

Dan Weston

2614 NE 17th Ave
Portland, OR 97212
503-381-2364
dweston@nerdworks.com

What Can I Do for You?

My specialty is making cutting-edge technology accessible to everyday people. With over thirty years of programming experience and strong communication skills, I can bring thoughtful design and solid implementation to any project, large or small.

Technology Skill Set

- iPhone application development
 - Mobile app service design
 - HTTP service wire formats: XML and JSON
 - Web Applications: Java, servlets, JSP, javascript,
 - Operating Systems: Macintosh OS X, Linux
 - Misc: cryptography, digital certificates
-

Software Engineering Experience

[Moovel Group](#), Portland, OR July 2017 - December 2017 - Senior Engineer

Working with moovel's R&D team, built iOS prototypes of various new features in the mobile transit payments domain. Technologies used included Bluetooth LE, NFC, cryptography, QR codes. Our emphasis was on rapid iteration, but with eventual code transfer to the production app teams, so we couldn't be sloppy.

[eBay](#), Portland, OR, 2011 - 2017 Principal Engineer

[Critical Path Software](#), Portland, OR, 2008 - 2010- Senior Engineer

Worked on iOS and Android [applications for eBay](#), with over 60 million downloads. Over the lifetime of the project (9 years) my responsibilities covered all aspects of project from feature specification through implementation to App Store submission. Work started in 2008 as contract assignment at Critical Path Software, and continued as eBay employee after acquisition in Dec 2010. For the last few years, I functioned mostly as an architect and advance scout, working with emerging features and services to make sure that they are

suitable to the mobile use case. The main thing I have learned from this work is how to do exacting and innovative work at massive scale, both technically and organizationally.

[Stumptown Game Machine](#), Portland, OR, 2009 Contracting Software Engineer

Assisted with the development of [Touch Pets](#), a 3D interactive digital pet game for the iPhone and iPod touch. I was brought in in the final phase of the project to help clean up massive memory leaks and crashes. The main thing I learned from this work is that it is much better to catch those kind of issues much earlier in the project lifecycle.

[Nerdworks](#), Portland, OR, 2008-2017- Independent software developer

Published [Pattern Blocks](#) iPhone application for sale on Apple iTunes App Store . The main thing I learned from this work was just to get the end-to-end lifecycle of a app in the app store. Plus, it recreated a favorite childhood set of wooden blocks that I used to play with on my grandparents floor.

[Critical Path Software](#), Portland, OR, 2007-2008- Senior Engineer

Web-hosted clipart libraries and flex and flash Rich Internet Applications and video streaming. Also worked on blog printing service for popular blog software using Java-based server-side components to render html blogs to PDF. The main thing I learned from this work was to build large libraries of javascript to run in the browser, which is not something I knew before.

[Thetus Corporation](#), Portland, OR, 2003 - 2006- Senior Engineer

Ground-up design and implementation work on server infrastructure for Thetus, which provides collection and classification software for non-text data, allowing distributed collaboration and searching. My work focused on an architecture for a dynamic, plug-in task framework to allow users to extend the classification system to fit their data domain. The main thing I learned from this work was how to build a large system from the ground up in a dynamic startup environment, but if I had to do it over again, I would probably take a much closer look at existing open source pieces to build some parts of the system.

[Netforest](#), Salem, OR, 2003 - Consulting Engineer

Helped build the web site for the [Oregon Cultural Trust](#) to allow donations over the web using a third-party credit card validation system. The main thing I learned from this work was how to handle payments in a web app.

[currenex.com](#), Menlo Park, CA, 2001 - 2002 - Consulting Engineer

Design and implementation work on a distributed, highly-interactive foreign currency trading application accessible over the web. Application involved time-critical response requirements and fault-tolerant transaction semantics. Technologies used included EJB, RMI, raw sockets, Oracle database, and BEA Weblogic server. The main thing I learned from this work was the fragility of the threading infrastructure in java servers as a service achieves larger scale. We were brought in to debug crashes that were occurring as the currenex system became more popular. It turns out that their original implementation just spawned a new thread for every new connection, with no buffering or throttling, and at about 900 active threads, the thread

scheduler went all to hell in the java VM. Adding connection pooling and throttling solved the problem.

WebGain, Portland, OR, 2001 - Consulting Engineer

Created a tutorial to help users learn to use WebGain Application Composer, a unique tool for creating applications from java components. The tutorial included using EJB components to build a medium-sized web application that modeled a mutual fund web site. The main thing I learned from this work was that programming UI's that use drag and drop components are really difficult to build, and especially hard to make relevant to non-trivial application development.

driveway.com, San Francisco, CA, 2000 - Consulting Engineer

Product planning and system design for driveway.com's business-to-business offerings. Product was implemented as a set of java servlets and JSP pages communicating with underlying Oracle database servers. Business-to-business communication was implemented with XML messaging over HTTP. The main thing I learned from this work was how to apply SOAP and service-oriented-software concepts, which were quite new at the time, to a real-world problem.

Sight and Sound Software, Portland, Oregon, 1999 - Consulting Engineer

Design and implementation of Fare Specials Manager, an application that allows employees inside American Airlines to manipulate and modify complex database relations that define airline pricing rules. The main thing I learned from this work was that airline pricing rules are evil, and I never wanted to do this kind of work again.

[Intel Architecture Labs](http://intel.com), Hillsboro, Oregon, 1995-2000 - Consulting Engineer

Design and implementation of application programs using Intel's Common Data Security Architecture (CDSA). The projects included an electronic checkbook demonstrating digital signatures on checks, a certificate manager for creation of X.509 certificates and certificate revocation lists, and software to utilize PKCS-11 smart cards for cryptographic services. The main thing I learned from this work was all the fundamentals of cryptography and digital security, but I also worked with real cryptographers who were developing new crypto algorithms, and they were way smarter than me.

Flight TECH, Inc. Hillsboro, Oregon, 1996 - Consulting Engineer

Design and implementation work on Intelliflight, an in-flight information system for corporate aircraft, including GPS-driven moving maps and MPEG movie playback. This was a fun project that involved lots of interesting technologies, and it was great to see the finished product installed in a corporate jet.

[Intel Architecture Labs](http://intel.com), Hillsboro, Oregon, 1994-1995 - Consulting Engineer

Design and implementation work on adaptation of existing UNIX-based content indexing tools to Windows 3.1 and Windows NT. Final products included both stand-alone applications and CGI/World Wide Web compatible modules.

[CACHe Scientific](http://cache.scientific.com), Beaverton, Oregon, 1993-1994 - Consulting Engineer

Design and implementation work on Windows 3.1 version of CAChe's computer aided chemistry applications. Key member of overall design team. Specific duties included design of data server library to facilitate shared access to chemical data among several concurrent applications. Much of the work involved adapting existing Macintosh and UNIX code to the Windows operating environment and MFC class library.

[CAChe Scientific](#), Beaverton, Oregon, 1991-1992 - Consulting Engineer

Designed and implemented various parts of a network-based client-server application suite for performing computationally intensive tasks on distributed processors, including Macintosh-hosted Motorola 88000 coprocessor cards and IBM RS-6000 workstations. System relied extensively on Apple Events and low-level network transport protocols for interprocess communication.

[Nerdworks](#), Portland, Oregon, 1990 - Independent Software Developer

Developed a C++ application framework for Macintosh programming. The software is sold as a companion product to The Elements of C++ Macintosh Programming.

Cogent Research, Inc., Beaverton, Oregon, 1989 - Consulting Engineer

Wrote low-level system utilities to implement clipboard and sound services for QIX, a UNIX-like operating system for parallel-processing computers.

[CAChe Scientific](#), Beaverton, Oregon, 1988-89 - Consulting Engineer

Provided design consultation and programming services for a group working on a computer-aided chemistry product for the Macintosh. Work included 3D color graphics firmware for Macintosh II and application software written in C++.

E-Machines, Inc., Beaverton, Oregon, 1987-88 - Software Engineer

Designed and implemented Nubus configuration firmware for Macintosh II video products, including monochrome, gray scale, and color display controllers. Work also included diagnostic and evaluation software.

[Nerdworks](#), Portland, Oregon, 1987 - Independent Software Developer

Developed the MacApp Browser, a Smalltalk-like browsing tool for object-oriented programming on the Macintosh, marketed by the MacApp Developers Association.

Technical Writing Experience

[Developer](#), Torrance, CA, 1999-2000 - Contract Technical Writer

Wrote technical documentation for SOAP, the Simple Object Access Protocol, an HTTP/XML-based method for remote procedure calls over the internet. SOAP defines an XML encoding format for requests and responses between clients and servers. SOAP acts as the glue between heterogeneous distributed software components. In SOAP, clients and servers send these requests and responses back and forth using the HTTP transport mechanism.

[Microsoft Corporation](#), Redmond, Washington, 1992 - Contract Technical Writer
Wrote technical documentation for Windows NT. Specific topics included interprocess communication mechanisms and virtual device drivers.

[Microsoft Corporation](#), Redmond, Washington, 1990-91 - Contract Technical Writer
Wrote technical documentation for MFC version 1.0, an object-oriented class library for C++ programming in DOS and Windows. Wrote a tutorial, cookbook, and sample programs.

Cogent Research, Beaverton, Oregon, 1989 - Contract Technical Writer
Wrote technical documentation for parallel programming and PostScript graphics programming for Cogent's parallel workstations. Work included the creation of sample parallel programs in C++.

[Microsoft Corporation](#), Redmond, Washington, 1988-89 - Contract Technical Writer
Wrote technical documentation for OS/2 Presentation Manager. Developed documentation and sample programs for software developers writing applications for OS/2.

[Intel Corporation](#), Hillsboro, Oregon, 1987 - Staff technical writer
Wrote technical documentation of iRMX 386; a real-time, multiprocessor operating system based on the 80386 CPU. Duties included planning, research, writing, creating sample programs, illustrations, and layout.

Technical Training Experience

[Developmentor](#), Torrance, CA 1995-2006 - Programming Instructor
Teacher for classes in java servlets, JSP, EJB, and Web Services. The classes last one week and include direct instruction and extensive lab experience. Classes are targeted at professional software engineers looking to gain competence in object-oriented programming.

Publications

- **Elements of Macintosh C++ Programming**, Addison Wesley, 470 pages, 1990. A detailed guide to programming the Macintosh in C++.
 - **The Complete Book of Macintosh Assembly Language, volume I and II**, Scott, Foresman and Company; volume I: 568 pages, 1986; volume II: 420 pages, 1987. A comprehensive guide to programming the Macintosh in assembly language.
 - **The Second Logo Book**, Scott, Foresman and Company; 212 pages, 1985: A book on advanced logo programming covering list processing, advanced graphics, and recursion.
-

Education

- [Grinnell College](#), Grinnell, Iowa, BA, Biology, phi beta kappa, May 1975
- University of California, Davis, K-12 teaching credential, June 1978