

Manitoba UNIX® User Group

MUUG Lines

Newsletter of the Manitoba UNIX® User Group

GroupKit - A Groupware Toolkit

By Mark Roseman — The University of Calgary

Groupware — A term for software which runs on a local area network and which allows people on the network (typically a team) to participate in a joint (often complex) project.

GroupKit is a freely-available groupware toolkit developed at the University of Calgary. It is used for developing real-time conferencing applications. These are groupware applications such as drawing tools, editors and meeting tools that are shared simultaneously among several users.

GroupKit runs on Unix workstations under the X11 window system. It is based on the Tcl/TK/Tcl-DP packages from the University of California at Berkeley, providing an easy to use but extensible interpreted environment for developing groupware. GroupKit has primarily been used as a research tool for prototyping groupware systems and investigating multi-user architectures and interaction techniques. It has also been used as a teaching tool in graduate level courses.

GroupKit's goal is to make developing groupware applications "only slightly harder" than single-user applications. At the same time, providing facilities — motivated by human factors work — to build features important to end users of groupware applications.

Applications built with GroupKit

GroupSketch — a multi-user stroke-based paint system. People can draw simultaneously, and multiple cursors are displayed.

FileViewer — lets people simultaneously scroll through and view a file.

BrainStorm — a simple brainstorming tool that lets people enter ideas (typed on the bottom onto a shared common list (seen on the top)).

Software Availability

GroupKit is available via anonymous ftp from `<ftp.cpsc.ucalgary.ca>` in the directory `/pub/projects/grouplab/software`. The following version is currently available (older ones are also available in the directory):

`groupkit-3.0.tar.Z`

The distributions contain a set of manual pages are a Postscript tutorial. Many simple groupware examples are enclosed as well.

This version requires TCL 7.3, TK 3.6, and TCL-DP 3.2. TCL and TK were developed by John Ousterhout (U. California Berkeley), now at Sun. TCL-DP was developed by Larry Rowe, Brian Smith et al, also from UCB. All three packages are available at many major ftp sites.

There are also many books, FAQs and PostScript documents available if you would like to learn more about any of these packages.

If you'd like to run GroupKit 3.0 with the beta releases of Tcl7.4 and Tk4.0, there is a patch to GroupKit available.

Mailing List

There is a mailing list for GroupKit users, `<groupkit-users@cpsc.ucalgary.ca>`. To join, just send a note. ➡

This Month's Meeting

Meeting Location:

Our June meeting is scheduled for Tuesday, June 13, at 6:30 PM. This meeting will be the traditional MUUG June BBQ. This year, Bary Finch will host the BBQ at his house on Manahan. A map is included in this month's newsletter.

Meeting Agenda:

Eat, drink, and be merry — but no computer talk!

Inside This Issue

Newsletter Editor's Ramblings	2
President's Corner	3
C++ Q & A	4
Ethernet Q & A	5
SIG Sideline	6
June 13th BBQ Map	6

Hacker Culture

By Andrew Trauzzi

RALEIGH (10 p.m.) — Kevin Mitnick was shackled as he came into U.S. District Court on Friday, but the case against the alleged computer hacker might be a good deal more difficult to lock up.

Proving Mitnick was behind the raids on data banks and thefts of at least 20,000 credit card numbers from computer systems across the nation will be a special challenge involving retracing the alleged hacker's steps in cyberspace.

And while federal authorities are relatively inexperienced in the complicated machinations of computer burglars, they are gaining expertise.

Dan Boyce, a former U.S. attorney who has started a private law practice in Raleigh specializing in computer crimes, said law enforcement agencies are unprepared to deal with the growing problem that sophisticated hackers present.

"Computer crime is the crime of the '90s," Boyce said. "You've heard the old expression that it's easier to rob a bank with a pen than with a gun, and now it's even easier to rob a bank with a computer because of the ability to gain access to information without being noticed."

Have a few hackers given the whole hacker community a bad name, or are hackers a malicious bunch? UNIX itself was created because a hacker became disgruntled with the state of operating systems! Jobs and Woz (the founders of Apple Computer), were making and selling black and blue boxes out of their garage before their first computer. Almost all early (and most current) video games were created because of hackers.

The motto that good hackers follow is: "Never do things the way that everyone else does them." Bad hackers will do anything that makes them a buck or, more importantly, makes them famous within the hacker community. This desire to "get a rep" can be paralleled with street gangs' desire to be on the front page of the paper.

The Internet has proliferated the spread of illegal and immoral information to good and bad hackers alike. In many crimes nowadays, the Police will claim that the criminals downloaded criminal information off the Internet.

In many ways, the computers and tools we use today (including the Internet) would not exist without hackers. If hackers continue to exploit the system they helped create, computing will not be safe for anyone. ♦♦

The 1993-1994 Executive

President:	Bary Finch	(W) 934-1690
Vice-President:	Rob Wright	(W) 488-5175
Treasurer/Secretary:	Doug Shewfelt	(W) 986-7111
Publicity Director	Rory Macleod	(W) 488-5168
Programs:	Doug McLean	(W) 985-1643
Mailing List:	Roland Schneider	1-204-785-9179
Membership Sec.:	Roland Schneider	1-204-785-9179
Newsletter editor:	Andrew Trauzzi	(W) 986-3898
Past President	Susan Zuk	(W) 989-3530
Information:	Bary Finch	(W) 934-1690
		(FAX) 934-1308
(or)	Andrew Trauzzi	(W) 986-3898

Copyright Policy and Disclaimer

This newsletter is ©copyrighted by the Manitoba UNIX User Group. Articles may be reprinted without permission, for non-profit use, as long as the article is reprinted in its entirety and both the original author and the Manitoba UNIX User Group are given credit.

The Manitoba UNIX User Group, the editor, and contributors of this newsletter do not assume any liability for any damages that may occur as a result of information published in this newsletter.

Advertising Rates

Quarter page	\$50
Half page	\$75
Full page	\$100
Insert (1-4 pages)	\$100

Above prices are per issue. The first ad is charged at the full price; each successive month is 1/2 price.

Ad copy must be submitted by the final copy deadline for an issue (usually 3 weeks prior to the monthly meeting) in a format acceptable to the editor. (Please make arrangements with editor beforehand.)

Internet E-mail: editor@muug.mb.ca

Group Information

The Manitoba UNIX User Group meets at 7:30 PM the second Tuesday of every month, except July and August. Meeting locations vary. The newsletter is mailed to all paid-up members one week prior to the meeting. Membership dues are \$25 annually and are due as indicated by the renewal date on your newsletter's mailing label. Membership dues are accepted at any meeting, or by mail.

Manitoba UNIX User Group
P.O. Box 130, Saint-Boniface
Winnipeg, Manitoba R2H 3B4

Internet E-mail: membership@muug.mb.ca
WWW: <http://www.muug.mb.ca>

The End of Another Season

By Bary Finch

Here we are, moving into our annual summer "recess". It has been another great year, with quite a variety of presenters over the season. Unfortunately we had a few cancellations that disrupted our program. We hope to have this corrected for the next season.

We will of course be reconvening in September, as usual, and on the second Tuesday of every month, as usual. The St. Boniface Research Centre will again be our "home" for the meetings.

In the meantime, it is definitely summer. (That was fast, as my last article noticed it was Spring.) This means it must be time for the Annual MUUG BBQ as our June meeting.

MUUG BBQ!

The BBQ will of course be on the usual Tuesday (June 13), only it will start an hour earlier than usual at 6:30 p.m. to let people actually eat at a reasonable time, and still get some sunlight to see what they're eating. I have a special interest in the details for this year's BBQ as I am hosting it at my house. (See the map for details on how to get there.)

So come on out and enjoy a relaxing time with your fellow MUUG members. As usual we (MUUG) will provide the "fixings" for the BBQ (salads, chips and soft drinks), and you just provide the BBQ meat (or whatever) of your choice. Of course you must also provide any alcoholic beverages that you desire. (If you want free liquor you should attend the Annual MUUG Wine and Cheese at Christmas!) Hope to see you all there!

Firewall

"In other news" (as they say) we are still considering how we can put on a Firewall seminar. So if there are any organizers out there in the MUUG membership, you still have a chance to get on board and

help us out. Just contact anyone on the board and say that you want to give a hand.

Another item to consider for the Fall is that our board will be changing as the MUUG year end (October) occurs. We will be looking for new board members to help us out, as a number of the existing board, as usual, will be retiring from their positions (without a pension of course) to pursue other interests. So if you have an interest in any specific position, or just want to help out and take whatever position the board needs filled most, let us know. We're always looking for new talent, as it were.

Wanted: SIG Coordinator

One position that has been vacant for a bit looks like it will be filled for the next season (by a volunteer who shall remain nameless for the time): the SIG Coordinator. I say the next season as we do not have a June meeting. It seems a bit suicidal to schedule a meeting for the third week in June just as the nice weather really gets going. So the SIG will reconvene in September, along with the regular meetings.

With our Fall program fast approaching, we'd appreciate any suggestions for fall speakers that are of special interest to people. Or maybe you've attended a conference lately that had an especially good presenter at it. Let us know! We'll chase down people from anywhere. Part of the MUUG board's job is to get great out of town speakers for our membership, especially for the members that get less of a chance to travel to the conferences where these speakers present the most.

That's about it for now. Enjoy the summer, and we'll see you at the BBQ! ➡

CORPORATE SPONSORS

The Manitoba UNIX User Group

gratefully acknowledges the
generous support of the following

Corporate Sponsors



Great-West Life Assurance
Company

THE DOCUMENT COMPANY



XEROX



ONLINE
BUSINESS SYSTEMS



sun
microsystems

C++ Q&A

By Marshall P. Cline

This month's C++ Q&A completes last month's comparison of C++ and Smalltalk. The complete C++ FAQ is now available in a book format — Addison-Wesley Publishers 0-201-58958-3 \$32.25.

Question 90: How do you use inheritance in C++, and is that different from Smalltalk?

There are two reasons one might want to use inheritance: to share code, or to express your interface compliance. I.e.: given a class 'B' ('B' stands for 'base class', which is called 'superclass' in Smalltalkese), a class 'D' which is derived from B is expressed this way:

```
class B { /*...*/ };
class D : public B { /*...*/ };
```

This says two distinct things: (1) the bits(data structure) + code(algorithms) are inherited from B, and (2) 'D's public interface is 'conformal' to 'B's (anything you can do to a B, you can also do to a D, plus perhaps some other things that only D's can do; ie: a D is-a-kind-of-a B).

In C++, one can use inheritance to mean:

- #2(is-a) alone (ex:you intend to override most/all inherited code)
- both #2(is-a) and #1(code-sharing)

but one should never use the above form of inheritance to mean

- #1(code-sharing) alone (ex: D really *isn't* a B, but...)

This is a major difference with Smalltalk, where there is only one form of inheritance (C++ provides 'private' inheritance to mean 'share the code but don't conform to the interface'). The Smalltalk language proper (as opposed to coding practice) allows you to have the *effect* of 'hiding' an inherited method by providing an override that calls the 'does not understand' method. Furthermore Smalltalk allows a conceptual 'is-a' relationship to exist *apart* from the subclassing hierarchy (subtypes don't have to be subclasses; ex: you can make something that 'is-a Stack' yet doesn't inherit from 'Stack').

In contrast, C++ is more restrictive about inheritance: there's no way to make a 'conceptual is-a' relationship without using inheritance (the C++ work-around is to separate interface from implementation via ABCs). The C++ compiler exploits the added semantic information associated with public inheritance to provide static typing.

Question 91: What are the practical consequences of diffs in Smalltalk/C++ inheritance?

Since Smalltalk lets you make a subtype without making a subclass, one can be very carefree in putting data (bits, representation, data structure) into a class (ex: you might put a linked list into a Stack class). After all, if someone wants something that an array-based-Stack, they don't have to inherit from Stack; they can go off and make effectively a stand-alone class (they might even *inherit* from an Array class, even though they're not-a-kind-of-Array!).

In C++, you can't be nearly as carefree. Since only mechanism (method code), but not representation (data bits)

can be overridden in subclasses, you're usually better off *not* putting the data structure in a class. This leads to the concept of Abstract Base Classes (ABCs), which are discussed in a separate question. You can change the algorithm but NOT the data structure. Bits are forever.

I like to think of the difference between an ATV and a Maseratti. An ATV [all terrain vehicle] is more fun, since you can 'play around' by driving through fields, streams, sidewalks and the like. A Maseratti, on the other hand, gets you there faster, but it forces you to stay on the road. My advice to C++ programmers is simple: stay on the road. Even if you're one of those people who like the 'expressive freedom' to drive through the bushes, don't do it in C++; it's not a good 'fit'.

Note that C++ compilers uphold the is-a semantic constraint only with 'public' inheritance. Neither containment (has-a), nor private or protected inheritance implies conformance.

Question 92: Do you need to learn a 'pure' OOPL before you learn C++?

The short answer is, No.

The medium answer length answer is: learning some 'pure' OOPLs may *hurt* rather than help.

The long answer is: read the previous questions on the difference between C++ and Smalltalk (the usual 'pure' OOPL being discussed; 'pure' means everything is an object of some class; 'hybrid' [like C++] means things like int, char, and float are not instances of a class, hence aren't subclassable).

The 'purity' of the OOPL doesn't make the transition to C++ any more or less difficult; it is the weak typing and improper inheritance that is so hard to get. I've taught numerous people C++ with a Smalltalk background, and they usually have just as hard a time as those who've never seen inheritance before. In fact, my personal observation is that those with extensive experience with a weakly typed OOPL (usually but not always Smalltalk) have a *harder* time, since it's harder to *unlearn* habits than it is to learn the statically typed way from the beginning.

Question 93: What is the NIHCL? Where can I get it?

NIHCL stands for 'national-institute-of-health's-class-library'. it can be acquired via anonymous ftp from [128.231.128.7] in the file pub/nihcl-3.0.tar.Z

NIHCL (some people pronounce it 'N-I-H-C-L', others pronounce it like 'nickel') is a C++ translation of the Smalltalk class library. There are some ways where NIHCL's use of weak typing helps (ex: persistent objects). There are also places where the weak typing it introduces create tension with the underlying statically typed language.

A draft version of the 250pp reference manual is included with version 3.10 (gnu emacs TeX-info format). It is not available via uucp, or via regular mail on tape, disk, paper, etc (at least not from Keith Gorlen). ◆◆

Ask Monsieur

A column in which our resident UNIX expert answers questions submitted by members.

Dear Monsieur Ex,

`/bin`, `/sbin`, `/usr/bin`, `/usr/local/bin`, `/etc`, `/usr/ucb`, `/usr/bin/X11`, `usr/sbin`? Where do I put that binary? When is something "local"? If I write it? If it doesn't come with the operating system?
`/var/spool/uucp`, `/usr/spool/uucp`? Where does that lock go? What does "var" mean anyway?
`/tmp`, `/usr/tmp`, `/var/tmp`? How many temporary directories does a person need in their life?
`/home` (or is it `/usr/home`) seems like a good thing. It gets the users out of `/usr`. Is `usr` short for `user`? I occasionally come across a `/user` directory on UNIX systems. Is `/user` short for `usr`?
 After many (too many?) years of puttering with UNIX I've never been able to decode the logic behind directory naming. Can you shed some light on this? A few rules of thumb perhaps? A pointer to the "big book" on UNIX naming? Next month I'm going to ask about groups. What use is a "bin" group anyway?

Sign me,
 Find command's biggest user

Mon cher "user,"

Premierement, Monsieur Ex would like to apologize for the long delay in replying. I've been "beaucoup trop" busy lately to handle my mail.

You ask a lot of questions, but the simple, one-word answer is: evolution!

In "le bon vieux temps," a UNIX system's root file system was usually very small, and only had room for the essential utilities to get the system up and running. Thus, there were counterpart directories under the much larger `/usr` file system for all the non-essential, or larger files. Hence, `/bin` (which is short for binary, since the binary images of programs are stored there) and `/usr/bin`, `/lib` and `/usr/lib`, `/tmp` and `/usr/tmp`. (Incidentally, `/usr`, which is short for "user," was also the file system containing users' home directories.)

As things started to evolve, other directories came into existence to further divide things, and reorganize them into more workable collections of files. As people started to add local software, (i.e. other software than what came with the system), it became apparent that this should go in a separate file system, to make it easy to keep this around even when the system software was upgraded (which sometimes involved a complete rebuild of the root and `/usr` file systems). Hence `/usr/local/bin`, et al were born.

Other collections of programs spawned the creation of

other directories, such as `/usr/ucb` for the Berkeley UNIX utilities, and `/usr/bin/X11` for X Window programs. It also became obvious that home directories should also be separated from any of the system software, so other directories like `/home`, `/user`, `/u`, and so on began to appear.

Ah, Networks!

Then, with the advent of networked file systems, another "grand changement" took place. The (now completely misnamed) `/usr` file system, which now contained the bulk of the system software, was divided up into two parts: one part (still called `/usr`) for all of the read-only, sharable files, and another part (called `/var`) for all the writable files (i.e. variable data) that used to be mixed in with the rest of `/usr`.

With this new structure, it was possible for a network of similar UNIX systems to share one copy of `/usr` over the network, by mounting it read-only, while each system would have its own private copy of `/var`, for all the variable data. For reverse compatibility, many of the old file names under `/usr` were maintained by setting up symbolic links to the file's new location under `/var`. Hence, `/usr/tmp` points to `/var/tmp`, `/usr/spool` points to `/var/spool`, `/usr/adm` points to `/var/adm`, and so on.

Sun — Oui, Oui!

Some systems, such as SunOS, also decided that the root file system should contain as few programs and system files as possible, so anything that could be shared was also moved to `/usr`, leaving only files that had to be either private to each machine, or absolutely essential to booting the system, on the machine's own root file system. So `/bin` now point to `/usr/bin`, `/lib` to `/usr/lib`, and so on. The few remaining binaries (which are statically linked) are placed in a new `/sbin` directory.

On other systems, `/sbin` and `/usr/sbin` are simply for system binaries, i.e. programs (such as daemons) run by the system, but not meant to be run as user commands. Such binaries used to be either in the usual `bin` directories, or mixed in with other support files in `/etc` or `/usr/etc`.

Monsieur Ex hopes that this has cleared up some of the confusion, and given you a better idea of where files should go. Of course, as the systems continue to evolve, the rationale may change, and so will the name and location of other files and directories.

Now I must go find some aspirin for a "mal de tete."

Salut, M. Ex.

Monsieur Ex, a mysterious Frenchman who claims to be a reformed hacker and an expert in UNIX, awaits further mental challenges! To reach him, e-mail
 <m-ex@muug.mb.ca>

Annual MUUG Barbecue

June 13, 6:30 pm

Host: Bary Finch
phone: 934-1690

Where: The Finchs'
1005 Manahan Ave.
It's the first house
beside the NCR
building on Pembina
Highway.
(See map)

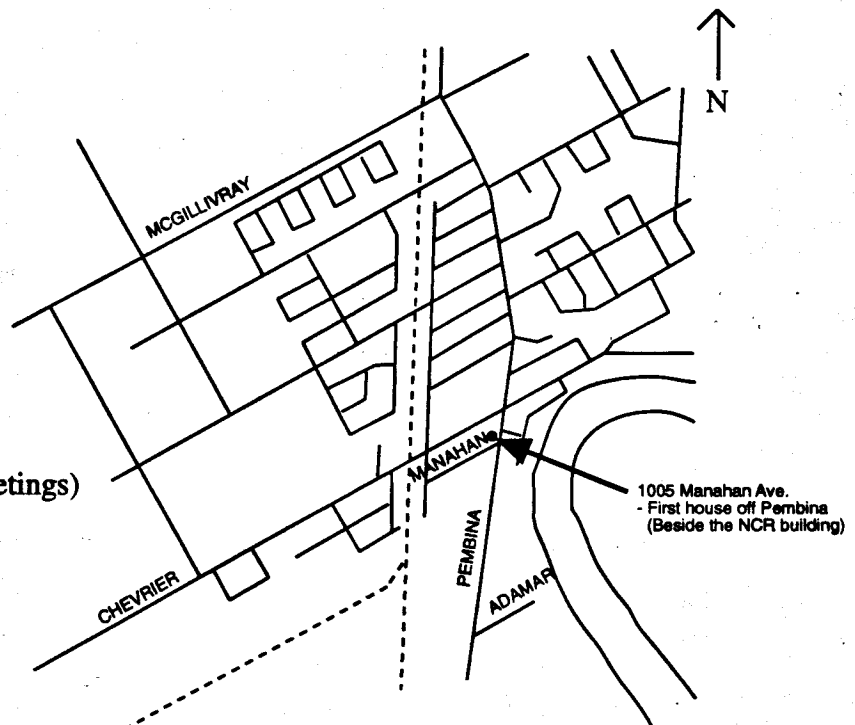
When: Tuesday, June 13, 6:30 pm
(1 hour earlier than our normal meetings)

Bring: Meat to cook
Beer
Lawn chairs

RSVP: Bary Finch
934-1690 (days and evenings)
e-mail: <bfinch@muug.mb.ca>
or
Andrew Trauzzi
986-3898 (days)

Notes: You can park in the parking lot beside
Bary's house if you so desire;

Please enter via the backyard, as that's
where the deck and BBQ are.



We will supply chips and other nibble food, soft drinks, salads, and a cake. (and insect repellent if necessary). Spouses or significant others, and children, are also welcome. ➡

SIG Sideline

By Andrew Trauzzi

The SIG group needs a new co-ordinator! If you are interested in taking an active role in MUUG activities, please contact the board at <board@muug.mb.ca>.

The SIG season has ended, therefore, the next SIG meeting will be September 19, 1995. The location is TBA. Check September's newsletter for details.

Coming Up

Meeting:

September's meeting is scheduled for Tuesday, September 12, at 7:30 PM. Meeting location will be the St-Boniface Research Centre, as usual. Stay tuned for details, and have a great summer!

Got any ideas for meeting topics? Any particular speaker, company, or product you'd like to see at one of our meetings? Just let our new meeting coordinator, Doug McLean, know. You can e-mail him at <dmclean@muug.mb.ca>.

Newsletter:

If you are interested in a particular topic, let me know. I'm sure I could coerce you into writing an article! I could use a few articles — especially shorter ones — half a page to one page (400 to 1000 words) would be fine.

Monsieur Ex has also let me know that his mail-box has room for more of your wonderful queries again — please submit your questions to the old guy via e-mail to <m-ex@muug.mb.ca>. He may be old, but he's not ready for retirement yet!