

# PAUL SORENSEN

*Software Engineer • Developer • Architect*

28 ALDERBROOK DRIVE • CRANSTON, RI • 02920  
401-862-9494 • PSORENSEN@ALUM.RPI.EDU • JORGANDER.COM/DEVELOPMENT

---

---

## QUALIFICATIONS

- I am a software engineer with a passion for excellence in software design and development, and am seeking an opportunity in which those skills could be used to develop outstanding applications. I am a self-starter, equally capable of working in a team or alone with minimal supervision.
- I have a strong background in both theoretical and applied data structures and algorithms, communication systems, and areas of applied mathematics related to real-time interactive environments such as 2D/3D graphics.
- The majority of my experience has been in the wireless communications industry, but I have also worked in the media industry and for an internet services firm.

---

---

## WORK EXPERIENCE

- **Software Engineer** (contract) · *Background Checks International, Inc* · 8/2007-1/2008
  - Fixed and maintained, as well as re-designed as I saw fit, a .NET web-based system to create background checks on individuals. The information in these checks was garnered from many different sources via a number of protocols (SOAP, XML, HTTP, etc.) and compiled seamlessly into a report for the end user. Optional parameters were available for the user to control and add report content.
  - Responsibilities consisted primarily of development using VB.NET although I did a small amount of web site design and system administration as necessary.
- **Software Engineer** (full time) · *SeaWave LLC, Middletown, RI* · 6/2003-8/2007
  - Designed, developed, and maintained an automated internal billing system in C++/Perl that performed record processing and reconciliation, as well as interfaced to other parts of the system. Also designed and administered the database containing the billing data.
  - Created online interface to the afore-mentioned billing system using C++/CGI that allowed coworkers to view and manipulate data and view/create reports.
  - Designed and developed a self-maintaining, self-updating system in C++ for retrieval and distribution of weather content, both textual and graphical, as well as a client-side process also in C++ to render graphical data into image form.
- **Flash Programmer** (contract) · *The Academy of Electronic Media, Troy, NY* · 2002-2003
  - Designed and developed interactive Macromedia Flash/Director animations, some of which utilized 3D content created either by myself or coworkers in applications such as Softimage XSI or Maya.

---

---

## TECHNOLOGIES

- **Languages:** I am especially strong in C/C++ with five years of experience, both functional and object-oriented, but have also used other languages such as C#, VB.Net, Perl, Java, HTML/JavaScript, Assembly, etc. Projects included server-only, client/server, and internet-based systems.
- **Applications:** Platform-permitting, I prefer MS Visual Studio for development. I use Softimage 3D/XSI and Adobe Photoshop/Premiere/After Effects to create and manage graphical content.
- **IPC + Synchronization:** Multi-threading, Events/Conditions, Mutexes/Semaphores (cross-platform)
- **Protocols:** TCP/IP, HTTP, HTTPS, FTP, and SMTP
- **Database Systems:** Postgres, MySql, and Access
- **APIs:** Windows API, Berkeley sockets, DirectX, OpenGL, GD, GDI+, and others
- **Operating Systems:** Well versed in both Windows and Linux

## EDUCATION

---

---

B.S. in Computer Science, Rensselaer Polytechnic Institute (1999-2003), GPA 3.47

## WORK ETHIC

---

---

While conscious of budgets and deadlines, my focus lies on system architecture and well written code. Features such as design, documentation, extensibility, scalability, portability, efficiency, etc. are all very important to me, from a high conceptual level right down to the code itself. I'm also aware that not all features are possible everywhere - scalability might be sacrificed for efficiency, or efficiency for readability.

Code review and if possible, design review, are essential. Unforeseen consequences and changing expectations encourage me to rethink original ideas and if time and budgets permit, I perform these on a regular basis. In general I prefer an event-driven system over one that polls and systemic design over special-case design.

## PROJECTS

---

---

- **Reconciliation:** Part of a multi-stage internal billing system that processed usage records and matched them from multiple sources before being processed by invoicing. *Technologies used: MS Visual Studio, C/C++, Perl, SQL, XML, TCP/IP, FTP, Cron, PostgreSQL (both client and server administration)*
- **Reconciliation Interface:** Online interface to the Reconciliation billing system that allowed coworkers to manually perform billing tasks, such as manipulate usage records, diagnose system problems, and review billing data prior to invoicing. *Technologies used: MS Visual Studio, C/C++, SQL, HTML/javascript, CGI, PostgreSQL, Apache*
- **Weather Content Distributor:** Cross-platform server process that comprised a downloader part and a distributor part. The downloader intelligently retrieved weather data from NOAA and converted it to a form more efficient for distribution. The distributor served the weather data in GRIB format to clients via a synchronous TCP/IP request, the format of which allowed for any combination of weather type, time, and geographical area regardless of how the data was retrieved by the downloader. *Technologies used: MS Visual Studio, C/C++, Multi-threading, Mutexes/Semaphores, TCP/IP, FTP, Windows API, GRIB, Cron*
- **Weather Content Renderer:** Heavily customizable cross-platform client process that rendered GRIB data to image form. A few of the parameters were image resolution/type/quality, land/water color, graphical API, and process priority. There were also numerous parameters to control how individual weather types (both vector and scalar) were rendered. Although never used in a production environment, a COM version was also written that included more functionality (e.g. writing image data to a Windows device context instead of a file); the intent was to overlay weather data on top of nautical charts. *Technologies used: MS Visual Studio, C/C++, Assembly, GDI+, GD, Multi-threading, Mutexes/Semaphores, TCP/IP, Windows API, COM, GRIB*
- **Game Engine:** A side project I work on in my spare time, it is more something that I use to teach myself game development than an actual project. Many technologies acquired and used here aren't included in the list as they are more or less exclusive to game development; some of these are vertex and pixel shaders, scripting (LUA), collision detection, soft skinning, skeletal animation (forward and inverse kinematics), and stencil shadow volumes. *Technologies used: MS Visual Studio, Adobe Photoshop, Softimage XSI, C/C++, DirectX, OpenGL, Multi-threading, Mutexes/Semaphores, TCP/IP, 3D modeling & animation, Linear Algebra*