Curriculum Vitae

Regan Russell BSc

December 3, 2025

Phone: +61 41 428 7577
Email: regan @ pymblesoftware.com
regan.russell @ gmail.com
https://www.linkedin.com/in/pymblesoftware/

Latest CV: https://www.pymblesoftware.com/cv.pdf

Latest CV: https://www.pymblesoftware.com/cv.pdf

Previous Experience

Contract roles through PymbleSoftware Pty Ltd since 2008 18 year(s), -1 month(s) .
(Personal contracts since 1998) 28 year(s), -1 month(s) .

Latest CV

Recent
Academic projects.
Business projects.
Personal software projects.
Real estate projects.
Working with publishers on technical manuscripts as part of the editorial cycle.

Cross-platform mobile developer with Flutter / PHP Laravel (Lumens) developer (Contract)

Charitable Organisation
Flutter (3.10.6), Flutter Realm (1.3.0), Dart, Stripe (flutter_stripe:9.3.0), credit_card_scanner (1.0.5), PHP, Laravel. Lumens.

GDPR (privacy by design) and PCI DSS (card data handling) compliant as possible. Only static content is kept in an in-memory database, not even in storage.
Over 12,000 lines of Flutter code. Over 800 lines of PHP web service code.
I read somewhere that the average US programmer averages 200 lines of code per day. In the first month, I averaged over 500 lines of code per day. **Greenfield** project from start to finish. Donation app for water and food aid projects. **Cross platform** with **flutter**. Did all the web services with Laravel/Lumens and MySQL database.

Payment gateway integration with **Stripe**.

---

iOS and Android Developer (Native)

Company **Humm Group**
https://humm-group.com
https://apps.apple.com/au/app/humm/id1455391873
https://apps.apple.com/au/developer/humm-pro-pty-ltd/id1532343413
https://apps.apple.com/ca/app/humm-ca/id1586818651

Apps: Bundll, QANTAS Pay, Humm Pro, Humm Canada, Humm Cards and legacy apps which included the following tools and libraries:

AFNetworking[1], FLAnimatedImage[2], Auth0[3], NVActivityIndicatorView[4], Floating Panel, JWTDecode[5], Kingfisher[6], Firebase, Dynatrace, AppFlyerFramework, CardIO[7], EMSMobileSDK, Alamofire[8], Lottie[9], KeychainSwift[10], Branch, Swipe, Validator, SwiftEntryKit[11], SVProgressHUD[12], SwiftLint, Quick, Nimble, IOSSecuritySuite, Snapkit, SwipeCellKit[13], ObjectMapper, SwiftyJSON[14], DateToolsSwift[15], RangeSeekSlider, NVActivityView, IQKeyboardManager, **SalesForce** market-

Latest CV: https://www.pymblesoftware.com/cv.pdf

[1] https://cocoapods.org/pods/AFNetworking
[2] https://cocoapods.org/pods/FLAnimatedimage
[3] https://auth0.com/
[4] https://cocoapods.org/pods/NVActivityIndicatorView
[5] https://cocoapods.org/pods/JWTDecode
[6] https://cocoapods.org/pods/Kingfisher
[7] https://cocoapods.org/pods/CardIO
[8] https://cocoapods.org/pods/Alamofire
[9] https://cocoapods.org/pods/lottie-ios
[10] https://cocoapods.org/pods/KeychainSwift
[11] https://cocoapods.org/pods/SwiftEntryKit
[12] https://cocoapods.org/pods/SVProgressHUD-0.8.1
[13] https://cocoapods.org/pods/SwipeCellKit
[14] https://cocoapods.org/pods/SwiftyJSON
[15] https://cocoapods.org/pods/DateToolsSwift

ing cloud[16], kotlinkit, cardview, androidx, Gson[17], threetenabp[18], brentrielly navigation, picasso[19], retrofit2[20], OkHttp3[21], biometric, rootbeer[22], fillrembedded, viewpagerdots[23], filetree, play services, swipe reveal layout, EML payments[24], RxAndroid/RxJava[25], epoxy, TMX, securebank, swiperefreshlayout[26], liveness

Humm Group white labels credit cards and provides financial services such as small to medium-enterprise equipment leasing. Specifically, Humm provides consumer credit cards / Store cards such as the Farmers[27] chain in New Zealand and the QANTAS Pay card.

Since I have had experience with financial services and credit card processing online in one form or another since working with Plink libraries at BDE in 1997; I was able to quickly adapt to the workflows, codebases/business practices of the credit card services provided by Humm Group to provide bug fixes and minor maintenance on 192342 lines of Swift, 879 lines of Objective C, 334037 lines of Kotlin, 47,982 lines of java, excluding libraries. To develop the changes to these codebases I reviewed Figma diagrams and incorporated changes to UI elements in Android and iOS code as per the requirements of the JIRA tickets.

To modernise the build environment, I added Azure DevOps / Fastlane CI/CD build pipelines for a few of the apps.

(Native) iOS and Android Developer
Company **Milieu Labs**. (Contract)
https://apps.apple.com/au/app/milieu-climate/id1566271872

Latest CV: https://www.pymblesoftware.com/cv.pdf

Consumer thermostat systems.

Pre-existing tech stack: Amazon **IoT** [28], AWS, SwiftFromat, RxSwift[29], RxCocoa[30], ReSwift[31], AWS SDK iOS, Alamofire, Crashlytics, Freddy, Butterknife[32], MQTT[33], RxJava[25], Redux[34], Mockito, Jetpack. AWS Cognito, AWS IoT, Dagger2, OkHttp[21], retrofit2[20]

Milieu Labs provides controls for domestic air conditioning systems. The control unit of the air conditioner is replaced with a module that receives MQTT messages from a server which is indirectly managed through a user's account on the app. The project added a Zone information screen. Zone state change handling which entailed unpacking MQTT Zone state messages, and translating that through a middleware layer onto the screen through the UI controls.

Android Developer
Company **Oracle** (Contract)

Worked on JSON parsing code for an Oracle internal project for external clients.
The project was under strict NDA. There was an intense level of security around the project. The project had

[16] https://developer.salesforce.com/docs/marketing/marketing-cloud/guide/mobile-push-sdk.html
[17] https://github.com/google/gson
[18] https://github.com/JakeWharton/ThreeTenABP
[19] https://github.com/square/picasso
[20] https://github.com/square/retrofit
[21] https://square.github.io/okhttp/
[22] https://github.com/scottyab/rootbeer
[23] https://github.com/afollestad/viewpagerdots
[24] https://www.emlpayments.com/
[25] https://github.com/ReactiveX/RxAndroid
[26] https://developer.android.com/jetpack/androidx/releases/swiperefreshlayout
[27] https://www.farmers.co.nz/
[28] https://aws.amazon.com/iot/
[29] https://github.com/ReactiveX/RxSwift
[30] https://cocoapods.org/pods/RxCocoa
[31] https://cocoapods.org/pods/ReSwift
[32] https://github.com/JakeWharton/butterknife
[33] https://mqtt.org
[34] https://cocoapods.org/pods/Redux

international security implications.

Technologies employed included OkHttp[21], Retrofit[20], bumptech/glide, Appsflyer, Facebook SDK, Firebase, Kakao, Line SDK, SnapChat SDK, Twitter SDK, VK[35] SDK.

Android Developer
Company **Australian Retirement Trust/Sunsuper** (Contract)

https://play.google.com/store/apps/details?id=com.sunsuper.prod

Java, Kotlin (Android)

Minor maintenance work.
Took Figma diagrams and updated UI elements in Android code.

Developer
Company **BP (British Petroleum)** (Contract)
No link: The company's internal app is only available on the corporate portal.

Flutter, Kotlin (Android), AWS CloudWatch, S3, node.js Lambdas, DynamoDB,
Azure DevOps

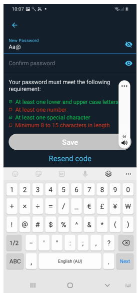
This was a suite of 3 apps that are four internal use by point of sale BP employees.

I introduced the **feature** of local notifications in Kotlin/Android with an Android AlarmManager boot time background service in less than 500 lines of MVVM architected code.
This was so that station employees could be reminded to rotate stock at certain points in the day.
Bug fixing in Kotlin, Flutter and node.js Lambdas.
Meetings with the UK are scheduled at 8 pm. Late-night meetings are scheduled for every night of the week.
Dealing with Teams messages at 12:46 am

Latest CV: https://www.pymblesoftware.com/cv.pdf

Android developer
Company **CoverMore** (Contract)

Kotlin, LiveData, Navigation graph, MVVM.
I **rewrote** the password reset screen. The design was such that the business logic was in a view model and as the user entered values the view model validated the input against the password rules. Publishing/observing, as the input matched the rules the label of each rule turned from red to green.
I introduced the **feature** of data binding into the app. Worked on the common button and some of the reusable widgets.
Android Material design, reusable software components and UI widgets.



New Password
Aa@

Confirm password

Your password must meet the following requirement:

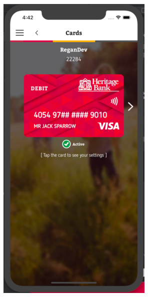
☑ At least one lower and upper case letters
□ At least one number
☑ At least one special character
□ Minimum 8 to 15 characters in length

CoverMoore Password Screen

Company **Nightlife music** (Contract)
https://apps.apple.com/us/app/crowddj/id911666442?ls=1
Java, Objective-C maintenance work and small projects including work on 8-year-old code base in Objective-C on iOS and Java on Android.

[35] VK is the Russian version of Facebook

Company **Heritage Bank** (Contract)
https://play.google.com/store/apps/details?id=au.com.heritage.app .
https://apps.apple.com/au/app/heritage-mobile-banking/id386772598?ls=1 .



Heritage Bank

Originally 100 per cent Objective-C, I **rewrote** scheduled payments screen in Swift adding new sections and **refactored** the design into separate legacy and MVVM modules.
Merged all the complex build steps and frameworks into folders. Reduced the build time from 8 minutes to under one minute.

Accessibility audit for vision impaired. Added **features** for Apple's VoiceOver support.
Provisioning credit cards into Apple Wallet via Apple APIs, encrypting on bank servers with the encryption keys from Apple's API. Leading and mentoring junior developers.
Cocoapod
Fixing the build system.
PassKit

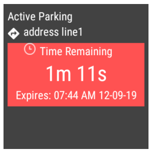
Company **Xinja Bank** (Contract)
https://xinja.com.au/
Android Developer.
Technologies: Kotlin

Maintenance and minor **bug fixing** of Kotlin NeoBank banking app. Koin and View Model, Live Data, Android material design.
There were over 20 onboarding screens in creating an account. I was tasked with adding more.

Technologies employed included androidx, biometric, firebase, Lottie, GSon[17], jodaTime, newRelic, ok2Curl, retrofit[20], zendesk

Latest CV: https://www.pymblesoftware.com/cv.pdf

Client **NSW Government DFSI** (Contract)
iOS and Android developer
Technologies: Swift / Kotlin
https://play.google.com/store/apps/details?id=au.gov.nsw.onegov.parknpay.release
https://apps.apple.com/au/app/parknpay/id1453474761?mt=8



Watch App

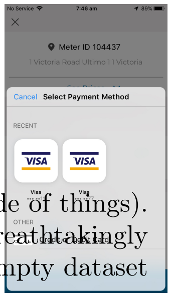
Added feature of Braintree payments integration in Swift and Kotlin in an app for NSW Government including Apple Pay and Google Pay.
The previous payment **integration** had reportedly been problematic and the payment gateway vendor for Braintree was almost effortless.
It took less effort and resources than expected so I added the **features** of Apple Watch and Android Gear Watch integration into the app for the NSW government.
Did a custom notification where the countdown timer and buttons appear in the phone notifications even on the lock screen.

The app communicates with an internal server, which acts as a proxy for another server that manages parking meters.
Part of the credit card processing integration occurred on the parking meter servers and part was within the app.

The idea of the app is to pay for parking meters from the phone and be notified of the need to top up the parking meter if it is about to expire by tapping on the watch and the phone notifications.
Complex integration due to the number of servers involved and the number of moving parts and the level of security.
All the watch app extensions, all the app to payment gateway integration.

Latest CV: https://www.pymblesoftware.com/cv.pdf

Client **Rydo Taxi app** . (Contract)
https://apps.apple.com/au/app/rydo/id1150318596
https://play.google.com/store/apps/details?id=com.Rydotechnologies.Rydo&hl=en
iOS and Android developer.
Technologies: Swift / Kotlin



Payments

Maintenance work and small projects involving C#, Swift, Java, and Kotlin (Mostly Java on the Android side of things). Fixed about a dozen bugs each week on Android then about a dozen on iOS the next week. The code was breathtakingly bad, if the server found no results in the database it would return 404 instead of 200 and an indication of an empty dataset in the JSON payload.
The server would 500 or 400 frequently on requests that worked a minute ago.

Somehow previous developers had considered the foreground and background of the app to be handled in view controllers instead of the app delegate, there was a lot of surreal code.
Whoever wrote it had no idea of the application life cycle or standard industry practices like not abusing HTTP error codes.

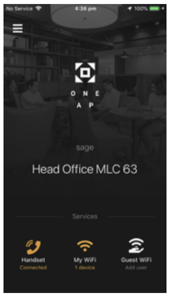
I was doing work on iOS, Android and ASP.NET and completing **features** in two weeks they had waited 2 years to be completed.

Client **ServCorp** (Contracts)
https://www.servcorp.com.au/en/oneap/
Technologies: Swift / Kotlin / Objective-C / Java / PHP / Larvel / CodeIgniter / LaTeX

Comments: Replaced back-end login process with Auth0[3] (an OAuth implementation). **Maintenance** work in PHP and iOS and Android code to replace naive login and session verification.
Did a **localisation** for the Japanese (日本語) version. I have some understanding of Japanese having once worked in (Citrix 2005/2006) and visited the country several times.
Wrote extensive documentation, and fixed bugs. **Refactored** local login to an OAuth login with Auth0[3].

Client: **eDale** / **ServCorp** (Contracts)
Technologies: Swift / Kotlin / Objective-C / Java / Node.js / Heroku / PHP / Larvel / CodeIgniter / LaTeX/ Android

Comments: Several overlapping short-term contracts. Some clean-up of previous work for Edale, bug fixing, and random ad hoc changes. Hand over and manage a developer I hired through my own company to take over work.
Short-term work on Bluetooth tile beacons for finding car keys - type tag beacons. Another USB-based Bluetooth beacon (Sensoro) and QR code project for allocating desktop handset phones and Wifi access. Worked 3 days a week at Servcorp doing maintenance work on an iOS and Android app that allows for provisioning of desktop handsets and (guest) wifi access



Servcorp OneAp

Sensoro beacons, translating Chinese comments into English. **Implemented feature of** Android circular reveal animation like the iOS animation. completed button handlers, differentiated between my wifi and guest wifi buttons and called different web service endpoints accordingly. Scale and slide up login animation to match the iOS version. Made UI consistent with the iOS version. Added disconnect handset screen in Kotlin. Added **feature** of pull to refresh the menu on the Android version to be consistent with the iOS version. Added site features refresh endpoint call and UI update. Added background handset polling. Used to be location dependent. Fixed refresh, device count wrong and other bugs in. both iOS and Android versions. A branch was created in the source code control system called allow_handset_choice to allow the choice of a beacon/handset other than the nearest found beacon. An extra button was added to the found handset screen. Touching this button launches another screen with a list of handsets detected. On both the iOS and Android versions all the beacons that were available in the area are stored in collections in the bowels of the Bluetooth code. In both versions, the collections were propagated up through the software layers until they are presented in a UITableView or a recycler view and the on-item selection handlers then put the selected MAC address in place of the nearest found beacon MAC address and then call the occupy handset code which calls the web service endpoint (Function 16).

Half of the endpoints of the ServCorp server were in CodeIgniter and half were in Larvel in another repository. I set up a local copy of the code bases on my development machine for debugging server-side and client-side issues.

Client: **Edale Holdings Pty Ltd** (Master contract)
The app has been removed from the app store.
Technologies: Android/iOS/Heroku/Amazon S3, Kotlin, retrofit2[20], Swift4, node.js, mongoDB.
Front-end and back-end developer. Team leader
Comments: **Greenfield development** Created a node.js web service with about a dozen endpoints. Created an iOS app that allows users to register, log in and like other users of a dating app. User images on S3 displayed on a dating app. Created an Android version of the same app to talk to the same web service. Full control over everything changing web services to fit the needs of mobile apps and changing mobile apps to deal with web service restrictions. Next to no formal specification. Lots of client hand-holding, random inconsistent changes, About 2700 lines of Kotlin, about 1000 lines of node.js and about 2200 lines of **Swift4** in the first week.
I eventually hired another developer through my own company. **Complete app start to finish**

Client: **Object consulting** (Contract)

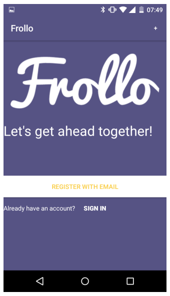
Technologies: Kotlin, Android, Bluetooth, Retrofit2[20]
Android developer
Comments: **Greenfield development**. Created an app that read from an RFID tag "wand". The RFID tags would be assigned as a collar or an animal tag. The list of tags was uploaded as JSON to the RESTful web service. Used retrofit2 for accessing the web service. Used Realm to store the tags. Used some example Bluetooth code to read the wand and send it simple commands.
Complete app start to finish

Bluetooth cattle scanner

Client: **Frollo** (Contract)

https://apps.apple.com/app/id1179563005

Technologies: Kotlin,
Android developer
Comments: **Greenfield development**. Created an app that integrates financial data from all bank accounts from all banks and hits about 60 endpoints on a single web service. Created multi-dimensional build variants so there was product A, production, staging, and develop and the same for product B. Wrote about 21,000 lines of Kotlin and about 800 lines of Java in 2 months. Initially not very idiomatic Kotlin but progressively more stylistic Kotlin. Wrote JUnit tests. Used com.github.PhilJay:MPAndroidChart to present data pulled from
com. square-up.retrofit2:retrofit:2.3.0 into recycler views. Used
com.github.vicpinm:krealmextensions:1.1.5 to store data from the endpoints. Used Picasso to fetch and display bank icons. Used bouncy castle to encrypt data put or posted to endpoints. Used Android Account manager to store account information. Created one-time passwords to send credentials and financial data to the server. Used SSL-pinning. Created code that read data from an end-point and dynamically created controls from it almost like a web browser. Used an Airbnb library on GitHub to do deep linking to handle frollo:// and https://m.frollo.us
Complete app start to finish The entire app had already between done on the other platform and all decisions had been made so it was a matter of making like for like.



Frollo

Client: **Seven Studio at the Easter Show (2 weeks), AMP Banking App** (Multiple Contracts)
Technologies: Android, iOS (iPad), AlamoFire, ffmpeg, Swift 3, AWS, S3., Carthage, mongoose.js, node.js, MongoDB, Kotlin,
iOS/Node.js, MongoDB, Android developer.

Comments:
AMP Android banking app - Kotlin, RxJava, Dagger dependency injection. A mess of deeply nested Kotlin templates. “Clean design” – model, view, presenter, use case, contracts.

7Studio, an App that records videos, displays a teleprompter. The recorded video gets an overlay graphic applied and then merged with a news break intro and news break closing theme. RecordedUI XC unit test. Memory leak debugging with instruments. Run at the Easter show for kids to be a reporter for a day.

Client: **Accenture/Foxtel** (Contract)
https://play.google.com/store/apps/details?id=com.foxtel.epg&hl=en_AU&gl=US
Technologies: Android, Web Services
Android developer
Comments: **Maintenance work** on 100,000 plus line of code program guide / billing-account management app with hundreds of thousands of users. Read crash reports, and patched code with defensive programming techniques like checking if things are null before attempting to use them. Fairly basic maintenance work. The code at times was textbook on what not to do.

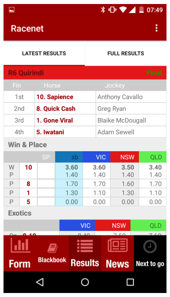
Latest CV: https://www.pymblesoftware.com/cv.pdf

Latest CV: https://www.pymblesoftware.com/cv.pdf

Client: **Invocare** (Contract)
No link: Internal Corporate portal app
Technologies: iOS. Objective-C (pre-dating ARC), iOS, Web Services
iOS developer, Architect. Team leader.
Comments: Maintenance work on an iPad app used to sell funeral services. **Salesforce**, Magic Record, SDWebImage, Masonary, Cocopods. There were about 100 cases in JIRA. I go through, pick a case, read a chain of emails going for 6 months, put in a fix for 5 lines of code in 20 minutes, stick it on a branch in bitbucket. I end up creating a dozen or so branches, all different fairly trivial fixes, never code reviewed, never tested. I’m told they’re doing a hybrid app now (Cordova, JavaScript) and my role is now “Application Architect”. I have to look at issues like the flow of data app to SalesForce to Epicor is truncating data and organise workshops with end users, and I digging around in T-SQL on SQL Server and APEX classes in SalesForce triggers and I end up churning out about 1000 pages of documentation and writing 50 to 100 lines of code in total. I hired a handful of developers. Lots of meetings, gathering requirements with end users.
Resolution of issues that had been outstanding for over a year

---

Client: **RaceNet** (Contract)
No link: The app was replaced after several years.
Technologies: Android and iOS. Java and Swift.
iOS, Android, Web Services developer.
Comments: **Greenfield development.** For the Android version I wrote about 20,000 lines of code in the first month. About 5,000 lines a week. GSon, Android studio. Got the basic version of the Android app completed. Worked with Web services developer on defining data models and endpoints. Used about 30 web service endpoints. I used Cocoapods and SwiftyJSON. Wrote the iOS version in Swift. I ended up churning out about 30,000 lines of Android code in Java and about 20,000 lines of code in Swift.
Complete app start to finish



R6 Quirindi Final

Fin	Horse	Jockey
1st	10. Sapience	Anthony Cavallo
2nd	8. Quick Cash	Greg Ryan
3rd	1. Gone Viral	Blaike McDougall
4th	5. Iwatani	Adam Sewell

Win & Place

		SP	sb	VIC	NSW	QLD
W P	10		3.60 1.40	3.60 1.40	3.50 1.40	3.40 1.40
P	8		1.70	1.70	1.60	1.70
P	1		1.30	1.10	1.30	1.10
P	5		0.00	0.00	0.00	0.00

Exotics

	VIC	NSW	QLD

RaceNet

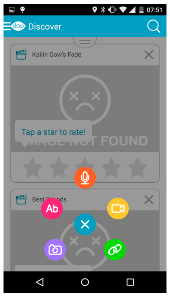
Client: **AiiMS** (Contract)

Technologies: iOS 8, CoreData, Xcode 6, XCTest unit testing framework.
Team leader iOS, Android, Web Services.
Comments: **Mentoring** Taught PHP developer Objective-C, the use of libraries like Cocopods and Cococontrols.com. Taught use of AFNetworking[1] and JSONModel. Taught C# .NET developer Android, Gradle, Maven, use of libraries and

search.maven.org. Taught both the use of git. Evaluation of web service frameworks such as node.js, and Ruby on Rails before settling back on C# .NET. **Technical leadership** on various things such as selecting CMS - bespoke vs Joomla vs others.

Client: **Macquarrie University finance department, TriBeeCam** (Multiple Contracts)
Technologies: Perl on Windows, NAB data source, Excel Spreadsheets and VS Script. iOS 8, CoreData, Xcode 6, XCTest unit testing framework.
Freelance project contractor.
Comments: Did maintenance work for a university that brings me back every couple of years.
Did AFNetworking[1] to JSONModel wrapper for a camera app. UIKit, everything is done in code, no storyboards, nib files, or anything. Git, JSONModel, JSON.

Client: **DoppelTime**. (Contract)
Link: https://apps.apple.com/gb/app/dopel/id1434777360
https://play.google.com/store/apps/details?id=com.dopel.dopel
Technologies: iOS 7, CoreData, Xcode 5, Unit testing framework.
iOS developer.
Comments: Completed an iOS app, added social elements, camera roll picker, camera control, voice recorder, bug fixing and finished off half-completed code. The company is a start-up in a start-up incubator. High pressure for quick results. Used Cocopods, AFNetworking[1], UIKit, GCD, Flurry Analytics, GoogleMaps, Core Graphics, added a bunch of UIViews to a core animation layer, did some cute explode-out pseudo-button animations and added gesture recognizers to the UIViews. Core Location, AVFoundation for Camera & Audio record/playback. Some node.js debugging of the services the iOS app used. Set up node.js for local testing of mobile code.



DoppelTime

Client: **SmartBill** (Contract)
No Link: Removed - App Store policies
Technologies: Android, Sqlite3, iOS 7, CoreData, Xcode 5s Unit testing framework.
iOS/Android developer.
Comments: **Greenfield development**. Developed an Android and iOS app to gather data usage and call log data and send it to a server for 'smart bill' analysis against phone plans. Used silent push notifications to wake the app up for processing. AFNetworking[1], working on ASP.NET code for Apple push notification servers. SQL Lite.
Complete app start to finish

Client: **Industrea/GE Mining** (Contract)
Technologies: ARM Embedded Linux, Sqlite3, rtrees, sockets.
C++ developer.
Comments: Added **features**. Developed a GPS Fence Daemon for a Collision Avoidance System (CAS). Point in Polygon, pulling fence data from web services. About 9000 lines of code in 3 months (about 100 – 200 lines of code a day). SQL Lite.

Client: **Kordia** (Contract)
No link: The app was available through the corporate portal only.
Technologies: JSONKit, YAJL JSON parser, Telerik C# controls, Visual Studio 2010, SQL Server 2010, Team Foundation Server. ASP.NET web services.
iOS Developer.
Comments: **Maintenance** work on KST, Kordia's iPod app that communicates with a web service pulling down JSON data

Latest CV: https://www.pymblesoftware.com/cv.pdf

Latest CV: https://www.pymblesoftware.com/cv.pdf

for Telco site planning, such as Telstra, Downer and NBN. Converted JSON requests to background SAX style streaming requests updating a UITableView as large projects with lots of assets were taking a long time to update with no indication to the user of any activity. Employed code blocks, ARC, Multi-threading and other more recent or advanced iOS techniques. A little C#/XAML but mostly Telerik controls.

Client: **ShuffleMaster (Star Games) (Poker machines)**
Technologies: OpenGL, Qt, C++, Linux and embedded Linux. SDL, Fedora Core 4, OpenSUSE 11.4
Linux C/C++ developer.
Comments: Worked on Roulette, Baccarat, and Sicbo, player terminals and dealer terminals. Bug fixing assisted with the development of the concurrent projects, implemented a couple of cute animations and some tab drawings, tab switching functions.

Client: **Mercurien** (Contract)
Technologies: OpenCV, Hadoop[36], ZooKeeper[37], Cloud Computing technologies, C++, JSON, REST, AJAX, Eclipse, Java, Ant. Maven, Cisco 3400E, Netgear FVS 336. Bamboo, Maven, Ant. Ruby
IT Specialist. (network administration, programming, cabling, etc).
Comments: Evaluated cloud computing technologies. Creating VPNs, production and internal Development/Test subnets. Building Java and C++ ANPR [38] software. Shell scripting in bash, creating Java build plans in Bamboo. Configuration and release management. OS-X Server administration. SVN management. Installing cameras in car parks, porting OpenCV (C++ computer vision) to OpenIndiana[39] (Open Solaris). Some OS-X Objective-C coding. Debugging and fixing some pre-existing Ruby code.

Latest CV: https://www.pymblesoftware.com/cv.pdf

Client: **Samsung Electronics Australia (SEAU)** (Contract) – Mobile development and wrote a mobile development book for a publisher
Technologies: C++, bada, JSON, REST, AJAX, OpenGL ES, Widgets, iPhone, Android. StarUML, PowerVR, Internet@TV Widget SDK. Eclipse. Flex/Flash Lite for mobile.
Bada (Mobile) Specialist.
Comments: Developer support specialist, helping people port applications from Android and iPhone. Digging through iOS, Android and Widget code. Site visits to companies like Blue Pebble (Essendon Football Club) and Fairfax Digital. Assisted with development of the MyCareer app, sole responsibility for the **Greenfield development** of Domain app working on-site at Fairfax offices working directly with Fairfax as Samsung's clients. Helping developers port from iOS to Bada, digging through iOS and bada code.
Greenfield development Wrote the entire Essendon Football Club App within one week, Shows match fixtures, with scores if played, from JSON data as logo –v- logo, Downloads thumbnail images from URL in JSON data, and shows images in news items. Player profiles, injury lists, scores, football club shop, live chat, statistics, etc. All are downloaded live from the official site. YouTube of app running on phone: http://www.youtube.com/watch?v=LvmnGqPC6Gw

---

Client: **Open Systems Consulting**
Technologies: RedHat Linux, Oracle, SCO UNIX Apache.
C/PERL Analyst/Programmer for iPaq mobile devices and infrastructure.
Comments: General UNIX **administration** activities, scripting, adapting cron jobs, setting up IMAP servers, Apache configuration, buying SSL certificates, SCO UNIX and Linux, maintaining very very old legacy C code called "carry" and "directbook". Creating WSDL specifications for SOAP::Lite interface and code for a system that accepts SOAP requests and transfers the XML data to whichever state it is destined for. Some PERL. 3 days a week, consulting work. Gathering requirements directly from managing directors of transport companies and implementing changes or making bug fixes and direct deployments to live systems.

[36] https://hadoop.apache.org/
[37] https://zookeeper.apache.org/
[38] (Automatic Number Plate Recognition)
[39] https://www.openindiana.org/

Company: **Open Systems Consulting**
Technologies: RedHat Linux, Oracle,
Analyst/Programmer for iPaq mobile devices and infrastructure.
Comments: **Maintenance** work covering for a staff member on leave. Data migration wrapped a SOAP layer around an XML-RPC-like application for interfacing with the SAP PI SOAP interface. Miscellaneous Apache/Linux fixing.

Company: **Telstra Bigpond**
Technologies: Solaris, C/C++, OpenLDAP
Analyst/Programmer.
Comments: Bug fixing and documentation of some LDAP [40] and RADIUS[41] code on the system that handles the leases on the Telstra Bigpond cable modems and interfaces into the billing system.

Latest CV: https://www.pymblesoftware.com/cv.pdf

Company: **Department of Innovation and Industry Research,**
National Measurement Institute.
Technologies: Linux, C++
Analyst/Programmer
. Comments: **Greenfield development** Linux daemon to certify the synchronisation of the Network Time Protocol with the atomic clocks for all of Australia. Provision of official time for all of Australia: Industry, Government, etc. Got to know the inner workings of the NTP protocol well. Stuffed data into the extensions fields of NTP packets. Wrote a network sniffer daemon program that sniffed the network for NTP packets and extracted and logged the extension fields. On completion of the project did a quick port of the ntpd to Windows, taking less than a week.

Company: **Infoplex**.
Technologies: Linux, mod_perl, CVS, **Blackberry mobile development** JDE, kSoap
PERL Analyst/Programmer.
Comments: **Maintenance** on Linux mod_perl code some C# ASP.NET, Visual Studio 2005, Visual Studio 2008, Java Server Pages, SOAP calls from Blackberry, ocr-xtr, AutoVue jVue, Apache configuration, AJAX, wrote an 877-page document explaining the current system. Reconfigure hylafax, postfix, created virtual machines, some Crystal Reports, SQL server 2008, RFC 2445. Lots of Postgres and CPAN and Perl DBI. Some IIS.

Company: **Saint George Bank Treasury Core Systems Development.**
Technologies: Solaris, Sybase, C++, PERL, Eclipse, Java, KSH scripting, CVS.
C++ and PERL Analyst/Programmer.
Comments: **Maintenance** on (www.misys.com) Risk Vision add-ons and (www.demica.com) Citadel extensions in C++, PERL, Java and KSH. Developed a patch in C++ for (Sybase, New Era Of Networks) NEON 2038 (32 bit) date problem [42]. Moved some systems from crontabs to AppWorks, and a lot of very small patches in PERL and Korn shell scripts.

Company: **Keycorp**
Technologies: Apache, CodeCharge, PHP, C++, RedHat
Enterprise Linux 4, XML, pThreads, sockets, Postgres 7.4.5. (Database) **Analyst/Programmer. (Manager of other**

[40] https://enwikipedia.org/wiki/Lightweight_Directory_Access_Protocol
[41] https://en.wikipedia.org/wiki/RADIUS
[42] https://en.wikipedia.org/wiki/Year_2038_problem

developers through my own company)
Comments: Digging for missing source code. Reconstructing missing and broken software, untangling mess, and editing image files with a hex editor because the image specification was a poorly worded paragraph in a long email chain that had not been understood. Hiring and firing programmers. Dealing with specifications that were still changing months after delivery had taken place. Moving columns back and forth between tables to match specs that changed daily. Fixing other programmers' code. Small amounts of Python and PERL in the XML2Db loader. Shell scripting and a lot of system administration of a few RedHat Enterprise Linux 4, servers running as virtual machines under VMWare ESX server. I hired two developers through my own company and managed their work including specification, verification and delivery of their work. I managed another notoriously difficult developer. Working on multiple projects with multiple project managers and allocating time.

Company **Macquarie Bank. Quantitative Applications Division.**
Technologies: Solaris, XP Pro64, Sybase, Orbix (CORBA[43]) Java and C++ sides of client server. Reuters SFC/SSL.
Quantitative Analyst/Programmer.

Comments: Imputation credits to Indextool Java/C++ CORBA-based client/server application and analysis of Sybase database. Some Reuters SFC/SSL code.

Company **VeCommerce.**
Technologies: Access, Visual C++

C++ Callflow Developer.
Comments: Built telephony application that handles entry of credit-card numbers and activation of SIM cards. This gave me the skills to build my DTMF-based phone system.

Company **SpamMATTERS.**
Technologies: SQL Server, Visual Studio, SysInternals tools, Ethereal. PostgreSQL, FreeTDS.

Team Leader of 5 C++ Developers

Comments: CGI-BIN/PERL scripting. **Maintenance** of web site. **Greenfield development**. Implemented a system that accepts mail into 20 accounts pulls the recipient field out registers them into PostgreSQL table posts values into a web page and parses the result page. Done in 3 days without prior knowledge of PostgreSQL.

Latest CV: https://www.pymblesoftware.com/cv.pdf

Company **Citrix.**
Technologies: Citrix Presentation Server 4.0, Web interface. Metaframe for Solaris, SQL server, Xinerama, Oracle, Gnome, WinDbg, ASP, Visual Studio, SysInternals tools, Ethereal.

Lead Escalation Engineer

Comments: Read kernel crash dumps and Dr Watson dumps with WinDbg. Debugging device drivers. Code investigation concerning trace logs. Dealt with customer issues. Business trips to Japan and Hong Kong.

Company **Optus / NCS – Part of SingTel.**
Technologies: C++ (aCC/cxx), Tru64 4.0D, AIX,
ORACLE 9.2, Mac OS9, AppleScript, VisualAge C++. Tuxedo8.1, Visual C#, HP-UX 11.0, Solaris, Mac OS9. Weblogic

[43] https://corba.org/

8.1, XML, ant, Enterprise Java Beans (EJBs)

C++/C#/Tuxedo Team lead/Programmer/ J2EE Programmer

Comments: **Porting/remediation** project replacing SII middleware with Tuxedo.
Boris remediation project. It is a 3-tier client/server application. Apple Mac client in OOPL communicates via OpenUI to the COGS middleware. The COGS talks (via SII replaced by Tuxedo) to the Boris server which contains Pro*C code to talk to the Oracle (7.3.4 replaced by 9.2) RDBMS. The SII section I was responsible for was the SIDL which is like the CORBA [44] IDL and is kept in a repository which is like the Windows registry. I wrote code to load SIDL to emulate the repository. The original target platform was HP and later moved to AIX. I did a partial port to Linux to do work on my laptop.
Testing of the interface to another system that communicates via ORACLE database pipes. Took a reworked C# client that communicated via OpenUI to the SII Boris and got it to call the Tuxedo Boris from the last two contracts. Led a team of three developers. Training developers, administration, project planning, architecting solution, etc, etc.
SNMP Support for the previous application. Wrote a server that polls a shared memory segment and dumps content to a log file to be retrieved by CA Unicenter Log Agent 3.0 and sent to the SNMP port. Also wrote a debug test harness that forces exceptions to be thrown for the “catch and send SNMP trap” code. Prototype for EJB interface for the previous application. Configuration of a WebLogic8.1 and Tuxedo8.1 server on Windows XP. WebLogic to Tuxedo (WTC) code. Java server pages to call the EJB. Ant scripts in XML to compile and deploy the EJB in the WLS. A C++ test program to call the test target service in “BORIS” as a prototype for the JSP/EJB prototype. WebLogic and Tuxedo domain configuration, on Linux, Windows XP and AIX. Documentation and support for previous projects. Maintenance work on the C++ and OpenUI OPL source code on the Macintosh client. Wrote C++ Tuxedo test harness for SIBEL interface.

Company **National Bank of New Zealand, Wellington, New Zealand.**
Technologies: C++, Solaris, Windows. Sybase, Sunsoft C++, Cytrix, Borland C++ Builder, Java, PERL/Tk, Tools++, DBTools++, GreenLeaf Comm++, SNA [45], LU-62, Systematics OWL, Paradox engine 3.0, Comms++, Protoview Datatable, Seagate Crystal reports, InstallShield, Borland C++ 4.52, 5.0, C++ Builder 5.0. Microsoft Word, Excel.
C++ Programmer.

Comments: I was one of two programmers responsible for “Direct Link” **maintenance**. Transaction processing in excess of $5 billion daily. Code for the $ 2.5 billion problem, numerous reports. The system contained several components including:

Client-side: monolithic 16bit application capable of running on Windows 3.1, an X.25 network interface, a 16-bit to 32-bit thunking layer [46] for connection to the server via Secure Socket Layer (SSL), CREEP protocol a modified form of DES encryption for a secure connection to the server. Maintenance on several attempts that had been made to port the OWL/Paradox-based 16bit application to 32 bits using various products including Borland C++ and C++ Builder and libraries/tools like DBTools++, etc.
Server-side: a modified form of DES (CREEP), an interface to other internal applications in the bank including updates to FOREX (FOReign EXchange) rate boards in the branches, connections to a Screen Scraper for communication to the IBM 3090 MVS mainframes via LU6.2 bridges. Access to Sybase DBMS and flat files. Access to DEC VMS systems, TCP/IP to SNA bridges, etc.
Disaster recovery systems. Communications via multiple networks to SSL authenticators, etc.
Sentinel a suite of packages in “XView” (XWindows UNIX dialogues) for the “Direct Link” Call centre to monitor transaction processing and accept/reject transaction batches and administrate client accounts. Call centre reports in Crystal Reports.

Company **Compaq** (On site at ADC broadband), Brisbane, Australia
Technologies: C++, Tru64. GNU CC. Tuxedo.
C++ Programmer.

Comments: **Maintenance** on-site development at ADC broadband. This was a conversion project for ADC on behalf of Compaq as a result of a request from one of ADCs’ Pacific clients. I ported code from Sun/SGI/AIX/HP-UX/Win32 to include conditional compilation for the Compaq (Now HP) version of UNIX (Tru-64). The code base is several millions

[44] https://corba.org/
[45] https://bitsavers.org/pdf/ibm/sna/GA27-3102-0_SNA_General_Information_Jan75.pdf
[46] https://learn.microsoft.com/en-us/windows-hardware/drivers/kernel/why-thunking-is-necessary

of lines of code for a telecommunication billing system. This included spotting known issues and fixing them. It required compiling on the new platform, rerunning unit tests and fixing compile errors and unit test failure bugs.

Company **Printrak**, Brisbane, Australia
Technologies: Visual C++ 6.0, MFC.
C++ Programmer.

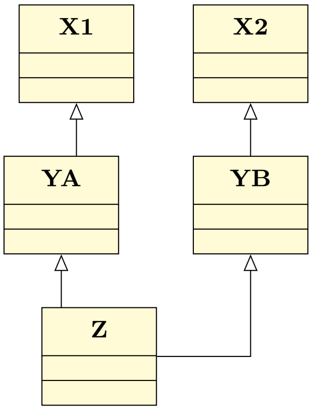
Comments: Emergency service response dispatch software, Fire, Ambulance, Police. 911 Call centre operations. Added additional dialogues for accessing Microsoft SQL Server 7.0 DBMS in C++/MFC/ODBC. Mostly this was to fetch some rows from a table and update the controls in the dialogue type code. Some critical systems accessed systems in TADEM/NON-STOP KERNEL subsystem.
Technologies: MFC, Microsoft SQL server 7.0. Tandem COBOL.

Latest CV: https://www.pymblesoftware.com/cv.pdf

Company **Active Sky**, Gold Coast, Queensland, Australia
Technologies: **Palm Pilot, Windows CE mobile development**, Solaris, Linux
C++ Programmer.

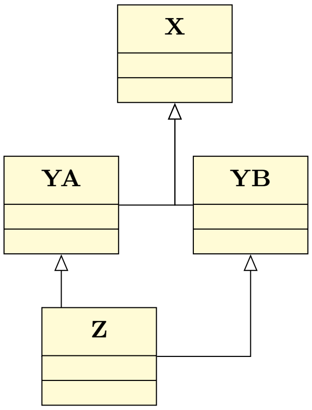
Comments: Video compression streaming to hand-helds. I did some architectural work regarding common file I/O libraries for both client and server-side communication and **mentored** some of the developers. I trained the Windows NT system administrators to configure and manage Solaris and Linux servers and deal with the programmer requests properly which they were not doing.

Company **Open Telecommunications**, Sydney, Australia
Technologies: Solaris and Tru64, SunSoft C++, GNATTS, PERL, expect, CORBA[47]. TAO, Orbix, ACE, pThreads, Rational Rose, UML / BOOCH Rmakeit, Nedit, Emacs.
C++ Programmer.



Comments: **Maintenance work**. This was a telecommunication company that mostly built digital switches (Signal Control Processors - SCPs). I scanned the bug list in the GNATTS database, resolved the bugs and submitted progress updates. One of the bugs in the systems was a multithreaded construction/destruction bug which was related to multiple inheritance and the C++ diamond problem as can be inferred from the following code snippet and opposing UML diagrams:-

```
class Y: public X {}; instead of class Y: virtual public X { }; and class Z: public YA, YB {};
```



Once the bug list was reduced, I migrated the code from the Orbix, ORB to the ACEs TAO ORB because the company wanted to use more open-source software and not pay for commercial licenses. I reran the unit tests for all relevant parts of the system and worked with other team members to resolve any issues. I rewrote the logging code which provided streams ("<<" and ">>" operators were overloaded) and URL style logging methods (such as "file:", "socket:", etc). We had internal seminars on SS7, Voice Over IP, etc. The main customer was One.Tel which was a spectacular "dot.bomb" failure. Open Telecommunications no longer exists. The documentation method was UML using Rational Rose, and Source code control was in CVS. Unit tests were in scripting languages such as expect, awk or PERL.

Latest CV: https://www.pymblesoftware.com/cv.pdf

[47] https://corba.org/

Company: **Thompson-CFS**, Dee Why, Sydney, Australia
Technologies: IRIX, DomainOS, Tru64 UNIX, Solaris, WinCenter, VME, VIMIC, MIL-STD-1553 [48] , ARINC-429 [49], MIL-STD498, Ada, C, DOORS, Interleaf.
C Programmer.

Comments: Defense training organization. I developed software to simulate and stimulate the MIL-STD-1553 and AR-Inc 429 buses. The environment was MIL-STD 498 documentation process. The buses interacted with the rest of the environment via VME boards.

---

after hours, part-time, concurrent with SMA below)
Company: **Transport Management Group**, CBD, Sydney, Australia.
Technologies: Windows/MFC, ORACLE, Windows NT, Visual C++ 6.0, QVCS
C++ Programmer.
Comments: Train scheduling. I produced graphical reports in MFC/C++, (eg zig-zag graphs that show when trains are scheduled to arrive/depart at points up and down the line). The environment was ORACLE Pro*C which was wrapped within smart pointers which loaded an internal cache, pre-fetching and lazy-evaluating as required. The development environment was initially extremely chaotic which I resolved to structure. I introduced, set up and maintained, QVCS as no source code control system was used and QVCS (a free/cheap product) was used at SMA where I was working concurrently (together from 6:30 am to 9 pm every day and sleeping through the weekends).

Latest CV: https://www.pymblesoftware.com/cv.pdf

Company: **Scientific Management Associates**, Lane Cove, Sydney, Australia
Technologies: Windows 98, Windows NT, Windows 98, Visual C++ 6.0, DirectX 6.0, QVCS, VME [50], MIL-STD-1397 [51], 3D Studio Max.
C++/3D Game Engine Programmer for Defence Project.

Comments: This was originally a 6-month contract which was extended to 11 months to coincide with the completion of the project. This was an extremely challenging project which required examining a real piece of equipment (EOSS) a system much like a periscope on a submarine and developing a design to simulate it. A director head with DLTV and Thermal imager sits about three-quarters the way up the mast of the Huon class mine hunters. On the bridge of the ship is a console that was simulated. I had to put 3 video cards into one computer and get all the device drivers to work together. Then I had to get hardware-accelerated Direct-3D to function on two video cards, and load textures and vertexes into each card, while the other video card displayed a menu system, I wrote that mimicked the controls on the real bridge. Some video cards would detect that they were not the primary display device and switch to software rendering. Other video cards would steal vertex lists or not load textures. The thermal imager had “White hot” and “Black Hot” modes and therefore two sets of textures had to be loaded for each object and flipped between them in the scene as the controls were accessed. The daylight TV camera had an intensity control on the touch-sensitive control panel and therefore I had to walk the vertex list on the video card and adjust the lighting intensity of each vertex in the scene. The glow and dim effect was quite spectacular and was quite cool to play with. Additional functionality included socket code to interact with an Instructors station (which someone else wrote). There was also a Digi I/O board added to the machine which enabled digital/analogue conversion of signals. The project was completed 2 two weeks ahead of schedule and I spent some time profiling and optimising it as much as possible. The Navy was happy to sign off on the project.
I tendered a project to replace the MADS (disk packs) on the submarine project. The small tender (roughly $100,000) was successful but the project ($100 million plus) was suspended indefinitely (pending a Royal Commission). If the tender was not revoked I would have had additional continuing contracts with SMA.
This was about the same time that I had the BeOS C++ Ray Tracer article published in Doctor Dobbs Journal. [52]

---

[48] https://en.wikipedia.org/wiki/MIL-STD-1553

[49] https://web.archive.org/web/20111029161330/http://www.holtic.com/category/352-arinc-429.aspx

[50] https://en.wikipedia.org/wiki/VMEbus

[51] http://www.interfacebus.com/Design_Connector_NTDS_Bus.html

[52] Russell R., (Nov 1999) "BeRays: A ray tracer for BeOS", Doctor Dobbs Journal. https://drdobbs.com/tools/the-berays-ray-tracer/184411102

Company: **TowerTechnology**, Lane Cove, Sydney, Australia
Technologies: Solaris, HP-UX, AIX, Digital Unix, Windows NT 4.0/5.0 beta, Windows 98. **UNIX/Windows NT, SCSI Device driver developer.**

Comments: TowerTechnology develops document and image processing systems and workflow solutions. I was responsible for maintaining the device driver code for the medium changers (large mag-optical disk libraries which contain crypts for disks and several drives). I became an expert on the SCSI bus protocol. I wrote a class factory pattern-based diagnostic tool. The class factory would generate objects of all medium changers that the company supported and dump all kinds of diagnostic information.
I completed a device driver for Sun Solaris to access the medium changers via a pass-through SCSI device driver, debugged multithreaded kernel panics on the Tru64 platform and debugged faults on the HP-UX, AIX, Solaris, Tru64, and Windows NT drivers.
One of the faults included reworking some of the drivers when the disk capacity increased from 2.6Gb to 5Gb, 32-bit limits were exceeded and block orientated seeks had to be replaced with ioctl()s on the devices. Some faults required some functionality to be moved from the upper layer of the kernel to the lower-level drivers or vice versa
Technologies: Purify, Clear Case, SunSoft Visual Workshop, GNU C++. SCSI-View (SCSI Analyser hardware). RCS. Rational Rose, Paradigm Plus, UML. Lotus Notes. Windows NT Kernel debugger (i386kd.exe) and crash dumps. PA-RISC, PowerPC, Intel and SPARC machine code and assembler.

Company: **Brilliant Digital Entertainment**, Double Bay, Sydney, Australia
Technologies: Windows NT 4.0 / Windows95 / Linux, ISAPI, Windows Registry, MFC, Visual C++ 5, COM, Automation, InstallShield 3/5, IntraBuilder, Borland C++ Builder, Delphi 3, Microsoft SQL server 6.5, Java, JavaScript, Perl, inline 80x86 assembler, WinSock32, Plink., SAMBA, PKware.
C++ Programmer.

Figure 1: CyberSwine game

Comments: BDE is a small dynamic games software house. I was involved in various aspects of real-time interactive movies. I took over the installer. I did all of the Unix work and wrote code to interact with other parts of the system including the ticket server. I also did the credit card validation code via Plink.

Latest CV: https://www.pymblesoftware.com/cv.pdf

Company: **Scientia Systems**, North Sydney, Australia.
Technologies: AIX / SunOS / Solaris / SCO / Windows NT 4.0 / Windows95. Visual C++ 5.0, Borland C++ Builder, C-ISAM, SAGA-C, ISDN, Win Gate, SMIT, Humming Bird Exceed XDK, Motif, SAMBA, Microsoft TCP/IP, POP3.
C/C++ Programmer.

Comments: Scientia is a software house that produces a scheduler used by manufacturing called “Synchro”. The main output is a Gantt chart with the capacity for drag and drop and running under the Motif system.
I had previously worked for the company in 1988 when it was known as Scientia-Whitehorse. The system was Accounting (invoicing, accounts receivable, payroll) and manufacturing (Just-In-Time (JIT) and MRP-II).

Company: **EyeOn Software**, Crows Nest, Sydney, Australia
Technologies: Windows NT 3.51/4.0 Intel/Alpha. Visual C++ 4.x RISC Visual C++ 5.0 Intel, MFC, MCI, Install Shield, OLE, Windows registry, ISDN, Notes, TCP/IP.
C++ Computer Graphics Programmer

.

Comments: The product "Digital Fusion" was an Object Oriented, multi-processor optimised, multi-threaded spline-based, resolution-independent video compositing system. My role was to design and implement features file format loaders and savers for the majority of graphic file formats (two dozen variants like JPEG, Sun Raster, PNG, TIFF, Gif, etc). I wrote Windows Registry code and various graphic processing code including Sobel and La Placian, edge detection, and blur filters. I wrote MFC/GUI code for custom controls like a “rubbery” range control (like 2 slider controls in the same control, which stretched and contracted at limits), screw control with infinite wrap-around looping behaviour. The entire GUI was based

on ray-traced images and was extremely slick. Amiga style "intuition layer" framework so that existing components would benefit from underlying extensions. DLLs could be dropped in so that at load time the system would recognise and register new components. Aspect-orientated/delegate style development. System administration of Lotus Notes, Novell Netware, network, and CISCO router. Build a system with Install Shield.
For some months all of the staff went to meetings with angel investors, and business partners overseas and attend SIGGRAPH conferences. I was left alone in charge of the company, answering phones, making sales, sourcing suppliers, deciding markups, arranging conferences, transferring money, paying bills, and my wages.

Organisation: **Department of Computer Science**. James Cook University, Townsville, Australia
Technologies: HTML, CGI, PERL, PASCAL, Novell 3.11, OSF/1, Digital UNIX. ULTRIX, SOLARIS, IRIX, C, PVM, MPI [53]. Various supercomputers including SGI, Cray. Processor farms.
Tutor. Semester-long Contract. Research Assistant/Programmer. Contract. Various supercomputers.

Comments: Taught students PASCAL, data structures. Did some WWW development. Wrote a 25-page Literature review on distributed data structures. Wrote a Scalable 3D torus distributed termination simulation on multiple networked workstations representing processing elements or nodes of the torus in PVM. The simulation is an accurate model of a distributed termination algorithm for the Cray T3D massively parallel processor[54]. Wrote a pseudo device driver for DEC Alpha under Linux. Modified Linux system to run OSF/1 binaries.
Dr B. Mans developed a distributed priority queue out of work from his PhD. thesis and work in Scotland. The Message Passing Interface (MPI) library was unavailable for any of the departments' equipment, so I was asked to rewrite the best part of a large project to use the Parallel Virtual Machine (PVM) library on a mixture of workstation virtual machine groups and supercomputers. The project was delivered ahead of schedule and very few modifications were requested.

Company: **Agire**. Townsville, Australia
Technologies: SCO UNIX/XENIX, SPARC Solaris.
Salesman/Technical support.

Comments: Having previously worked on small projects in Informix, XENIX and C for AGIRE. I was asked to join as front office sales and handle local technical support while most of the team travelled.

Organisation: **Department of Psychology**, James Cook University. Townsville, Australia.
Technologies: DOS, Windows, Turbo PASCAL, Borland C++.
Research Assistant/Programmer. Contract.

Comments: Stereopsis is the post-processing of images by the front of the brain giving the 3D effect found by squinting at the images in MAGIC EYE books. The Psychology department was interested in different effects, shapes and reaction times. The initial main focus was to develop different tests for subjects using in-house developed libraries. METAcode for Windows, a real-time multi-pass event logger, that produced graphs of statistics of events, was designed by Dr Ryan and myself. [55] [56]

Helicopter Pilot, Outback Australia. Consulting as a University student.
Part-time Real Estate agent. Various student jobs.
(AGIRE, BTC, FNQEB, MCD Consulting, etc) Short-term contracts. Technologies: Informix, Pick, C-ISAM, Zinc, C, DOS,

Latest CV: https://www.pymblesoftware.com/cv.pdf

[53] https://www.mpi-forum.org/
[54] https://en.wikipedia.org/wiki/Cray_T3D
[55] Russell, R., & Ryan C (1994) "METAcoder for windows: real-time and multi-pass event logging and analysis in the social and behavioural sciences." Psychology Teaching Review.
[56] Russell, R., & Ryan C. (1994) "METAcoder for windows: real-time and multi-pass event logging and analysis in the social and behavioural sciences." Psychology Software News.

UNIX, BASIC, 8051, 8052. Windows, ORACLE Pro*C, METAwindows, AS/400, DECSystem10, VMS, CP/M. MP/M, PC-MOS, XENIX
Comments: While studying and flying I worked on small projects for various companies and organisations.

Company: **Scientia WhiteHorse**. Crows Nest, Sydney Australia
Technologies: NCR UNIX (Tower 32, Tower XP), XENIX, DOS, C, SAGA. VMS, DIBOL
Title: C Programmer.

Comments: Scientia White Horse was a software development company producing accounting and manufacturing systems. I assisted in the development of the Dental front office system. The dental front office system had additional features such as a history-sensitive teeth-charting system. From existing designs I implemented accounts payable account reconciliation, accounts receivable, invoicing, and general ledger postings. The shipping container system was a very specialised stock control system. Shipping containers each have there own ID with a check-digit and may reside in yards for years or get written off such as getting lost at sea.

Latest CV: https://www.pymblesoftware.com/cv.pdf

Latest CV: https://www.pymblesoftware.com/cv.pdf

Summary

≥ 30 iOS apps in the app store, several apps on other app stores, wrote a book on mobile development (iBooks), consulted on app development (e.g. AFL, Newspapers). Broad spectrum of expertise: UNIX, Windows, Mainframe, mobile and Embedded, Middleware (WebLogic, Tuxedo, CORBA, SII, sockets client/server, SOAP, RESTful WS) Various languages Objective-C, C/C++,C#, Java, PERL, PHP, Scripting, Cocoa, Swift, UIKit, XML, SQLite, Facebook/Twitter SDK integrations, REST with AFnetworking, MapKit, Quartz2D, CoreAnimation, CoreData, Magic Record and Mogentator, Multi-threading and GCD, XCTest, native C/C++ code, Interface Builder, HTML5, JS, JQuery Mobile, AJAX, PHP, Magento, Node.js, Neo4j. Experience in professional software development since 1986. Team leader of 3 (twice) and team leader of 5, management and mentoring skills. Agile, Scrum, MIL-STD-498 and MIL-STD-1267A. Published on several app stores/marketplaces, including current iOS apps:
https://itunes.apple.com/au/artist/pymble-software-pty-ltd/id553990081
Android apps on Google Play:
https://play.google.com/store/apps/developer?id=PymbleSoftware+Pty+Ltd&hl=en
C# / XAML Windows Phone 8 Apps:
http://www.windowsphone.com/en-US/store/publishers?publisherId=PYMBLE%2BSOFTWARE%2BPTY%2BLTD.

Education

Web3
Certificate Blockchain for Developers: Hyperledger Fabric on Azure **LinkedIn Learning**
Certificate Learning Bitcoin and Other Cryptocurrencies **LinkedIn Learning**
Certificate Blockchain: Learning Solidity **LinkedIn Learning**
Certificate Blockchain: Beyond the Basics **LinkedIn Learning**
Certificate Top 10 Skills for Blockchain Developers **LinkedIn Learning**
Certificate Build an Ethereum Smart Contract with Go and Solidity **LinkedIn Learning**

Certificate Ethical Hacking: Cryptography **LinkedIn Learning**
Certificate ISO 27001:2013-Compliant Cybersecurity: Getting Started **LinkedIn Learning**
Certificate Secure Coding in Python **LinkedIn Learning**
Certificate React: Securing Applications **LinkedIn Learning**

Publications

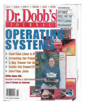
Russell R., (2024) “Thirty-five years in Software Development”, Kindle from
https://www.amazon.com.au/dp/B0DK4FHMZ4 .

Russell R., (2024) “Thirty-five years in Software Development”, Paperback from https://www.amazon.com/Thirty-five-years-Software-Development-Leadership/dp/B0DP7CR7SV .
ISBN 979-8301444326

Russell R., (2024) “Thirty-five years in Software Development”, Hardcover from https://www.amazon.com/Thirty-five-years-Software-Development-Leadership/dp/B0DPDLWPRC/ .
ISBN 979-8301885365

Russell R., (2012) “Programming bada”, Kindle, iBooks and PDF file from www.pymblesoftware.com/book .
www.pymblesoftware.com/book/bada-short.pdf .
http://itunes.apple.com/au/book/programming-bada/id543013439?mt=11&ls=1
https://www.amazon.com/Programming-bada-Regan-Russell-ebook/dp/B007LFX608

Russell R., (Nov 1999) "BeRays: A ray tracer for BeOS", Doctor Dobbs Journal.
Russell, R., & Ryan C (1994) “METAcoder for windows: real-time and multi-pass. event logging and analysis in the social and behavioural sciences.” Psychology Teaching Review.
Russell, R., & Ryan C. (1994) “METAcoder for windows: real-time and multi-pass. event logging and analysis in the social and behavioural. sciences.” Psychology Software News.



Doc Dobbs Journal

Review of Windows NT Device Driver Development Doctor Dobbs electronic review of computer books (ERCB).
Review of The Windows NT Device Driver Book: A Guide for Programmers ERCB.
Review of Developing Windows NT Device Drivers ERCB.
Review of Writing a UNIX Device Driver, Second Edition. ERCB.
Review of Panic: Unix crash dump analysis, ERCB.
Review of Advanced animation and rendering techniques, ERCB
Review of Windows TCP/IP. ERCB
Review of Open source development with CVS. ERCB
Review of System performance tuning. ERCB
Review of Learning the vi editor. ERCB
Review of Ada for experienced programmers. ERCB.

BeRays Article

Publications Officier, James Cook University SCUBA Dive Club 1992 - 1993.

Interests

Tennis, Tango, Reading Non-fiction like ’Stolen Focus’, Harvard Business Review, Podcasts like ’The Knowledge Project’, ’The Finn (Review)’, ’Sengoku daimyo’, and ’NTI’s Japan Real Estate’, General Psychology and Economics podcasts, Travel, Outdoors, Playing Ice Hockey, Playing Squash, Snow Skiing, Hiking, Aviation, Sailing, Surfing, Computer Graphics and Parallel and distributed processing and Computer Architecture (especially SIMD, MIMD), hypercubes, CUDA, MPI, PVM, FPGAs, Antique super computers and Mainframes, UNIX kernel internals. Japanese, and French. Playing with Raspberry Pis, Arduinos, Robotics, and managing 6 server racks, routers and various subnets around the house. PADI qualified diver.

Latest CV: https://www.pymblesoftware.com/cv.pdf

Other information

Previous Defence Clearance, Heavy vechile license (National), Civilian private helicopter pilots license (CAA ref #425348).

Board memberships, committes

- Executive committee Ocean Beach Malibu Club 2016 - 2018
- Treasurer, Strata Plan SP11881
- Director, Macquarie University School of Business Consulting club.
- Director, Three Australian companies.

Latest CV

Book: 35 Years in Software Development

Book: Prograqmming bada

Book a meeting with me

Latest CV: https://www.pymblesoftware.com/cv.pdf

Latest CV: https://www.pymblesoftware.com/cv.pdf