

# John A. Pagakis

Cave Creek, AZ

(602)820-6493    thebfh@toolsmythe.com

## Technical Summary

### **Operating Systems**

Windows 7, Windows Vista, Windows Server 2008, Windows Server 2003, Windows XP, Windows 2000 Server and Professional, Windows NT 4.0, Linux (various), VMWare (Server and ESXi), Novell NetWare 3.x, UNIX (Solaris), Windows 98/95/ME, Windows 3.1, DOS x.x, OS/2 2.x and Warp, MVS

### **Platforms**

Intel, Sun, IBM Mainframe

### **Languages**

C#, JAVA, C++, C, Visual Basic, ASP.NET, Java Server Pages, SQL, XML, XSLT, HTML, Perl, UML, PHP 4.x, VBScript, JavaScript, Bourne Shell Script, REXX, DOS Batch, COBOL/COBOL II, MVS JCL

### **Compilers**

Microsoft Visual Studio .NET (all releases through 2013), Eclipse (Luna), Borland jBuilder (various), GNU C++

### **Libraries and Frameworks**

.NET Framework, jQuery, Microsoft Office OpenXML SDK 2.0, Infragistics Web Controls, VisiFire Graphs and Charts, Struts, C++ Standard Template Library, Active Template Library 3.0, Windows API, MFC, CodeIgniter, Silverlight

### **Portals**

SharePoint 2007 & 2010, DotNetNuke, XOOPS

### **Database**

MS SQL Server 20xx/7.0/6.5, Microsoft Entity Framework 6.0, PostgreSQL 9.x, - 7.x, MySQL (various), Oracle, BTRIEVE, Sybase, Informix, IDMS, IMS/DB, ADO, RDO, DBLib, DTS, DMO, DBI

### **Desktop Tools**

Microsoft Office Suite, SharePoint Designer 2007

### **Miscellaneous**

FrontRange ITSM 6.x, FrontRange IPCM, CommVault, Microsoft Word Content Control Toolkit, MS SQL Server Enterprise Manager, WebRidge, Sheridan Software Calendar Widgets, AutoSys (Solaris). PCQUOTE and Telesphere(Telekurs) Prices APIs, Xcellenet, Sybase ISQL, TCP/IP, Sockets programming, TSO

### **Publications**

"Compensating for Unimplemented Features in PostgreSQL 7.1", Pagakis & Gauthier, 2001

([http://wiki.postgresql.org/wiki/Compensating\\_for\\_Unimplemented\\_Features\\_in\\_PostgreSQL\\_7.1](http://wiki.postgresql.org/wiki/Compensating_for_Unimplemented_Features_in_PostgreSQL_7.1))

Original Author and moderator of the ConfigEncryptor workspace on GotDotNet

(<http://gotdotnet.com/Workspaces/messageboard/home.aspx?id=4b3ed7d5-204a-4afc-aaf9-c778127eee08>)

### **Awards**

CARA Corporation Pride Award - 4<sup>th</sup> Quarter 1998

DevelopOnline.com Hero Of the Month – January 2001

## Professional Experience

### **2/12 – 1/15: Developer – OneNeck IT Services, Inc. (a TDS Company)**

- Wrote a data import facility to pull data from a subsidiary's proprietary Incident Tracking database and add it to our Frontrange ITSM incident tracking database. Written in C#, this multi-threaded command-line program utilizes Microsoft Entity Framework 6.0 for both source and target databases.
- Responsible for maintenance of and enhancement to a C# ASP.NET customer portal back-ended by SQL Server 2010 and eventually 2012. The database was the back end to FrontRange's ITSM Helpdesk Ticket Management package. When OneNeck was purchased by TDS, the enhancements proved instrumental in this portal being selected over those of other TDS acquisitions in becoming the TDS customer portal. These enhancements included:
  - Designed and implemented an Employee Management Facility allowing our customers to self-administer their employee list. Employees can be added, terminated and their profile attributes edited, all within a jQuery Lightbox pup-up.

- Duplicated ITSM's Incident management functionality allowing customers to self-service creation of Incidents, Tasks, Journal Entries, and Attachments. Customers may also resolve tasks and assignments and have multiple views and filters to track existing incidents.
- Designed and implemented an Incident Reporting facility. This is an extremely flexible facility providing the customer many options for filtering and a sophisticated and very intuitive jQuery based interface to choose fields to be included in the report and the order of those fields. Checkboxes are provided for inclusion and the fields (including the checkboxes) can be dragged into the desired order. Data displayed in an on-screen grid that can be exported to Excel format.
- Designed and implemented a Project Management reporting facility. A selected project's Summary, Project Plan and library of Project-related files are displayed in a jQuery tabbed interface. The Project Plan is grouped into phases, each phase segregated into a node within a jQuery Accordion control. The project plan can also be displayed as a Gantt Chart.
- Designed and implemented an Incident Statistics facility providing basic filtering and selection criteria. Data displayed as VisiFire Graphs (mostly Bar and Pie Charts) and Infragistics Grids, within a jQuery tab control.
- Designed and implemented an Incident Survey Reporting facility showing summary and allowing the customer to drill down to detail on the results of their customer satisfaction surveys.
- Duplicated part of FrontRange's IPCM product which tracks and displays phone support statistics. Reverse-engineered a rather complex and sophisticated back-end to produce the same reports as IPCM. Data displayed in VisiFire Graphs and Infragistics Grids.
- Designed and implemented a Customer Asset Management Reporting facility.
- Designed and implemented a Backup Reporting Facility that queries and reports against CommVault's back-end database.
- Designed and implemented a site-wide error-handling and error logging methodology including the creation of a logging class.
- Created a self-scrolling horizontal menu using jQuery when our horizontal menu expanded beyond the available space.
- Introduced jQuery Calendar plugin for all date and date range selection throughout the site.
- Converted all site grids over to use Infragistics' Grid Control.
- Designed an intranet Knowledgebase Search Facility and a data access web service for use by the Helpdesk. Flexible search criteria, all data access handled via a web service, search includes Full Text search on Text fields and results are ranked and ordered via a probability algorithm.
- Designed and implemented various reports that utilize the Microsoft Office OpenXML SDK 2.0 and the Word Content Control Toolkit to produce reports in Word format, some with embedded Excel Charts.

**6/00 – 2/12: Independent Consultant - Toolsmythe Software Services, Inc.**

- **08/02 – 01/15**

- **Arizona Health and Wealth Raffle (<http://www.healthwealthraffle.org/>)**

Original and only remaining member of the team that developed the Arizona Health and Wealth Raffle. The Raffle is run twice a year and netted over \$60 million dollars for the Barrow Neurological Institute and St. Joseph's Hospital and Medical Center.

*As Database Architect:*

- Designed the original schema and all subsequent schema modifications.
- Developed the algorithm for generating a random ticket number to isolate users from any sense of order in their ticket purchases.
- Devised a ticket vending strategy that prevents duplicate sales of a ticket and prevents overselling. Wrote various stored procedures including: the ticket vending sproc, the ticket cancellation sproc, the ticket number generation sproc.
- Incorporated the ability of the site to offer multiple ticket packages (i.e. single tickets and three-packs). This was designed and implemented on two weeks' notice.
- Designed the post-raffle data dump process and wrote the stored procedure supporting this process.
- Created a data warehouse implementing a Star Schema in PostgreSQL. Created a load application that initially took 15 hours to complete and tuned it so that it now completes in under an hour.
- Created a process to create various types of customer data files for email and postal mail marketing that utilizes both data warehouse historical data and current raffle data.
- Developed a SQL algorithm to "normalize" customer-entered postal addresses for matching purposes.

*As Database Administrator:*

- Responsible for generation of schema at the beginning of every raffle and verification that the correct raffle parameters are in place.
- Create the ticket drawing file for each drawing.
- Ad hoc queries to research customer questions and credit card fraud claims.
- Monitor over-all health of the database and ensure that replication is working.
- Run post-raffle data-dump and backups.

*As Developer:*

- Wrote all Data Access Classes.
- Wrote all Data Transport Classes.

- Wrote all Back-office reports.
- Modified the system to allow the sale of multiple ticket packages (as of the second raffle, the Raffle offers single tickets at one price and a limited number of three-packs at a discount). This was designed and implemented with two weeks' notice and would have been a public relations nightmare had the date been missed or the implementation failed.
- Created a transaction editor and ticket re-print facility.
- Wrote the ticket purchase calculator on the Customer Information page.
- Automated the count-down calculator.
- Implemented code to prevent ticket sales after sales cut-off in the event of a non-sellout.
- Created a Windows-based database monitoring application written in C# (2003) and connecting to a PostgreSQL database.
- Designed and implemented a highly secure credit card portal to allow the Raffle's accounting firm to process Credit Card refunds and charges. Each transaction is secured by two single-use passwords, one for the requestor and one for the approver. Also wrote a .NET application to create the password files for the requestor and approver and corresponding SQL files to be loaded by the DBA. These files contain hashes of the passwords, so the DBA has no password access.
- Oversaw upgrade of web server from Tomcat 5.0 to Tomcat 8.0.
- Converted from using Borland jBuilder compiler to Eclipse Luna.

Written in Java compiled with Borland jBuilder/Eclipse Luna using the Struts framework and front-ended with Java Server Pages.

- **02/11: Maricopa County Department of Public Health**  
Developed a strategy to help the Maricopa County Department of Public Health migrate sensitive data from a legacy PostgreSQL 7.0 database to Microsoft SQL Server. Created SQL to generate the migration database in SQL Server; created the required migration dump and created and supplied the development team with an ERD of the existing database.
- **12/09: Apollo Group**  
Created an Executive Dashboard within Apollo's SharePoint 2007 portal. Data was displayed both as list views and graphically using the VisiFire Graphics Library and allowed Apollo to filter on department or view data across all departments. Also created documentation on how to modify the dashboard. Engagement resulted in Apollo Group contracting Ensynch to develop and administer their SharePoint portals.
- **12/04 – 04/05: Arizona State University School of Business**  
Joined an existing team writing an ASP.net application for ASU's School of Business. Wrote a data conversion module to port data from Access databases to the new SQL Server schema. The two schemas were too disparate to effectively use DTS, so the conversion was written in C#.
- **05/05 – 06/05: Maricopa County Senior Aided Independent Living Project**  
Joined an existing team writing an ASP.net application for Maricopa County's Senior Aided Independent Living project. Wrote code to back various database lookup/update/insert pages on the site. Wrote a tool to read a SQL Server database schema and generate C# data access and data transport classes. Written in C# and Microsoft SQL Server 2000.
- **01-06 – 05/06 Honeywell, Inc. – Flight Management Systems**  
Honeywell's Flight Management Systems unit absorbed several departments, each having their own servers, their own home-grown website portal and their own methodology for inter-team communication and work-flow. Conducted interviews with each team to identify intra-team communication problems and requirements for a high-level portal. Performed a comparative analysis of available portals and helped the team select and prototype SharePoint as a solution. Created a Microsoft Virtual Server, installed SharePoint 2003 and Microsoft SQL Server 2000. Created a web part using Visual Studio 2003.
- **Webmaster for JerryRiopelle.com ([www.jerryriopelle.com](http://www.jerryriopelle.com))**  
Coded and deployed site for JerryRiopelle.com. Implemented the PERL-based YABB bulletin board for the sites forums. Applied modifications to the board to customize behavior. Written in HTML and PERL.
- **Akesis, Inc ([www.akesis.com](http://www.akesis.com)).**  
Coded and deployed site for Akesis, Inc. Akesis is a start-up drug company specializing in drugs addressing diabetes. Akesis needed a web presence within a week and that requirement was met. Written in HTML and PHP (for the stock quote facility). All graphics were created by a third party hired by Akesis.
- **01/02 – 8/02: Honeywell, Inc.**  
Designed, wrote and installed a transaction processor framework, a database access .NET web service, and a subsystem for monitoring transaction throughput. The transaction processor is a multi-threaded Windows service. The framework is defined in an MS SQL Server 2000 database, and allows recognition, processing and monitoring of new transaction types without the need to modify code. The monitoring subsystem consists of a website and an

email notification program. The website lists transactions and their status and allows drill-down to detail. The website allows filtering of any combination of Transaction Type, Processing Status and Timeframe. The email notification program is a command line application that looks for errors and issues emails to interested parties (defined in the database). The same program also can run in a summary mode, which will show statistics by Transaction type and Processing type for the past 24 hours. A third mode automatically resubmits transactions that failed due to database connectivity issues. All database access is encapsulated in a .NET web service and all the aforementioned components consume the services provided by that web service. Written using C#, ASP.NET, MS SQL Server 2000, Dynamic SQL in Stored Procedures.

Authored two thread-safe C++ ATL COM classes.

The first, the Authentication Payload Cryptographic Services class, creates an encrypted authentication payload that includes information obtained from an XML configuration file and also provides decryption services. The class uses Microsoft's Crypto API for encryption/decryption services, and Microsoft's XML object. Wrote a generic (compiles both Unicode and ASCII) Base64 encoding class to handle Base64 encoding/decoding of the encrypted payload.

The second class is a low-level SOAP (Simple Object Access Protocol) client that provides a toolkit to web developers wishing to use Honeywell Web Services. Uses Microsoft's XML object both for reading XML and for XSLT translations. Uses the Authentication Payload Cryptographic Services object (see above) to provide encrypted authentication payloads.

Various bug-fixes and enhancements in the Webridge-based ICHome and Corporate Connect web sites.

Miscellaneous ad hoc queries against Oracle databases.

- **7/00 – 7-01: DevelopOnline.com, Inc., Tempe, AZ**

Administration of an object-oriented database (WebRidge), documented existing object model using UML. (RationalRose). WebRidge is an object-oriented middle layer that sits on top of SQL Server 7.0, disguising SQL Server as an OODB.

Identified a bug in the WebRidge database that was causing gradual database corruption (queries were returning rows that did not actually exist). Worked with the vendor to create a SQL Stored Procedure to clean up the corruption. Wrote a batch process to automate the running of this Stored Proc. For this process, wrote a Visual Basic 6.0 app that used SQL Distributed Management Objects (DMO) to script out some triggers that had to be dropped prior to running the Stored Procedure, and re-inserted after completion.

In preparation for a migration from the WebRidge object-oriented database to a relational database (PostgreSQL), performed the following tasks:

- Conducted a product comparison. Based on the new server environment (Debian Linux) and the new Web Server language (PHP), chose PostgreSQL as the new SQL Database.
- Schema design and normalization.
- Designed, built and tested prototype data off-loaders from the WebRidge database (VBScript).
- Designed, built and tested prototype data loaders to the new Schema using Perl and the DBI interface.
- Supervised the creation of the remainder of the off-loaders and loaders.
- Executed and validated the conversion.
- Assisted Quality Assurance in an independent validation of the conversion.
- Wrote two C Stored Procedures to provide services that would have been otherwise difficult to create in PHP.
- Educated the programming staff in the mechanics of using the database in the test environment.
- Provided guidance to the programming staff on complex multi-table joins in SQL.
- Wrote a Perl utility to script out foreign key constraints in the database. The utility writes SQL to add constraints back for tables that will be dropped and redefined.
- Established a change control and migration process during the design phase of the conversion.
- Created and provided timely statistical reports to marketing and the executive team.
- Authored a white paper, "Compensating for Unimplemented Features in PostgreSQL 7.1", Pagakis & Gauthier, 2001 (<http://techdocs.postgresql.org/techdocs/compensating4features.php>). This web site is the official online repository for PostgreSQL technical information.

The conversion from WebRidge to PostgreSQL and PHP took a little more than one month and has resulted in a system that is stable and easy to modify.

**11/97 - 6/00: ACS Technology Solutions (formerly CARA Corporation) - Phoenix, AZ  
Consultant**

- **American Express - Phoenix, AZ**

1/99 - 6/00

Designed and implemented a Visual Basic 6.0 ActiveX server encapsulating End Of Day Processing for the American Express point-of-sale system for Travel Service offices in the United States, United Kingdom, Europe, and Singapore.

Performs all steps required for EOD processing, provides visual feedback and status to the users, logs start and end times and completion status to the MS SQL Server 6.5 database, is automatically step restartable and provides detailed error messaging (including call stack) to a log file.

Conceived, designed and implemented a monitoring program to check the health of servers at offices in the US, UK and Europe. This monitor enumerates drives, indicates the capacity and free space of each, processor type, OS version, latest service pack, and number of days the server has been up. From SQL Server, enumerates the databases on the server and for each indicates the database size and free space, and the size and free space of syslogs (if held in a separate data device). Captures general SQL Server information such as how much memory is allocated to SQL Server, number of Open Objects, etc. Captures errors from the Windows NT System and Application Event Logs. Output is an XML data file that is transmitted back to a central location. These data files are loaded into a star schema, providing Phoenix personnel the ability to identify errors, perform predictive failure analysis, and trend analysis across over 500 offices worldwide. Monitor is written in Visual C++ 6.0 and uses Windows API calls extensively. ODBC was used for SQL Server access.

Designed and implemented a database maintenance manager for offices that operate 24 hours. Detects users logged into the database, asks them to log out for maintenance and locks them out as they comply. Once everyone is locked out, it assembles and executes a docket of scheduled tasks and monitors their completion. When all tasks are complete, it unlocks all users and broadcasts a message that the maintenance cycle is complete and users may log back in. Written in Visual Basic 5.0 and uses RDO for database access. Logs all errors to the Windows NT Application Event Log.

Designed and implemented a Windows NT Service to provide a print spool utility for a system being migrated from Novell to DOS on Windows NT 4.0. Error messaging is done to the Windows Event Log. It was written in Visual C++ 5.0 and uses Windows API calls to perform Windows printing.

Designed and implemented a product installation tool. This tool has been used to roll out upgrades to an in-house developed product that has been deployed at over 250 offices in the UK and Europe. The tool is configurable, and controls upgrades of executables, dlls and SQL Scripts. It also auto-detects workstations attached to the server, propagates itself to those workstations, alters the registry so that the install will run the next time the workstation is logged on, forces the workstation to re-boot and auto login with the correct permissions to run the install. This utility has been used in three upgrades and has saved American Express over \$250,000 over similar installs. Written in Visual Basic 5.0.

#### 2/98 - 12/98

Performed analytical and coding tasks to design, build and implement the Travel Financial Services back-office data consolidation process.

The system was written in C++ (Visual C++ 5.0) with extensive use of the Standard Template Library, and runs under Windows NT 4.0. The Graphical User Interface is written in Visual Basic 5.0. Any classes in the batch portion of the system that could be reused by the GUI were wrapped in Automation Servers built using the Visual C++ 5.0 Active Template Library. This technique allows the C++ class services to be shared with the Visual Basic Applications.

Wrote a batch job scheduler in Perl for Win32 to control and monitor the execution of the batch jobs.

Several programs were ported from the old OS/2-based Consolidation process to Windows NT.

The old system ran in 1 hour 10 minutes, the new system runs in under 10 minutes. This provides the back office staff more data collection time, and greater re-run / recovery flexibility.

Provide support of front-end Visual Basic 5/MS SQL Server data collection front-end. This support includes assisting in rollouts to offices throughout Europe and England, gathering information from office staff to troubleshoot problems, modify MS SQL Server Stored Procedures and co-ordinate mass rollouts to all offices, and troubleshoot and correct errors in the application.

Wrote a utility that that changes the NT login password by generating a seed and passing it through a hash algorithm. The password is then programmatically changed, and the seed is reported to the operator. A second program takes the seed as an argument and reports the password. This utility will be used to grant temporary supervisor access to office staff at remote sites.

#### 11/97 - 2/98

Responsible for tier two support of the Travel Financial Systems back-office operations. This system consolidates office transaction information for the 120+ American Express offices in the United States. Responsibilities include tier two support of the Travel Financial Systems at all US offices, support of the Xcellenet process that transmits data to and from each of the offices and support and operation of the back-office data consolidation process. Platforms involved are Windows NT 4.0, OS/2 2.1 and Novell Netware 3.1. The system is written in C and Turbo Pascal.

Wrote an automated hard drive clean-off utility that determines what files should be archived and what files are simply no longer needed based on file age. The age and file mask parameters are configurable within a configuration file.

This has taken a process that used to take an hour over a modem line and reduced it to 20 minutes. The development of this process makes unattended hard drive clean-off via an Xcellenet-triggered process a possibility for the future. This utility was written in Borland C++ v5.0 and runs on any DOS platform (Novell in this case). Wrote a 32-bit version as well using Visual C++ v5.0 and various Standard Template Library container classes.

**1/83 - 10/97: The Options Clearing Corporation Chicago, IL**  
**Senior Technical Specialist**

- Established the Micro Production Support team. This team is responsible for the support and enhancement of application software developed in-house. Directly responsible for the support of 35 systems. Responsible for hiring and training a staff of five during this time. Supported systems run on Novell NetWare 3.1, Windows NT 4.0, Windows 3.1, Solaris and OS/2, most written in C++ & C using Visual C++ & MFC, Borland, IBM C-Set++ and Sun C++ compilers, controlled by batch files or REXX scripts.
- Pre-install support, testing and debugging of OCC's next generation Clearing System. Project has a Windows NT front-end written in Visual C++ 4.2 & MFC. The backend resides on SUN (Solaris) platform, and utilizes Sybase as the database engine. The system is transaction-driven, and all transactions are conveyed via Orbix+Iris bus technology. Responsible for system control and first line support during shakedown testing of alpha version of system. Debug and correct problems on all involved platforms and technologies.
- Responsible for support and enhancement of the Option Pricing System. Enhanced the system over a three-year period to improve error-recovery and logging. During that interval, reduced the execution time of the Start of Day process by 50% and reduced manual editing by 35%. Also developed and refined a facility that establishes a test environment and captures data files quickly and reliably. System runs in a NOVELL NetWare 3.1 environment, is written in C, utilizes BTRIEVE as its database management system and extracts prices from two vendors - PCQUOTE & Telesphere.
- Provided support for the UNIX-based Theoretical Value Calculation System. During this period, learned UNIX, supported an upgrade from SunOS to Solaris and supported an upgrade in Momentum XIPC software (inter-process communication package). Met production deadlines nightly throughout support period. System runs in a Solaris environment, is written in C and uses Informix as its database management system.
- Developed an application intended to improve the performance of an existing production system by multi-processing transactions against an Informix database. Written in C++ on Solaris and implements the Vector container from the Standard Template Library.
- Responsible for maintenance and upgrades to an in-house developed sockets program. This program reads packets from a Telesphere price feed, parses, reformats and writes the information to a PCQUOTE memory-resident database. Process runs on an OS/2 Warp machine, is written in C and compiled with IBM's C-Set++ compiler.

**Senior Team Leader**

- Senior Team Leader for the rewrite of the Position Accounting System. This one-and-a-half year development effort consisted of 335 programs (over a quarter of a million lines of code), 111 Procs and 183 Jobs. Directed the efforts of five to twelve programmers in addition to programming and analytical duties. Platform was IBM Mainframe, MVS operating system, Computer Associates' IDMS and ADS/O, COBOL II.
- Conceived, designed and wrote a mainframe-based change tracking utility that alerts development teams of changes in production libraries. The system was written in COBOL and is fed by various IBM utilities.

**Team Leader**

- Lead a team of five from high level design through implementation of interfaces for projects converting existing system Master Files and Position Accounting files to IDMS-DB tables. This project served as the base for the rewrite of the Position Accounting system.

**Senior Programmer/Analyst**

- Designed, wrote and implemented the Position Adjustment system, the first CICS application within The Options Clearing Corporation.

**7/80 - 1/83: W.W. Grainger Niles, IL**  
**Programmer/Analyst**

- Participated in Requisition System rewrite. COBOL Master File conversions from VSAM to IMS/DB and extensive CICS development. Involved in a VS/2 to MVS conversion.

**7/79 - 7/80: Bell & Howell Skokie, IL**  
**Programmer**

- Responsible for installation and documentation of MSA financial packages in company branches. Participated from Requirements through Install of an on line distributive system interfacing IBM System 34's to the mainframe. Involved in a VS-1 to VS-2 conversion.

**Education**

**1997 DePaul University Chicago, IL**

Windows 95 Development Using Visual C++ and MFC

- 10 Credit Hours. Grade: 4.0

**1995 DePaul University Chicago, IL**

C++ Program

- 10 Credit Hours. Grade: 4.0

**1991 DePaul University Chicago, IL**

Principles of Computer Science I & II

- 4 Credit Hours. Grade: 4.0

**1990 DePaul University Chicago, IL**

C for Programmers

- 4 Credit Hours. Grade: 4.0

**1979 Control Data Institute Chicago, IL**

Computer Programming & Operations

- 630 Hours
- Graduated with High-honors
- Afternoon class Operations Manager

Designed, wrote and implemented the school's Library Inventory package

**2013 – Microsoft Visual Studio Live**