**Profile**

**Experience**

My 38 years of working in the IT industry has given me a wealth of IT know-how, broad exposure to many different environments, platforms, operating systems and languages, the flexibility to adjust and adapt to the ever changing technologies driving our industry, and the ingenuity and creativity to apply the innovative techniques I have accumulated over the years to streamline operations, automate time-consuming manual procedures, and to contribute to the company’s success by reducing costs and improving productivity. I love to learn and continuously investigate new technologies and new techniques. Extensive experience and expertise in server automation using Perl, CGI, HTML, CSS, JavaScript, UNIX/Linux shell scripting, data encryption, secure application data transport and design, data mining, data acquisition, data parsing, web robotics, and browser-neutral web design. Over the years, I have saved my customers and users untold hours by automating manual and/or repetitive tasks, while reducing input errors and increasing security in the process.

**04/2010 – 09/2019**

**DXC Technology, Falls Church VA**

**Role: Senior Staff Augmentation Server Tools Architect/Developer for Boeing DCIX System**

**Duties and Responsibilities:** Currently System Architect of the DCIX (Distributed Configuration Information Examiner) suite of applications. Lead a team of 3 developers tasked to complete a total system rewrite of the DCIX system code. Lead designer/architect of the new system. Create badly-needed interactive web-based reports for Boeing utilizing Perl, CGI, DBI, HTML, CSS, JavaScript and UNIX shell scripting on HP-UX and Linux servers. Write/maintain cross-platform agent applications and shell scripts collecting server security configuration compliance data which runs on AIX, HP-UX, Linux, Sun and Windows servers. Support a large and complex web-based enterprise server reporting system (well over a million lines of code) providing comprehensive historical reporting of SOx data and extensive reporting of server security compliance with company computing security standards. Write applications and scripts that detect and report installed software and patch levels on a large server population across a large global network, and many other aspects of automated server management as well. Mentor less experienced and less knowledgeable personnel. Considered the go-to guy for UNIX shell scripting, Perl, CGI, HTML, JavaScript, CSS, data mining, data encryption, web robotics and other interesting topics. Work in a highly dynamic, fast-paced and challenging technical environment.

**Accomplishments:** Created an interactive report showing who has what elevated privileges on which Boeing servers. Wrote an authentication framework to allow clients to view data only for the servers their own managers are responsible for. Wrote an interactive report showing the installed software, software version and software customization on each server. Created a system that nightly captures organizational data from the corporate electronic directory, populates an organizational data structure so DCIX reports can show rollups of server data by manager or senior manager or director. This data appears on the Boeing executive scorecard. Created reusable Perl modules to accomplish common tasks, such as sorting any HTML table by column by clicking on the column header. Created a portable authentication scheme so that scripts and applications can access account and password information from a securely encrypted database.

**06/2003 – 04/2010**

**Client: The Boeing Company**

**Location: Bellevue, WA, USA**

**Role: Senior Server Tools Developer supporting The Boeing Company**

**Duties and Responsibilities:** Performed the same role as a Boeing employee as I currently perform as a CSC employee. Supported the same extensive and complex enterprise reporting system briefly described above. Designed, coded tested, deployed and supported a variety of automated server tools and integrated them into the corporate infrastructure. Supported DCIX, ANT, AWLM, WebSR, SRMS and other Boeing-written tools. Subject matter expert for UNIX shell scripting, Perl, HTML, CSS, JavaScript.

**Accomplishments:** Created interfaces to several corporate systems and databases. Reverse engineered and enhanced an extensive and complex Sox historical reporting tool (32,000+ lines of Perl code) to customer specs with no documentation, written by a contractor no longer with the company. Took a simple prototype created by a co-worker and transformed it into an automated workload manager tool to scan the corporate ticketing system (BARS) for new tickets, compared the information in the ticket against rules customized for each customer, and automatically dispatched the ticket to the correct group or individual to work. Participated in development of a user-friendly end-user interface to BARS (SRMS) which was later chosen to be the standard tool over competing commercial products and internally written products.

**04/1997 – 06/2003**

**Client: The Boeing Company**

**Location: Bellevue, WA, USA**

**Role: UNIX System Administrator**

**Duties and Responsibilities:** Member of a two-person system administration team on call 7 x 24 x 365, managing 250+ Sun servers supporting various internal Boeing business functions. Responsible for OS and application software installation, configuration, upgrades and patches.

**Accomplishments:** Automated many system administration tasks using UNIX shell scripting and Perl. Taught myself Perl and began using it in daily operations. Wrote a script to remotely login to all 250+ servers to create a new account. Wrote backup scripts as well as many other scripts. Was asked to join a team assembled to create an interface between a vendor’s commercial product (which provided dynamic web content on Palm Pilot and Compaq PDA’s) and BARS, allowing users to download and view BARS tickets on their PDA, as well as allowing them to update the ticket and upload the changes back to BARS. As the only Perl team member, I did all the design and coding of the interface to BARS. After the project was completed, I used the BARS interface I had created and wrote ANT (Automated Notification Tool) to monitor BARS for incoming tickets and provide enhanced rules-driven notification and escalation to service providers that they have a ticket in their queue. The system provided much greater visibility of forgotten tickets and measurably reduced lost workstation hours as service providers responded to tickets sooner and missed fewer service level agreement commitments thanks to the three levels of notification escalation provided by ANT.

**06/1989 – 04/1997**

**Client: The Boeing Company**

**Location: Bellevue, WA, USA**

**Role: UNIX System Administrator**

**Duties and Responsibilities:** Managed multiple Apollo token rings of high-end engineering workstations supporting Propulsion, Noise, Weights, Maintenance Tech Pub and Large Airplane Engineers. Responsible for all OS, application software and hardware on the ring. Supported both BSD and Sys 5 UNIX. Managed complex Interleaf document publishing software. Isolated and repaired breaks in the token ring by reconfiguring the data closets. Performed routine hardware maintenance by swapping out scavenged spare parts. Reconfigured the token ring as needed to support new workstations, moves and workstations leaving the ring. Performed all OS upgrades and patches as well as all software installs patches and upgrades. Assisted engineering users as needed.

**Accomplishments:** Automated many system administration tasks using UNIX shell scripting. Trained new Apollo system administrators. Wrote a computer-based training system compiler in shell scripts which allowed me to write training material utilizing simple keywords and content to display on the screen. The student would run the script, which would display the training material for each topic on the screen and ask the student questions. The student would respond and the script would compare the student’s answer with the expected answer, providing the student feedback. The student could actually enter OS commands from the script, allowing the student to see the results of the command entered. The script kept track of correct/incorrect student responses and provided the totals at the end of each training session. Wrote a unique complete end-to-end OS load script which formatted the hard drive, loaded the OS, installed patches, loaded all software, created the necessary user accounts and restored their data from the most recent backup. Several reboots were required at various stages of the procedure, and the script automatically performed the reboot and resumed running when the machine came back up, allowing the entire procedure to run uninterrupted even across reboots. Reduced flatten and reload time from 6-8 hours of dedicated manual work to around 45 minutes of unattended run time which I could use for other purposes.

**01/1987 – 06/1989**

**Client: The Boeing Company**

**Location: Everett, WA, USA**

**Role: System Administrator PDP 11/70 Mini-Computer**

**Duties and Responsibilities:** Supported Noise, Weights and Propulsion CAD/CAM engineers. Manage system configuration, software and hardware. Manage Gould liquid toner plotter. Keep system running. Call vendor support in event of hardware failure.

**Accomplishments:** Read cryptic system manuals and taught myself how to use system macros. Wrote a macro to print tape labels for nightly system backups written to tape. Taught myself Fortran and wrote a Fortran application which provided a simple help system to assist new engineers in learning the cryptic system commands required to get their job done. Wrote various macros to automate various aspects of managing the system. Purchased an allen wrench which I kept on site which saved 30-40 minutes of downtime for all system users whenever the system would freeze or go down (which it did rather frequently). When the system had a glitch, it was unusable until the maintenance vendor was called and would come in, use an allen wrench to open the back panel on the computer, press a reset button, close the panel and reboot the system. It took at least 30-40 minutes for the vendor to arrive. After I purchased the allen wrench, I performed those functions myself, saving many valuable engineering hours.

**09/1978 – 12/1987**

**Client: Various**

**Location: Various**

**Role: Computer Operator**

**Duties and Responsibilities:** Operated IBM 360/30, DataPoint, IBM 4331 and Prime computer systems. Operated peripherals such as tape drives and card reader. Loaded punched cards into card hopper required for each batch job. Re-punched damaged cards and restarted job in event of a card jam. Ran all batch jobs according to schedule. Staged jobs so computer began the next job immediately after the previous job finished. Did not allow computer to sit idle in between jobs. Loaded appropriate paper stock in printer appropriate for the job being run (check stock, report stock, etc.)

**Accomplishments:** Most memorable accomplishment came while employed with Vacation Internationale. Taught myself the Databus programming language. Came up with an idea for a Databus program to completely automate my job. Began work on it and mentioned it to my boss, who liked the idea and had a staff programmer finish it. After debugging and refining the concept, the program did indeed completely automate my job, at which time they no longer needed me and laid me off. First and only time I’ve ever automated myself out of a job, learning a great deal in the process.

* 06/1977-06/1978 Diploma, ITT Peterson School of Business, Seattle, WA, USA
* 09/1976-06/1977 Seattle Pacific University, Seattle, WA, USA
* 09/1974-06/1976 Shoreline Community College, Shoreline, WA, USA

|  |  |  |
| --- | --- | --- |
| **Development Tools**   * **VIm** * **UNIX tools** * **Various debug tools** * **Others** | **Hardware**   * **Sun** * **HP** * **AIX** * **Linux** * **Windows** | **Languages**   * **Perl** * **JavaScript** * **Shell scripting** * **Windows batch file** |
| **Special Technologies**   * **HTML/XHTML** * **CSS** * **Ajax** * **CGI** * **DBI** * **Web design** * **XML** * **SSH** * **SSL** | **Operating Environments**   * **Sun** * **AIX** * **HP-UX** * **Linux** * **Windows** | **Software Packages**   * **Cygwin** * **MinGW** * **OpenSSL** * **MS Office** |