Erik Max Francis

Staff Software Engineer

San Jose, CA, USA

max-resume@alcyone.com
http://www.alcyone.com/max/
/in/erikmaxfrancis/
@erikmaxfrancis
@erikmaxfrancis.bsky.social
JABBER erikmaxfrancis

2025 May 07

OBJECTIVE

To obtain gainful employment in a challenging position at a forward-looking company, utilizing my particular skills and talents under a Unix, Linux, Unix-like, or platform-agnostic environment. I'm passionate about learning and particularly enjoy building frameworks and complex software from scratch.

SKILLS

- **Programming languages:** C++, C, Python
- **Operating systems:** Linux, *BSD, Solaris, and other Unix-like operating systems (both System V and BSD)
- **Other languages:** Go, Io, Java, JavaScript, Common Lisp, Emacs Lisp, Mathematica, Perl, Tcl, Scheme
- **Tools:** Bourne-like shells (sh, bash, zsh, etc.) and shell utilities, gcc, gdb, GNU Make, Ninja, vim, emacs, m4, awk, sed, telnet, ssh, rsync, wget, curses, screen, gnuplot, CVS, SVN, Perforce, git, Bugzilla, Jira, Confluence, Status Hero, Trello, DefectDojo, Snyk, Apache, sendmail, qmail/ezmlm, procmail, BIND, named, tinydns, X Window System, ctwm, sawfish, GNOME, Netpbm
- **Technologies:** TCP, UDP, HTML/XHTML, HTTP, SSI, CGI, XML, RSS, Atom, SVG, GraphViz/dot, JSON, Markdown, AsciiDoc, reStructuredText, wikitext, LaTeX, RTF, PDF, SMTP, NNTP, DNS, SSL/TLS, CORBA, SQL & SQLite3, OpenGL, HTML Canvas, jQuery, Unicode, IRC, XMPP, Slack, POSIX threads, regular expressions, client/server networking, object orientation (including patterns), domain-specific languages, system administration, system security, API design
- **Python technologies:** NumPy & SciPy, matplotlib, pandas, TensorFlow, BeautifulSoup, PIL/Pillow, PyOpenGL, pygame, pydoc, HappyDoc, Sphinx, docutils, unittest, PyPy, Python 2 vs. 3 compatibility
- **DevOps:** agile programming; unit, integration and system testing; continuous integration (CI) and continuous delivery (CD); microservices and containers (Docker, LXC, etc.); configuration and infrastructure as code
- **Advanced concepts:** analytic geometry; differential, integral, vector and tensor calculus; linear algebra; statistics; predicate calculus; physics (classical mechanics and quantum mechanics, special and general relativity); astronomy; astrophysics; artificial neural networks; machine learning and deep learning; cellular automata; Lindenmayer systems; finite state machines; genetic algorithms and genetic programming; evolutionary computation; fractals; artificial life
- **Writing:** technical documentation, how-to, non-fiction, technical advice on popular science fiction novels

WORK EXPERIENCE

Staff Software Engineer — *CloudBees, Inc. (San Jose, CA)* **(2019–)**

Contributed to **CloudBees Build Acceleration** (formerly Accelerator), a mature product with substantial annually recurring revenue that accelerates GNU Make and Ninja builds via automatic parallelization of jobs. Build Acceleration consists of a client (eMake) written in C++, an agent also written in C++, and a coordinating server (Cluster Manager) written in Java. Also involved are two **kernel modules**, EFS (Electric Filesystem) and LOFS (Loopback Filesystem), to help detect affected files during a build and to separate build sandboxes for agents running on the same host, respectively. Eventually took over **sole maintenance** of the entire product, investigating and addressing customer bug reports, adding support for new versions of operating systems and kernel versions as they were released, implementing new features when needed or requested, keeping up with features in newer versions of GNU Make, and updating the documentation and release notes for each new release. Implemented ongoing support for acceleration of new releases of **Android** (AOSP) (largely written in Go) and **Yocto** (BitBake) builds (written in Python), including a great deal of feature additions and modernization for Yocto support. Maintained and updated **documentation** for new releases and on an ongoing basis. Also took over quality assurance role. Got up to speed on already-existing **continuous integration** (CI) system (using CloudBees CD/RO), including unit, integration and system tests and took over full maintenance, expansion, and updating of said system, all running on Google Compute Platform. Prototyped acceleration support for Java tests via **JUnit 4 and 5**. Designed and implemented **Moxie**, a portable build simulator written in C89 that can simulate any build given a specification file to demonstrate real-world parallelization on a Build Acceleration cluster but requires almost no CPU usage. Build specifications can be created from existing builds or from scratch. Telecommuted.

Senior Software Engineer — *Ultimate Gaming [formerly CyberArts], Inc. (Emeryville, CA)* **(2003–2014)**

Designed, implemented and maintained a highly-multithreaded **poker server** as part of a server suite for a next-generation online gaming platform, Foundation, written in Standard C++ under Linux. Handled the full lifecycle from design, implementation, bugfixing, release, through to maintenance. Was personally and solely responsible for all poker game logic and user interface associated with the poker table, given an extremely thin and portable (Windows, Macintosh, Flash) client. Engineered the server to be very modular, easily accommodating the addition of new poker variants, resulting in a poker gaming platform offering the widest variety available in the market. Designed the system to be fully localizable and skinnable from a data-driven source. Assisted with the design and structure of the XML-based client-server protocol, as well as the architecture of Foundation, the full server suite. Advised on the design of the other server components, including the CORBA interfaces and SQL schemas needed for their interactions. Assisted with the creation of a Game Developers Kit. Helped with user interface design. Delivered stable software with very long uptimes, even initially with no dedicated quality assurance resources. Met demands for customization and feature requests for numerous customers with aggressive, rapidly-changing, and often overlapping schedules. Wrote a high-performance **client library** in Python for testing the server through the use of bots, as well as for other ad hoc tools. Wrote emulator for backward compatibility with legacy Python scripts. Designed and implemented the **Player Segmentation Module** API designed to allow administrators to efficiently specify groups of players through well-defined criteria. Created a domain-specific language to specify these criteria inspired by set theory which would compile the queries to actual SQL queries and wrappers. Integrated an XMPP client library and exposed an **XMPP API** for developers. Switched over to agile methodologies and converted the poker server to support a later thick HTML5 client using a **JSON-based protocol**. Also performed technical interviews. Telecommuted.

Senior Computer Scientist — *Adobe Systems, Inc. (San Jose, CA)* **(2003–2004)**

Returned to the Advanced Technology Group as a contractor to design and implement **SVG Renderer**, a portable (Linux, Solaris, Windows, Macintosh) Python application using the SVG core library, in order to render SVG documents to a variety of image formats via the Python Imaging Library (PIL). Created the application to be used as a standalone rasterizing utility; an HTTP server which, when provided an SVG document via a `POST` query, would rasterize the document and respond with the resulting image; as an HTTP client that would interact with an SVG Renderer server, and locally manipulating the resulting rasterized image; and as a Python module. Telecommuted.

Computer Scientist — *Adobe Systems, Inc. (San Jose, CA)* **(2000–2001)**

Ported the **Scalable Vector Graphics (SVG) core engine** written in C++ to Linux and Solaris (from Windows and Macintosh), making possible the SVG Linux viewer for Mozilla. Implemented and maintained a company-wide, portable (Linux, Solaris, Windows, Macintosh) API for SVG rendering and importing, utilized by multiple Adobe applications, including Illustrator and Photoshop. Ported an automated build validation tool written in Python to Linux and Solaris, which ran against every change entered into source control. Wrote an **automated test system** in Python to render a suite of test files and interactively display differences from the blessed testcases through a simple GUI. Provided expertise in C and C++ Standards conformance and portability. Partially telecommuted.

Computer Scientist — *Adobe Systems, Inc. (San Jose, CA)* **(1995–1998)**

Quickly and simultaneously learned the **Adobe Illustrator** 5.5 and 6.0 plugin API (Macintosh), the latter of which was under heavy development at the time. Worked with a small team to obtain basic **testing of the entire API** of over 500 calls, as well as depth and applied testing over the most important calls, including several complete plugin features. Also developed a rapid prototype in Java for a research project which converted Illustrator documents to class files runnable as browser applets. Later, developed and maintained **internal functionality and plugins** for Illustrator 7.0 and 8.0 (Windows and Macintosh) in C and C++. Responsibilities included general rasterizing libraries, raster file formats (most prominently GIF and JPEG), Web features (including hyperlinks and imagemaps), **Smart Guides**, and transformation and shape tools.

RELEVANT ACTIVITIES

- **Open source author and contributor**, including numerous Python modules and applications available at *http://www.alcyone.com/* **(2002–)**
  - EmPy, a powerful templating system used widely enough that is available as a system package for most Linux distributions; a revamped major version (4.0) was released in late 2023
  - CAGE, a cellular automata library
  - chess, a chess moves adjudicator
  - ZOE, an OpenGL-based 3D graphics engine for rapid prototyping and visualization
  - BOTEC, an astronomical and astrodynamics library
  - fauxident, an identd security system
  - uid, a customizable unique ID generator
- **Freelance technical writer**; wrote two articles for *Linux Journal* **(1999–)**
- Offered pre-publication **scientific and literary comments** on the novels *Bowl of Heaven* and *Shipstar* by Gregory Benford and Larry Niven **(2010–2013)**

au:Erik Max Francis te:emf.em si:24001 ge:EmPy/4.2 do:1786f686 co:cc58260f ti:2025-05-07T18:30:12