills used: Object-oriented programming, Database driven application, Data munging

Instinet/Reuters

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. The program used a 3-level hierarchy of object-oriented iterable behaviors. In retrospect, one level of protoyped-based objects would have been equally powerful. 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.

Dot-coms

- **mcy.com:** Developed scripts and modules to parse and generate fixed-width text files for various EDI (electronic data interchange) feeds. Extended Text::FixedLength with Text::FixedLength::Extra by creating a simpler API and adding support for leading zero and floating point formatting.
- **angryman.com:** 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.
- **end70.com:** 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
- **hosting.com:** 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.