Andrew M Dunstan

1917 Castleburg Drive Apex, NC 27523-5118
+1.919.414.5634
andrew@dunslane.net

Summary

•A Core Committer on the PostgreSQL project and PostgreSQL Specialist
•An experienced Solutions Architect, Development Manager, and Developer / Consultant who is strongly oriented to helping organizations achieve their IT goals on time and on budget

Recent Experience

October 2015 – present: Pinogy – Senior Database Architect
Deployed sqitch migration tool. Designed and deployed backup and recovery for critical databases. Extended and modified numerous databases in accordance with changing business requirements. Designed new databases for greenfields developments. Mirrored SQLServer into PostgreSQL for back office data analysis. Skills/Technology used: PostgreSQL, C, git, sqitch, Perl, JSON, SQLServer, ODBC, Linux, Windows, AWS.

January 2009 – October 2015: PostgreSQL Experts Inc. – Co-Founder and Senior Consultant.
Board Chairman for several years, previously Treasurer.
Project Description: The company consists of approximately ten highly competent consultants, expert in the whole range of software development, management and troubleshooting, focusing particularly on PostgreSQL based applications.
Skills/Technology used: PostgreSQL, C, git, Perl, XML, JSON, PHP, SQLServer, ODBC, Linux, Windows

April 2007 – October 2016: Dunslane Consulting, LLC – President/Principal Consultant:
Project Description: Started my own consulting business focused on improving usability and function of the legendary PostgreSQL database software. The business has kept a full roster of clients and billing time from the start. Skills/Technology used: PostgreSQL, C, git, Perl, XML, XSLT, FOP, JSON, Redis, PHP, node.js, Tcl, SQLServer, ODBC, Linux, Windows

August 2006 – April 2007: Anchorage Capital Group - Senior Developer / Administrator:
Project Description: maintain, improve and extend a trading system for a hedge fund, interfacing with third party systems and trading partners. Implement sound backup and disaster recovery procedures. Develop new systems as required.
Skills/Technology Used: PostgreSQL, SlonyI, Perl, C, Subversion, Trac, SQLServer, ODBC, XML, Linux, Bacula, rsync, Catalyst

July 2004 – July 2006: HCS Systems / Ringsphere Technologies - Consultant / Senior Developer:
Project Description: renovate, maintain, and produce next generation of Ringsphere, a tool for gathering, replicating and displaying time series metrics, principally network metrics. Also provide consulting services to HCS clients as needed.
Skills/Technology Used: TCP/UDP/ICMP/IP, HTTP, HTML, Apache, SSL, FastCGI, Perl, Template Toolkit, SNMP, nmap, RRDTool, tcpdump, tcptrace, vtun, Subversion, Gforge, Linux, VMWare, PostgreSQL, PHP

June 2003 – June 2004: HCS Systems / North Carolina State Highway Patrol - Consultant / Architect:
Project Description: Design, develop and deploy a pilot program to deliver facial images of persons of interest to law enforcement officers in the field over a low bandwidth network, using XML based on Justice XML Data Dictionary as medium of data exchange, and using only free / open source technology. Skills/Technology Used: TCP/IP, HTTP, HTML, FTP, Tomcat, Java, Ant, Servlets, JSP, JSTL, JAI, Jpeg2000, rsync, ssh, XML, XSL, Xerces, Xalan, JusticeXML, PostgreSQL

July 2000 - April 2003: AnyDevice / HiddenMind - Senior Solutions Architect, Development Director/Manager, Technical Lead, Senior Developer:
Company Description: Software development, tools and platform, both custom and shrink-wrapped, targeted at small, wirelessly/intermittently connected devices (Palm, PocketPC, RIM pager, WML/HDML phones, etc.) Skills/Technology Used: TCP/IP, HTTP, HTML, HDML, WML, WAP, Apache, SSL, FastCGI, Perl, Tomcat, CVS, Linux, Solaris, VMWare, Windows, WindowsCE, ADOCE, PostgreSQL, MySQL, Oracle, MS SQLServer, HSQL, Javascript, Java, Servlets, JSP, Tiles, JSTL, XML, XSL, Xerces, Xalan.

PostgreSQL Project

I first became involved in the PostgreSQL community in 2003. In 2004, in addition to my development work, I created the PostgreSQL Build Farm, a distributed build and test system which has had a profound effect on our development processes. I was also one of the original administrators of pgFoundry, a site for third party community developments. I am still the principal maintainer of the Build Farm software. In 2005, I was made a committer for the core PostgreSQL code. In 2009 I served as a board member of PostgreSQL.US.

Below is a list of my major feature contributions (sometimes in cooperation with others) to the PostgreSQL project since 2003:

Version 10 (expected release Sept 2017)

•Support for enums in btree_gin and btree_gist modules (work in progress)
•Improve maintainability of Postgres catalog code (work in progress)

Version 9.5 (released 2016-01-07)

•Add JSONB functions jsonb_set() and jsonb_pretty
•Add several generator functions for JSONB that exist for JSON
•Allow TEXT, TEXT array, and INTEGER values to be subtracted from JSONB documents
•Add JSONB operator ||
•Add json_strip_nulls() and jsonb_strip_nulls() functions to remove JSON null values from documents
•Add statistics for minimum, maximum, mean, and standard deviation times to pg_stat_statements
•Add psql setting pager_min_lines setting to control pager invocation
•Improve psql line counting used when deciding to invoke the pager
•Add support for INSERT, UPDATE And DELETE to the Redis Foreign Data Wrapper
•Add support for Redis cursors to the Redis Foreign Data Wrapper.

Version 9.4 (released 2014-12-18)

•Add JSONB, a more capable and efficient data type for storing JSON data
•Add new JSON functions to allow for the construction of arbitrarily complex JSON trees
•Render JSON dates in a format compliant with ISO 8601
•Reduce vacuumlo client-side memory usage by using a cursor
•Rewrite duplicate_oids Unix shell script in Perl
•Add make targets check-tests and installcheck-tests, which allow individual tests to be run
•Improve support for VPATH builds of PGXS modules

Version 9.3 (released 2013-09-09)

•major overhaul of JSON parsing
•new JSON generating functions
•new JSON extraction and processing functions
•allow addition of values to an enum if they don't exist
•rewite of pgindent utility in Perl
•allow use of Assert() in client code

Version 9.2 (released 2012-09-10)

•add JSON generating functions to complement new JSON type
•backport new JSON features to release 9.1
•add –-exclude-table-data option to pg_dump
•add –-section option to pg_dump and pg_restore
•preserve column names in row expressions
•improve pretty printing of view definitions
•add environment variable overrides for psql startup and history files
•add \setenv command to psql
•enhanced Redis Foreign Data Wrapper to handle many different types of foreign table mappings

Version 9.1 (released 2011-09-12)
•cleanup and publicly release buildfarm server code
•Allow addition of values to enum types efficiently, greatly expanding their possible use
•Created Foreign Data Wrapper extensions for reading fixed length record files and CSV files with arbitrary numbers of fields per line
•Support MinGW64 compiler on Windows and for cross-compiling for Windows.
•Allow generic record arguments to PL/Perl functions

Version 9.0 (released 2010-09-20)
•revamp buildfarm client to support git instead of CVS
•Verify that PL/Perl return values are valid in the server encoding
•New Makefile targets world, install-world, and installcheck-world,
•Add query text to auto_explain output

Version 8.4 (released 2009-07-01)
•Add suppress_redundant_updates_trigger() trigger function to avoid overhead for non-data-changing updates
•Allow pg_restore to use multiple concurrent connections to do the restore, reducing restore times in some cases by up to 90%.

Version 8.3 (released 2008-02-04)
•Allow server log output in comma-separated value (CSV) format which can easily be loaded into a database table
•Arrays of composite types
•Improve efficiency of LIKE/ILIKE, especially for multi-byte character sets like UTF-8
•Close all known holes by which invalidly encoded data could be stored in tables.
•Allow type-name arguments to PL/Perl spi_prepare(), PL/Python plpy.prepare() and PL/Tcl spi_prepare to be data type aliases in addition to names found in pg_type
•Enable \timing output for \copy
•Enable server core dump generation in pg_regress on supported operating systems
•Add a pg_ctl option to control generation of server core dumps
•Interpret the dbName parameter of PQsetdbLogin() as a conninfo string if it contains an equals sign

Version 8.2 (released 2006-12-05)
•Increase default values for shared_buffers and max_fsm_pages
•Add DROP object IF EXISTS for many object types
•For security's sake, modules used by a PL/PerlU function are no longer available to PL/Perl functions
•Add TG_table_name and TG_table_schema to trigger parameters for PL/PgSQL and table_name and table_schema to trigger parameters for PL/Python
•Make $_TD trigger data a global variable in PL/Perl
•Run PL/Perl and PL/PerlU in separate interpreters, for security reasons

Version 8.1 (released 2005-11-08)
•Properly process carriage returns and line feeds in COPY CSV mode
•Add COPY WITH CSV HEADER to allow a header line as the first line in COPY
•Add a validator function for PL/Perl
•Allow PL/Perl to return a Perl array when the function returns an array type
•Allow Perl nonfatal warnings to generate NOTICE messages
•Allow Perl's strict mode to be enabled
•Add a PGPASSFILE environment variable to specify the password file's filename
•Allow pg_config to be compiled using MSVC - This is required to build DBD::Pg using MSVC
•Allow IPv6 connections to be used on Windows

Version 8.0 (released 2005-01-19)
•Allow the database server to run natively on Windows
•Rewrite initdb, pg_ctl and other shell script utilities completely in C
•Allow COPY to read and write comma-separated-value (CSV) files
•Implement dollar quoting to simplify single-quote usage
•Major PL/Perl overhaul and feature boost
•Allow logging of session disconnections using server configuration log_disconnections

•Add new server configuration parameter log_line_prefix to allow control of information emitted in each log line
•Replace the virtual_host and tcpip_socket parameters with a unified listen_addresses parameter
•Listen on localhost by default, which eliminates the need for the -i postmaster switch in many scenarios

Version 7.4 (released 2003-11-17)

•Allow pg_hba.conf to accept netmasks in CIDR format
•Allow IPv6 server connections

Other

•Devised abstract schema mechanism used by Bugzilla project to support multiple database backends, with first support for PostgreSQL.