

# AUUGN

The Journal of AUUG Inc.

Volume 18 • Number 2  
May 1997

**AUUG '97**  
4 months and  
counting!

The long lost '96 Index

Java Goes Orange:  
live from AUUGWet'97

A new column:  
Solaris Musings

Inc.

Plus Book Reviews, Chapter News  
and more AUUG Exec Bios!

UNIX & OPEN SYSTEMS USERS



# AUUGN

The Journal of AUUG Inc.

Volume 18 • Number 2

May 1997

---

## AUUG Membership and General Correspondence

The AUUG Secretary  
PO Box 366  
Kensington NSW 2033

Tel: 02 9361 5994  
Fax: 02 9332 4066  
Toll Free: 1800 625 655  
Internet: [auug@auug.org.au](mailto:auug@auug.org.au)

---

## AUUG Executive

**President:**  
Michael Paddon  
[Michael.Paddon@auug.org.au](mailto:Michael.Paddon@auug.org.au)  
Australian Business Access  
723 Swanson Street  
Carlton VIC 3053

**Vice President:**  
Lucy Chubb  
[Lucy.Chubb@auug.org.au](mailto:Lucy.Chubb@auug.org.au)  
Softway Pty. Ltd.  
79 Myrtle St  
Chippendale NSW 2008

**Secretary:**  
David Purdue  
[David.Purdue@auug.org.au](mailto:David.Purdue@auug.org.au)  
Sunsoft Pacific  
Level 4, Sunsoft Building  
44 Hampden Road  
Artarmon NSW 2064

**Treasurer:**  
Stephen Boucher  
[Stephen.Boucher@auug.org.au](mailto:Stephen.Boucher@auug.org.au)  
MTIA  
509 St. Kilda Road  
Melbourne VIC 3004

**Committee Members:**  
Malcolm Caldwell  
[Malcolm.Caldwell@auug.org.au](mailto:Malcolm.Caldwell@auug.org.au)  
Northern Territory University  
Casuarina Campus  
Darwin NT 0909

Alan Cowie  
[Alan.Cowie@auug.org.au](mailto:Alan.Cowie@auug.org.au)  
Northern Territory University  
Casuarina Campus  
Darwin NT 0909

Frank Crawford  
[Frank.Crawford@auug.org.au](mailto:Frank.Crawford@auug.org.au)  
ANSTO  
Private Mail Bag 1  
Menai NSW 2234

Mark White  
[Mark.White@auug.org.au](mailto:Mark.White@auug.org.au)  
Tandem Computers  
143 Coronation Drive  
Milton QLD 4064

Pauline van Winsen  
[Pauline.van.Winsen@auug.org.au](mailto:Pauline.van.Winsen@auug.org.au)  
Uniq Professional Services  
PO Box 70  
Paddington NSW 2021

---

**AUUG Business Manager:**  
Elizabeth Egan  
[busmgr@auug.org.au](mailto:busmgr@auug.org.au)  
Level 4, 90 Mount Street  
North Sydney NSW 2060

---

## Table of Contents

---

Editorial.....	3
President's Column.....	3
AUUG Membership Rates.....	5
Call for Papers.....	5
AusWeb97 .....	7
Java Goes Orange: The Bounds of Trust.....	8
Has NT's Time Come? .....	16
Computer Magic .....	18
The Network is Down .....	19
Confessions of a Road-Trip Tutorial Presenter.....	21
Book Reviews.....	23
Solaris Musings.....	28
Meet the AUUG Exec.....	29
AUUG News: New South Wales .....	31
AUUG News: Queensland .....	32
AUUGN Back Issues.....	33
AUUGN Volume 17 Index Papers (Sorted by title).....	33
AUUGN Volume 17 Index Book Reviews (Sorted by author).....	37
AUUGN Volume 17 Index Reports (Sorted by author).....	38



## Contribution Deadlines for AUUGN in 1997/98

---

Volume 18 • Number 3 • August 1997 :	<b>July 7th 1997</b>
Volume 18 • Number 4 • November 1997 :	<b>October 7th 1997</b>
Volume 19 • Number 1 • February 1998 :	<b>January 7th 1998</b>
Volume 19 • Number 2 • May 1998 :	<b>April 7th 1998</b>

---

### AUUGN Correspondence

Please send all correspondence regarding AUUGN to:

AUUGN Editor  
PO Box 366  
Kensington NSW 2033

Internet: [auugn@auug.org.au](mailto:auugn@auug.org.au)

---

### Submission Guidelines

Submission guidelines for AUUGN contributions are regularly posted on the [aus.org.auug](http://aus.org.auug) news group.

They are also available from the AUUG World Wide Web site at:

<http://www.auug.org.au>

Alternately, send email to the above correspondence address, requesting a copy.

---

### AUUGN Back Issues

A variety of back issues of AUUGN are still available; for price and availability please contact the AUUG Secretariat, or write to:

AUUG Inc.  
Back Issues Department  
PO Box 366  
Kensington NSW 2033  
Australia

---

### Conference Proceedings

A limited number of copies of the Conference Proceedings from previous AUUG Conferences are still available. Contact the AUUG Secretariat for details.

---

### Mailing Lists

Enquiries regarding the purchase of the AUUGN mailing list should be directed to the AUUG Secretariat.

Tel: (02) 9361 5994  
Fax: (02) 9332 4066

During normal business hours.

---

### Disclaimer

Opinions expressed by the authors and reviewers are not necessarily those of AUUG Inc., its Journal, or its editorial committee.

### Copyright Information

Copyright © 1997 AUUG Inc.  
All rights reserved.

AUUGN is the journal of AUUG Inc., an organisation with the aim of promoting knowledge and understanding of Open Systems, including, but not restricted to, the UNIX® operating system, user interfaces, graphics, networking, programming and development environments and related standards.

Copyright without fee is permitted, provided that copies are made without modification, and are not made or distributed for commercial advantage.

---

## Editorial

Günther Feuereisen <gunther@ibm.net>

Here we are again folks! It's been a busy year for everyone, and I apologise for this issue being a little late.

This issue we have some very interesting articles, along with a paper from AUUGWet on Java and Security. A must read for anyone interested in Java.

On a conference note, don't forget AUUG'97 is not far away! If you are thinking of attending, and you are not yet registered, wait no further!

We're working here at AUUGN, new ideas, new columns. This month, "Traps & Tricks" is taking an issue off. I haven't had the time to put anything together, and I haven't heard from any of you out there! I'm sure somebody has had to perform some magic in the last 3 months! C'mon, let me know your secret! ;-)

However, while "Trap & Tricks" takes a holiday, David Purdue has volunteered his services with the inaugural column of "Solaris Musings" - a column directly aimed at Solaris questions.

We're hoping to see more of this kind of column in the future, particularly with the major version of UNIX out there: Digital UNIX, Linux, HP-UX, AIX and IRIX. If anyone is interested in sub-editing any of these, please let me know.

The Volume 17 index has finally appeared - somehow it was misplaced it for the last issue. Thanks to Stephen Prince for all his efforts. There is now an online version of the index - check out it out on the AUUG web page <<http://www.auug.org.au>>.

As always my special thanks to all our regular contributors, without whom we wouldn't have an AUUGN. If you've never contributed, now's the time. Drop us a note at: [auugn@auug.org.au](mailto:auugn@auug.org.au). Whatever your interest, we'd like to hear from you!

That's about it for this issue - see you in August!

❖



---

## President's Column

Michael Paddon <Michael.Paddon@auug.org.au>

Just a day or so ago I ordered another server for my network. This time I needed a machine that could really kick some butt, loads of CPU, disk and i/o. Two years ago, I'd have picked up the phone and spoken to a friendly salesperson at one of the major UNIX vendors. These days, as you'd expect, I ended up buying a fast PC with a SCSI controller, etc.

Naturally, I'll spin up a real operating system on the box (NetBSD; though FreeBSD, OpenBSD and Linux would do just as well). Then I'll load up a bunch of really useful software such as the GNU tools, Apache, nethack :-), tex, xv... you name it, it's out there somewhere.

This is quite probably a familiar scenario. We've all known for years that you don't have to pay for the best software in the world, and we've tended to try and spread the word to those who are ignorant and pity those who don't understand the new dynamics of the real information economy.

"It doesn't cost anything? That just doesn't make sense!", is the common lament of someone outside of our subculture. Some people just aren't equipped to understand that software can be worth more than money; that an ancient technology is being supplanted by new currencies of interaction and exchange.

I tend to be particularly amused when the aforementioned confused person tends to make the next obvious leap, and conclude "free software simply can't be any good, then". After all, the creators aren't asking for any of the paper survival tickets that I value so much in exchange for their work!

One of the true visionaries of our industry (where collecting paper tickets seems to be valued highly, at least in the short term), is Richard Stallman, a founder of the Free Software Foundation. Mr Stallman saw, and more importantly believed, all of this years ago, and his vision has led to the creation of the GNU software suite amongst other things. It's probably impossible to quantify the value of contributions to our industry and hence the world at large, but it he has inarguably made a gigantic impact.

What does this mean to you (apart from being able to get hold of really neat software)? Think about the

long term effects of cheap as dirt hardware and cool free software...

Our industry is no longer going to support vendors. UNIX hardware vendors are already in trouble, and have been for a long time. I've been hearing whispers that another major vendor may be just about to pack it in. But it's only a matter of time until the same fate befalls the operating system vendors. And I didn't use the adjective "UNIX", there.

Most people's initial response to this is "Good, I hope the bastards burn. We'd still be running 7th Edition with ed as our text processor if it hadn't been for Berkeley, anyway". The oncoming troubles of certain other OS vendors is usually discussed with what can only be described as enthusiastic glee.

But in this brave new world where corporations can only afford to focus on applications, what happens to OS development? Who codes, tests, documents and supports the UNIX of the future?

The answer is very simple. And you're sitting with that person right now. It can't happen without your involvement, and the alternative is an unattractive shade of corporate grey. This new economy, new community and new world that has started evolving needs your help, and it needs it now.

If you can code, this means writing something useful and making it available. The GNU and Berkeley licences are good examples of different schemes of "giving" away your work without getting screwed over. Consider joining a development team of one of the numerous free projects, from systems to graphics to databases to word processors.

If you can't code, volunteer to test. Or maybe to write documentation... programmers need all the help they can get in that department. Or even just donate your ideas.

Look at what's been achieved by a small and hardy group of developers to date in the free software world. I doubt is they number more than one developer in one thousand users. Imagine what can be done if we all donate a little of our time similarly, and imagine the richness of the rewards.

This brings me on to a shameless plug for yet another piece of free software. The second version of the Berkeley DB Database is on the cusp of public release. I've been fortunate to get to play with a pre-release of this code and it is truly wonderful stuff.

Many of you will be familiar with the first version, which provided a choice of btree, dynamically hashed or sequential record databases, with a C API. The new

version adds logging, locking, transactions and access for multiple readers and writers, amongst other improvements. In short, a fully functional free database. Check out [www.sleepycat.com](http://www.sleepycat.com) for more information.

My initial work has shown that I can exceed three hundred insert transactions per second on an ordinary PC, which leaves most commercial products looking somewhat anaemic. When someone writes a schema and an SQL layer for this package, I'd start to worry if I were in that business.

I also found the Sleepycat copyright rather interesting. This software is free so long as you distribute, for free, the source of any program which uses it. Not a bad model for promulgating the concept of free code.

No president's column is complete without a quick report on the state of AUUG. Your conference committee, headed by Mark White, has been busy getting the wheels turning for the Winter Conference. We have secured major sponsorship for the event, and are busy liaising with the large number of vendors who wish to be involved in the Exhibition.

George Michaelson and his programme committee already have the call for papers in circulation and we need your submissions. Yes, what you are doing \*is\* interesting and we have members who want to hear about it. An abstract is all that lies between you and fame and fortune. Maybe you can tell us about a piece of free software you are working on?

I've visited the venue in Brisbane several times now, and I can only say that it is brilliant for our event. The networking support, in particular, is going to allow us to attempt new heights in sophistication (and fun toys). AUUG already runs the best network of all the conferences, but this year is going to be special.

The venue is also great for it's space and layout. No long walks between conference and exhibition, a magnificent hall for the conference dinner and superb facilities in the presentation rooms will go a long way to making the event our best yet. There's even a beach nearby on the Brisbane River (what else did you expect?)

I certainly hope you are planning to make it to this year's conference. If you've never been to one before, I can honestly say that it provides a unique mix of the best technical information in the industry combined with an enormously enjoyable social atmosphere.

❖

---

# AUUG Membership Rates

Michael Paddon <Michael.Paddon@auug.org.au>

One of AUUG's major functions is to spend money on you, our member. In fact, our expenditure on AUUGN, conferences, chapter activities, and the like far exceeds simple membership revenues. Contributions from our corporate sponsors, profits from our events and premiums charged to non members for those events all go towards giving you back more than you put in.

That's what a user group is all about.

By aggressively seeking funds outside of our organisation, AUUG has managed for a number of years to keep the cost of membership unchanged. Realistically, of course, this could not continue indefinitely. The cost of printing, venues, postage, catering, advertising, maintaining an office, and indeed all of our activities has only crept upward over the years, and to ignore this fact would be foolhardy. Nevertheless, it was with something less than enthusiasm that your committee concluded that now is the appropriate time for revising dues.

When reviewing the various membership rates, we were mindful of the need to minimise any increase for all of our membership, but especially for our student and individual members, whom we felt were most susceptible to any changes. The new rates are:

Individual	\$100
Student (rate unchanged)	\$25
Institutional	\$390
Additional institutional representatives (over the first 2)	\$80
AUUGN subscription only	\$80

These rates will become effective from (and including) the upcoming renewal period. We will continue, of course, to actively raise monies from outside the membership to subsidise our activities.

We hope that you understand our decision to make these changes and that you approve of them. With your support, we can continue to improve and extend the quality of AUUG's services.

This is an appropriate place to acknowledge major sponsors (in alphabetical order) Digital Equipment Corporation, Sun Microsystems and Tellurian for

their generous support of AUUG. Without their help, AUUG would be a much less effective organisation.

Thank you.



---

## Call for Papers

*AUUG97 Conference  
September 3-5, 1997  
Brisbane Convention & Exhibition Centre,  
Queensland,  
Australia*

### **THEME: "TECHNICAL SOLUTIONS"**

The 1997 AUUG winter conference will be held at the Brisbane Convention & Exhibition Centre, Queensland, Australia, between September 3rd and 5th.

The conference will be preceded by two days of tutorials, on September 1st and 2nd.

The program committee invites proposals for papers and tutorials relating to:

- Technical aspects of UNIX and Open Systems
- Networking, Internet (including the World Wide Web)
- Business Experience and Case Studies

As the theme of this years conference is "Technical Solutions", papers with a strong technical flavour are particularly welcome.

Presentations may be given as tutorials, technical papers, or management studies. Technical papers are designed for those who need in-depth knowledge, whereas management studies present case studies of real-life experiences in the conference's fields of interest.

All presentations must be accompanied by a written paper for the conference proceedings.

Speakers may select one of two presentation formats:

Technical presentation:

a 25 minute talk, with 5 minutes for questions;

Management presentation:

a 20-25 minute talk, with 5-10 minutes for questions (ie a total 30 minutes);

Panel sessions will also be timetabled in the conference and speakers should indicate their willingness to participate, and may like to suggest panel topics.

Tutorials, which may be of either a technical or management orientation, provide a more thorough presentation, of either a half-day or full-day duration.

Representing the largest UNIX and Open Systems event held in Australia (with an average 600 attendees based on the 1995 and 1996 conference attendance) this conference offers an unparalleled opportunity to present your ideas and experiences to an audience with a major influence on the direction of computing in Australia.

September is a very good time for being in the southern hemisphere, and you would be well advised to timetable additional travel within Australia and take the chance to see some more of the country. Brisbane is ideally placed for further travel on the eastern seaboard.

## SUBMISSION GUIDELINES

Those proposing to submit papers should submit an extended abstract (1-3 pages) and a brief biography, and clearly indicate their preferred presentation format.

Those submitting tutorial proposals should submit an outline of the tutorial and a brief biography, and clearly indicate whether the tutorial is of half-day or full-day duration.

## SPEAKER INCENTIVES

Presenters of papers are afforded complimentary conference registration.

Tutorial presenters may select 25% of the profit of their session OR complimentary conference registration. Past experience suggests that a successful tutorial session of either duration can generate a reasonable return to the presenter.

Tutorial presenters who are interested in arranging a follow-on tour of Australia with repeat presentations of their course should indicate this in submitting a proposal. This can only be undertaken by joint arrangement with AUUG, and must follow after the conference to ensure its financial viability.

## IMPORTANT DATES

Abstracts/Proposal Due:

May 15, 1997

Authors notified:

June 4, 1997

Final copy due:

August 1, 1997

Tutorials:

September 1-2, 1997

Conference:

September 3-5, 1997

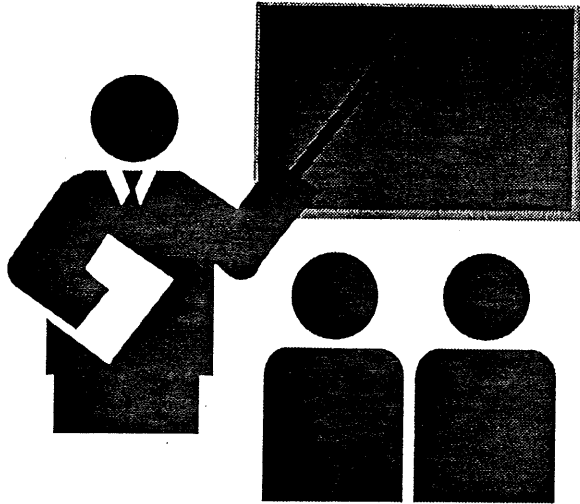
Proposals should be sent to:

**AUUG Inc.  
PO Box 366  
Kensington NSW 2033  
AUSTRALIA**

Email: [auug97@auug.org.au](mailto:auug97@auug.org.au)







---

## AusWeb97

The Third Australian World Wide Web Conference - AusWeb97, is hosted by the AusWeb team at Southern Cross University. It will be held at Conrad Jupiters on Queensland's Gold Coast from 5-9 July 1997.

Full details of the conference, the keynote speakers and workshops, special interest groups and social programs are available at the AusWeb97 web site at:

<http://ausweb.scu.edu.au/>

The conference will have paper presentations and keynote speakers in the four theme areas:

- Technical Futures
- Business Opportunities
- Education and Learning
- Culture, Media and Social

Keynote speakers at AusWeb97 include Robert Cailliau from CERN in Switzerland, who is one of the co-founders of the Web and David Barbagello, Chief Executive of the Distributed Systems Research Centre (DSTC). Robert will be speaking on the Past, Present and Future of the Web. Robert has been with the Web since before there was a Web as we know it .. he is, along with Tim Berners-Lee, uniquely capable of providing a perspective on the amazing growth of the Web and the future of the Web. Robert is the chair of the International WWW Conference Committee (IW3C2) which will be meeting in Australia.

David's team at the DSTC is conducting some of the most important Internet research in Australia. David will present an overarching view of Internet and Web research and development in Australia. Three other keynote speeches, one on each of the other theme areas, are being finalised - details will be posted to the AusWeb site shortly.

All of the usual features of the AusWeb series of conferences will be at AusWeb97 including the ability to interact with presenters and other AusWeb participants in a very intensive and productive fashion .. but equally to have an enjoyable time on the Gold Coast. A full conference exhibition will be associated with AusWeb97. Nine workshops will be featured at AusWeb97 as well as a Developers Day and an Intranet Day on the Wednesday following the conclusion of the conference.

For further information please visit the AusWeb site at <<http://ausweb.scu.edu.au>>, or email the organisers at <[ausweb97@scu.edu.au](mailto:ausweb97@scu.edu.au)> or call Norsesearch Conference Services on Freecall 1800 649 202 or (066 20 3932 (from inside Australia) (066) 20 3932 (from outside Australia) or Fax (066) 221 954.



[ Editor's Note: This is the first of two papers presented at AUUGWet97, the recent NT Chapter conference. Many thanks to Malcolm Caldwell for obtaining this for AUUGN's use. ]

---

# Java Goes Orange: The Bounds of Trust

Thomas O'Daniel <Lozoya@ibm.net>  
Faculty of Information Technology  
Universiti Malaysia Sarawak  
February, 1997

*Transfer of executable content across the Internet has, in the past, been limited to file transfers from definable sources. The advent of the Java programming language has changed the paradigm; executable content can now be transferred to and run on the recipient's machine with little or no user intervention. The system security issues that this change implies have been heavily debated in both the academic world and the business world. This paper is an attempt to evaluate the possibility of using Java applets in a trusted environment. The rationale behind the analysis is based on Sun's efforts to provide tools for creating a secure environment for applets, and their response to the community's efforts to point out weaknesses in the implementations.*

## WHY JAVA?

Standards for evaluating trusted systems are clearly defined and internationally accepted. The difficulty in evaluating Java's security mechanisms lies in the characterisation of Java as a product or as a system [Hsec]. From security point of view, the main difference between systems and products is certainty about the operational environment. A system has a real world environment which can be defined and observed; particular characteristics and requirements of its end-users will be known. A product must be suitable for incorporation in many systems; the precise operational environment is not known to its developer.

The Java language, in its present form, is designed to serve both as a general-purpose programming language, and a way to provide distributed applications across a network. In the first role, the development of 'applications', an analysis of Java from the standpoint of security is no more

meaningful than an analysis of C++ or Perl, for example. The basic safety features of the Java language and Java Virtual Machine -- strict typing, garbage collection, lack of pointer arithmetic, access modifiers, immutable strings, bounds-checked arrays - - make it easier to write "safe" code, and are well explained in the Java Security FAQ [JSFAQ]). These applications maintain a direct relationship with the operating system, and the user trusts the OS to enforce its own security policies.

The case of distributed applications is distinct. They are designed to run in an environment provided by a browser, which incorporates the Java bytecode interpreter. They may originate from one or more external sources, and are executed on the local machine with little or no interference on the part of the user. The prospect of executable code from unknown sources being run on a system with sensitive or confidential information is enough to make some system administrators ban the use of Java applets altogether. The magnitude of the risks of code of unknown origin running uncontrolled on a system was recently demonstrated by members of the Chaos Computer Club, who showed how Microsoft's ActiveX could be used to send unauthorised bank transfers [R18.80].

Recognising the impact of this on their vision that "the network is the computer", Sun has attempted to provide the basic mechanisms for providing a secure applet environment. Web Applets cannot be trusted with the full authority granted to a given user, and so require that Java define and implement a protected subsystem with an appropriate security policy. It is here that it becomes difficult to distinguish between Java as a system -- providing an environment for the applet to run in -- and Java as a product, incorporated into a web browser.

## SOURCES OF STANDARDS

The original document that addressed the explicit specification of standards which could be used to rank the security offered by a computer system is the Trusted Computer System Evaluation Criteria (TCSEC), more commonly known (and hereafter referred to) as the Orange Book. The current version is dated 1985. In the Orange Book, systems are divided into seven security classes (D through A1, lowest to highest) grouped into four divisions. Criteria C1, for example, is suitable for "cooperating users processing data at the same level(s) of sensitivity" [Osec]. Windows NT Workstation has been certified to meet the slightly higher C2 criteria; Silicon Graphics' IRIX provides standard C2 with a B1 (next level higher) option.

The Orange Book has a number of companion publications for official interpretations and guidelines, all with different colour covers, which are sometimes known collectively as the "Rainbow Series". Where the Orange Book focuses on the evaluation of a monolithic data processing system, the "Red Book" (Trusted Network Interpretation) provides perspective on the partitioning of networks into discrete parts for certification, and the "Lavender Book" (Trusted Database Management System Interpretation) does the same for applications.

The Information Technology Security Evaluation Criteria (ITSEC) published in 1991 represents the "Harmonised Criteria" of France, Germany, the Netherlands and the United Kingdom. In ITSEC six assurance levels (E1 through E6) are defined, indicating increasing degree of trust in the Correctness and Effectiveness of the security functions of a computer system. In addition, ten predefined Functionality Classes (F1 through F10) are specified. A rating of F2/E2 corresponds to the Orange Book class C2 [Efaq], and F5/E6 corresponds to A1. E0 corresponds to class D [AK91].

The Canadian Trusted Computer Product Evaluation Criteria (CTCPEC) is the Canadian equivalent of the ITSEC (CTCPEC products rated with a C2 functionality profile and T1 assurance correspond to an Orange Book C2 rating [Efaq]). Both the ITSEC and the CTCPEC are more prescriptive than the Orange Book and its interpretations, which make them more practical for the organisation which wishes to have a product certified, but less colourful reading for those interested in the logic that forms the foundations of the criteria.

The Common Criteria for Information Technology Security Evaluation (Common Criteria) is a multinational effort to write a successor to these documents, and standardise the methodology for evaluation. The sponsors of the Common Criteria project, the European Community (EC), the U.S. National Security Agency (NSA), the U.S. National Institute of Standards and Technology (NIST), and the Canadian Communications Security Establishment (CSE) released an initial version in January of 1996.

The structure and terminology of the Common Criteria are closer to the ITSEC than the Rainbow Series. The Common Criteria include the concept of a "profile" to collect requirements into easily specified and compared sets. This is a significant refinement in the process of cumulative evaluation of systems composed of any number of products. The intent of the project

is to deliver the results to the ISO as a contribution toward an international standard for computer system security evaluation results.

## A GENERAL MODEL

In the case of a system, security objectives can be formed by considering actual security threats, and implementing countermeasures in the form of external and internal constraints. In the case of a product, the developer must give the necessary information for a prospective purchaser to decide whether it will help satisfy his system security objectives, and to define what else must be done for those system security objectives to be fully met.

In general, secure systems will control access to information such that only properly authorised individuals (users), or processes operating on their behalf, will have access to read, write, create, or delete information. A secure system will also attempt to maintain availability of the services it offers, and the integrity of processes while they are executing.

Security functions are those parts of the system that either directly or indirectly enforce the rules that govern access to system resources (security policy). Use of security functions may, in some cases, be negotiated and controlled by the service user with service primitives and their parameters; otherwise they are only visible to the security profile and need not be activated by the user (they are either always active or are activated by the systems management guided by the security policy).

System resources can be structured and used in any number of ways; an entity refers to one grouping of resources. Subjects are active entities that cause operations to be performed on information; objects are passive entities, either the container from which information originates or to which it is stored. In certain cases, (e.g. interprocess communication) a subject may be acted on as an object.

A reference monitor is an abstract machine which enforces the authorised access relationships between subjects and objects of a system. A reference validation mechanism checks each reference to an object by a subject against a list of authorised types of reference for that subject. According to the standards, such a reference validation mechanism must be tamper proof, it must always be invoked, and it must be small enough to be subject to analysis and tests, the completeness of which can be assured [Ccse].

As control is based on authorisation, the identity of the subject must be authenticated before being trusted (allowed access to resources). The degree of trust granted to a subject has been called the security context [AK91]. Because the user has a degree of direct physical control, local applications are considered trustworthy by their owners. However, a user does not trust other users of the same system, nor a foreign system he may be using. Furthermore, systems do not trust one another, unless they take on the role of a specific user.

The life span of a security context can be divided into three consecutive phases: set-up, use and termination. A security context can be set up by means of prior agreement, negotiation, and management (security policy). In the case of connection-oriented communications, a security context can be set-up by negotiation or prior agreement when establishing the connection, and remain in effect throughout the session. If services are not based on an end-to-end connection between the communicating parties, a security context has to be set-up by prior agreement or management, and associated with each message or request-reply pair.

## THE ORANGE BOOK AND THE CC

The model described above is consistent with both the Orange Book and the CC; however, their statements of criteria reflect the difference in their style. The CC specify a specific set of "functional requirements that users can detect by direct interaction....or response to stimulus", laid out in a hierarchy of functional classes. The topmost level of the hierarchy consists of nine classes:

- Security Audit
- Communication
- User Data Protection
- Identification and Authentication
- Privacy
- Protection of the Trusted Security Functions
- Resource Utilisation
- Access
- Trusted Path/Channels

Each of these classes contains families of functions, which in turn contain components that can be used to create a profile of the security policy and its implementation.

The Orange Book, on the other hand, simply states six fundamental requirements:

1. Security Policy: Given identified subjects and objects, there must be a set of rules that are used by the system to determine whether a given subject can be permitted to gain access to a specific object.
2. Marking: It must be possible to mark every object with a label that reliably identifies the object's sensitivity level, and/or the modes of access accorded those subjects who may potentially access the object.
3. Identification: Each access to information must be mediated based on who is accessing the information and what classes of information they are authorised to deal with. This identification and authorisation information must be securely maintained.
4. Accountability: The system must be able to record occurrences of security-relevant events in an audit log. Audit data must be protected from modification and unauthorised destruction. The capability to select the audit events to be recorded is necessary to minimise the expense of auditing and to allow efficient analysis.
5. Assurance: The computer system must contain hardware/software mechanisms that can be independently evaluated to provide sufficient assurance that the system enforces requirements 1 through 4.
6. Continuous Protection: The trusted mechanisms that enforce these basic requirements must be continuously protected against tampering and/or unauthorised changes.

It is perhaps worth noting at this point that the Orange Book has required substantial interpretation!

## THE JAVA SECURITY MODEL

Sun's published Java Security Reference Model (SRM) applies to the JDK 1.0.2, and is designed to be abstract enough to apply to any valid JDK implementation [Jsrn]. The SRM defines constraints on the set of policies that may be implemented within a Java-Enabled application.

("Java-Enabled applications" are the environment that Java applets run in, generally a web browser. "Java applications" are written in the Java programming language, but do not require a browser to run. Initially the SRM makes this distinction, but later ignores it. Here, "Java" will refer to the Java language and runtime system, and "application" will be used in the limited sense of a Java-enabled application designed to run applets.)

Cybec is a specialist in the provision of anti-virus software for the Australian and international domestic and corporate markets. Their proven security products have gained a very high level of penetration of these markets.

## ***INTERNET SYSTEMS DEVELOPER***

### ***• LEADING MELBOURNE TECHNICAL TEAM***

This is an outstanding opportunity to join an excellent team of security software developers. Our anti-virus software has already gained a significant penetration of local and international markets.

We are seeking a top rated designer to lead the development of new security products for the corporate and business sectors. You will be responsible for initial design and development work and then build a small team to continue development of the product.

You should have well proven design skills and a very good understanding of communication protocols including TTP, FTP, SMTP and TCP/IP. You should have very good skills with C, C++ assembler and preferably have some exposure to encryption technology. Experience with major mail client file formats will be an advantage.

**For a strictly confidential discussion**

**phone or fax your résumé to**

**Owen Morgan at:**

**Level 2, 22 William Street**

**Melbourne Vic 3000**

**Phone 03 96497466**

**Fax 03 96212181**

**E-mail: [owen@cma.com.au](mailto:owen@cma.com.au)**

**vet**  
*it first*

The global security invariant defines at the highest-level the abstract security property preserved by the system. The JDK global security invariant is:

Downloaded applets are constrained by the application policy as configured by the client system administrator or end user.

This abstract definition of JDK security is supported by lower-level (more concrete) security properties enforced by other system components. The SRM defines the six components of the model as the Java applet, Java-enabled application, Java Virtual Machine (JVM), client platform, server platform, and server. Each of these components is given a degree of responsibility for preserving overall system security.

The lower-level security properties are defined in terms of security invariants for each security-relevant transition. A demonstration of effectiveness would show that all possible security transitions of the JDK implementation have been described, and that the combination of invariants and assumptions are sufficient to demonstrate that the global security invariant is met. A demonstration of correctness would show that the assumptions are preserved within the implementation of the invariants, and that a secure initial state can be created.

## SECURITY TRANSITIONS

The security relevant transitions defined in the SRM can be divided into three groups:

- Client and Server actions
  - Secure Initial State
  - Application Access Device Attempt
  - Application Manipulate Process Attempt
  - Server Platform to Client Platform Data Stream
  - Client Platform to Server Platform Data Stream
  - Server Platform to Server
- Actions taken by the application to load a class
  - ClassLoader constructor
  - Load Class
  - Verify/Link class
  - Class Initialisation
- Interactions between the application and the applet
  - Applet Access Device Attempt
  - Applet Manipulate Thread Attempt
  - Applet Manipulate Thread Group
  - Applet Manipulate Process Attempt
  - Applet Modify NameSpace Attempt

The first set of transitions are fundamental services that underlie the interactions between the applet and the application. The Operating System and system administrator are entirely responsible for these functions. The application's security policy must mediate interactions between the applet and the OS, and security policy implementation begins when classes are found and loaded.

The secure initial state is a key assumption. The user or the administrator must configure the application, JVM, and platform to properly enforce security policies. The application's system identity and device access privileges should be configured appropriately, and other processes running on the user's machine should not interfere with the application. Obviously if the system security policy is more restrictive than the application's (opening connections to port 25, for example), the system policy will be enforced. The application's security policy simply filters the requests that can be made by the applet.

## APPLICATION SECURITY POLICY

The application's security policy is established when the application starts up, and initially implemented when the applet requests a class. The JVM is prepared to handle three distinct types of code: system classes, native code, and downloaded classes. System classes are found using the CLASSPATH environment variable, which points to Java code stored on the local file system. Native code is not written in the Java language; the applet calling a native method causes the JVM to load an external library (.DLL or .so) to resolve the call. Neither system classes nor native code are checked for compliance with the application's security policy; the security context in this case is set by management, and from the application's point of view the code is completely trusted.

Downloaded code is subject to various checks on its integrity, and to the application's security policy. If a requested class is not found in the CLASSPATH the JVM calls the application's `loadClass` method to download an applet from a server. Instances of the class `ClassLoader` enforce the Java name space hierarchy, guaranteeing that a unique namespace exists for each source. To resolve references by name, the runtime system calls the `ClassLoader` that originally created the class. The runtime system must search the system class namespace first, to avoid using a downloaded class that mirrors a trusted class.

(Downloaded, in this sense, is slightly misused; any code which is not loaded from the CLASSPATH is loaded with an instance of `ClassLoader`).

An applet that is loaded by the `ClassLoader` is subject to the application's security policy, defined by the class `SecurityManager`. The `SecurityManager` class which is a part of the core Java library is an abstract class, so its implementation disallows everything, no matter what the source. To implement a security policy, the application must subclass `SecurityManager`, and override the methods that determine resource access. The application can have only one `ClassLoader` and one `SecurityManager`<sup>1</sup>, which are established at start-up and thereafter cannot be extended, overloaded, overridden or replaced. The combination of the `ClassLoader` and `SecurityManager` are meant to create a 'sandbox' for downloaded applets; an environment where the sources of code are clearly labelled and interactions can be mediated by the application's security policy.

In addition, the JVM doesn't assume that every class file was produced by a "friendly" or "trusted" compiler. During `loadClass`, the application in turn calls the JVM verifier (accessed through the methods `defineClass` and `resolveClass`) to verify and link the code. To pass the verifier, bytecode must conform to the strict typing and predicability of the runtime stack that are defined by the Java language. The verifier checks the class file format and object signatures, does a data-flow analysis of the bytecode instruction stream, and does a special analysis of the clauses that are used for exception handling<sup>2</sup>.

---

<sup>1</sup> A special class of downloaded code is a remote method stub. The RMI facility enables creation of distributed Java-to-Java applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts. It relies on an RMI Registry to supply a proxy (stub) for the class; when a remote method is invoked, the stub forwards the call to a server-side proxy (skeleton), which in turn handles communication with the remote object implementation.

Stub security policy for applications (not applets) is defined by the `RMISecurityManager` class, an extension of `SecurityManager` that overrides many of its methods. Typically the `RMISecurityManager` will disable all functions except class definition and access. A class can define methods that are not specified in the remote interface, but those methods can only be invoked within the virtual machine running the service and cannot be invoked remotely. The remote object implementation must explicitly create and install a security manager, or no loading of RMI classes will be allowed [Jdkd].

<sup>2</sup> Other languages can be compiled into the class format. The bytecode verifier is not specifically tied

to the Java language; any set of bytecode that satisfy the structural criteria will be certified by the verifier. However, The bytecode verification process itself has introduced security weaknesses. The SuperClass constructor bug and the and an attack based on package names described by the Safe Internet Programming Team [Pr96] have been fixed [Jfaq], but that does not rule out the discovery of similar inconsistencies.

Class initialisation is done in two steps. The first time an instruction that references a class is executed, the verifier 'prepares' the class by initialising statics to default values. The first time an instruction calls a method, or accesses or modifies a field, the verifier executes the initialisation code for the static fields declared in the class, and the initialisers for the class overall. The verifier prevents code from using the new object before it has been initialised, and from initialising the object twice [FY94].

## BUILDING A TRUSTED COMPUTING BASE

In Orange Book terms, the `ClassLoader` provides Identification that can be used by the `SecurityManager` to enforce security policy, and the JVM verifier provides some degree of Assurance that the constraints of the language itself are enforced. Marking is one of these constraints; if no modifier (`public`, `protected`, `private`) is specified, a method is accessible only within its defining package. Additionally, a variable or method can be declared `static`, which cannot be overridden in a subclass, classes declared `final` cannot be subclassed, and methods declared `final` cannot be overridden.

The latest JDK (1.1.b3) provides interfaces for signed "Java Archives". Essentially, a digital signature is appended to a group of classes combined into a single file, and run through a one-way hash function<sup>3</sup>. In conjunction with routines for key and function management, these facilities can provide Continuous Protection for both system code and code transferred across the network.

Unfortunately, the security manager doesn't have all the information that it needs to do a good job of resource tracking. The security manager can investigate the current execution environment to learn

---

to the Java language; any set of bytecode that satisfy the structural criteria will be certified by the verifier. However, The bytecode verification process itself has introduced security weaknesses. The SuperClass constructor bug and the and an attack based on package names described by the Safe Internet Programming Team [Pr96] have been fixed [Jfaq], but that does not rule out the discovery of similar inconsistencies.

<sup>3</sup> It is essential to sign before encrypting if one is to assume the signer has any knowledge of the data. A third party cannot assume that a signature attached to encrypted data is genuine, so non-repudiation is lost [RA95].

what classes currently have methods executing on the execution stack; it can also learn about the ClassLoader objects that loaded those classes into the runtime environment. However, sockets don't keep track of the number of bytes that have passed through them, and threads don't keep track of their CPU utilisation. No provisions are made for recording access to resources, even when this information does exist.

Through careful design, these basic facilities can be combined to implement an acceptable reference monitor. Current implementations have limited their efforts to confining the applet to a 'sandbox', because the security policy is rather easier to define, and the Java system is in a state of flux. Even the sandbox has been more difficult to implement than might be expected, because developers are anxious to take advantage of the potential of distributed applications. For example: The sandbox limits an applet to opening a port on the server where it originated, either by IP address or the CodeBase associated with the web page. If there is no way to see the number and purpose of active threads, it is possible to enlist every client that visits the page in a distributed computation without the client's knowledge. The author of one such applet points out [Guru]:

*This possibility raises a tangled web of unaddressed legal and ethical questions, and it is at least conceivable that running such an applet might be illegal in some situations. For example, suppose that a federal employee, say from NASA, happens to download an applet which begins using government resources for private ends. Have any laws been broken, and if so, who is the guilty party? Now suppose that instead of factoring integers, the applet farms out pieces of a brute force attack to decrypt some financial information, and suppose that an FBI agent, doing a little lunchtime browsing, happens to download the applet running this decryption program. ....The present example shows that even allowing applets to establish connections to other ports on their home hosts entails risks.*

In this case, even a 'ps' type summary of executing threads may not be sufficient; the Internet Worm of 1988 deliberately killed some of its processes in an attempt to avoid detection [Spaf].

## SECURITY POLICY AND ASSURANCE

- It is up to the application designer to ensure that the application always calls the SecurityManager to see if a requested access is permitted. The JDK provides default implementations for many access

methods which call the SecurityManager, but the application designer is free to override the default implementations through subclasses. A failure to call the SecurityManager will result in access being granted, contrary to the security engineering principle that dangerous operations should fail unless permission is explicitly granted. In addition, the default implementations are not comprehensive. For example, the SecurityManager only checks that applets cannot alter system threads; there are no restraints on applets altering other applet threads.

- Security policy must also ensure that the applet cannot access resources by any means other than calling the application. In essence, this means that access to native methods must be strictly controlled. When a native method is entered, the internals of the JVM and the environment outside it are equally exposed. Java itself has no mechanism for assuring that the integrity of the system is respected by native methods. A design for an Extensible Security Manager which uses "factory objects" to register and enforce security policy for native methods has been proposed [Guru], but the burden of assurance that the security manager is always called is left with the application developer.
- Security policy must prohibit untrusted code from defining new classes into a package that is protected by the application security manager. In the current implementation, a package is just a loose association of classes, and the classes of a given package can be loaded from different locations [JB96]. If no modifier (public, protected, private) is specified, a method is accessible only within its defining package. It is the responsibility of the ClassLoader to identify the package in which the class belongs and call SecurityManager.checkPackageDefinition before actually loading the class as part of that package. An applet could declare new classes in packages which are not systematically protected against package extension, and gain access to attributes with package scope.
- Accountability must be addressed. Java's simplistic default model of trust (CLASSPATH vs. ClassLoader) can easily be extended, and the assurance that trust has not been misplaced or abused can come only from the analysis of audit information. The lack of accountability leaves open possibilities for denial of service attacks, and the use of covert channels<sup>4</sup>. Unanticipated

---

<sup>4</sup> A covert channel is any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy.



interactions which cause security breaches can also be uncovered. For example, an applet with access to the key management database and the ability to open a port can send the database object to the server it came from, even if it does not have direct access to some of the contents [JB97].

## CONCLUSION

Java as a system provides most of the basic tools necessary for building a product which could be incorporated into a secure system. While the developer of such an application is left with complete responsibility for assuring that a security policy is defined and enforced, Java provides sufficient mechanisms for marking, labelling, and continuous protection of the security functions. The most serious single deficiency is the lack of auditing information on resource access, which the system should take the lead in helping the developer to address.



## REFERENCES

- [AK91] Karila, Arto. Open Systems Security - an Architectural Framework. June, 1991.  
Available via  
<http://www.urec.fr/Ftp/secure/Docs/Reseaux/OpenSsystemsSecurity.txt>
- [Ccse] Common Criteria for Information Technology Security Evaluation - Version 1.0. Common Criteria Implementation Board, January, 1996  
Available via  
<http://www.radium.ncsc.mil/tpep/library/ccitse/>
- [Efaq] Computer Security Evaluation FAQ, Version 2.1 Apparently maintained by the NIST.  
Available via  
<http://www.radium.ncsc.mil/tpep/process/faq.html>
- 
- There are two types of covert channels: storage channels and timing channels. Covert storage channels include all vehicles that would allow the direct or indirect writing of a storage location by one process and the direct or indirect reading of it by another. Covert timing channels include all vehicles that would allow one process to signal information to another process by modulating its own use of system resources in such a way that the change in response time observed by the second process would provide information. [Osec]
- [FY94] Yellin, F. Low Level Security in Java. Fourth International World Wide Web Conference, Boston MA, Dec. 1995.  
Available via  
<http://www.w3.org/pub/Conferences/WWW4/Papers/197/40.html>
- [Guru] Vanderburg, G. et al. Tricks of the Java Programming Gurus. Sams.net Publishing, Indianapolis, Indiana. 1996  
Available via  
<http://www.mcp.com/390157412285691/sams/books/102-5/httoc.htm>
- [Hsec] Information Technology Security Evaluation Criteria. Department of Trade and Industry, Publisher. London, June 1991  
Available via  
<http://www.radium.ncsc.mil/tpep/library/non-US/ITSEC-1.2.html>
- [JB96] Billon, J. Java Security: Weaknesses and Solutions. December, 1996  
Available via  
[http://www.dyade.fr/actions/VIP/JS\\_pap2.html](http://www.dyade.fr/actions/VIP/JS_pap2.html)
- [JB97] Billon, J. Security Holes in the Java Security API. January, 1997  
Available via  
<http://www.dyade.fr/actions/VIP/SecHole.html>
- [Jdkd] JDK 1.1 Beta Documentation Sun Microsystems, Inc. 1997.  
Available via  
<ftp://ftp.javasoft.com/docs/jdk1.1/jdk1.1-beta3-docs.html.tar.gz>
- [JFaq] Frequently Asked Questions - Applet Security. Apparently maintained by Sun Microsystems.  
Available via <http://www.javasoft.com/sfaq>
- [Jsrn] Erdos, M. et al. Security Reference Model for the Java Developer's Kit 1.0.2. Sun Microsystems, November 1996.  
Available via  
<http://www.javasoft.com/security/SRM.html>
- [Osec] Department of Defence Trusted Computer System Evaluation Criteria. (US) DOD 5200.28-STD, Library No. S225,711, December, 1985.  
Available via  
<http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html>
- [Pr96] Dean, D. et al. Java Security: From HotJava to Netscape and Beyond. Proceedings of 1996 IEEE Symposium on Security and Privacy, May 1996.  
Available via  
<http://www.cs.princeton.edu/sip/pub/secure96.html>

[R18.80] Electronic Funds Transfer without stealing PIN/TAN. Risks-Forum Digest, Volume 18:Issue 80 (1 February, 1997). ACM Committee on Computers and Public Policy, Peter G. Neumann, moderator. Available via <http://catless.ncl.ac.uk/Risks/18.80.html>

[RA95] Anderson, R. and Needham, R. Robustness Principles for Public Key Protocols. Published in Crypto 95. Available via <ftp://ftp.cl.cam.ac.uk/users/rja14/robustness.ps.gz>

[Spaf] Spafford, E. The Internet Worm Program: An Analysis. Purdue Technical Report CSD-TR-823, November, 1988. Available via <ftp://ftp.win.tue.nl/pub/security/worm.report.ps.Z>

---

## Has NT's Time Come?

*Frank Crawford <Frank.Crawford@auug.org.au>*

Microsoft's publications regularly claim that this is the year that NT will take off, as if these repeated statements will make it happen. However, in the last few months there appears to be some evidence that NT may in fact be beginning to be accepted, but this may not be exactly the way Microsoft would like.

In the last six months the number of people actively studying NT, finding design flaws, bugs and security issues has increased dramatically. Making it worse for Microsoft, this concentrated attention has started to filter through to the non-technical people involved in decision making, countering many of Microsoft's own press releases.

In fact, NT is starting to suffer from exactly the same public pressures that most UNIX systems have for years (and has even been used by Microsoft as points against UNIX).

Things started quietly late last year, when it was noted that by connecting to the NT Web server (IIS), by typing a specific string you could cause the IIS server to crash. Nothing too dramatic, and it turned out to be fixed in the next release of IIS, however, not everyone was ready to upgrade to the new version, or even knew it was necessary.

Over the Christmas period, things began to get worse. Firstly, it was discovered that connecting to NT's RPC server (an essential service) and typing random characters would cause the system load to approach 100% and lock out all those network processes to that machine. The way to fix the problem was either to stop and start the process controlling the RPC service (rpcss) or reboot the system. Either way this would drop most active network services.

Shortly after this, it was discovered that the RPC service was not the only one susceptible to such attacks. Further, this bug was in both NT version 3.5 and NT version 4.0. To it's credit, Microsoft acted fairly quickly and had a fix out for the problem within a week, however, this fix was only for Intel based NT version 4 systems with the current service pack installed (SP2), and only for the original RPC problem.

After these problems surfaced, a few users started investigating other such shortcomings. Recently it has been discovered that the DNS server (i.e. the program that handles Internet name to address conversion) can be easily killed if it is sent a response that it never requested.

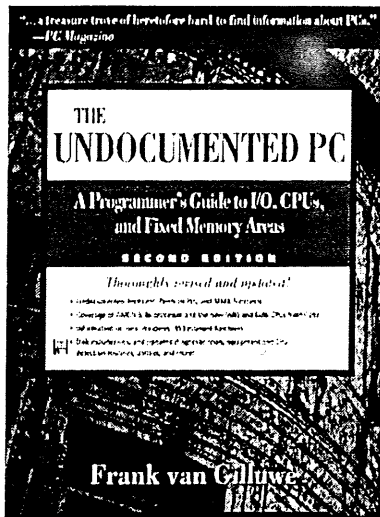
Now, while it may seem that all these problems are unlikely to occur in real life in a protected environment, on the Internet they are common place. All of these can (and are) used by malicious users to perform what are known as "Denial of Service" attacks against systems. For a system used within an Intranet the likely-hood of such an attack is small, but for any system connected directly to the Internet, it is a certainty.

If the existence of these problems is not bad enough, they all seem to exhibit a common feature, and one which raises even more concerns. They all seem to stem from poor data and error checking by the programs involved, and by extension, probably many others. This is something that is taught from day one in most programming courses and is essential in all programs that have any security implications. Unfortunately, it appears that many of the programmers at Microsoft have forgotten it.

Taken back a step further, this problem may be a cultural one at Microsoft rather than just poor programming. Most of the codes developed by Microsoft are not available for public inspection, are developed under extremely tight deadlines and little effort is put into working with those outside a very tight circle.

If you contrast this with the UNIX world you see why this culture is a problem. Firstly, most new

# NEW! from Addison Wesley Longman

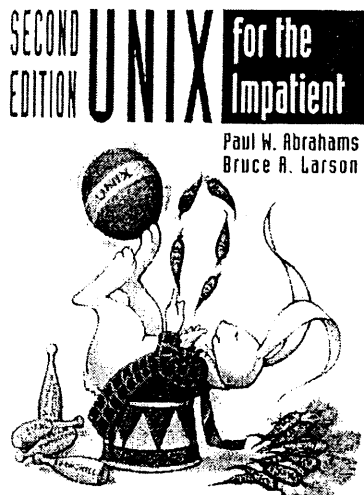


## The Undocumented PC 2nd Ed

Fully updated and revised, this new edition contains even more undocumented and hard-to-find information than the bestselling first edition. This includes comprehensive coverage of Pentium, Pentium Pro, and MMX functions, AMD's 5k86 processor, 5x86 and 6x86 CPUs from Cyrix, the Windows 95 Keyboard functions and more.

Each major subsystem of the PC is explained in detail, followed by extensive reference sections that show and explain both documented and undocumented BIOS functions. This book also covers all the major system architectures and is still the only available reference to input/output ports, undocumented CPU instructions, fixed memory areas and more.

ISBN: 0 201 47950 8 RRP: \$62.95



## UNIX for the Impatient 2nd Ed

Thoroughly updated to account for recent developments, the second edition of this bestselling book is an in-depth, comprehensive guide to UNIX—a handbook that you can use both for learning and as a ready reference. This book is written for the technically oriented UNIX user who doesn't want to wade through verbose tutorials, but isn't already an expert.

Its functional organization makes it easy to find the right tool for any task, with a complete alphabetical summary providing fast lookup of commands, options and subcommands.

ISBN: 0 201 82376 4 RRP: \$46.95

**AUUG Members receive a 20% Discount off the RRP**

Please post or fax to Alison Barr, Trade Marketing Coordinator at Addison Wesley Longman  
95 Coventry Street, South Melbourne Victoria 3205 Ph (03) 9697 0664 Fax: (03) 9699 2041.

Title \_\_\_\_\_ Price \_\_\_\_\_

Title \_\_\_\_\_ Price \_\_\_\_\_

Total less 20% \_\_\_\_\_

Postage and handling charges 1 book \$4.00 2 or more books \$7.00 Plus Postage \_\_\_\_\_

Send your order with a cheque, or bill your credit card.

Bankcard  Mastercard  
 Visa  Amex

Card No

Expiry Date \_\_\_\_\_ Amount \_\_\_\_\_

Name \_\_\_\_\_ Signature \_\_\_\_\_

Address \_\_\_\_\_

Suburb \_\_\_\_\_ State \_\_\_\_\_ Postcode \_\_\_\_\_

Prices are correct at time of printing, but are subject to change without notice.

**ORDER FORM**

features within UNIX have been developed though the free availability of source code. First through the original Bell Laboratory versions, then to BSD, and now with Linux, FreeBSD and other public developments. The availability of source has two effects, firstly, the original programmer is conscious that others will be seeing their work and are unlikely to take short cuts, and secondly, even if they do miss something, a later developer is very likely to pick it up. It is interesting that while many of the commercial UNIX vendors make their own modifications, the vast majority of the code base still comes from the original freely available source.

The pressures of deadlines is an obvious cause of errors, and while it is present in the commercial UNIX market place, so much of the development is still done in the public domain, it is not as much of an issue. At the same time, because there are a number of different UNIX vendors and developers, while one may be under the pressure of a deadline at any one time, others will not, and hence have some time to note the problems and correct them.

The limited exposure that Microsoft products get before release is a further cause for concern. As any developer will tell you, the users of a program do not do as you expect. When the process is tightly controlled, it is unlikely that anyone will do one of those stupid things that happen in the real world. The code will work correctly when given correct data, but it is unlikely to be heavily tested on its response to incorrect data. Again, the wide distribution of code at a very early stage in the UNIX world, quickly fixes that.

So, while NT may have started to come of age, like any child this means that it has also started to encounter unexpected problems. While the basic design appears to be sound, the implementation leaves a lot to be desired, and there are likely to be many growing pains ahead.

❖

---

## Computer Magic

*Frank Crawford <Frank.Crawford@auug.org.au>*

Everyone involved in computer support knows of people who just have to walk by a computer for it to play up. They also know people who just seem to have everything go right for them, such as only ever having to install Windows 95 once (or maybe never). At the same time anyone who is seriously involved in computer support has to firmly believe in

Murphy's Law, and often take unusual actions to mitigate it.

Now, while those not seriously involved may think that the statements above shouldn't be taken seriously, in fact, in modern computing there are good reasons to follow them.

Today's computers are by far the most complex piece of equipment that people encounter, and not just from the hardware components. If you look at the average PC, it has all the equipment of a stereo, telephone, typewriter/printer, fax machine, copier and television, as well as the "specialist" items such as a hard-disk, network card and mouse. Then on top of this can change it's personality at the touch of a button, from an educational toy for a pre-schooler, to a Formula 1 racing car simulation, from a challenging chess opponent, to something to chat between friends, and even on to something for solving complex scientific problems that are hard to describe let alone understand.

Yet, at the same time these things are being deployed into every corner of the country with the assumption that anyone can use them. Also built within this widespread distribution is the assumption that people understand what a computer is doing. Unfortunately, this is totally false. It may have been possible for people to understand all the steps involved in any computational activity, in the days when computers only ran a single job at a time, which included all the instruction for every activity, but today with multitasking, hardware and software interrupts, multiple devices, and even multiple CPUs, at any tick of the clock, your computer may suddenly run off to undertake a different and unexpected activity.

All these different activities imply an exponential number of possible interactions, and it is these that make any computer unpredictable. In fact, today, much of the work by computer support staff involves experimentation to see how a system reacts in a given situation. The good support people have a feel for it, just like the old bushman has a feel for the country.

At one time, this experimentation was only needed for high end systems, unfortunately, it is now getting more and more common right across the entire computer spectrum. Despite the open nature of most UNIX systems, security analysts often have to experiment to see the effects on different security policies and options. Even worse, NT security people are unsure which settings in the operating system are significant, and which are irrelevant. To make matters more difficult, the NT documentation does not even fully list what all the security options are!

The problems faced by these groups is no different to scientists trying to explain the world we live in, unfortunately, the world that is being explored here is man made, yet we still can't comprehend it.

When you take a step back and look at how this affects what the average user sees, it is not surprising that unusual problems occur, and that some people are just naturally better at adjusting or correcting them than others. Some people will instinctively take the correct approach, others will blindly follow some preconceived ideas.

Ultimately, it comes down to the simple issue that anything to do with a computer was developed by a person. Unarguably, this person was not able to comprehend all the interactions that may affect their work, and as such may have left out an important detail. On the positive side, you should always assume that this was unintentional, and in general, is only a small deviation from the truth (however, the lower down this is the bigger can be the effect by the time it affects you).

In other words, what doesn't work was not aimed at you, but was only caused by them being human, and, at the same time, understanding and correcting that problem may take intuition and "magic" rather than a totally logical process.

❖

---

## The Network is Down

*Craig Bishop  
President of SAGE-AU*

How many times do we hear these words uttered in frustration on a weekly basis? The "thing" that is down often varies and we could easily substitute "system" or "computer" to cover the majority of cases. Regardless of the word used, the answer is very much the same. It depends. It depends on how good your system administrators are at running your site.

As computers continue to invade the workplace and all forms of work practices, our personal reliance and business reliance on the network and systems which connect us to the corporate Intranet and on to the Internet, is making them mission critical. Therefore the administration of these networks and systems has become, or is fast becoming, a mission critical function.

So, who are we entrusting with this mission critical function?

A function which requires the following skills:

- complex problem solving skills, often under pressure;
- performance of installations, upgrades and maintenance;
- capacity planning and capital expenditure budgets;
- project management skills, so that planned activities are completed - on time and on budget with minimal impact on the day to day business;
- can understand complex technical manuals;
- can bridge the technology gap for your general staff.

I started my career as a system administrator 15 years ago at Deakin University. I had just finished an honours degree in Computer Science and I decided to stay on at the University to administer a system for honours students and research staff. Yet, I found myself very badly equipped by my study to take on this role. I had studied many useful subjects related to computer hardware, software design, graphics, programming and information systems. Unfortunately I learnt nothing about being a system administrator. After ploughing through many manuals and much trial and error, I connected the University to the beginnings of the Internet in Australia where I obtained access to many other people working on the same problems. I had email and network news (yes they did exist back then) and the ability to make mistakes and learn. Eventually I became a system administrator.

Things have changed now, or have they?

At this time there are only two Australian Universities known to me which offer significant training in system administration. For the majority of people who become system administrators in this country, there is still no formal training or exposure to the knowledge and skills it takes to fulfil this job. The only difference now is that the Internet and the technology are more accessible, but this is offset by technology which can be substantially more complex.

This situation creates a problem for employers of system administrators. Experience is costly in two separate ways, employing an experienced system administrator can be expensive (\$50k upwards), employing a graduate or inexperienced system administrator will cost in training, recovery from honest mistakes and ultimately in down time on your network and systems.

This is where SAGE-AU can help. SAGE-AU is the "System Administrators Guild of Australia".

SAGE-AU is an organisation which is affiliated with SAGE organisations around the world. We exist to promote the profession of system administration across all operating systems and provide services to system administrators.

Some of the services provided to members by SAGE- are listed below:

- mailing list for asking questions about system administration -Questions range from system specifics, disk configurations for different applications to the ethics of email administration and the Internet;
- a network of helpful system administrators who provide advice and answers based on their experience to this mailing list;

- a newsletter which comes out every two months;
- a yearly conference with talks and tutorials from experienced systems administrators;
- frequent regional meetings with talks by experienced systems administrators, discussion forums and technology updates;
- a world wide web site with summaries of regional meetings, useful bookmarks for system administrators and other valuable information;
- a code of ethics.

System Administration is an old job but a new profession, it is a profession which has not yet matured and is often not recognised. SAGE-AU can help you and your system administrators to bridge the training gap and ensure your network and systems run smoothly and efficiently. Have a look at what we have to offer at <http://www.sage-au.org.au/>



## Skillsearch Computing Pty Ltd

Call us for your next Contract or Permanent position .  
We know the market for your skills in:



- ☐ *Telecommunications*
- ☐ *Banking & Finance*
- ☐ *SCADA*

Good Unix , C & C++ developers are always in demand !!  
So call Farah Shah or Geoff Drury today and let us help you find the job you want..

fshah@skillsearch.com.au  
gdrury@skillsearch.com.au

Phone: +61 2 9955 4001  
<http://www.jobnet.com.au/>

---

# Confessions of a Road-Trip Tutorial Presenter

Danny Smith <danny@auscert.org.au>  
Operational Manager  
AUSCERT

When Pauline van Winsen from Uniq Professional Services first approached me on behalf of AUUG to ask about delivering a tutorial series, my first reaction was "What a great idea!".

Now that the trip around Australia has been completed, I am not so sure it was such a great idea. Mostly, it was my own fault.

Liz Egan and her supporting staff from AUUG did an absolutely wonderful job of organising the travel, logistics, and a million other fine details. Everything ran very smoothly, but many people may not be aware of just how close it came to all falling apart! Let me tell you how it all happened.

Liz and I started to discuss the dates, places, and the various schedules. Trying to get rooms available on certain days, getting a sane flight schedule, and keeping costs within reason proved to be an interesting time. I also added a stricter criteria: I did not want to be travelling away from home for extended periods of time, so I wanted to tighten the schedule up.

After much discussion, it was felt that Perth was just out of reach for this trip, so we finally settled on the following schedule:

11 Nov	Darwin
13 Nov	Adelaide
14 Nov	Melbourne
15 Nov	Hobart
18 Nov	Sydney
19 Nov	Canberra
20 Nov	Brisbane

So we started in Darwin. A fairly lengthy flight via Cairns and Gove, to arrive in Darwin with around 34 degree heat at 2100. Sometime during the evening, it was discovered that the tutorial notes and AUUG banner had not arrived as scheduled. You see, "Overnight Capital City Delivery" doesn't really mean overnight when it comes to Darwin.

Fortunately, all items arrived with about 30 minutes to spare, so we were back on track. The best part was the sticker that was placed over the "Overnight Delivery" sticker (after the articles were submitted) that ruled out Darwin.

The day went very well, until around 1630 when the fire alarm started to go off. We were instructed to stay in the room while it was investigated. The alarm was finally cleared, only to go off again (several times). Consequently, the afternoon session was an interesting time of speaking and listening (to fire alarms).

Tuesday I headed to Adelaide. This was the most leisurely part of the travelling. When I arrived in Adelaide, I was advised by the taxi driver that there had been a bus strike that day, and that taxis were kept rather busy. I inquired about the likelihood of having troubles the next day and he assured me that all would be well.

Wednesday morning, I ordered my taxi, but it failed to arrive within a reasonable time (like half an hour). Since time was getting away, I shared a taxi with another gentleman who, unfortunately, was heading in the wrong direction. Finally arrived with around five minutes to spare, set up and started my presentation. At 1630, an impact drill started somewhere in the building, making it difficult to hear anything. I decided that there was a trend starting here - a trend I didn't like.

Departed Adelaide and headed for Melbourne, arriving rather late (damn time zones!). The room for the presentation was downstairs with a set of bar fridges along one wall behind me. These fridges continually started and stopped, making it almost impossible for me to hear anything. We had them terminated (thanks Steve)!

Next stop was Hobart. The flight from Melbourne to Hobart had a few "lads" on board who were heading down to watch the International Cricket match at Bellerive. These "gentlemen" had consumed a few refreshments on their flight from Sydney, and were keen to get a few more down before disembarking in Hobart - so keen, they continually pestered the cabin crew for beers - long before we had even taken off from Melbourne!

Arrived at Hobart at 1900 and it was seven (7!) degrees when walking across the tarmac. Bit of a shock after Darwin.

The hotel in Hobart was lovely, and the room that the tutorial was held in was very picturesque. Mind you, we had to close the curtains so that the sun didn't shine in! By this time, I was starting to feel a little

tired, and looked forward to a good night's rest. So I retired at 2030.

At 2100 that evening, the band at a function started playing - right under my room. I really enjoyed the music, the dancing shaking the floor, the loud voices calling to each other. They finally desisted some time after midnight.

I flew back to Brisbane on Saturday and managed to repack for the next onslaught.

Back to Sydney on Sunday night. The taxi driver decided I needed a tour of the Kings Cross area as we drove past. He very carefully pointed out "the wall" if I wanted any boys, the street where I would find the girls, and a plethora of shops to check out. I don't think I really got through to him that all I was interested in was getting to bed - alone!

Sydney started in an interesting fashion - there was no slide tray to load the slides into. After much ringing around, we decided to start without it, and fly "on a wing and a prayer". By the morning break, we had our slide tray and could continue normally. I started having fun with the people sitting near the back that had to bend one way or the other to see me around the various concrete pillars in the room. By moving backwards and forwards across the front of the room, I could almost start a swaying motion in the room - much like a rock concert!

On to Canberra. This was the largest audience with around 50 participants. The room was long and narrow, and only seated about six people across. Hence, it was a long way to talk to people at the back. By this time, I knew I was starting to lose my sanity. For the first time in the road trip, I actually found myself seriously questioning "What city is this?". During my presentations, I often use anecdotes and stories to highlight my points. I had reached a situation where I could not remember if I had already told a particular story today, or was it yesterday? I had definitely lost the plot.

Out to the airport at Canberra, only to find that my flight had been cancelled. Because I was a little early, they managed to get me the last seat on an earlier flight, so I made it to Brisbane without many more difficulties.

Finally, I presented in Brisbane, and I thought the road trip had ended. I had two days in the office to recover before heading back to Sydney on my next assignment.

It turned out that the road trip had not ended. Various people in Perth inquired as to why they were omitted. It was explained that the logistics of the time simply

did not allow it. The Perth people banded together and requested that they be included at a later date. I am very pleased to see that they could organise sufficient attendance to allow the road trip to continue to Perth and so I completed Perth on 2 May 1997.

Perth had its own share of troubles, not least was the strike in Western Australia that threatened the entire trip. I managed to get there, but the tutorial notes did not. Fortunately, a frantic phone call from Liz Egan to me about an hour before I left to catch the plane allowed me to photocopy more notes and carry them with me.

Seven capital cities in 10 days, and one capital city six months later.

A range of interesting problems arose along the way, and were solved by dedicated people. One problem that could not be solved by anyone was that the 35mm slides I was using started to flex and buckle when heat was applied (that is, when they were in front of the slide lamp). Hence, one minute they were in focus, and the next they would snap out of focus. Of course, every second slide was different.

In the aftermath of the road trip, the post-mortem indicated that we got extremely lucky. Since the agenda was extremely tight, any problems with transport along the way could have had a significant domino effect on the rest of the trip. Fortunately, most flights were on-time and the rest of the travel occurred without any problems.

My sincerest gratitude goes to the various volunteers that assisted me along the way to make the running of the series so smooth. In particular, I thank Liz Egan and her staff at AUUG for the excellent organisation and logistical support. I should thank Pauline for suggesting the road trip, I really should. Finally, thank you to all the people that attended the tutorial series. Without your support, none of this would have happened. I hope you got as much out of the day as I tried to put into it.

Finally, if any of you are thinking of presenting a road trip, I counsel you to talk to me first. Let me see if I can talk you out of it!

❖



---

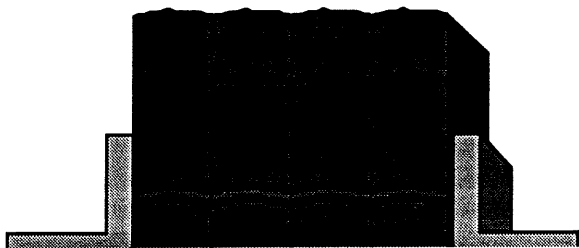
# Book Reviews

Sub-editor: Frank Crawford  
<Frank.Crawford@auug.org.au>

Welcome to another book review section, unfortunately it is a bit thin this month. This isn't due to a lack of books, but rather due to a lack of responses. Generally, most books are expected to be reviewed in 6 weeks to two months, but lately, some of the reviews have taken far longer. Anyway, I hope to have many more by the next review.

This month we have reviews on Linux, Java and Shell Programming, certainly a wide range showing the breadth of topics covered by the UNIX community.

At the moment, I don't have many books available, as I'm waiting for outstanding reviews to be returned, however, I will shortly be getting more. The current practice is to post a note to the mailing list <auug-books@auug.org.au> and the newsgroup `aus.org.auug` when we have new books available. Unfortunately, this disadvantages members without network connections, or on the end of a low speed link. For people in such a position, either mail, via the AUUG PO Box, or fax me on (02) 9717 9273, with your contact details and preferences.



## LINUX SECRETS

by Naba Barkakati  
IDG Books Worldwide (WoodsLane)  
1996, 902 pages + CDROM, \$99.95  
ISBN 1-56884-798-X

Reviewed by: Charlie Brady  
<cbrady@ind.tansu.com.au>  
Telstra

The title of this book struck me as rather odd - what "Secrets" can there be in a system developed and

distributed on the Internet with all source code available? Well, none really; "Linux Secrets" is just another title in a line of Secrets books, covering Windows 3.1, Windows 95, the Internet, Network Security, etc. The author, Naba Barkakati, is listed as "Linux Software Developer" on the cover, but a quick search of the Web reveals him as a professional computer book writer on subjects ranging from Microsoft Macro Assembler to Java, Windows Games to X Windows system to Object-Oriented programming and Windows 95 to Linux.

"Linux Secrets" is targeted at the beginner, but aims to go beyond the installation and setup of Linux and show how Linux can be used for practical tasks such as an Internet host, a workgroup server or a software development platform.

The first 300 pages of the book covers the installation and general running of Linux. No superficial overview, it includes, for example, sections on shell, perl and tcl/tk programming. The writing is generally clear and illustrated by many examples. The margins are dotted with icons indicating that the adjacent paragraph contains a Tip, a Note, Caution, a Cross Reference, a Secret ("facts which are not well-documented but important to know") and Wizard ("information which will be of interest to an advanced user". Scattered through the text are sidebars containing useful background material (e.g. a short history of X) or pointers to additional material.

Another 200 pages is devoted to "Exploiting Your Hardware in Linux". This section seems to be just a distillation or rehashing of various HOWTOs, FAQs and kernel documentation and will quickly become dated.

The rest of the book consists of descriptions of how to use Linux for a variety of practical tasks. Most of these chapters could be expanded to a book of their own - "Setting up a Linux Internet Host" and "X Programming in Linux", for instance, are hardly covered in great detail in about fifty pages.

Given the scale of the task, the author succeeds reasonably well in presenting Linux as a useful and usable system for users new to UNIX. There really is a lot of stuff covered in the book, which is generally well-written and no doubt interesting and useful to many Linux users.

The book, however, has major failings. For a start, it is too expensive. There are few technical errors, but one I found is a real howler - "Linux drivers use the BIOS to access the peripheral devices". This is so fundamental that I find it telling; the author does not know and understanding Linux well. This is reflected

in some of the omissions - no mention of sticky bits, setuid and setgid programs and directories, block and char devices and mknod, job control, the cron daemon, scant mention of ps and kill, hardly a mention of security.

The CDROM which accompanies the book contains the Slackware 3.0 distribution. Much of the text is closely tied to the CDROM contents - down to 12 pages listing the Slackware "disk sets" and their contained packages. Although one could criticise the choice of Slackware, it is the lack of any reference at all to alternatives that is the real failing. Not only is there no mention of the increasingly popular RedHat and Debian distributions, but where there are differences in the distributions, such as the init package and gettys, there is no information about the alternatives. This could certainly be a source of confusion.

A Linux beginner who was happy to use the Slackware distribution would certainly find this book useful. Eventually though, they would almost certainly need information that this book doesn't even hint at. Users of other Linux distributions would find "Linux Secrets" even less satisfactory. Look elsewhere, I'm sorry to say.

❖

## JAVA PROGRAMMER'S LIBRARY

*by Suleiman "Sam" Lalani and Kris Jamsa, Ph.D.*  
*JAMSA Press (WoodsLane)*  
*1996, 556 pages + CDROM, \$99.95*  
*ISBN 1-884133-26-6*

*Reviewed by: Craig Macbride <craig@rmit.edu.au>*

This is yet another in the plethora of Java books now available. It is interesting to read, from the point of view of someone who has had little to do with Java, but how well does it achieve its aims?

The book includes a CD-ROM with the JDK for MS Windows 95 and NT, states that its goals include giving programmers everything they need to be instantly productive with Java, and launches into installation instructions for the JDK. It includes instructions for running the compiler and appletviewer and, of course, how to find the examples from the book on the supplied CD.

The 50 chapters which follow describe various elements of the Java language, how to use these elements, and give sample programs. Every sample program is listed in full, along with a line by line commentary on what is going on and what features of the language are being used. The explanations are quite detailed early on, so as to give the novice a fair idea of what the parts of each program is doing.

There is basically one program or applet per chapter, with a series of learning goals at the beginning of each chapter and suggestions at the end of programming exercises for enhancing the given example. The chapters cover blinking graphics, cute buttons, playing music, using trigonometric functions in Java to bounce a ball from letter to letter on a displayed message, displaying various styles of real-time clocks, server/multiple client messaging, two-way chat clients, graphical techniques for card games and so on.

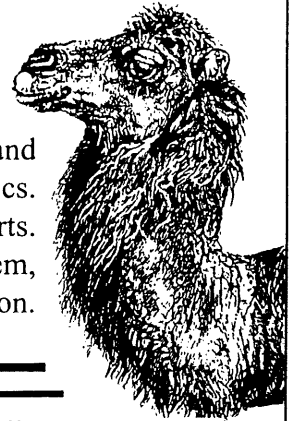
There is one error very early in the book, when using packages from the JDK is being discussed. The following is stated: "Note that like the C/C++ #include statement, you don't place a semicolon at the end of an import statement." Every example of an import statement in the book, except some on that particular page, has a semicolon at the end of it. Indeed, the Java compiler is most unhappy if you leave the semicolon off.

It is this sort of error, as well as omissions of answers to questions, which makes the book not feel as complete and precise as it could be. I often found myself wondering "what if" and "why" questions about methods used in this book. While the answer in many cases is to read the Java language and class library definitions, this requires jumping from one reference to another too often. Broad but accurate answers could be given to many matters that would prevent most need to seek a more precise reference. For example, I would have expected some coverage of security, but there is none. I would have expected some comment on the practical use of Java, such as what happens when text-only browsers come along, but again there is none.

It is the subtleties of programming languages that cause programmers to make subtle mistakes, so the most important thing is often not just showing a way to achieve something, but showing why not to do it other ways. Java has numerous class libraries available, containing many class definitions. I don't feel that this book really gives enough information on what resources are available to the programmer.

The book does make you instantly able to produce useful Java code, but only for some applications. A part of the introduction to the book says: "most

# Animals will always rule the jungle...



O'Reilly is the key to helping you master Internet publishing and programming topics, going above and beyond mere mechanics. O'Reilly books explore subjects in-depth and are written and reviewed by experts. The next time you want to tackle an animal of a problem, consider O'Reilly as the only solution.



1565921496

## Programming Perl 2/e

Wall  
O'Reilly & Associates  
\$79.95

Perl has emerged as the language of choice for World Wide Web programming. This heavily revised second edition of Programming Perl contains a full explanation of the features on Perl. It covers version 5.0 Perl syntax, functions, debugging, efficiency and the Perl library. It also includes a Perl cookbook and a quick reference card.

1565922603

## Unix Tools 2/e (Bk/CD)

Peck  
O'Reilly & Associates  
\$120.00

Loaded with even more practical advice about almost every aspect of UNIX, this new edition of UNIX Power Tools addresses the technology that UNIX users face today. It presents a combination of tips and tricks with links to other articles presented in a hypertext format. The CD-ROM includes all of the scripts and aliases from the book, plus Perl, GNU emacs, netpbm (bitmap manipulation utilities), ispell, screen, the sc spreadsheet, and about 60 other freeware programs.

1565921526

## Learning GNU Emacs 2/e

Cameron  
O'Reilly & Associates  
\$59.95

Learning GNU Emacs second edition describes GNU Emacs Version 19.29, which has many new features, including support for Internet services (email, Usenet news, File Transfer Protocol, and the World Wide Web), Revision Control System (RCS), and X Windows integration.

1565923049

## Java in a Nutshell - Deluxe Edition (Bk/CD)

O'Reilly  
O'Reilly & Associates  
\$140.00

A Java programmer's dream come true in one small package. The heart of this Deluxe Edition is the Java Reference Library on CD-ROM: Volume 1, a comprehensive electronic library for Java developers and programmers. This invaluable on-screen resource is a complete reference describing all aspects of version 1.1, all fully indexed and searchable. Included in this five volume package is the bestselling second edition of Java in a Nutshell.

1565922360

## DNS and BIND 2/e

Albite  
O'Reilly & Associates  
\$65.00

This book is a complete guide to the Internet's Domain Name System (DNS) and the Berkeley Internet Name Domain (BIND) software, the UNIX implementation of DNS. In this second edition, the authors continue to describe Bind version 4.8.3, which is included in most vendor implementations today.

1565922298

## Webmaster in a Nutshell - Deluxe Edition (Bk/CD)

O'Reilly  
O'Reilly & Associates  
\$140.00

WebMaster in a Nutshell takes all the essential reference information for the Web and pulls it together into one slim volume. This book is a quick reference for anyone who works on the Web-content providers, programmers and administrators alike.

1565921323

## Networking Personal Computers with TCP/IP

Hunt  
O'Reilly & Associates  
\$59.95

This book offers practical information as well as detailed instructions for attaching PCs to a TCP/IP network and its UNIX servers.

1565921518

## Running Linux 2/e Welsh

O'Reilly & Associates  
\$59.95

This second edition of Running Linux covers everything you need to understand, install and start using your Linux system.

1565921003

## Sed & Awk 2/e

Dougherty  
O'Reilly & Associates  
\$59.95

Linux is the most exciting development today in the world of UNIX. This book is a broad overview of Linux with specific instructions on installation and introductions to tools for users and programmers.

1565922395

## Mastering Oracle Power Objects (Bk/Dsk)

Greenswald  
O'Reilly & Associates  
\$79.95

This is the first book to cover Power Objects Version 2.0. It provides a wealth of developer tips and techniques, as well as understandable explanations of the internal workings of Power Objects. The accompanying disk contains practical and complete examples that will help you build working applications.

1565922220

## Sendmail 2/e

Costales  
O'Reilly & Associates  
\$79.95

Sendmail 2/e, covers sendmail version 8.8, as well as the standard versions available on most systems. Other topics include the error delivery agent, sendmail's exit values, MIME headers and how to set up and use the user database, mailertable and smrsh.



The best tool for programming on the web? **O'REILLY™**

DISTRIBUTED IN AUSTRALIA BY WOODSLANE  
Phone (02) 9970 5111 Fax (02) 9970 5002 Email: techbooks@woodslane.com.au

importantly we created several really cool applets." It is true that they have, but it's unfortunate that this was, to the authors, the most important point. I would prefer to have a better breadth of data from which to create a greater variety of my own really useful applets.

In the end, I found this book similar to a number of movies that I've seen recently, such as "Mars Attacks" and "Metro". Despite enjoying them, I can't help thinking about them afterwards and pondering how much better they could have been, given the resources available.

❖

## PORTABLE SHELL PROGRAMMING: AN EXTENSIVE COLLECTION OF BOURNE SHELL EXAMPLES

by Bruce Blinn

Prentice-Hall, Hewlett-Packard Professional Series  
1996, 270 pages  
ISBN 0-13-451484-7

Reviewed by: Con Zymaris <conz@cyber.com.au>  
Cybersource Pty. Ltd.

The programability of the Bourne Shell forms one of cornerstones of UNIX's power and success as a serious enterprise Operating System. Bruce Blinn's "Portable Shell Programming" is part of the Hewlett-Packard Professional Series published by Prentice-Hall PTR. Don't let this mislead you in terms of target audience and systems. This text covers many popular commercial UNIX flavours, not just HP-UX. This book lists its scope and purpose as providing serious scripting examples for current practitioners, and tips for maintaining portability between the UNIX flavours. An understanding of UNIX is assumed.

Why do any programming in Shell? Firstly, availability. You're guaranteed that all UNIX systems will include the standard Bourne Shell (named after Stephen R. Bourne, formerly of Bell Labs). Secondly, simplicity. By comparison to other scripting languages, such as Perl, Shell's language constructs can be described in a few pages, making it easy to learn. Thirdly, applicability. Shell scripting makes the perfect tool for numerous system automation, system activity monitoring and software

installation applications, all of which can be made portable, with no need for re-compilation.

The text focuses on the Bourne Shell (/bin/sh), due to its default implementation across different vendors' systems, and the fact that some of the major scripting successors to the Bourne Shell, such as the Korn Shell, the POSIX Shell and BASH are supersets. C Shell is mentioned, but not delved into.

Among the topics covered are: Syntax, Variables, Shell Functions and Built-in Commands, Manipulating Files, Environment Variables, as well as more advanced topics such as Trapping Signals and garnering information through Remote Command Execution (rsh). Covered are Command Line Parameters, Using Filters, Shell Utilities (such as arithmetic/floating point operators, string manipulation and user interaction). There are also interesting chapters on Portability (between commercial UNIX variants, generally limited to comparisons between System V and BSD derivatives) and Debugging techniques as well as an FAQ section.

The text is easy to read linearly as a tutorial, as well as something to be dipped into as a reference. There is a clear emphasis on knowledge transfer through techniques, tips, and the accumulation of numerous Shell scripts for your toolbox, to be used as templates for new projects. The text comes with all scripts on floppy disk, as well as a pointer to an FTP site for downloads.

There are numerous techniques presented throughout on maintaining portable code using known UNIX variant idiosyncrasies, the `uname` command and the `OPTFIND` variable. Developers of portable Shell scripts need to be well versed in the differences between UNIX commands (such as command line parameters and values returned). Here, Blinn assists by providing lists of the most standardised commands, plus example code for handling known variations.

It is worth noting that a common paradigm used in the book is to determine the type of Operating System in use (e.g. Solaris, SunOS, HP-UX, AIX, Ultrix) and then look that up with a case statement to work out, for instance, which arguments to give to "ps". The problem with this approach is that when you port your shell scripts to a new O/S, you need to go through each shell script checking for these case statements, and add your new O/S to either the list of systems which use "-ef" or the list of systems which use "ax".

The examples also make extensive use of shell functions. This makes the code more modular and easy to understand, but does have the dis-advantage of

making the job of porting the code to ancient V7 or BSD based shells (such as the standard /bin/sh on Ultrix) very difficult. However in most such cases another shell which does support functions will be available, in addition to the built-in /bin/sh.

Overall "Portable Shell Programming" is a very clear, understandable book, which can reveal information to

the novice and professional practitioner. The book's title would seem to imply that there would be a sole focus on portability to the exclusion of other aspects of Shell scripting, but this isn't the case. Regardless, this is a title that can be recommended to beginning Shell programmers.

❖

*Combine Business with Pleasure*

**UniForum NZ '97**

**Untangling the Web**

20-24 May 1997  
Rotorua, New Zealand

**Featuring:**

Integration  
Linux  
Intranet

Network Security  
Data Warehousing  
Java

Registration forms available March 1997

Contact: Roger De Salis, UniForum New Zealand  
PO Box 397      email: [roger@newzealand.sun.com](mailto:roger@newzealand.sun.com)  
Wellington      Ph: 64-25-481-452  
New Zealand      Fax: 64-4-389-1214

[ Editor's Note: This is the first in a series of new columns we're hoping to bring to AUUGN. If anyone out there is interested in acting as Sub-editor for a column based on a particular flavour of UNIX, please let us know! ]

---

# Solaris Musings

Sub-editor: David Purdue  
<David.Purdue@auug.org.au>  
SunSoft Pacific

The recent AUUG Membership survey showed that 74% of members use Solaris, with 51% using SunOS.

So, after a discussion with the AUUGN Editor, I volunteered to edit a Solaris column, and you are reading the first one.

This column will, I envisage, look a lot like the "UNIX Tricks and Traps" column, except that it will concentrate on how things work in the Solaris environment. There may be some overlap - these things may work on other versions of UNIX, your mileage may vary. And, as always, contributions to this column are most welcome.

Yes, I do work for Sun. However, I promise not to let the marketing people run rampant with this column, and I promise not to drop a contribution just because it is critical of Sun's products.

Contributions should be sent to <David.Purdue@auug.org.au> or faxed to me on (02) 9904 7057. Please put "Solaris Musings" in the subject.

## LOGGING FAILED LOGIN ATTEMPTS

In managing the security of a host system, one thing you like to know is when failed login attempts occur. The occasional failed login can be ascribed to clumsy fingers, but repeated, frequent failed logins probably indicate an attempt to gain unauthorised access to a host.

Solaris can log this information, but by default this logging is turned off. To turn it on, you just have to create the log file: /var/adm/loginlog. This file has to have particular ownership and permissions, or logging will not occur. Run these commands as root:

```
# cd /var/adm
# touch loginlog
# chown root loginlog
# chgrp sys loginlog
# chmod 600 loginlog
```

Logging starts as soon as the files exists - there is no need to reboot.

Each log record will record the user name entered, the tty that the attempt came from and the time of the unsuccessful login, all separated by colons.

There is a down side - this logging does not do all it could. For one thing, if the login attempt came over the network, then all you will see in the loginlog file is a pseudo-tty entry, you get no indication of where the attack is coming from (although once you notice this happening you could use snoop to find the source).

Also, you do not get any log records unless there are five unsuccessful attempts on a single connection. At this stage the login process drops the tty connection and writes the 5 unsuccessful records into loginlog. Thus an attacker could avoid logging by disconnecting after 4 failed attempts.

## CHANGING THE LOGIN PROMPT

While we are talking about logging in - are you tired of the old "hostname console login:" prompt? Do you want to add some more information before you even see a login prompt? Do you want to revert to the more traditional ";login:"?

Well you can.

The actual login prompt for the console is set in /etc/inittab. Edit that file and look for the line that starts "co:". You will notice that the console is run by the ttymon program, and the -p flag gives the console prompt - change it as you like. This will, however, only change the prompt for the console.

Another thing you can do is create a file called /etc/issue. The contents of this file will be displayed by the login program before it issues the login prompt. This file is commonly used to warn people of the dire legal consequences of logging in to a machine for which you do not have explicit permission to log in.

❖

---

# Meet the AUUG Exec



## PAULINE VAN WINSEN

*Senior Technical Consultant  
Uniq Professional Services*

My parents claim I was born a geek. I was the only kid at school to have bi-focals at age eight. By the time I was in high school I was lugging home one of the school's Microbees to muck around on in the school holidays I knew I wanted to have some sort of career in computers.

I moved to Sydney to study Computing Science at NSWIT (now called UTS) & was introduced to UNIX by way of a parity error. The first year students in my year were told we were in a position of privilege & we were to learn Pascal on a new UNIX system, a Gould running BSD4.2.

Our first tutorial was on logging in, basic commands & using ex. I stalled at logging in. My surname had been munged by one of those great username standards designed to remove conflicts into a username consisting of all the same parity characters.

Little did I know that my funky username had tickled a feature of the microcode of the async board of the UNIX system I was attempting to login to. The upshot of that was that whenever I attempted to login I received the magic words: "Login incorrect".

So I spent the first lab session banging down the door of the "Interface Room" as it was called, saying I

couldn't login. The lab assistant spent many hours ignoring me & calling me a fool for forgetting my passwd. I finally was sent to see the Systems Programmer who lived in a dark, messy room at the end of the corridor who tried changing my passwd to no avail. He managed to track it down to a weird feature of the microcode of the async board which controlled all the terminals & had it's own processor. The microcode made the assumption that if it saw all the same parity characters from the username it would assume the tty was set to the same parity, & it cheerfully turfed out characters of a differing parity, hence I could never login due to my passwd containing characters of both parity. My username was changed to something which would work & my curiosity about UNIX was born.

I owe a lot to the Systems Programmer who helped me out that day (Hi Tony!), who gave me my first job working in the Interface Room as a trainee Systems Programmer when he got sick of me asking too many questions. He also insists I was the only student this feature ever effected.

Those were the days of ACSnet, the beginnings of AARNet & vnews, that great news reader that read every article in every newsgroup.

By the time I finished my degree I was working fulltime at NSWIT as a Systems Programmer. My friends at Uni all said I was wasting my time with UNIX & that I should try & get a COBOL programming job, there were lots of them [ *And its still true today! - Y2K Compliant Ed* ] going. Business programming didn't have much appeal, particularly COBOL.

Luckily I knew someone in Sun & he asked me if he could recommend me for a position at Sun, there was also the small matter of a headhunters fee, but he always claimed I would be great for the job. I didn't get that job, but another guy at Sun saw my resume & I think his words went something like this: "Anyone who puts 'Ability to quote Monty Python & Goon Show scripts on demand' in their list of skills on their resume is someone I want to employ". So I started work as a National Software Support Specialist for Sun a month later, which was a much better position than the one I originally applied for. I had a great time at Sun, learnt a lot & had my first real exposure to UNIX source code.

About five years ago I started with Uniq, along with some people I'd worked with at Sun. Simon, the guy with the penchant for Python is our MD. I now spend my days consulting, generally in the computer security field.

I'm a Director of ISOC-AU as well as AUUG and am involved with SAGE-AU.

I do have a life outside of work, though my SO sometimes argues that point. Though he's the one who bought a new ether hub so he could connect his new Linux box to the rest of the network at home & is looking at ways of making the monitors in our fish tank manageable from the network. Apart from the geeky stuff, sailing & reading are my two favourite pastimes.

So now you know I'm thoroughly broken & next time you see me please don't ask me to quote the Parrot sketch, I much prefer the Bookshop sketch.

❖

## MEET THE COMMITTEE: FRANK CRAWFORD

*Site Systems Manager*  
ANSTO

Well, this is my last chance to write something about myself as a member of the AUUG Executive, as I have decided to step down. After more than seven years on the committee, it is time to move on to other activities.

During my time with AUUG, I've seen a great many changes, from Conferences held in a lecture theatre in Sydney Uni, through to Australia's Premier Open Systems Event, from an organisation of a few hundred techies to one of over a thousand members from all areas of the computing industry.

So, back to history, I first joined AUUG around 1984 (not sure of the exact date), was elected to the committee in 1988, lost the following year, and then reelected from 1990 through to the present. I was also Treasurer from 1991 to 1995, and boy wasn't I glad to hand it over to Stephen Boucher.

Aside from my commitments on the Exec, I've also been involved in numerous other activities with AUUG, from the original ACSnet SIG (i.e. the precursor to AARNet) in 1988, to AUUG-AARNet Co-ordinator (i.e. paper shuffler) and now to the Co-ordinator of the AUUG Column in the Tuesday's Australian. At the "local" level, I am the current Treasurer of the NSW Chapter.

However, despite all that, the most fun I've had with AUUG has been related to publications and presentations. I've been involved with the production of AUUGN since 1991, and currently I'm acting as

the Book Review Editor. Most of all I like writing and presenting, having presented both papers and tutorials at many National and Chapter conferences in the past 10 years. This has even lead to co-authoring a book with Berny Goodheart, OzInternet, which now seems to be out of print.

Outside of AUUG, I'm heavily involved in the System Administrators Guild of Australia (SAGE-AU), and a supporter of ISOC-AU. Maybe with my reduced commitments to AUUG, I'll get more time for them. Even further afield, I work for ANSTO, as a System Administrator (although my current title is Site Systems Manager).

Finally, and most importantly, I'm married with two young girls, who are now getting me involved in all sorts of school activities, and local committees, so I don't have any free time.

Over the years, I've met many interesting people through AUUG, most of whom I still contact from time to time, but more especially, I've found that the old adage that you only get out of an organisation as much as you put in, is very true. I've enjoyed most of the activities I've been involved in, and will continue to do so for some time yet. After all, I'm only standing down from the Executive, not from my other activities, and I expect all the rest of the members to support me by writing heaps.

❖



---

# AUUG News: New South Wales

Peter Chubb <peterc@sw.oz.au>  
Secretary AUUG NSW

Minutes of the AUUG NSW Chapter AGM held 20 Mar 1997 at the Occidental Hotel, 43 York St., Sydney, NSW

Meeting commenced at 7:10pm after some unofficial discussion as we tried to remember how many we needed for a quorum.

Present were:

Catherine Allen, Wayne Bell (in the chair), Charlie Brady, Peter Chubb, Paul Colquhoun, Frank Crawford (taking minutes), Peter Gray, David Herd, Peter LayMan (??), Robin McDonald, Glenn Rickersley, Regan Russell and Pauline van Winsen

Apologies were received from:

John O'Brian, Chris Mugden, Lucy Chubb, and Matthew Dawson.

The President, Wayne Bell, gave his report. He briefly summarised the meetings held during 1996, and expressed his thanks to the outgoing committee, especially to the two members co-opted to the committee last November --- Catherine Allen, for her work on the Summer Conference, and Matthew Dawson for his work on the AUUG-NSW web site.

MOTION: that the President's report be accepted. PvW/PC -- Carried.

The secretary David Purdue being absent, an unofficial report was given by Frank Crawford and Catherine Allen:

- We have around 350 members, spread fairly evenly though all postcode areas, except for some higher numbers in the city centre, North Sydney and North Ryde.

The minutes of the last AGM are unavailable at this time.

The Treasurer Frank Crawford gave his report:

- There have been changed financial arrangements with AUUG-national. This mainly affects the way that banking is done, rather than the amount of funds available.
- The exact amount in the AUUG Westpac account at the time of the AGM was \$3976.30, with

expenses outstanding of \$478.53, giving a total balance of \$3497.77. It is unclear as to the exact amount in the Advance Account, and won't be until it is closed.

MOTION: that the Treasurer's report be received: PvW/PC -- Carried.

The President then vacated the chair, and a returning officer (Peter Chubb) and an assistant returning officer (Pauline van Winsen) volunteered from the floor.

After some friendly coercion the following people were nominated:

POSITION	Who	Nom/Sec
President	Peter Gray	Peter Chubb/ Wayne Bell
Secretary	Peter Chubb	Self/ Charlie Brady
Treasurer	Frank Crawford	Charlie Brady/ Catherine Allen
Committee	Paul Colquhoun Wayne Bell	Self /Catherine Allen Peter Gray/ Frank Crawford

There being exactly one nomination per position, the returning officer declared all nominees elected unopposed.

The new president, Peter Gray, then took the chair.

General Business:

A general discussion followed, initiated by Pauline van Winsen, on how to increase the membership's involvement. Points made (in no particular order, and sorry I can't remember who said what) included:

- Changing the venue.
  - Venue requirements include
    - Smoke Free!
    - Handy for Public Transport, and with parking
    - Internet access, good speaker's facilities
    - In the city.
- Changing the time
  - Maybe a lunchtime or breakfast meeting?
- Getting overseas speakers
  - A bit expensive except as a once-off -- and then what do you follow it up with?
- Getting interstate speakers -- swapping with other chapters.
- Getting the vendors to provide speakers, and perhaps sponsorship.

There were comments on the interrelationship between AUUG-NSW and AUUG\_national at this point. It was agreed that Liz Egan, the AUUG Business manager, should be asked about this matter.

- Improved communication with members.
- The mailout often costs more than the meeting, so is restricted to once a quarter.
- The email reminder is meant to be just that -- so should be timely. It could be sent earlier?
- AUUGN is cheaper for us -- we should take out a full-page ad to give our programme.
- Increased Membership
- Membership drive
- Talk to Unis about student memberships
- Talk to SLUG and Sage
- Summer Conference -- There's still a desire to hold something, maybe a shoe-string one-day affair where AUUG members get in free if they bring a non-member (to tie in with the membership drive).

The meeting closed at 7:55pm, and almost everyone went to the Bowlers Club for a meal. Dress regulations there meant that they lost our custom, and we proceeded to a Korean Restaurant in Clarence St., where the food was very good and very cheap.

❖

---

## AUUG News: Queensland

Mark White <m.white@pacific.tandem.com>

[ *Editor's note: this report was submitted prior to the advertised deadline. I need to acknowledge this, as noone will believe me otherwise! Well done! Mark ;-)* ]

A double-booking at the Inn on the Park forced the March Chapter meeting back to the Regatta Hotel - our old stomping ground. Despite the (now) non-regular venue a fair-sized crowd was present, indicating that most caught the news about the venue swap. To anyone who missed it - our apologies, the monthly meetings will return to the Inn on the Park (507 Coronation Drive, Milton) on Tuesday 27th May at 7:00pm.

Our topic for the evening was security, and guest speaker Brian Meilak gave a solid overview of PGP, SSH, and SSL. As Senior Systems Programmer for Queensland University of Technology Brian has to use and maintain these and other security packages

and protocols, and this experience was reflected in his presentation.

There was much discussion during and after the presentation over the various licensing mechanisms for security products (eg what precisely constitutes "commercial use"?) as well as the different versions of these products required to comply with the United States' cryptography export restrictions. Several other experienced security types were present in the audience (most notably Tim Hudson and Gary Gaskell) which made for a rich and informative forum.

The next scheduled event for AUUG's Queensland Chapter is of course the Chapter Technical conference, to be held on Thursday 26th April (from 8:30am) at Brisbane's Park Royal Hotel, Alice Street, Brisbane. The Queensland Committee has been furiously planning this event for the past several months and have assembled a great line up of speakers, including Bill Segall, David Hughes, and Pauline Van Winsen. The Keynote Address will be delivered by David Barbagello, CEO of DSTC Pty Ltd. The Queensland CTC has been a tremendous success in recent years, and we hope this to continue in 1997.

Due to the CTC, no April Chapter meeting will be held. Instead, the next meeting will be on Tuesday, 27th May from 7pm at the Inn on the Park. Both AUUG members and non-members are welcome to partake in discussions, presentations, networking and, of course, free beer and munchies. We normally send out meeting notifications a week or two before the meetings, via an e-mail list and also by facsimile. If you'd like to receive e-mail notifications of forthcoming meetings and AUUG Queensland news, send a message to majordomo@auug.org.au with the message body containing "subscribe qauug <your.email.address>". To receive facsimile notifications, call either Rick Stevenson (3270 4242) or Mark White (3218 3618).

Finally, we're always looking for presenters for the monthly meetings, so if you have something interesting to say, promote, or have just done some funky hacks at work lately we'd love to hear from you. Drop us some e-mail at <qauug-exec@auug.org.au>.

❖

[ Editor's note: for purchasing information call the AUUG Secretariat on 02 9361 5994. ]

## AUUGN Back Issues

Year	Issue	No. available
1989	Feb	3
	June	1
	Dec	1
1990	Feb	1
	April	1
	June	1
	Dec	1
1991	Feb	5
	Oct	3
	Dec	3
1992	Oct	3
	Dec	3
1993	April	3
	June	3
	August	3
	Oct	3
1994	Feb	3
	April	3
	June	3
	August	3
	Oct	3
1995	Feb	3
	April	9
	June	2
	Oct	4
1996	Feb	4
	April	5
	June	1
	August	30
	Oct/Nov	30

❖

[ Editor's note: somehow this was omitted from the last issue. My special thanks to Stephen Prince for all his efforts and for making it very easy for me to add this to AUUGN. ]

This index has now been added to the AUUG web page, where you can search through Volumes 4 through 17! Special thanks again to Stephen for his efforts. Check out the index on the AUUG web page <<http://www.auug.org.au>> for more details. ]

## AUUGN Volume 17 Index Papers (Sorted by title)

Compiled by: Stephen Prince  
<[S.Prince@clcs.com.au](mailto:S.Prince@clcs.com.au)>

Crawford, Frank, "AUUG96 Conference & Exhibition," AUUGN, vol. 17, no. 4, pp. 12-13, AUUG Inc., Melbourne, VIC, August 1996.

Winsen, Pauline van, "AUUG Key Signing Service," AUUGN, vol. 17, no. 5-6, p. 14, AUUG Inc., Sydney, NSW, October 1996.

Ranum, Marcus J., "Can existing security solutions scale cost effectively?," AUUGN, vol. 17, no. 3, p. 13, AUUG Inc., Melbourne, VIC, June 1996.

McCrea, Phil, "Client-Server Internet," AUUGN, vol. 17, no. 2, p. 12, AUUG Inc., Melbourne, VIC, April 1996.

Crawford, Frank, "Comments on the Future of the Internet," AUUGN, vol. 17, no. 3, p. 12, AUUG Inc., Melbourne, VIC, June 1996.

Crawford, Frank, "Computer Room Archaeology," AUUGN, vol. 17, no. 1, p. 24, AUUG Inc., Melbourne, VIC, February 1996.

Crawford, Frank, "Do Non-Technical Managers Work?," AUUGN, vol. 17, no. 4, pp. 11-12, AUUG Inc., Melbourne, VIC, August 1996.

Crawford, Frank, "The End of AUSCERT?," AUUGN, vol. 17, no. 4, pp. 5-6, AUUG Inc., Melbourne, VIC, August 1996.

McCrea, Phil, "Formation of 'The Open Group,'" AUUGN, vol. 17, no. 2, pp. 17-18, AUUG

- Inc., Melbourne, VIC, April 1996. Email: philip.mccrea@syd.dit.csiro.au
- Crawford, Frank, "**Formation of ISOC-AU**," AUUGN, vol. 17, no. 5-6, p. 7, AUUG Inc., Sydney, NSW, October 1996.
- Crawford, Frank, "**How Do You Make Use of Petaflops?**," AUUGN, vol. 17, no. 5-6, pp. 14-16, AUUG Inc., Sydney, NSW, October 1996.
- Crawford, Frank, "**How Long Does Security Last**," AUUGN, vol. 17, no. 3, pp. 13-14, AUUG Inc., Melbourne, VIC, June 1996.
- Allen, Catherine, "**How to Handle Spamming**," AUUGN, vol. 17, no. 5-6, pp. 16-17, AUUG Inc., Sydney, NSW, October 1996.
- Egan, Liz, "**Images from AUUG '96**," AUUGN, vol. 17, no. 5-6, pp. 40-42, AUUG Inc., Sydney, NSW, October 1996.
- Whittle, Robin, "**Internet Censorship**," AUUGN, vol. 17, no. 3, pp. 15-16, AUUG Inc., Melbourne, VIC, June 1996.
- McCrea, Phil, "**Internet Industry**," AUUGN, vol. 17, no. 4, pp. 6-7, AUUG Inc., Melbourne, VIC, August 1996.
- Winsen, Pauline van, "**Internet Security - what technology can't solve!**," AUUGN, vol. 17, no. 5-6, pp. 17-18, AUUG Inc., Sydney, NSW, October 1996.
- Crawford, Frank, "**Is a Standard UNIX a Good Idea?**," AUUGN, vol. 17, no. 3, p. 11, AUUG Inc., Melbourne, VIC, June 1996.
- Crawford, Frank, "**Is the Internet Anti-Commercial?**," AUUGN, vol. 17, no. 4, pp. 10-11, AUUG Inc., Melbourne, VIC, August 1996.
- Flanagan, David, "**Java Enters New Evolutionary Phase**," AUUGN, vol. 17, no. 4, p. 5, AUUG Inc., Melbourne, VIC, August 1996.
- Crawford, Frank, "**Linux - The Choice of a New Generation**," AUUGN, vol. 17, no. 1, p. 20, AUUG Inc., Melbourne, VIC, February 1996.
- Rose, Greg, "**A Major Contributor**," AUUGN, vol. 17, no. 2, p. 9, AUUG Inc., Melbourne, VIC, April 1996.
- Kolstad, Rob, "**Modern Operating Systems Unleash Pentium Hardware's Power**," AUUGN, vol. 17, no. 5-6, pp. 18-19, AUUG Inc., Sydney, NSW, October 1996.
- Egan, Liz, "**More Conference Pix!**," AUUGN, vol. 17, no. 3, p. 7, AUUG Inc., Melbourne, VIC, June 1996.
- Stock, Andrew Van der, "**A Moving Story**," AUUGN, vol. 17, no. 1, p. 19, AUUG Inc., Melbourne, VIC, February 1996.
- Purdue, David, "**Name.Space vs DNS**," AUUGN, vol. 17, no. 5-6, pp. 19-20, AUUG Inc., Sydney, NSW, October 1996.
- McCrea, Phil, "**The Network Computer**," AUUGN, vol. 17, no. 4, pp. 9-10, AUUG Inc., Melbourne, VIC, August 1996.
- Chubb, Lucy, "**On Tour: Security Seminar**," AUUGN, vol. 17, no. 5-6, p. 8, AUUG Inc., Sydney, NSW, October 1996.
- McCrea, Phil, "**Paying on the Net**," AUUGN, vol. 17, no. 3, pp. 16-17, AUUG Inc., Melbourne, VIC, June 1996.
- Paddon, Michael, "**The Physics of Java**," AUUGN, vol. 17, no. 4, pp. 7-8, AUUG Inc., Melbourne, VIC, August 1996.
- Sheehan, Kevin, "**Preface to 'Why aren't you using mmap() yet?'**," AUUGN, vol. 17, no. 1, p. 6, AUUG Inc., Melbourne, VIC, February 1996. Email: kevin@uniq.com.au
- Crawford, Frank, "**Presenting at AUUG96 Conference**," AUUGN, vol. 17, no. 3, p. 10, AUUG Inc., Melbourne, VIC, June 1996.
- Stock, Andrew van der, "**Professionalism**," AUUGN, vol. 17, no. 2, p. 16, AUUG Inc., Melbourne, VIC, April 1996.
- AUUG, "**Rules of AUUG Incorporated**," AUUGN, vol. 17, no. 1, pp. 36-41, AUUG Inc., Melbourne, VIC, February 1996.
- <unknown>, "**Santa Claus Is Coming To Town in C-Shell**," AUUGN, vol. 17, no. 5-6, p. 39, AUUG Inc., Sydney, NSW, October 1996.
- Salter-Duke, Brian, "**Scientific computing: A forgotten discipline?**," AUUGN, vol. 17, no. 5-6, pp. 13-14, AUUG Inc., Sydney, NSW, October 1996.
- Winsen, Pauline van, "**Searching for needles in haystacks: an Internet detective story**,"

AUUGN, vol. 17, no. 5-6, pp. 12-13, AUUG Inc., Sydney, NSW, October 1996.

Purdue, David, "**Standard API for Windows,**" AUUGN, vol. 17, no. 2, p. 11, AUUG Inc., Melbourne, VIC, April 1996.

McCrea, Phil, "**Standards Groups do not stand still,**" AUUGN, vol. 17, no. 5-6, p. 11, AUUG Inc., Sydney, NSW, October 1996.

McCrea, Phil, "**Summary of IDC Industry Briefing,**" AUUGN, vol. 17, no. 3, pp. 14-15, AUUG Inc., Melbourne, VIC, June 1996.

Egan, Liz, "**Summer Conference Reflections,**" AUUGN, vol. 17, no. 2, pp. 5-8, AUUG Inc., Melbourne, VIC, April 1996.

Crawford, Frank, "**UniForum NZ'96,**" AUUGN, vol. 17, no. 3, pp. 17-18, AUUG Inc., Melbourne, VIC, June 1996.

Feuereisen, Gunther, "**UNIX Tricks & Traps - Debugging Ports: A Daemon of a Problem,**" AUUGN, vol. 17, no. 3, pp. 19-21, AUUG Inc., Melbourne, VIC, June 1996. Email: [gunther@agsm.unsw.edu.au](mailto:gunther@agsm.unsw.edu.au)

Huxtable, Glenn, "**UNIX Tricks & Traps - Getting the exit status of any command in a pipeline,**" AUUGN, vol. 17, no. 1, p. 25, AUUG Inc., Melbourne, VIC, February 1996. Email: [glenn@fs.com.au](mailto:glenn@fs.com.au)

Feuereisen, Gunther, "**UNIX Tricks & Traps: How to move or copy large volumes of data - The tar(1) command,**" AUUGN, vol. 17, no. 4, pp. 15-18, AUUG Inc., Melbourne, VIC, August 1996. Email: [gunther@agsm.unsw.edu.au](mailto:gunther@agsm.unsw.edu.au)

Turnbull, Peter, "**UNIX Tricks & Traps: Tar - The Story Continues,**" AUUGN, vol. 17, no. 5-

6, pp. 38-39, AUUG Inc., Sydney, NSW, October 1996. Email: [petert@daedalius.com.au](mailto:petert@daedalius.com.au)

Feuereisen, Gunther, "**UNIX Tricks & Traps: TR(1) - More than just a two letter acronym,**" AUUGN, vol. 17, no. 4, p. 15, AUUG Inc., Melbourne, VIC, August 1996. Email: [gunther@agsm.unsw.edu.au](mailto:gunther@agsm.unsw.edu.au)

Marsh, Arthur, "**Website 1.0/1.1 for Windows NT,**" AUUGN, vol. 17, no. 1, pp. 4-5, AUUG Inc., Melbourne, VIC, February 1996. Email: [arthur@gateway.dircsa.org.au](mailto:arthur@gateway.dircsa.org.au)

Crawford, Frank, "**What is a Network?,"** AUUGN, vol. 17, no. 1, p. 23, AUUG Inc., Melbourne, VIC, February 1996.

Crawford, Frank, "**When will NT learn?,"** AUUGN, vol. 17, no. 2, p. 14, AUUG Inc., Melbourne, VIC, April 1996.

Sheehan, Kevin, "**Why aren't you using mmap() yet?,"** AUUGN, vol. 17, no. 1, pp. 7-17, AUUG Inc., Melbourne, VIC, February 1996. Email: [kevin@uniq.com.au](mailto:kevin@uniq.com.au)

McCrea, Phil, "**Will the real Information SuperHighway please stand up!,"** AUUGN, vol. 17, no. 1, p. 21, AUUG Inc., Melbourne, VIC, February 1996.

Crawford, Frank, "**Wither Go HPC?,"** AUUGN, vol. 17, no. 3, pp. 9-10, AUUG Inc., Melbourne, VIC, June 1996. Email: [frank@ansto.gov.au](mailto:frank@ansto.gov.au)

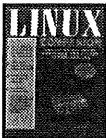
Crawford, Frank, "**A Wrapup of SAGE-AU 96,"** AUUGN, vol. 17, no. 4, pp. 8-9, AUUG Inc., Melbourne, VIC, August 1996.

❖

# AUUG BOOK CLUB & PRENTICE HALL AUSTRALIA



**UNIX System Programming Using C++** focuses on real-life challenges in developing network and client/server applications, databases, compilers, operating systems and CAD systems.



**Linux Companion: The Essential Guide for Users and System Administrators** explains Linux from beginning to end, showing how it compares to DOS and Windows, how to install and use it and how to set up and administer a Linux multi-user system.



**Program Development in UNIX: Case Study Approach**  
How to improve your productivity as a UNIX programmer by knowing the tools available to UNIX programmers and identifying those programming improvements that will give you the most cost benefit.



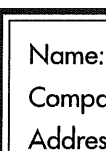
**UNIX Shells By Example.** The only book that covers all the important UNIX utilities in one neat package. The CD ROM contains all example programs, a repository of additional source code and a suite of shell programming utilities.



**LINUX A-to-Z** - for any computer-literate person, this will give all the information needed to make the most of LINUX, the freely available UNIX variant that is skyrocketing in popularity.



**SCO Companion: The Essential Guide for Users and System Administrators.** A comprehensive trouble shooting guide which answers the most commonly asked questions and concerns of SCO users.



**SCO System Administration** - the complete guide to administering the SCO UNIX system - the leading X86 UNIX system.

## 20% DISCOUNT TO AUUG MEMBERS

Please send me a copy/copies of the following books -

<input type="checkbox"/>	Unix System Programming Using C++ 0133315622	Paper	<del>\$49.95</del>	\$39.95
<input type="checkbox"/>	SCO Companion 0134516834	Paper	<del>\$69.95</del>	\$55.95
<input type="checkbox"/>	UNIX Shells by Example 0134608666	Paper	<del>\$69.95</del>	\$55.95
<input type="checkbox"/>	LINUX Companion 0132318385	Paper	<del>\$39.95</del>	\$31.95
<input type="checkbox"/>	Program Development in UNIX 0132373971	Paper	<del>\$49.95</del>	\$39.95
<input type="checkbox"/>	SCO System Administrator 0134436644	Paper	<del>\$69.95</del>	\$55.95
<input type="checkbox"/>	LINUX A-to-Z 0132347091	Paper	<del>\$69.95</del>	\$55.95

AUUG members receive 20% discount below recommended retail price . Please send me the selected book/s on 30-Day approval

Name: \_\_\_\_\_ Position: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

Telephone: \_\_\_\_\_

**PLEASE SEND MY  
BOOK/S ON  
30-DAY APPROVAL**

Enclosed cheque for \$ \_\_\_\_\_ (Payable to 'Prentice Hall Australia')

Charge to me **OR**  Company purchase Order No. \_\_\_\_\_

Please charge my:  Bankcard  Visa  MasterCard  AMEX

Expiry Date: \_\_\_\_\_ Credit Card No:

Signature: \_\_\_\_\_



**PRENTICE HALL AUSTRALIA**  
14 Aquatic Drive, Frenchs Forest NSW 2086  
Tel: (02) 9454 2200 • Fax: (02) 9453 0117 A.C.N. 000 383 406

# AUUGN Volume 17 Index

## Book Reviews (Sorted by author)

Compiled by: Stephen Prince <S.Prince@clcs.com.au>

Author	Title	Reviewer	Issue
Aspinwell, Jim	IRQ, DMA and I/O (resolving conflicts and preventing PC system conflicts)	Dave Horsfall	1
Barrett, Daniel J.	Bandits On The Information Superhighway	Alex Kowalenko	4
Bradner, Scott and Allison Mankin	IPng: Internet Protocol Next Generation	Kate Lance	2
Brooks, Frederick P. Jr.	The Mythical Man-Month: 20th Anniversary Edition	Adrian Booth	1
Campbell, Dave and Mary	The Student's Guide to Doing Research on the Internet	Jon Wright	1
December, John	Presenting Java: An Introduction to Java and HotJava		4
Drake, Chris and Kimberly Brown	PANIC! Unix System Crash Dump Analysis	David Denton	1
Flanagan, David	Java In A Nutshell	Peter Gray	5-6
Freeman, Adam and Darrel Ince	Active Java: Object-Oriented Programming For The World Wide Web	Danny Yee	5-6
Huitema, Christian	IPv6: The New Internet Protocol	Mark Delany	3
J.Perry, Paul	Creating Cool Web Applets With Java		5-6
Kline, Jennifer	Managing & Using UUCP	Paul Burgess	2
Kopka, Helmut and Patrick W Daly	A Guide to LaTeX 2e: Document Preparation for Beginners and Advanced Users	Jagoda Crawford	5-6
Leben, James and Martin, Joe	Client/Server Databases: Enterprise Computing	Greg Black	3
Lehey, Greg	Porting UNIX Software	Glenn Huxtable	1
Lemay, Laura and Charles L. Perkins	Teach Yourself Java in 21 Days	Peter Gray	4
Lions, John	Lion's Commentary on UNIX 6th Edition With Source Code	Lucy Chubb	5-6
Lowe, Sue	On-Line in Oz	Michi Henning	2
Martin, James, Kathleen Kavanagh Chapman, and Joe Leben	Enterprise Networking: Data Link Subnetworks	Peter Lane	4
Nelson, Stephen	Field Guide to the Internet with Windows 95 (quick, easy answers)	Catherine Allen	2
Niemeyer, Patrick and Joshua Peck	Exploring Java	Danny Yee	5-6
Nye, Adrian	Volume 0: X Protocol Reference Manual	John Chalk	3
P.J.Plauger, and Jim Brodie	Standard C: A Reference	Jamie Honan	3
P.Kehoe, Brendan	Zen And The Art Of The Internet (4th Ed)	Jon Wright	5-6
Pendergrast, Stephen Jr.	Desktop Kornshell Graphical Programming	Jamie Honan	1
Perry, Paul J.	World Wide Web Secrets	Andrew Wenn	1

Pesce, Mark	VRML: Browsing And Building Cyberspace	Kirk Barrett	5-6
Peterson, Larry L. and Bruce S. Davie	Computer Networks: A Systems Approach	Danny Yee	5-6
Prochak, Michael	On The Road - Pervasive Portable Computing: PowerBooks, PDAs, and Beyond	Adrian Booth	2
Ritchey, Tim	Programming With Java!	Lawrie Brown	4
Rose, Marshall T.	The Simple Book: An Introduction to Network Management	Michael Haldey	4
Siyan, Karanjit and Chris Hare	Internet Firewalls and Network Security	Craig Macbride	5-6
Vahalia, Uresh	UNIX Internals: The New Frontiers	Warren Toomey	3
Vahalia, Uresh	UNIX Internals: The New Frontiers	Michael Usher	4
Vormans, Johan	Perl 5 Desktop Reference	Frank Crawford	4
Waters, Frank	AIX Performance Tuning	Matthew See	3




---

## AUUGN Volume 17 Index Reports (Sorted by author)

Compiled by: Stephen Prince <S.Prince@clcs.com.au>

Author	Title	Issue	Page	Month
Barlow, Jon	Canberra Chapter: June 96 Newsletter	3	31	June
Barlow, Jon	Canberra Chapter	1	35	February
Bell, Wayne	News from the NSW Chapter	4	29	August
Bell, Wayne	AUUG News: NSW Chapter	5-6	35	October
Bishop, Jeremy	Canberra Chapter: September 96 newsletter	4	28	August
Brown, Michael	Tasmanian Chapter	2	26	April
Caldwell, Malcolm	AUUG News: Darwin	5-6	34-35	October
Egan, Liz	Images from the Chapter Council Meeting	5-6	37	October
Hallam, Tom	From the Western Front	2	25	April
Hallam, Tom	From the Western Front	4	27	August
Hallam, Tom	From the Western Front	1	34	February
Hallam, Tom	From the Western Front	3	29-30	June
White, Mark	Queensland Chapter	2	26	April
Wright, John	AUUG News: Canberra	5-6	34	October





AUUG Inc is the Australian UNIX and Open Systems User Group, providing users with relevant and practical information, services and education through co-operation among users.

**AUUG OFFERS SOMETHING FOR YOU!!**

**AUUGN**  
*Technical Newsletter*  
 AUUG's bi-monthly publication, keeping you up to date with the world of UNIX and open systems.

**Education**  
 Tutorials  
 Workshops

**Events.....Events.....Events**

- Annual Conference & Exhibition
- Overseas Speakers • Local Conferences
- Roadshows • Monthly meetings

**DISCOUNTS**  
 to all AUUG events and education.  
 Reciprocal arrangements with overseas affiliates.  
 Discounts with various internet service providers, software, publications and more...!!

**Connections**  
 • Newsgroup  
 aus.org.auug

# Application for Individual or Student Membership

**Section A: PERSONAL DETAILS**

Surname \_\_\_\_\_ First Name \_\_\_\_\_  
 Title: \_\_\_\_\_ Position \_\_\_\_\_  
 Organisation \_\_\_\_\_  
 Address \_\_\_\_\_  
 Suburb \_\_\_\_\_ State \_\_\_\_\_ Postcode \_\_\_\_\_  
 Telephone: Business \_\_\_\_\_ Private \_\_\_\_\_  
 Facsimile: \_\_\_\_\_ E-mail \_\_\_\_\_

**Section B: MEMBERSHIP INFORMATION**

Please indicate whether you require Student or Individual Membership by ticking the appropriate box.

**RENEWAL/NEW INDIVIDUAL MEMBERSHIP**  
 Renewal/New Membership of AUUG  \$100.00

**RENEWAL/NEW STUDENT MEMBERSHIP**  
 Renewal/New Membership of AUUG (Please complete Section C)  \$25.00

SURCHARGE FOR INTERNATIONAL AIR MAIL  \$60.00

Rates valid as at 07/96

**Section F: PAYMENT**

Cheques to be made payable to **AUUG Inc**  
*(Payment in Australian Dollars only)*

*For all overseas applications, a bank draft drawn on an Australian bank is required. Please do not send purchase orders.*

-OR-

Please debit my credit card for A\$ \_\_\_\_\_  
 Bankcard  Visa  Mastercard

Name on Card \_\_\_\_\_  
 Card Number \_\_\_\_\_  
 Expiry Date \_\_\_\_\_  
 Signature \_\_\_\_\_

Please mail completed form with payment to: Or Fax to:  
 Reply Paid 66 AUUG Inc  
 AUUG Membership Secretary (02) 9332-4066  
 PO Box 366  
 KENSINGTON NSW 2033  
 AUSTRALIA

**Section C: STUDENT MEMBER CERTIFICATION**

For those applying for Student Membership, this section is required to be completed by a member of the academic staff.

I hereby certify that the applicant on this form is a full time student and that the following details are correct.

NAME OF STUDENT: \_\_\_\_\_  
 INSTITUTION: \_\_\_\_\_  
 STUDENT NUMBER: \_\_\_\_\_  
 SIGNED: \_\_\_\_\_  
 NAME: \_\_\_\_\_  
 TITLE: \_\_\_\_\_  
 DATE: \_\_\_\_\_

**Section G: AGREEMENT**

I agree that this membership will be subject to rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.

Signed: \_\_\_\_\_  
 Date: \_\_\_\_\_

**Section D: LOCAL CHAPTER PREFERENCE**

By default your closest local chapter will receive a percentage of your membership fee in support of local activities. Should you choose to elect another chapter to be the recipient please specify here:

\_\_\_\_\_

**Section E: MAILING LISTS**

AUUG mailing lists are sometimes made available to vendors. Please indicate whether you wish your name to be included on these lists:

Yes  No

**AUUG Secretariat Use**

Chq: bank \_\_\_\_\_ bsb \_\_\_\_\_  
 A/C: \_\_\_\_\_ # \_\_\_\_\_  
 Date: \_\_\_\_\_ \$ \_\_\_\_\_  
 Initial: \_\_\_\_\_ Date Processed: \_\_\_\_\_  
 Membership#: \_\_\_\_\_

# Application for Institutional Membership

## Section A: MEMBER DETAILS

The primary contact holds the full member voting rights and two designated representatives will be given membership rates to AUUG activities including chapter activities. In addition to the primary and two representatives, additional representatives can be included at a rate of \$70 each. Please attach a separate sheet with details of all representatives to be included with your membership.

NAME OF ORGANISATION: \_\_\_\_\_

### Primary Contact

Surname \_\_\_\_\_ First Name \_\_\_\_\_  
 Title: \_\_\_\_\_ Position \_\_\_\_\_  
 Address \_\_\_\_\_  
 Suburb \_\_\_\_\_ State \_\_\_\_\_ Postcode \_\_\_\_\_  
 Telephone: Business \_\_\_\_\_ Facsimile \_\_\_\_\_  
 Email \_\_\_\_\_ Local Chapter Preference \_\_\_\_\_

## Section B: MEMBERSHIP INFORMATION

Renewal/New Institutional Membership of AUUG  \$390.00  
 Surcharge for International Air Mail  \$120.00  
 Additional Representatives Number  @ \$80.00

Rates valid as at 07/96

## Section C: PAYMENT

Cheques to be made payable to **AUUG Inc** (Payment in Australian Dollars only)

*For all overseas applications, a bank draft drawn on an Australian bank is required.  
Please do not send purchase orders.*

-OR-

Please debit my credit card for A\$ \_\_\_\_\_  
 Bankcard  Visa  Mastercard

Name on Card \_\_\_\_\_  
 Card Number \_\_\_\_\_  
 Expiry Date \_\_\_\_\_  
 Signature \_\_\_\_\_

Please mail completed form with payment to: \_\_\_\_\_ Or Fax to: \_\_\_\_\_

Reply Paid 66 AUUG Inc  
 AUUG Membership Secretary (02) 9332-4066  
 PO Box 366  
 KENSINGTON NSW 2033

## Section D: MAILING LISTS

AUUG mailing lists are sometimes made available to vendors. Please indicate whether you wish your name to be included on these lists:

Yes  No

## Section E: AGREEMENT

*I/We agree that this membership will be subject to rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.*

*I/We understand that I/we will receive two copies of the AUUG newsletter, and may send two representatives to AUUG sponsored events at member rates, though I/we will have only one vote in AUUG elections, and other ballots as required.*

Signed: \_\_\_\_\_  
 Title: \_\_\_\_\_  
 Date: \_\_\_\_\_

### AUUG Secretariat Use

Chq: bank \_\_\_\_\_ bsb \_\_\_\_\_  
 A/C: \_\_\_\_\_ # \_\_\_\_\_  
 Date: \_\_\_\_\_ \$ \_\_\_\_\_  
 Initial: \_\_\_\_\_ Date Processed: \_\_\_\_\_  
 Membership#: \_\_\_\_\_



UNIX® AND OPEN SYSTEMS USERS

# Membership Application

### AUUG Inc Secretariat

PO Box 366, Kensington NSW 2033, Australia

Tel: (02) 9361 5994  
 Free Call: 1 800 625 655  
 Fax: (02) 9332 4066

email: [auug@auug.org.au](mailto:auug@auug.org.au)

ACN A00 166 36N (incorporated in Victoria)

<http://www.auug.org.au>