By using the multi-resolution seamless images database (MRSID), developed at Los Alamos, very large images can be compressed for viewing on workstation monitors. The images shown here depict the region around the Lincoln Monument extracted at each resolution. At the coarsest resolution (64 meters/pixel) the entire Washington, D.C., area can be displayed on a workstation monitor. At the finest length scale (1 meter/pixel), the user can distinguish features as small as an automobile. Some seams are apparent where the digital orthophoto quadrangle (DOQ) images are joined. This is not a result of MRSID processing, but rather is due to mismatches in registration and radiometry between the individual DOQs. This work was developed by Jonathan Bradley of CIC-3 and is part of the Los Alamos SUNRISE/National Information Infrastructure project.

Inside this issue:

Feature Articles
New Access Number for Off-Site Paging .....................1
Information Architecture Teams Forming ..................2
Distributed Computing Team Supports PVM Software
and Initiates Parallel Tools Users’ Group ...............4
Using C++ For Scientific Computing
Through Array Classes ..................................6
Tips from the Consultants
PAPER or PLASTIC? LOGIN or .CSHRC? .............11
Cluster Corner
The Coming of Network Licensed Software ............13
Microcomputing News
Apple Introduces New Version of Apple PhotoFlash ....14
In the Classroom
CIC Computing Classes ................................15
Lab-Wide Systems Training ............................19
LANL Research Library Training .......................22
ICNchanges .............................................23
Forms and Schedules ..................................31
Index .................................................37
Integrated Computing Network (ICN)

Consulting
- Centralized scientific and engineering computing ...........................................7-5745
- Lab-wide administrative and business systems .................................................7-9444
- Passwords (required for access to ICN) ..............................................................5-1805
- Systems documentation (local and vendor supplied) ...........................................7-6992

Central Computing Facility (CCF) ....................................................................... 7-4584

Advanced Computing Laboratory (ACL) ...............................................................5-4530

Local Area Network (LAN) system administration services ................................. 5-2220

Desktop Support Center (DSC) ............................................................................. 7-4357 (7-HELP)
- (PC Help for IBM and Macintosh personal computers)
- For questions about PC software: PCSW-help@lanl.gov
- For questions about PC hardware: PCHW-help@lanl.gov
- For questions about Mac software: MacSW-help@lanl.gov
- For questions about Mac hardware: MacHW-help@lanl.gov

Telephone Services Center ...................................................................................... 7-3400
- (includes voice mail)

Computer training
- Lab-wide systems support training ................................................................. 7-9444
- Computer/workstation training ........................................................................ 7-9399
- Personal computer training ............................................................................... 7-9071
- Microcomputer support facility seminars ....................................................... 7-4357
  - (Macintosh/IBM software, lending library)

List of Forms and Schedules

Course Registration Form for CIC Computing Classes ........................................... 15
CCF Machine Availability and Downtime ................................................................. 30
DSC Software Order Form ..................................................................................... 31
Accessing Computing Machines through the INC ................................................. 32
Accessing the ICN through Dialup Modem .............................................................. 32
ICN Validation Request Form .................................................................................. 33
Reader Feedback Form ............................................................................................ 35
You may now dial (505) 665-9800 from anywhere outside the Laboratory to access the LANL radio paging system. Known as “ring-up” or “over-dial” type paging, 665-9800 is a standard telephone line (actually, there are multiple lines on rotary, terminated directly into the paging terminal). Upon answering, the paging terminal will prompt you: “Please enter the pager ID number.” At this point, the caller should enter a 7-digit pager number.

Since the 118-series pagers only have six digits (118-xxx), the caller will be required to enter an additional 1, making the call a 7-digit (1118-xxx) page instead of a 6-digit (118-xxx) page. The method for entering the 7-digit 104 pager numbers has not changed (i.e., 104-xxxx).

The new outside paging number will not affect the current operations of the LANL paging system. The old outside access number, 665-0062, will remain operational for now for those who are familiar with its use. The 665-0967 alpha-numeric port will also remain available for dialing in from your laptop PC equipped with modem and paging software. The telephone device for the deaf (TDD) access number, 665-DEAF, is not affected either.

The 665-9800 access mode should work from any touch tone telephone anywhere. This accessibility will be especially helpful when calling long distance from such places as Lawrence Livermore National Laboratory. The old 665-0062 access number does not work from such locations because of translation/supervision glitches in the telephone system when certain long distance carriers are used.

Additionally, as of this past summer, all 104 paging is now being transmitted from Pajarito Mountain, with back up transmitters at the TA-55 tower.

*Eric Powers, 665-3461, ep@lanl.gov
Telecommunications Group (CIC-4)*
Information Architecture Teams Forming

If you would like to participate in the development of guidelines and standards for the Laboratory’s information architecture (IA), the time to get involved is now. Phase Two of the IA project, which started in June, has been rapidly picking up speed with the formation of standards development teams.

As discussed in previous issues of BITS, the objective of an IA is to provide a framework for overall planning and coordination of computing, information, and communications activities. Earlier this year, the Laboratory Leadership Council (LLC) approved a set of 20 high-level principles to guide the IA. Part of the current IA effort is to develop guidelines and standards—specific instructions for implementing the principles. (For details about the IA standards development process, see the August and September 1994 issues of BITS.)

The IA effort is being performed by a number of self-directed teams, each with membership drawn from the Laboratory-wide community (as illustrated below).

The IA review team is responsible for coordinating the overall IA effort. This team, which was formed in July, identifies general areas where standards are needed, charters standards development teams to address those areas, and reviews the work of the standards development teams.

Each standards development team is responsible for investigating issues related to its area, surveying user needs and available technologies, and recommending guidelines and standards to the IA review team. As it deems appropriate, each of these teams can also generate additional teams to focus on specific areas.

There are currently five standards development teams that have been chartered by the IA review team:

- Applications
- Data
- Data Warehouse
- Desktop
- Infrastructure

The scope of work for each team is discussed on the following page.

Although these teams have already started, their work is still in the early stages and there is still time to join them. Indeed, because the success of the IA effort depends on the active participation of a wide range of people, the IA review team encourages you to join.

If you see a particular area of interest, please contact the leader for that team. If you would like more general information about the IA project, please contact one of the project’s co-leaders: Diane Weir (DDCIC, 7-9337, drw@lanl.gov) and Karl-Heinz Winkler (DDCIC, 5-5530, khw@lanl.gov).

Tad Lane, (505) 667-0886, tad@lanl.gov
Communications Arts and Services (CIC-1)
IA Standards Development Teams

Applications Team
Scope of Work: The IA applications team addresses issues related to software applications systems developed at the Laboratory for internal Laboratory use. Some specific areas covered by this team are

- legacy system access
- on-line forms
- standards for client/server applications
- usability and user interfaces
- standards for “groupware” applications

Team Leader: Gerald Reisz
(CIC-12, 5-2468, greisz@lanl.gov).

Data Team
Scope of Work: The IA data team addresses issues related to Laboratory-wide data management, format, and interchange. Some specific areas covered by this team are

- Laboratory corporate information directory (LCID)
- geographical information systems (GIS)
- data interchange
- Laboratory information policy

Team Leader: Fred Montoya
(CIC-12, 5-0912, montoya_frederic_d@lanl.gov).

Data Warehouse Team
Scope of Work: A data warehouse is a mechanism for improving access to information by consolidating the storage of that information in a central place and consistent format. By creating an integrated library of historical information, the data warehouse can ensure that information is more quickly, easily, and consistently available. Some specific areas covered by this team are

- data warehouse strategies
- data warehouse standards
- data warehouse processes
- information priorities
- data warehouse pilot project

Team Leader: Steve Blair
(NIS-3, 5-5895, sgblair@lanl.gov).

Desktop Team
Scope of Work: The IA desktop team is responsible for developing guidelines and standards for desktop tools for the Laboratory as a whole. The goal is interoperability, which will enable Laboratory citizens (i.e., Laboratory employees and contractors) to share information in native electronic form, to collaborate more effectively in problem solving, and to produce results that are compatible and consistent. Some specific areas covered by this team are

- establishment of an acceptable set of standard multiplatform tools to improve information flow across the Laboratory
- identification of acceptable processes that will
  —formalize the implementation and continuation of desktop standards, and
  —foster an environment that encourages Laboratory-wide participation
- collaboration with other teams in areas where desktop product availability affects the selection of technical standards (E-mail, encryption, etc.)

Team Leader: Frank Bobrowicz
(CIC-2, 5-5531, fxb@lanl.gov).

Infrastructure Team
Scope of Work: The IA infrastructure team addresses issues related to providing the physical connectivity and other infrastructure needed for easy, seamless access to information, services, and products. Some specific areas covered by this team are

- E-mail services
- encryption and digital signatures
- automated desktop backups
- LAN topology
- LAN management standards
- a strategy for services in the secure partition
- provision of access to Gopher, Mosaic, and the World Wide Web

Team Leader: Doug Gatchell
(CIC-5, 7-5119, dgg@lanl.gov).
Distributed Computing Team Supports PVM Software and Initiates Parallel Tools Users’ Group

The Distributed Computing group (CIC-8) would like to introduce one of its newly formed teams—the Parallel Distributed Computing Tools Team. The team’s mission is to provide parallel computing tools to the scientific community at Los Alamos. The members of this team are Jeff Brown (team leader), Bob Boland, Richard Barrett, Suresh Damodaran-Kamal, Jonathan Greenfield, and MaryDell Tholburn. This article will focus on the team’s support of PVM software and the creation of a new users’ group for developing distributed memory parallel computing applications.

Support for PVM Software
Parallel Virtual Machine (PVM) is a software package that allows users to configure multiple UNIX-based workstations into a single computational resource. This software is now being supported by the Parallel Distributed Computing Tools Team. PVM is the first of many parallel tools to be released and supported by the team. Team support for PVM includes the following:

• Library Upkeep: PVM is now available throughout the Laboratory via a link to the Info-Server. (Note to system administrators: For details on establishing this link, refer to the Mosaic entry Software Archives, which is listed on the LANL home page). Users can link to PVM through the Info-Server by adding /usr/lanl/pvm/lib to their path. This link will provide access to the most up-to-date and architecture-specific version of PVM. It will also free the user from the need to set the environment variables PVM_ROOT and PVM_ARCH (although if these variables are set, they will not be overwritten and PVM will make use of them). For additional information on accessing current and previous versions of PVM, see the Mosaic entry described on the next page.
• **Mosaic Entry:** This entry includes an overview of PVM, frequently asked questions, documentation, user experiences, listings of software updates, and CIC-8 contacts. To access this entry use the URL (uniform resource locator) code below:


• **Staff Support:** This includes startup assistance, and debugging and performance enhancement tools, such as XPVM and AIMS.

• **LANL Mailing List:** This list is dedicated to the discussion of, and announcements pertaining to, parallel computing. To subscribe, send E-mail to listmanager@lanl.gov and include subscribe ptools@lanl.gov in the body of your E-mail.

For details of other services provided by CIC-8, which include data-parallel compilers, development tools, load balancing, math software, message passing, and object-oriented methods, see our team’s home page via the Mosaic URL:

http://www-c8.lanl.gov/dist_comp2/cic8-distcomp2.html

To navigate to this page from the LANL home page, click on “Computing at LANL” and “CIC-8 Distributed Computing.”

---

**PTools Users’ Group**

The Parallel Distributed Computing Tools Team is forming a LANL PTools Users’ Group. This users’ group will focus on facilitating the development of portable, parallel, scalable programs in a heterogeneous distributed memory environment, and on providing a forum for education, discussion, and exchange of experiences in parallel/distributed environments. The users’ group will provide CIC-8 with input on requirements for software tools and programming environments that will be needed by the emerging parallel/distributed computing community.

The first meeting of the LANL PTools User Group will be held Thursday, December 8, at 9:30 a.m. in the Center for Nonlinear Studies (CNLS) conference room. Our initial meeting will focus on PVM, but we will also discuss the various software libraries available for distributed memory parallel computing at LANL, the advantages and disadvantages of using explicit message passing in application programming, user experiences, and future directions for the group.

Topics for future meetings include data-parallel compilers, parallel computing development tools, load sharing issues, object-oriented methods, math software, and user experiences. For further information concerning the users’ group, as well as announcements of future meetings, refer to our team home page by clicking on “PTools User Group” on Mosaic.

Feel free to contact the author of this article if you have questions about PVM software or the PTools Users’ Group.

Richard Barrett, rbarrett@lanl.gov, 667-6319

Distributed Computing Group (CIC-8)
Introduction
The computing environment of today places many demands on the scientific programmer. Algorithms are becoming more complex and therefore more difficult to program. The platforms on which the programmer works are changing more rapidly than ever before, requiring a portable programming technique. Finally, the programming style used by the implementer is also in great flux with the advent of new parallel architectures. These architectures include massively parallel computers, workstation clusters, and even heterogeneous collections of workstations used within a message passing environment. In response to these challenges, Los Alamos is developing a new programming environment. Our ultimate goal is to create an environment in which users can develop and debug numerical methods in the user-friendly domain of a serial computer and then recompile the source code for a parallel production machine. This environment will use C++ as its primary implementation language. In this article we describe an array class library, called A++, that is used as the basis for this portable programming environment.

What is C++?
C++ is a superset (with some very minor exceptions) of the programming language C. For those who appreciate C, the postfix operator “++” is used to increment by one the variable it follows. Therefore C++ is literally an add-on to C. Before describing C++, it is useful to start with a description of C. C is a general purpose programming language closely associated with the UNIX operating system. It was originally designed to be the high-level “assembly language” used to implement the UNIX operating system and its large number of utilities. C has since evolved into one of the more popular languages for most application areas. Because of its dynamic memory management capabilities and available mathematics libraries, C even has its foot in the door of scientific computing.

C has the basic data types to represent and manipulate single or double precision floating point numbers (floats and doubles respectively), integers (ints), and bytes (chars). Arrays of these types can be formed within a program stack (initialized within a function) or allocated off of the memory heap (initialized by getting a pointer to a block of memory). Arrays and pointers are virtually interchangeable within C as well. In fact, multidimensional arrays are implemented as arrays of pointers to other arrays of pointers or basic types. This has both advantages and disadvantages as pointer access is quick and flexible but pointers require extra memory.

On the logical side C offers various conditionals and looping constructs that are necessary for the use of structured programming practices. One can test all inequalities with the various types of numbers. Integers do double duty performing as logical variables. Looping constructs include whiles, do ..., whiles, and for loops, giving the user flexibility to test at the end or beginning of loops. Even a goto statement exists allowing the user to jump anywhere within a module in a pinch.

Finally, C can take individual pieces of data and bind them together. These collections are called structures. Structures can contain floats, doubles, ints, chars, or other structures. Like the components of the structure, structures themselves can be allocated on the stack or from the heap. Structures are a powerful way of organizing data and many high-level languages have this capability to the same or lesser extent. PASCAL records are similar to structures while FORTRAN 77 COMMON blocks have the same general functionality without dynamic allocation.

Although not part of the language, most C compilers come with a defacto standard collection of libraries and associated interfaces to perform many chores. In particular, there is a library to execute most of the mathematical functions found in FORTRAN, a library to do string manipulation, a collection of functions to supply I/O, and many others. There is a surprising amount of compatibility between various C compilation systems, but one must be aware that it is not perfect.

C++ has all the aforementioned features plus some new capabilities that bring it into the realm of what is called object oriented programming (OOP). OOP is a technique, discipline, or style of writing programs. Here algorithms are organized around data structures called objects that both hold data and supply the functions needed to manipulate the data in safe ways. Encapsulation is often used to describe this process. The goal of OOP is to generate reusable program modules with few side effects. The idea seems like a good and simple concept from a common sense viewpoint, but the implementation of flexible and reusable object libraries requires a great deal of forethought and design. Practice shows us that several design iterations are often required to “get it right.”

The primary way in which C is augmented to become a language that supports OOP is through the introduction of the data structure called a class. A class, in its most simple form, is a structure. However, classes become objects by associating “member functions” with the class. Member functions are simply functions that operate on the data within the
Inheritance is a mechanism for the reuse of objects. New objects can be created from currently available objects by inheritance. The inheriting objects will have all the properties of the inherited objects plus whatever is added (data or more member functions). Inheritance facilitates a rich structure of objects through multiple inheritance (inheriting an inherited class which may in turn inherit other classes) yet allows the developer to encapsulate data at all levels. However, remember that no programming paradigm will prevent people from writing sloppy code.

Inheritance is further enhanced by Polymorphism. Polymorphism literally means many shapes. In C++ the same function name or even operators such as "+", "/", etc., can be used for many purposes. A simple example is to consider the type double, a double precision number. Doubles can be added, subtracted, multiplied, along with a host of other arithmetic operations. A simple example of polymorphism is to define a new class that would represent complex numbers as a pair of real numbers. Many of the same arithmetic operators that are used to manipulate doubles can be "overloaded" to manipulate complex numbers as well. The array class library described below is another example of polymorphism.

The last feature of C++ that we will discuss is templates. Templates are a relatively new feature of the language and may undergo further change because of the brittleness of the current implementation. The concept of a template is simple no matter how it is implemented. There are many times in which the same code is duplicated with different types of data. For example, the programmer often has to keep track of data through lists. Different types of data require different list functions even though most of the code will look the same. The C programmer can often save time and duplication of effort by using macro expansion or pointer casting (casting exploits the fact that pointers to different objects are handled uniformly since they are merely addresses of the objects). Both of these techniques can often lead to problems. Macro expansion is cumbersome to debug while casting of pointers may not be "type safe." Type safe means that different types of data may inadvertently be treated uniformly because the compiler is blinded by the casting operation, i.e., the compiler can no longer distinguish between objects since it can only see addresses. With the brief summary of C++ complete, we can now discuss array classes.

**What is an Array Class?**

In C or C++, doubles, floats, ints, and chars are treated in a complete and consistent manner. An important feature of C++ is its ability to create new objects that can also be treated in a complete and consistent manner. To make this concept more concrete, consider FORTRAN 90 and how it has evolved from FORTRAN 77. A major enhancement of FORTRAN has been the addition of array syntax. Arrays and parts of arrays can now be handled in a complete and consistent manner much like double precision, single precision, and integer scalars. That is, one can perform a large collection of operations on these arrays much like one can do now for scalars in FORTRAN 77. The same environment can be implemented using the built-in extensibility of C++. Below we describe a class library for array objects called A++ (and P++), which is designed and implemented by Dan Quinlan of CIC-3. This class library is used extensively within the Parallel Computing and Numerical Analysis Cell of CIC-3. Array objects are defined and operations through polymorphism and operator overloading can be defined. The array class design process is not easy if the target implementation is going to be efficient and robust, but there are rewards and even a few surprises in following this path. The following is a brief tutorial of the array class.

There are integer, float, and double array objects. These are declared on the stack or off of the heap (see the example below).

```c
#include <A++.h> // Include the array interface // definitions.

doubleArray A(10,10), B(20); // Stack based allocation.

floatArray F(10,10,N); // N is an integer.

intArray *C = new intArray(7,13,11); // Heap based allocation.

. .

delete C; // Explicitly delete C.
```
The arguments in parenthesis naturally set the dimensions of the arrays. The memory for the arrays is dynamically allocated at run time. For the stack based arrays, the memory is deleted at the end of the array’s scope. Once arrays are allocated they can be initialized in a number of ways (see the example below).

```c++
#include <A++.h>

doubleArray A(10); // Stack based allocation.
A = 10.0 // Set every element of A to ten.
A(1) = 5.0; // Set the second element of A to five (C base zero indexing).
```

Of great importance, arithmetical operations can be applied just like scalar operations. Even mathematical functions are overloaded and array operations can be mixed with scalars to get the expected result (see the example below).

```c++
#include <A++.h>

doubleArray A(31), B(31), C(31);

B = 2143.0;
C = 22.0;

A = sqrt(sqrt(B/C)) + (B + C)/2165. - 1.0;

// A has a good approximation
// to pi in each of its elements.
```
Operations on parts of the arrays can be succinctly expressed and efficiently carried out using other objects associated with arrays called Index and Range classes. The following is an example of a simple relaxation algorithm using an Index class.

```cpp
#include <A++.h>

doubleArray A(32,32), F(32,32);

double h2 = 1.0/(32.0*32.0);

Index I(1,30,1);  // Construct an Index object
                 // starting at location 1 of
                 // length 30 and stride 1.
Index J = I;       // Construct another copy a I.

A(I,J) = 0.25*(A(I+1,J) + A(I,J+1) + A(I-1,J) + A(I,J-1)) + h2*F(I,J);

// Relax the interior of A
// which is (1:30,1:30) using
// Fortran triplet notation. The values at the edges of A
// are the boundary conditions. Note also that the base
// value of arrays default to 0 (verses 1 in FORTRAN) in C
// and C++. The array class uses the same convention
// although the user can change the bases at will.
```

A FORTRAN implementation might look like the following:

```fortran
real*8 a(0:31,0:31), f(0:31,0:31);
real*8 h2
parameter(h2 = 1.0/(32.0*32.0))

integer i, j

do 20 j = 1, 30
   do 10 i = 1, 30
      a(i,j) = 0.25*(a(i+1,j) + a(i,j+1) + a(i-1,j) + a(i,j-1)) +
               h2*f(i,j)
   10   continue
  20   continue
```
There are many other features of the array class language, but in an article such as this there is not enough room to describe them all. Needless to say there is a complete set of arithmetic operations available to the user within the array class. In addition, there are constructs for handing conditionals through masking. Finally there are many support functions one would expect in the maintenance and development of an array syntax application such as very flexible access to the structure of arrays themselves. But there are other interesting characteristics of array classes besides the syntactical features. Perhaps the most important aspect of the array class may be its portability. Since the class is written in C++ using standard features of the language, it should compile and run on many platforms with few changes (if any). In fact A++ runs on a variety of platforms ranging from PCs running OS/2 to UNIX workstations such as Suns and SGIs to vector supercomputers like the YMP. Source code written using A++ and C++ then also becomes portable.

But array classes may enable us to have a more general concept of portability. The array class has also been ported to parallel architectures such as networks of workstations and massively parallel machines using message passing. The C++ code remains the same for the application written in the array language but the implementation of the actual classes changes in order to distribute and interchange the data within the arrays between processors. To distinguish the parallel array class library from the serial version, the parallel version is often called P++. Therefore, code written using the array classes on serial machines can be recompiled on a parallel architecture and run. This can be done since the communication and partitioning algorithms are available to the class implementer instead of being embedded within the compiler. But then why stop there? Since array classes are really nothing more than interfaces to a machine dependent architecture, many possibilities can be explored. Further optimization can be done at run time through analysis of the machine and data configurations. Analysis of dependencies at run time could yield even more parallelism by allowing the spawning of several tasks per processor. The possibilities are great.

**Are You Getting Something for Nothing?**

If things sound too good to be true then they usually are too good to be true. What are the problems? Perhaps the most obvious problem with high-level implementations is the overhead associated with maintaining and hiding complexity. For small arrays, the overhead from handling the arrays becomes significant with respect to the array operations themselves. Another effect of a high-level implementation is that the compiler cannot “see” enough to use register optimizations. Highly optimized code may run faster than the equivalent array library code because of more efficient register utilization and less fetches to memory. To alleviate the overhead problems, some care should be taken to make sure the array sizes are sufficiently large. The register allocation problem is more difficult. To get peak performance one can optimize by extracting the pointers to the array data and dropping into more low-level constructs in C++ or some other language so that the compiler can optimize using registers. Work is going forward on doing this automatically using precompilation targeted for the array class.

**How to get started.**

Although the structure of A++ is intuitive, one should have some working knowledge of C++. Like most computer languages and libraries, a programming model is always implied. Object libraries for A++ are publicly accessible on Crays Zeta and Gamma (see the example below). Workstation versions of A++ object libraries are available through CIC-8 and will eventually be loaded onto a Laboratory-wide NFS server.

```
Cray Directory:
/usr/local/usys/A++   Includes library libA++.a, and header files
/usr/local/usys/A++DOCS Includes a quick reference card
and other documentation.
```

Jeffrey Saltzman, (505) 667-4285, jss@lanl.gov
Computer Research and Applications (CIC-3)
While not as vital a question as to what to name the baby, fairly violent arguments can be started by asserting what should or should not be in the various shell initialization files. What should be where and when it should be there depends on the use that is made of these files; this, in turn, depends on whether the user is carrying out an interactive login (telnet, rlogin, klogin), is running a UNICOS production job, or is accessing the machine via one of the remote commands such as rcp, kcp, rsh, ksh, or xon.

**Interactive Login**
The files read by the various login shells are given in the order read by the shell at login time:

- **CSH** /etc/cshrc, $HOME/.cshrc, $HOME/.login
- **FCL** /usr/local/etc/flogin, $HOME/.flogin, $HOME/.fclrc
- **SH** /etc/profile, $HOME/.profile
- **KSH** /etc/profile, $HOME/.profile

For this shell, if the environment variable ENV is given a value in the $HOME/.profile file, then the file specified is read after .profile. Conventionally, this file is named according to

```
ENV=$HOME/.kshrc
export ENV
```

In the above, $HOME evaluates to the user’s home directory, for example /u0/joe_user. It is somewhat edifying to take a look (use “cat” or “more”) at the “system” initialization files /etc/cshrc, /etc/profile, and /usr/local/etc/flogin. These files provide the starting point for the user’s login files. Thus, on the UNICOS machines, these system files establish the initial value of the PATH variable to

```
/usr/ucb/bin:/usr/bin:/usr/local/bin:
```

as well as setting up a number of other environment variables.

Once the system file is read, the login shell proceeds to read the user’s initialization files; after that the user receives the shell prompt and is truly “logged in.”

**The C-Shell (CSH)**
CSH first reads the file $HOME/.cshrc and then the file $HOME/.login; the writer may believe this to be a design error—it would make better sense if this reading order were reversed. However that may be, keep in mind that the file .login is read but once at login time, while .cshrc is read whenever a sub-shell is spawned by, for example, executing a C-shell script.

Since .login is read but once, it makes sense to include in that file commands that need to be read but once. Among these are:

a) stty commands that control the terminal settings;

b) the “set term” command that sets the terminal type;

c) setenv commands such as setenv DISPLAY (for remote X-window-based displays) and setenv FILENV that on UNICOS controls the location of the infamous .assign file;

d) any alias commands that you know will be needed only in the login shell; although, generally alias commands are placed in the .cshrc file; and

e) any set commands that define shell variables that need be available only in the login shell.

While there is (at present, at any rate) no federal law against placing such commands in the file .cshrc, there is really no need to do this and burden the shell with repeatedly executing (upon spawning of a sub-shell) gratuitous commands.

This leaves a small category for the .cshrc file:

a) the “set prompt” command,

b) alias commands that you want available in all subshells, and

c) set commands that define shell variables that you want available in all subshells.
In the interactive session, the Bourne sub-shell invoked by the login C-shell to run the script “inherited” all the environment variables defined by the login shell. In the production job, the Bourne shell has available only the environment variables set by `/etc/profile` and the user’s `$HOME/.profile` files. And, most likely, the C-shell user does not even have a `.profile` file. The best way for this user to find out what is missing in the production environment is for him to submit a 1 minute Bourne shell production script that contains the one line `printenv` and compare the output of that command with that obtained by an interactive invocation of `printenv`. Then, the user can either construct a `.profile` file that supplies the missing information or, even better, modify the script itself to include this information at the very beginning.

Remote Invocations

The remote commands such as kcp, rcp, ksh, rsh, and xon all read the user’s remote `$HOME/.cshrc` file; this is totally independent of what the user may be using as a login shell on the remote machine. (The file `$HOME/.login` is not read by these remote commands.) Consequently, if the user accesses the remote machine mainly via these remote commands rather than by interactive logins, it behooves him to concentrate definitions of environment variables, shell variables, and aliases in file `.cshrc`. This is unlike the strategy recommended for a user whose use of the remote machine is through interactive logins.

John Wood, (505) 667-5745, consult@lanl.gov
Customer Service Group (CIC-6)
The Coming of Network Licensed Software

Actually, it’s not coming—it’s already here! In today’s software market, where “bootleg” software is often available and software distribution and licensing are costly to manage, many software vendors are taking steps to validate licenses in new and unique ways.

IBM has begun to use Network Licensing System (NetLS) for validating licenses. Under the current operating system, AIX 3.2.5, on the Open Network Compute Server (ONCS) (also known as CIC Open Cluster), there are just a handful of products that require NetLS. In AIX 4.1, however, every product from IBM is licensed using NetLS.

What is NetLS?
NetLS is a distributed application that uses communications technology based on the Network Computing System (NCS). NCS is a remote procedure call system that allows applications on one machine to invoke procedures that run on another machine in a client/server environment. NCS utilizes a naming service, referred to as a location broker, to enable the client application to locate procedures on remote machines.

NetLS is used in a client/server fashion to manage the licensing of software used on machines connected to a network. It provides the following licensing schemes:

• Node-locking: (also known as CPU locking) a software product is installed so that it can be used only on a specific node.

• Site Licensing: the number of licenses is limited only by the number of nodes available.

• Concurrent Access Licensing: a specific number of users on a node or network can use a product simultaneously.

• Single Execution Licensing: a product can be used only once before a license ticket expires.

As mentioned previously, IBM began using NetLS for a handful of products on AIX 3.2.5. Fortran and Cset++ are licensed using the concurrent access method. On the ONCS, 10 concurrent Fortran accesses (licenses) are available. For Cset++ there are 2 concurrent accesses. Any node in the network that looks to the server (in this case ibm-08) for its license validation can participate.

What should you know about NetLS administered licenses?
One of the few drawbacks of NetLS is that if the server or servers are down so are all the client nodes. In such a case you should get a message that indicates that the licensed product is unable to contact the server. If you get such a message or any other message that seems to indicate a problem with the server, please capture it and send it through E-mail to cluster_team@lanl.gov so that we can quickly take appropriate corrective action.

If all of the concurrent access sessions are in use you will get the message “Waiting for license.” As soon as one becomes available your command will execute. So far on the ONCS we have not experienced even one waiting for license condition (unless we artificially created it), and in the three months that we have been using NetLS there have been about 10,000 compiles per week. If a “wait” does occur, it will be logged so that we can monitor the need to increase the number of concurrent licenses.

How does this scheme benefit both vendors and their customers?
• NetLS provides for ease in distribution of software. IBM can now send a tape or CD-ROM that contains every AIX product they produce with the confidence that it will only be used by a machine that is licensed for its use. The same is true for software provided by anonymous ftp. For example, today the most current version of Fortran can by obtained by anonymous ftp, which provides quicker access to new releases.

• A single machine or group of machines can now provide licenses to all the other machines in the network, so adding or removing client nodes is very simple as far as software licensing goes.

• Demonstration licenses are easy to obtain. You can use FAX, E-mail, or toll-free calling to quickly receive a time bound and node-locked license key to test a product.

• If an organization chooses to look at its software licensing globally, they should be able to significantly reduce the total outlay for software by licensing only the concurrent needs network wide.

• The key registration center is always available to help licensed users. For example, we recently upgraded the ONCS workstations from the Model 560s to 590s. Consequently, the hardware CPU IDs changed. We found out on a Sunday night that we could not contact the server from any node. After a few minutes on the phone with the key registration center, we had new keys for the new hardware.

Network licensing is definitely a change in culture and as such requires some planning, forethought, and "getting used to.” It also promises to provide many benefits.

Doug Lora, dlora@lanl.gov, 665-3321
IBM/Computing Group (CIC-7)
Apple Introduces
New Version of
Apple PhotoFlash

Reprinted with permission of the author—Mary Devincenzi.

Product Description
Apple Computer, Inc. recently announced availability of Version 2.0 of Apple PhotoFlash, a digital photo cataloging and enhancement tool that quickly and easily enables computer users to add digital photos to their documents. Apple PhotoFlash 2.0 has, in a single application, been optimized for the increased performance of Apple Power Macintosh personal computers, while being fully compatible with Macintosh systems based on the Motorola 680x0 processor. Version 2.0 of Apple PhotoFlash also supports several key components in Apple’s new operating system, Macintosh System 7.5.

Significance
This enhanced version of Apple PhotoFlash, for Apple Macintosh® personal computers, now opens and saves individual catalogs of image “thumbnails” for easy photo organization and searching. With Apple PhotoFlash 2.0, it is now possible to search for photos by their similarity to an image or a sketch made by the user, as well as to adjust the color of your photos. The support for Macintosh Drag and Drop enables photos to be easily moved between catalogs and into applications supporting Macintosh System 7.5. In addition, the integration of Apple PhotoFlash 2.0 with Aldus PageMaker 5.0, QuarkXPress 3.2 and 3.3, Aldus Persuasion 3.0, Microsoft Word 6.0 and Claris FileMaker Pro 2.1, enables digital photos to be placed into documents in a single-step. Extensions are included for Aldus PageMaker and QuarkXPress that enable image editing and placement to be easily initiated from within those page layout applications.

Apple PhotoFlash 2.0 continues to support Apple’s AppleScript scripting technology by making scripts easier to record and play, and includes ready-to-run scripts to automate common tasks. In addition, the new version of Apple PhotoFlash continues to enable users to directly control and automate the Apple QuickTake 100 camera using AppleScript from within the PhotoFlash application, which enables further customization for vertical solutions.

Apple PhotoFlash 2.0 will come bundled with a Visual Symbols Sampler Photo CD, which contains a library of 140 everyday objects whose symbolic meaning makes them useful to enhance visual communication in a variety of documents.

System Requirements
Apple recommends Apple PhotoFlash 2.0, for customers with an Apple Macintosh, Power Macintosh, Macintosh Performa® or PowerBook computer with a 68020, 68030, 68040 or PowerPC processor with at least 8MB of RAM (3.5MB of application RAM), and any version of the System 7 operating system.

Availability
Apple PhotoFlash 2.0, is available in the United States through software resellers and Apple authorized resellers, as well as through the Apple Volume Licensing Program. The suggested retail price is U.S. $129.00. Existing Apple PhotoFlash customers can upgrade for U.S. $39.95, plus shipping and handling by calling 800-769-2775 x5929.

Customers who purchased Apple PhotoFlash 1.0 or Apple PhotoFlash 1.1 between August 3, 1994 and December 3, 1994 can receive an upgrade for only the cost of shipping and handling by submitting an order form and proof of purchase that is a dated, itemized sales invoice showing the purchase of Apple PhotoFlash 1.0 or Apple PhotoFlash 1.1 between these dates.

For more information about the upgrade program, or to obtain an order form, customers can call 800-769-2775 x5929. Apple plans to introduce international English and French versions of Apple PhotoFlash 2.0 later in 1994 and a Kanji version in 1995.

Headquartered in Cupertino, California, Apple Computer, Inc., (NASDAQ: AAPL) develops, manufactures and markets personal computer, server and personal interactive electronic systems, software and interactive services for use in business, education, the home, science, engineering, and government. A recognized pioneer and innovator in the personal computer industry, Apple does business in more than 120 countries.

Apple, the Apple logo and Macintosh are registered trademarks; PowerBook, PhotoFlash and QuickTake are trademarks of Apple Computer, Inc. PowerPC is a trademark of International Business Machines Corporation, used under license therefrom. Additional company and product names may be trademarks or registered trademarks of the individual companies and are respectfully acknowledged.

Mary Devincenzi, (408) 974-4562
Apple Computer, Inc.
CIC Computing Classes

CIC offers a variety of computing courses for the professional development of Laboratory employees. The courses listed in Table 1 will meet at the time and the date shown. Time and date for the courses in Table 2 are not known at this time. Courses in Table 3 run continuously from month to month.

Course Registration

To register: (1) check the box beside the appropriate course, (2) complete the Enrollment Information section located on the back of this form, and (3) follow the mailing instructions also on the back of this form. Submittal of a Course Registration form does not guarantee participation in an advertised class, but it is the only way to get into the queue for notification of upcoming classes. Classes are conducted in a secure area unless noted; uncleared participants require escorts. Call the Training Coordinator at 667-9399 for more information.

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Instructor</th>
<th>Cost</th>
<th>Dates/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Programming (Beginning)</td>
<td>Marilyn Nelson</td>
<td>$1000-$1400</td>
<td>10/31/94 to 11/4/94 8:30-5:00</td>
</tr>
<tr>
<td>C Programming (Advanced)</td>
<td>Michael Chase</td>
<td>$1000-$1400</td>
<td>12/5-9/94 8:30-5:00</td>
</tr>
<tr>
<td>DCE Applications Programming</td>
<td>TBA</td>
<td>$1000-$1500</td>
<td>11/28/94 to 12/2/94 8:30-5:00</td>
</tr>
<tr>
<td>FrameMaker (Basic)</td>
<td>Phoebe Wright (Aquila)</td>
<td>$700-$900</td>
<td>11/7-8/94 or 11/9-10/94 8:30-4:00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Instructor</th>
<th>Cost</th>
<th>Dates/Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris 1.1.1 (SunOS 4.1.X) System Admin. (Beginning)</td>
<td>Sun Microsystems, Inc. Expert</td>
<td>$1500-$1800</td>
<td>TBA (5 days)</td>
</tr>
<tr>
<td>Solaris 2.X System Admin. (Level I or II)</td>
<td>Sun Microsystems, Inc. Expert</td>
<td>$1500-$1800</td>
<td>TBA (5 days)</td>
</tr>
<tr>
<td>C Programming (Beginning)</td>
<td>Marilyn Nelson</td>
<td>$1000-$1400</td>
<td>10/31/94 to 11/4/94 8:30-5:00</td>
</tr>
<tr>
<td>C Programming (Advanced)</td>
<td>Michael Chase</td>
<td>$1000-$1400</td>
<td>12/5-9/94 8:30-5:00</td>
</tr>
<tr>
<td>DCE Applications Programming</td>
<td>TBA</td>
<td>$1000-$1500</td>
<td>11/28/94 to 12/2/94 8:30-5:00</td>
</tr>
<tr>
<td>FrameMaker (Basic)</td>
<td>Phoebe Wright (Aquila)</td>
<td>$700-$900</td>
<td>11/7-8/94 or 11/9-10/94 8:30-4:00</td>
</tr>
</tbody>
</table>

Note: Detailed course descriptions are provided on the following pages.
Name ________________________________

Phone ________________ Z-Number ________________

Group ________________ Mail Stop______________

Program Code* __________ Cost Code*____________

Group Leader Signature __________________________

* Enter program code and cost code for all courses. If you need to withdraw from a class fewer than 5 working days before the class is scheduled to begin, your group will still be charged. Substitutes may be sent, but please let the CIC Division Training, Development, and Coordination Office (667-9399) know who your substitute will be.
ALL-IN-1 Electronic Mail System Training

Prerequisite: ICN password and access to the OFVAX ALL-IN-1 mail system.

Location: TA-3/SM-261/Rm P180 (Otowi Bldg./open area)

Enrollment: Minimum 6/Maximum 8

Topics: Accessing OFVAX; Accessing On-line Training; Creating a Distribution List; Creating a Nickname; Creating and Sending Mail; Reading and Answering Mail; Sending Messages to FAX or PAGES; Sending Paper Mail; Editing Message Headers; Using the Spell Checker; Granting Mail Access; Setting Mail User; Forwarding Mail; Editing the Text of a Memo; Setting the Ruler; Printing a Document; Using Standard Mail Folders; and Managing Files.

C Programming (Beginning)

Prerequisite: Proficiency with another high-level programming language.

Enrollment: Minimum 12/Maximum 18

Topics: Introduction and Fundamentals; Basic Semantic Constructs; Base Level I/O with C; The Preprocess-Compilation Environment; Operators, Data Types, and Storage Classes; Control Flow Constructs; Conditional Constructs; Higher Level Data Constructs in C; File I/O; UNIX Software Tools; and POSIX System Calls.

C Programming (Advanced)

Prerequisite: Beginnin C Programming or equivalent experience.

Enrollment: Minimum 12/Maximum 18

Topics: Data Structures, Algorithms, and OOP; Advanced Clinic for C Programmers; ANSI C Recommendation X3.159; C and ANSI C War Stories; Data Structure and Assessment of Algorithms; Arrays; Structures; Unions; Stacks; Queues; Linked-Lists; Recursive Functions; Binary Trees; Hashing; File Organizations Using the C Runtime Library; Standard Interprocess Communication Mechanisms; Introduction and Overview of AT&T’s C++ 3.0; and References for Periodicals, Journals, and Texts.

Distributed Computing Environment (DCE) Applications Programming

Prerequisite: C programming skills and DCE component services and their functions.

Location: TA-3/SM-200/Rm 210

Enrollment: Minimum 10/Maximum 16

Topics: An Overview of DCE Application Development; The Interface Definition Language (IDL); Writing a Server; Writing a Client; Error Handling; Using Threads; Using DCE Distributed Time Service; Using Security; Using Objects; and Using the DCE Director Service.
FrameMaker (Basic)

Prerequisite: None.
Location: TA-3/SM-200/Rm210
Enrollment: Minimum 6/Maximum 8

Topics: Editing and Formatting Text; Applying/Modifying/Creating Paragraph and Character Formats; Search/Change Text & Formats; Using Thesaurus and Spell Checker; Creating/Editing Graphics Using Draw Tools; Using FrameMaker Templates and Clip Art Files; Tables and Table Formatting; Anchored Frames (adding illustrations); Changing Basic Layout of a Document on a Master Page; and Creating Custom Documents.

Note: This course will be presented on the Sun workstation platform.

ICN2 Split and How It Will Affect You—Because of the lack of attendance at the last scheduled presentation of this topic, the advertised seminar to be held on November 10 is questionable. PLEASE CALL TO VERIFY THAT IT WILL HAPPEN. At this point in time there will be NO seminar unless a minimum of ten requests are received.

The ICN2 Network Split and How It Will Affect You

Prerequisite: Current usage of the Integrated Computing Network (ICN).
Enrollment: 42 maximum
Topics: Overview of the New ICN2 Network Environment; ICN to ICN2 Differences: CFS changes, UNICOS changes, FOCUS changes, PAGES Changes, and Authentication Changes; Benefits of the New Environment; and Tradeoffs of the New Environment.

Solaris 1.1 System Administration (Level I & II)

Prerequisite: Useful UNIX skills, familiarity with Sun OpenWindows, vi or Text Editor skills, and Sun mouse/keyboard skills.
Enrollment: Minimum 10/Maximum 16
Topics: Sun Machine Terminology/Types; Format Utility for Partitioning; Installing Solaris 1.X Operating System with SunInstall; System Start Up and Shut Down; Customizing a Kernel; Adding Users; Ethernet Hardware/Networking Commands; Configure NFS Server; Add/Remove Diskless Clients; Backup and Restore Files; Repair Bad Blocks; and Manage Defects List.

Solaris 2.X System Administration (Level I & II)

Prerequisite: Useful UNIX skills, familiarity with Sun OpenWindows, vi or Text Editor skills, and Sun mouse/keyboard skills.
Enrollment: Minimum 10/Maximum 16
Topics: Perform Quick Install of Solaris 2.X; Client/Server Environment; OpenWindows; Change Run Levels; Navigate through Solaris File System; vi; Change Permissions; Monitor System Access; Restrict Root Access; Set and Use Shell Variables; Customize Start Up Files; Monitor Disk Space/Usage; and Using Utilities for Process Control, Locating Files, and Automating Maintenance Tasks.

Beginning UNIX—This course has been restructured to address generic UNIX information. There is no longer a focus on Sun operating systems and tools. Additional topics are being added. This course will be presented the third week of each month, so long as a demand continues.

UNIX (Beginning)

Prerequisite: None
Enrollment: 9 maximum
Topics: Getting Started; The UNIX File System; Manipulating Files; Customizing Your Environment; The C-Shell; Editing and Writing with vi; and Using the Network.
The Customer Service Group (CIC-6) offers training for users of Laboratory information systems. The CIC-6 courses offer training for a variety of personnel including property administrators; group secretaries; training coordinators; budget analysts; group leaders; or anyone needing to access training records, property records, costs, employee information, travel, chemical inventories, etc. Refer to the table on pages 22 and 23 for specific information about courses currently offered.

Course Registration
You must have a valid “A” or “U” level ICN password before taking any of the courses shown in the table. To register for a course, call CIC-6 Training, Development, and Coordination section at 667-9444. You will be sent a registration form to be completed and returned.
### IN THE CLASSROOM

<table>
<thead>
<tr>
<th>Course Title</th>
<th>Date</th>
<th>Time</th>
<th>Cost</th>
<th>Course Number</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Automated Chemical Inventory System (ACIS):</strong></td>
<td>Scheduled Upon Request</td>
<td>$410</td>
<td></td>
<td>Course #7480</td>
</tr>
<tr>
<td>Participants receive hands-on instruction to update the status (end-user, location, quantity) of chemical containers. Participants will also learn to generate chemical inventory reports by chemical name, end-user, location, and organization.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Budget Computing System (BUCS):</strong></td>
<td>Scheduled Upon Request</td>
<td>$410</td>
<td></td>
<td>Course #3527</td>
</tr>
<tr>
<td>This training is an introduction to the Budget Computing System (BUCS). Students practice generating “quick reports” and reports requiring parameter files. An introduction and demonstration of (no “hands-on”) allocating and forecasting procedures are given during the three-hour session.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Directory Information System (DIS):</strong></td>
<td>Scheduled Upon Request</td>
<td>$410</td>
<td></td>
<td>Course #7072</td>
</tr>
<tr>
<td>Lab-wide customers responsible for maintaining the Laboratory directory in the Employee Information System will receive hands-on instruction to update Laboratory employees, update and add non-Laboratory employees, retrieve location and address information for any employee, and print reports.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Employee Development System - Basic Training (EDS I):</strong></td>
<td>11/9/94</td>
<td>8:30 - 12:00</td>
<td>$410</td>
<td>Course #5289</td>
</tr>
<tr>
<td>The course provides hands-on instruction to request course enrollment, use the on-line course catalog, retrieve training transcripts, and assign EDS authorities. The student will learn to create courses, add students to the courses, and generate several training reports.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Employee Development System - Training Plans (EDS II):</strong></td>
<td>11/30/94</td>
<td>8:30 - 12:00</td>
<td>$410</td>
<td>Course #7155</td>
</tr>
<tr>
<td>Participants receive hands-on instruction to create and maintain training plans, assign assignment codes, and generate training plan reports. Attendees must have prior training in the Employee Development System (course #5289).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Facilities Project Information/Work Orders (FPI/WO):</strong></td>
<td>Scheduled Upon Request</td>
<td>$410</td>
<td></td>
<td>Course #6996</td>
</tr>
<tr>
<td>Lab-wide users with a need to view the status of work orders and tickets in their organizations will receive hands-on instruction to request, print, and review work order, ticket and project summary information reports.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Financial Management Information System (FMIS):</strong></td>
<td>11/22/94</td>
<td>8:30 - 12:00</td>
<td>$410</td>
<td>Course #8338</td>
</tr>
<tr>
<td>Participants receive hands-on instruction to “explode” and “transfer” through the costs, allocations, and outstanding commitments screens. In addition, participants will create/review reports, access the Information Manager Utility for printing reports, and learn how to assign authorities in the system.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hazardous Materials Transfer Tracking System for Non-Radioactive Material (HMTTS/NRAM):</strong></td>
<td>Scheduled upon request</td>
<td>$410</td>
<td></td>
<td>Course # 7907</td>
</tr>
<tr>
<td>Participants receive hands-on instruction to create, update, and print the non-RAM Hazardous Materials Transfer Form (HMTF). Attendees must have completed “Completing the HMTF for Non-RAM,” course #7512, sponsored by HS-8.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Hazardous Materials Transfer Tracking System for Radioactive Material (HMTTS/RAM):</strong></td>
<td>Scheduled Upon Request</td>
<td>$410</td>
<td></td>
<td>Course #7993</td>
</tr>
<tr>
<td>Participants receive hands-on instruction to create, update, and print the Radioactive Materials Transfer Form (RMTF). Information about the non-RAM Hazardous Materials Transfer Form (HMTF) is included. This course is appropriate for people who fill out both RAM and Non-RAM forms. Attendees must have completed “Completing the RMTF,” course #7517, sponsored by HS-8.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Title</td>
<td>Date</td>
<td>Time</td>
<td>Cost</td>
<td>Course Number</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------</td>
<td>---------------</td>
</tr>
<tr>
<td>Information Access System (IAS):</td>
<td>Scheduled upon request</td>
<td>$410</td>
<td></td>
<td>Course #5223</td>
</tr>
<tr>
<td>Students receive hands-on instruction to search, retrieve, and print information in the Financial Management Handbook, material safety data sheets (MSDS’s), and Department of Energy (DOE) orders. As time permits, other options in the Information Access System (IAS), such as Update News, meetings, visitors, colloquia, and PC Help are demonstrated.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Introduction to Lab-Wide Systems:</td>
<td>11/28/94</td>
<td>8:30 - 10:30</td>
<td>No Fee</td>
<td>Course #2900</td>
</tr>
<tr>
<td>This introductory class is an overview of Lab-wide information systems. During the 90 minute session, students learn how to become Lab-wide system users and access those systems. Several Lab-wide systems are demonstrated and their functions are discussed. Optional hands-on exercises are offered at the end of class.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property Accounting, Inventory, &amp; Reporting (PAIRS):</td>
<td>11/29/94</td>
<td>8:30 - 12:00</td>
<td>$410</td>
<td>Course #7411</td>
</tr>
<tr>
<td>This course is for Property Administrators (PA’s) and Lab-wide customers with a need to view property record information. PA’s receive hands-on instruction to update property element and location information. All participants will receive hands-on instruction to generate and print a variety of property reports. The MAT-2 Property Administrators course is recommended before PA’s attend this course.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secretarial/Contract Services (SE):</td>
<td>Scheduled Upon Request</td>
<td>$410</td>
<td></td>
<td>Course #7481</td>
</tr>
<tr>
<td>Students receive hands-on instruction for requesting temporary secretarial services and for entering VOLT attendance on-line. Students will also learn to review notifications and use the Information Manager utility for printing reports. A training database will be used for the class.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Signature Authority System (SAS):</td>
<td>11/17/94</td>
<td>8:30 - 12:00</td>
<td>$410</td>
<td>Course #7582</td>
</tr>
<tr>
<td>Managers or their designees receive instruction to assign, view, and change signature authorities (purchase request, chemical purchase, and handling hazardous material). Participants will also learn how to generate and print authority reports for their organizations.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STORES:</td>
<td>Scheduled upon request</td>
<td>$410</td>
<td></td>
<td>Course #3529</td>
</tr>
<tr>
<td>Participants receive hands-on instruction to search for an item in the on-line catalog by key word, part number, or exact name. Participants learn how to select items from the catalog, and place, change and cancel an order. Several methods for reviewing orders are also taught including reviewing an order in detail, scanning all orders, and reviewing back-orders.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Travel Reporting Information Planning System (TRIPS):</td>
<td>11/30/94</td>
<td>1:30 - 4:30</td>
<td>$410</td>
<td>Course #4369</td>
</tr>
<tr>
<td>Class participants receive hands-on instruction to prepare travel requests (TRs) on-line and learn the print, revise, and cancel options. The participants also learn how to use the on-line approval function. The various reports available in TRIPS-II are reviewed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All-In-One Electronic Mail System</td>
<td>11/18/94</td>
<td>8:15 - 12:00</td>
<td>$410</td>
<td>Course #6882</td>
</tr>
<tr>
<td>Participants receive hands-on instruction to create, read, and print electronic mail. Participants also learn how to edit mail, create distribution lists, send mail to a FAX machine, and grant mail access to others. Prerequisite: an ICN password and an account on the OFVAX.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The LANL Research Library provides training for its users, free of charge, for the specialized databases owned by the library. Please call the Research Desk at 7-5809 or e-mail ref@lanl.gov for reservations, to arrange a special session or tour, or for more information. Sessions begin at times and dates indicated below. Each session is 30 minutes, except the Gopher class which is 2 hours.

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Subject Matter</th>
</tr>
</thead>
<tbody>
<tr>
<td>11-1-94/1:00</td>
<td>Materials/Engineering Resources</td>
</tr>
<tr>
<td>11-3-94/1:00</td>
<td>Biosciences and Biotechnology Resources</td>
</tr>
<tr>
<td>11-8-94/1:00</td>
<td>Chemical Resources</td>
</tr>
<tr>
<td>11-10-94/1:00</td>
<td>Multidisciplinary Science Citation Index</td>
</tr>
<tr>
<td>11-15-94/1:00</td>
<td>Earth and Environmental Systems</td>
</tr>
<tr>
<td>11-17-94/1:00</td>
<td>High-Performance and Supercomputing</td>
</tr>
<tr>
<td>11-17-94/10:00</td>
<td>Information Sources on the Internet via Gopher</td>
</tr>
<tr>
<td>11-22-94/1:00</td>
<td>Business/Technology Transfer Resources</td>
</tr>
<tr>
<td>11-23-94/11:00</td>
<td>Melvyl (University of California catalog and</td>
</tr>
<tr>
<td></td>
<td>associated databases)</td>
</tr>
<tr>
<td>11-29-94/1:00</td>
<td>Physics/Weapons Resources</td>
</tr>
<tr>
<td>11-30-94/11:00</td>
<td>Melvyl (University of California catalog and</td>
</tr>
<tr>
<td></td>
<td>associated databases)</td>
</tr>
</tbody>
</table>
### Change Control for November 1994

#### Deletions

INITFCL ............................................................................................................................... 24

#### Changes

- Change Control CFS Nodes .............................................................................................. 24
- COST (UNICOS) .................................................................................................................. 24
- COSTPP (UNICOS) ............................................................................................................. 25
- PPAGES (DEC_Alpha, DOS, HP, IBM, IRIS, NeXT, Sun, UNICOS, ULTRIX) ............... 25

#### System Information

- Machine Delta ..................................................................................................................... 26
- Machine Epsilon/E/S .......................................................................................................... 26

#### Documentation

- Information About Change Control .................................................................................... 28
- Online Information ............................................................................................................ 29

#### December Deadline

- CCF Machine Availability and Downtime .......................................................................... 30

---

## Schedule for Change Control

<table>
<thead>
<tr>
<th>Date</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 1 (First Tuesday)</td>
<td>New or changed software is available in experimental (X) files on CFS for testing. This initial testing period is for uncovering problems in the software before the software is put into production. If you find a problem, please call the ICN Consulting Office at (505) 667-5745.</td>
</tr>
</tbody>
</table>
| November 8 (Second Tuesday) | The changes become production version on,  
  - Machines **rho, epsilon/e/s** and **zeta** (UNICOS)  
  - Distributed processor **beta** (ULTRIX)  
  - Distributed processor **ccvax** (VMS)  |
| November 15 (Third Tuesday) | If no problems are reported to the ICN Consulting Office (505) 667-5745, changes are installed on  
  - Machines **delta and gamma** (UNICOS) |

Note: A stop sign in front of a title is significant:

**STOP** = incompatible changes; please read!
This section lists utilities and services that are targeted for deletion or have already been removed from the systems listed. If removal of these files or services will cause you a problem, please contact Marj Johnston at (505) 667-5745 or send e-mail to mdj@lanl.gov immediately.

### Deletions

**INITFCL (UNICOS)**
- **Function**: Initializes or changes your login environment to run FCL as your login shell.
- **Change**: In December 1994 INITFCL will be removed.

### Changes

**CFS Nodes**
In January 1995 the CFS nodes `/unixx`, `/vaxx`, and `/sunx` will be moved to subdirectories under the `/ccx` node. The new subdirectories are already defined as `/ccx/unix`, `/ccx/vax`, and `/ccx/sun`.

**COST (UNICOS)**
- **Function**: Produces a monthly or yearly summary of CCF charges for a specified user, group, program, division, or charge code.
- **Change**: COST now picks up a file that contains user numbers vs. user names as a UNIX text file, rather than the standard text file version; this reduces running time. The notes at the end of the output from COST now contain the fiscal 1995 rates for CCF services.
- **X File Access**: On CFS as: `/unicosx/bin/costx` for all UNICOS machines.
- **Online Documentation**: To display the man page (dated 10/93), enter: `man cost` For the built-in help package, enter: `cost` and respond with `help` at the prompt.
### COSTPP (UNICOS)

**Function**
Annotates output files produced by AUTOSUM and COST with user names, where appropriate.

**Change**
COSTPP now uses a native UNIX text version of the file that contains user numbers vs. user names, instead of the corresponding standard text file version.

**X File Access**
On CFS as: `/unicosx/bin/costppx` for all UNICOS machines.

**Online Documentation**
To display the man page (dated 4/92), enter: `man costpp`

---

### PPAGES (DEC_Alpha, DOS, HP, IBM, IRIS, NEXT, SUN, UNICOS, ULTRIX)

**Function**
Submits jobs to the Print and Graphics Express Station (PAGES) for paper and film output.

**Change**
If an error occurs on UNIX systems an `exit` code of 1 is returned. On VAX/VMS systems, an `exit` code of 44 is returned. In such cases, `ppages` will terminate abnormally when called from an external program.

Now when `ppages` is running on a machine within the administrative partition it will allow a user to specify the output marking Unclassified Sensitive (US).

The number of files allowed on the command line has increased to 999 to better accommodate video requests.

Internal code changes have been made to fix bugs and to meet requirements for ICN2. Users will now have to make sure that these environment variables are set:

- `ICNZ <your z-number>`
- `ICNSL <default marking for this session>` i.e. U, P, C or S
- `ICNCHARGE <your cost center and program code>`

**X File Access**
On CFS as: `/ccx/alpha_osf/ppages.tar` for DEC_Alpha machines.
On CFS as: `/ccx/dos/ppages.exe` for PC/DOS machines.
On CFS as: `/ccx/hp/ppages.tar` for Hewlett-Packard workstations.
On CFS as: `/ccx/ibm_rs6000/ppages.tar` for IBM RS6000 workstations.
On CFS as: `/ccx/next/ppages.tar` for NeXT workstations.
On CFS as: `/sunx/ppages.tar` for Sun workstations.
On CFS as: `/unicosx/bin/ppagesx` for Machines Delta and Epsilon/E/S.
On CFS as: `/unicosx/bin7/ppagesx` for Machine Rho.
On CFS as: `/unicosx/bin7c/ppagesx` for Machine Gamma.
On CFS as: `/unixx/ppages.tar` for the ULTRIX Machine Beta.

**Online Documentation**
To display the man page (dated 11/94), enter: `man ppages`
This section provides information and a record of changes to the ICN operating systems. When changes are announced here, they may already be included in the production versions of the indicated operating systems and machines. Most of the changes are strictly internal to the systems and should not affect users. However, if you detect a problem, please call the Consulting Office at (505) 667-5745, or send electronic mail to consult@lanl.gov.

### Machine Delta

The conversion of Machine Delta to UNICOS Version 7.0 is scheduled for Saturday, November 5, 1994. You are reminded that this version of UNICOS supports no SIMP services.

Please monitor news updates on all the Open and Secure machines as well as each new issue of the ICNchanges.

For more information, contact Ray Miller at (505) 665-3222 or e-mail at rdm@lanl.gov or contact the consultants at (505) 667-5745 or e-mail consult@lanl.gov.

### Machine Epsilon/E/S

The conversion of Machine Epsilon/E/S to UNICOS Version 7.0 is scheduled for Saturday, December 3, 1994. At this point, no SIMP services will be supported on any Cray hardware in either the Open or the Secure Partition.

Please monitor news updates on all the Open and Secure machines as well as each new issue of the ICNchanges.

For more information, contact Ray Miller at (505) 665-3222 or e-mail at rdm@lanl.gov or contact the consultants at (505) 667-5745 or e-mail consult@lanl.gov.
New and Updated Man Pages

The following online information has been added or updated.

**UNICOS Man Pages**

To access a UNICOS man page, enter: `man command_name`, where `command_name` is the name of the command, library, routine, or utility whose man page you wish to view.

### Man Page Description

<table>
<thead>
<tr>
<th>Man Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ppages</strong></td>
<td>PPAGES executes without interaction. This utility accepts input parameters, then sends the file through the network to PAGES for processing. On ULTRIX and Sun UNIX systems, if no filename is given, or if the argument '-' is given, <strong>ppages</strong> reads from the standard input.</td>
</tr>
</tbody>
</table>

To create ASCII files of the Cray man pages, use one of the following commands to remove the special characters for bold and underlining:

- UNICOS 7.0: `man command_name | col -bx > filename`
- UNICOS 6.0: `man -bulo command_name > filename`

Barbara Ritchie (bxr@lanl.gov), (505) 667-7275
Communication Arts and Services (CIC-1)
ICN Change Control is the set of procedures that coordinates changes in the ICN to ensure quality control and smooth operation and to avoid introducing additional problems. In an environment as dynamic as the ICN, control must be imposed on the scope and timing of changes that involve many components. Please report any problems as soon as they occur by calling the Consulting Office at (505) 667-5745.

The following CFS nodes are used for software that is maintained or announced through Change Control procedures. The files under /unicosx are deleted the last Friday of each month because these experimental versions become the production versions on all machines by the third Tuesday of the month. The other nodes keep the most recent versions of their respective software.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/unicosx/bin</td>
<td>executables for ymp CIC-Division supported software on UNICOS Version 6.</td>
</tr>
<tr>
<td>/unicosx/bin7</td>
<td>executables for Machine Rho, UNICOS Version 7.</td>
</tr>
<tr>
<td>/unicosx/bin7c</td>
<td>executables for Machine Gamma, UNICOS Version 7c.</td>
</tr>
<tr>
<td>/unicosx/ubin</td>
<td>executables for user-supported software on UNICOS Version 6.</td>
</tr>
<tr>
<td>/unicosx/lib7c</td>
<td>libraries for Machine Gamma, UNICOS Version 7c.</td>
</tr>
<tr>
<td>/unicosx/ulib</td>
<td>user-supported libraries for UNICOS Version 6.</td>
</tr>
<tr>
<td>/unicosx/xlib</td>
<td>X-mode libraries for UNICOS CRAYs, UNICOS Version 6.</td>
</tr>
<tr>
<td>/unicosx/ylib</td>
<td>Y-mode libraries for UNICOS CRAYs, UNICOS Version 6.</td>
</tr>
<tr>
<td>/unixx</td>
<td>tar files for VAX/ULTRIX machines.</td>
</tr>
<tr>
<td>/vaxx</td>
<td>backup save sets for VAX/VMS machines.</td>
</tr>
<tr>
<td>/sunx</td>
<td>tar files for Sun/UNIX workstations.</td>
</tr>
<tr>
<td>/ccx/...</td>
<td>workstation directories of current versions of application in tar or backup save sets as applicable.</td>
</tr>
<tr>
<td>/ccx/alpha_osf</td>
<td>tar files for DEC Alpha OSF/1 machines.</td>
</tr>
<tr>
<td>/ccx/alpha_vms</td>
<td>backup save sets for DEC Alpha VMS machines.</td>
</tr>
<tr>
<td>/ccx/convex</td>
<td>tar files for Convex machines.</td>
</tr>
<tr>
<td>/ccx/dec_risc</td>
<td>tar files for DEC RISC workstations.</td>
</tr>
<tr>
<td>/ccx/dos</td>
<td>executables for PC/DOS machines.</td>
</tr>
<tr>
<td>/ccx/hp</td>
<td>tar files for Hewlett-Packard workstations.</td>
</tr>
<tr>
<td>/ccx/ibm_rs6000</td>
<td>tar files for IBM RS6000 workstations.</td>
</tr>
<tr>
<td>/ccx/mac</td>
<td>binhex (.hqx) or MacBinary (.mbin) files for Macintosh computers.</td>
</tr>
<tr>
<td>/ccx/next</td>
<td>tar files for NeXT workstations.</td>
</tr>
<tr>
<td>/ccx/sgi</td>
<td>tar files for Silicon Graphics workstations.</td>
</tr>
<tr>
<td>/ccx/solaris</td>
<td>tar files for Sun Solaris workstations.</td>
</tr>
</tbody>
</table>

If problems are discovered during the cycle, defective hardware or software is corrected, replaced, removed, or backed off.
DECEMBER DEADLINE

The deadline for articles for the December Change Control is 8:00 am, Monday, November 21, 1994. Please submit items to bulletin@lanl.gov

Note: please submit your articles on-time because of the Holiday schedule.
## CCF Machine Availability and Downtime

<table>
<thead>
<tr>
<th>Machine Name(s)</th>
<th>Machine Type</th>
<th>Operating System</th>
<th>Security Partition</th>
<th>System Availability (Sept. 1994)</th>
<th>Scheduled Downtime**</th>
</tr>
</thead>
<tbody>
<tr>
<td>delta*</td>
<td>CRAY Y-MP8/128</td>
<td>UNICOS 7.0</td>
<td>Secure</td>
<td>99.6%</td>
<td>Nov. 15/1900-2300</td>
</tr>
<tr>
<td>epsilon/e/s*</td>
<td>CRAY Y-MP8/128</td>
<td>UNICOS 6.1</td>
<td>Secure</td>
<td>100%</td>
<td>Nov. 8/1900-2300</td>
</tr>
<tr>
<td>rho</td>
<td>CRAY Y-MP8/64</td>
<td>UNICOS 7.0</td>
<td>Open</td>
<td>97.4%</td>
<td>Nov. 1 &amp; 29/1900-2300</td>
</tr>
<tr>
<td>zeta</td>
<td>CRAY Y-MP8/2-64</td>
<td>UNICOS 8.0</td>
<td>Secure</td>
<td>100%</td>
<td>Nov. 9/1900-2300</td>
</tr>
<tr>
<td>gamma</td>
<td>CRAY Y-MP/M98/82048</td>
<td>UNICOS 7.0</td>
<td>Open</td>
<td>100%</td>
<td>Nov. 22/1900-2300</td>
</tr>
<tr>
<td>pi***</td>
<td>CRAY Y-MP EL92/1-256</td>
<td>UNICOS 8.0</td>
<td>Open</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>cluster</td>
<td>IBM Workstation Cluster</td>
<td>AIX</td>
<td>Open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>beta</td>
<td>VAX 6320</td>
<td>ULTRIX</td>
<td>Open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCVAX</td>
<td>VAX 6410</td>
<td>VMS</td>
<td>Open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OFVAX</td>
<td>VAX 6410</td>
<td>VMS</td>
<td>Open</td>
<td></td>
<td></td>
</tr>
<tr>
<td>canyon</td>
<td>Thinking Machines Corp. CM-200</td>
<td>SunOS</td>
<td>Secure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tres</td>
<td>Thinking Machines Corp. CM-200</td>
<td>SunOS</td>
<td>Secure</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* In the IP networks, Cray machines are addressed by their Greek names.
** Additional downtime for the Cray machines may occur as a result of Network Dedicated Systems Time (NDST). The schedule for possible NDST is from 0600-0700, Monday through Friday. Should NDST become necessary, a message listing the scheduled downtime will be broadcast on the applicable machines before the actual downtime occurs. For additional information contact the shift supervisor at (505) 667-4584. All times listed are Mountain Time.
*** Access restricted.

## Questions About Announced Changes?

All scheduled downtime will be broadcast on the machine before the downtime. For up-to-date machine status and scheduled downtime call: CCF Status Message (505) 667-5588

Only on Delta or Epsilon/E/S, type **focus dst**

Please report any problems as soon as they occur.

## Publication Information

ICNchanges Editor/Publication Coordinator
Barbara Ritchie (CIC-1)
Mail Stop B295
Telephone (505) 667-7275

Change Control Coordinator
Marjorie Johnston (CIC-6)
Mail Stop B252
Telephone (505) 667-7309
To order free software, fill in your name, group, and mail stop; check the software you would like to have and mail this form to

Free Software
Desktop Support Center (CIC-2) MS D445

Name_________________________ Group________________
Mail Stop____________________ Z-Number_________________

Please send the correct number of replacement high-density diskettes with your request. This software can also be retrieved via FTP from ftp.lanl.gov. (The Macintosh software is in Mac Binary format.)

Macintosh Software Order Form

_____ FREEWARE DISKETTE (Include 1 high-density disk.)
This diskette contains the following software:

  **Alias Finder**: Quickly finds the original of an alias when the alias is dragged on top of the Alias Finder icon.

  **Color Cursor**: INIT turns arrow cursor into the Apple colors.

  **Disinfectant**: Virus protection for the Macintosh.

  **Disk Copy**: Creates copies of diskettes using one floppy drive.

  **Mode 32**: Allows Macintosh II, IIx, IICx, and SE/30 to access more than 8 Mbytes of memory with System 7.0 or later.

  **SCSI Probe**: Shows connected devices on the SCSI bus.

  **StuffIt Expander**: Unstuffs BinHex 4.0, StuffIt, and other types of compressed files.

Note: THE FOLLOWING TWO APPLICATIONS COME WITH SYSTEM 7.5:

  **Extensions Manager**: Allows selection of which INITs to load.

  **SuperClock**: Puts a clock in the upper right corner of your Macintosh.

_____ INTERNET DISKETTE (Include 1 high-density disk.)
This diskette contains the following software:

  **Fetch**: Easy-to-use for FTPing files from FTP archives.

  **TurboGopher**: Gopher client application for the Macintosh.

  **StuffIt Expander**: Unstuffs BinHex 4.0 and other types of compressed files.

_____ HARDWARE SYSTEM UPDATE 3.0 (Include 2 high-density disks.): Update for System 7.1

_____ MACINTOSH SYSTEM 7.5 (Include 9 high-density diskettes.)
Indicate number of systems on which this System 7.5 will be used: ______

_____ SYSTEM 7.5 POWERTALK AND QUICKDRAW GX. (Include 4 high-density diskettes.)
Note: We recommend that you do not install these parts of System 7.5 unless you have a specific need to do so.

IBM Software Order Form (Include 1 high-density disk for each selection.)
Note: More IBM PC Freeware will be added next month.

_____ DATA PHYSICIAN: Virus detection programs.

_____ PC GOPHER III: Gopher client software for DOS systems.
   Includes PKZip: File compression and decompression program.

_____ HGOPHER: Gopher client software for Microsoft Windows systems.
   Includes PKZip: File compression and decompression program.

_____ LIST: Searches for character strings within files.
   Includes Machinist’s Assistant: Calculates cut angles, etc., for machinist work.


## Accessing Computing Machines through the ICN

This table shows how to access open machines on the ICN through MICOM lines, TCP/IP hosts, and DECnet hosts. Additional machines outside the ICN are accessible through TCP/IP and DECnet. To access any of these machines, except for LIS, you must first establish an ICN account, which includes obtaining an ICN password and registering as an ICN user (contact the CIC Customer Service Center for details).

Example: Suppose you want to access the REGISTER machine from MICOM. By referring to the table, you can see that the appropriate command to enter is tig. Once you connect to the tig, enter your ICN usernumber and password as prompted. At the tig prompt (tig>) enter register and login to the register machine.

<table>
<thead>
<tr>
<th>TO</th>
<th>FROM</th>
<th>MICOM Lines</th>
<th>TCP/IP Hosts: (BETA, CCVAX, IBM Cluster IOVAX, OFVAX, REGISTER, UNICOS, ACL Hosts, etc.)</th>
<th>DECnet Hosts: (BETA, CCVAX, IOVAX, OFVAX, etc.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hosts reachable from MICOM Lines: (BETA, CCVAX, IOVAX, OFVAX, STORES, TYMNET, LIS)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIG</td>
<td>hostname</td>
<td>TELNET</td>
<td>TELNET DIG</td>
<td>TELNET DIG</td>
</tr>
<tr>
<td>MICOM Lines</td>
<td>hostname</td>
<td>MICOM</td>
<td>SET HOST</td>
<td>SET HOST</td>
</tr>
<tr>
<td>TCP/IP Hosts</td>
<td>TELNET</td>
<td>hostname</td>
<td>hostname</td>
<td>hostname</td>
</tr>
<tr>
<td>(e.g., TIG)</td>
<td>TELNET</td>
<td>hostname</td>
<td>TELNET</td>
<td>TELNET DIG</td>
</tr>
<tr>
<td>DECnet Hosts</td>
<td>SET HOST DIG</td>
<td>TELNET</td>
<td>SET HOST DIG</td>
<td>SET HOST</td>
</tr>
<tr>
<td></td>
<td>MICOM</td>
<td>hostname</td>
<td>MICOM</td>
<td>hostname</td>
</tr>
</tbody>
</table>

Dialup access to the ICN is available through the Terminal Internet Gateway (tig). The tig is a gateway to the internet and allows you to telnet to ICN machines as well as other machines. Configure your modem and terminal for 8 bit, no parity, one stop bit. Based on your modem, select the appropriate number listed in the table to dial into the tig. Then enter your ICN usernumber and password as prompted. At the tig prompt (tig>) enter a machine name or IP address.

Report problems to the Network Control Center at 667-7423 Monday through Friday, 6 am to 6 pm or at 667-4585 during non-business hours.

## Accessing the ICN through Dialup Modem

Type of Access | Phone Numbers
---|---
Racal Vadic Modems 300 through 2400 | (505) 667-9020, 9021 (Number of Lines 32)
Microcom Modems 300 through 38400 | (505) 667-9022, 9023, 9024, and 9025 (800) 443-1461 (Number of Lines 48)

Note: Use the second phone number if the first does not answer properly.

Revised April 1994
INTEGRATED COMPUTING NETWORK (ICN) VALIDATION REQUEST

To access ICN Computing resources, please complete all parts of this form that apply to you, including "Special Requirements." Mail your completed application to:
ICN Password Office (PWO)  
Mail Stop: B271  
Los Alamos National Laboratory  
Los Alamos, NM 87545

If you have questions:  Call: (505) 665-1605  
E-mail: validate@lanl.gov

All Laboratory computers, computing systems, and their associated communication systems are for official business only. By completing this request, users agree not to misuse the ICN. The Laboratory has the responsibility and authority to periodically audit user files.

**Owner Information**

<table>
<thead>
<tr>
<th>LNLS Group</th>
<th>LANL Mail Stop</th>
<th>Citizenship (Foreign National see Special Requirements)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Phone Number</th>
<th>Cost Center</th>
<th>Program Code</th>
</tr>
</thead>
</table>

Check LANL affiliation:
- [ ] LANL employee
- [ ] Contractor  
  (specify contract company)
- [ ] Consultant, VSM, associate
- [ ] External user  
  (specify employer)
- [ ] Other (specify)  

Send password / smartcard to:
- [ ] Mail Stop  
  or  
- [ ] Mail to address indicated below

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>City, State, Zip Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Access**  Check needed partitions:

- [ ] Open partition (e.g., email systems, open machines)
- [ ] Administrative partition (e.g., IA [BUCS, Stores, Travel], IB [EIS, FMS, PAIRS])

Access method:
- [ ] Password
- [ ] Password / Smartcard (permits Administrative access from an Open port)

*NOTE: If you are not a C-cleared LANL employee, see required steps in "Special Requirements" section.*

- [ ] Secure partition (i.e., Secure Machines)

Highest level of data to be processed:
- [ ] Secret
- [ ] Confidential
- [ ] PART
- [ ] Unclassified

*NOTE: A C-clearance is required. All classified computing must be performed within the Secure environment.*

<table>
<thead>
<tr>
<th>PWO Use Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
</tr>
<tr>
<td>------</td>
</tr>
</tbody>
</table>

Comments:

07/94  Supersedes previous versions (Rev. 8/28)
Special Requirements  Check all that apply and follow the procedures.

Administrative Partition
(U.S. Citizens Only) Lab-Wide Systems (e.g., IA [BUCS, Stores, Travel], IB [EIS, FMIS, PAIRS])

☐ Under 18 years of age If you need to access Administrative systems, your group leader must provide a memo accepting responsibility for your actions and justifying your need for access. This memo is to accompany all forms taken to the security briefing (see "Contractor or Non-Q-Cleared") section below. You may not access the Secure Partition.

☐ Contractor or Non-Q-Cleared Phone (505) 667-9444 to obtain Access Authorization packet.
Phone (505) 667-9153 to schedule a security briefing.
Bring all forms including this ICN Validation Request to the security briefing for approval.

Security Briefing Approval Signature Date

☐ Foreign National
Attach a copy of Form 982 (REQUEST FOR UNCLASSIFIED VISIT OR ASSIGNMENT BY A FOREIGN NATIONAL) with all approval signatures. Be sure Box #11 of Form 982 is completed. If you are not a visitor/assignee under a LANL/DOE approved Visit/Assignment Request, attach written justification from your host Division Director describing your need to access the ICN.

Authorization (required)

Print Manager Name  (Group Leader or above) Manager Z-Number Group

Manager Signature  (Group Leader or above) Mail Stop Date

If you are NOT a LANL employee, obtain your LANL contact’s signature in addition to the contact’s manager’s signature.
NOTE: LANL contacts are regular Laboratory employees. Contacts are responsible for obtaining annual re-authorizations, forwarding renewals, and notifying the ICN Password Office of changes in user or contact status.

Print LANL Contact Name Contact Z-Number Phone Number Group

LANL Contact Signature Mail Stop Date
Reader Feedback

Feedback helps us to provide a document that responds to the changing needs of its readership. If you have comments or questions about this publication, please let us hear from you. We have reserved the back of this form for that purpose. We also accept articles for publication that are of interest to our readers. Contact the managing editor for more information. This form is also used for new subscriptions, deletions, or changes. Instructions are on the back. If you prefer to contact us by E-mail, send your comments and/or subscription request to finney@lanl.gov.
## Feedback

If you would like to be added to or deleted from our mailing list, please check the appropriate line, complete the form below, and mail us the form.

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Mail Stop</td>
</tr>
<tr>
<td>Group</td>
<td>Organization</td>
</tr>
<tr>
<td>City</td>
<td>State</td>
</tr>
<tr>
<td>Phone</td>
<td>Number of copies</td>
</tr>
</tbody>
</table>

---

**New Subscriptions, Deletions, and Changes**

**BITS** is published by Los Alamos National Laboratory. If you would like to be added to or deleted from our mailing list, please check the appropriate line, complete the form below, and mail us the form.

- Add my name to the **BITS** mailing list.
- Delete my name from the **BITS** mailing list.
- Change my name/address as indicated below.
This index is organized according to keywords taken from the original titles of BITS articles. Keywords are listed in alphabetical order and the coverage of articles goes back one year from the date of the current issue.

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Title of BITS Article</th>
<th>Date/Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audience Analysis</td>
<td>Audience Analysis for Computing and Communications News</td>
<td>Mar. ’94/p. 3</td>
</tr>
<tr>
<td>Binary File Transfers</td>
<td>Binary File Transfers between Workstation and Supercomputer</td>
<td>July ’94/p. 8</td>
</tr>
<tr>
<td>Bit By A Byte [UNICOS]</td>
<td>Bit By A Byte</td>
<td>Apr. ’94/p. 14</td>
</tr>
<tr>
<td>BITNET Connection</td>
<td>The Demise of BITNET Connection</td>
<td>Apr. ’94/p. 2</td>
</tr>
<tr>
<td>CFS (Common File System)</td>
<td>Separation of CFS into Secure and Open Systems is Coming!</td>
<td>Dec. ’93/p. 2</td>
</tr>
<tr>
<td></td>
<td>Reminder for Owners of Classified Files on CFS</td>
<td>Dec. ’93/p. 14</td>
</tr>
<tr>
<td></td>
<td>CFS Transitions Tools Available to Aid the Network Separation</td>
<td>May ’94/p. 2</td>
</tr>
<tr>
<td></td>
<td>TCP/IP Based Open CFS Gateway Now Available</td>
<td>Dec. ’93/p. 1</td>
</tr>
<tr>
<td>CIC (Computing, Information, and Communications)</td>
<td>Organizational Structure of CIC Division</td>
<td>June ’94/p. 1</td>
</tr>
<tr>
<td>CIC-8</td>
<td>Streamlined, Efficient, and Flexible—CIC-8</td>
<td>Sept. ’94/p. 6</td>
</tr>
<tr>
<td>CLAMS</td>
<td>CLAMS Now Available on the CIC ICN Tools</td>
<td>May ’94/p. 1</td>
</tr>
<tr>
<td>ClariNews</td>
<td>ClariNews Now Available at LANL</td>
<td>July ’94/p. 7</td>
</tr>
<tr>
<td>Cluster</td>
<td>Introducing Cluster Corner Articles</td>
<td>June ’94/p. 10</td>
</tr>
<tr>
<td></td>
<td>IBM AIX XL Fortran (Version 3) Installed on CIC Cluster</td>
<td>July ’94/p. 10</td>
</tr>
<tr>
<td></td>
<td>CIC Cluster Update</td>
<td>Oct. ’94/p. 5</td>
</tr>
<tr>
<td>Code Portability</td>
<td>Code Portability Using CrayPointers</td>
<td>May ’94/p. 5</td>
</tr>
<tr>
<td>CPIO</td>
<td>CPIO—An Alternative to TAR</td>
<td>Dec. ’93/p. 19</td>
</tr>
<tr>
<td>CF90</td>
<td>Cray CF90 Programming Environment Tools</td>
<td>July ’94/p. 4</td>
</tr>
<tr>
<td></td>
<td>Cray CF90 Programming Environment</td>
<td>June ’94/p. 6</td>
</tr>
<tr>
<td></td>
<td>CF90 Programming Environment Now</td>
<td>Sept. ’94/p. 3</td>
</tr>
<tr>
<td></td>
<td>Available on All Open Crays</td>
<td>Sept. ’94/p. 10</td>
</tr>
<tr>
<td>CrayDoc</td>
<td>CrayDoc On-line Documentation</td>
<td>Sept. ’94/p. 4</td>
</tr>
<tr>
<td>EIS (Employee Information System)</td>
<td>Keeping Your EIS Data Up-to-Date</td>
<td>Oct. ’94/p. 4</td>
</tr>
<tr>
<td>E-mail</td>
<td>Distribution Lists for E-mail Users</td>
<td>June ’94/p. 4</td>
</tr>
<tr>
<td></td>
<td>LANL E-mail to Print Gateway</td>
<td>June ’94/p. 7</td>
</tr>
<tr>
<td></td>
<td>E-mail to DOE and Other Sites</td>
<td>Dec. ’93/p. 6</td>
</tr>
<tr>
<td></td>
<td>E-mail to FAX Gateway</td>
<td>Nov. ’93/p. 1</td>
</tr>
<tr>
<td></td>
<td>New E-mail Server: POP+</td>
<td>Oct. ’94/p. 2</td>
</tr>
<tr>
<td>Electronic Phone Book</td>
<td>Using the LANL Electronic Phone Book</td>
<td>Dec. ’93/p. 8</td>
</tr>
<tr>
<td>Eudora</td>
<td>A Look at Eudora</td>
<td>July ’94/p. 12</td>
</tr>
<tr>
<td></td>
<td>Another Look at Eudora</td>
<td>July ’94/p. 11</td>
</tr>
<tr>
<td>FileMaker Pro</td>
<td>Cross-Platform Databases Using FileMaker Pro</td>
<td>Nov. ’93/p. 7</td>
</tr>
<tr>
<td>Keywords</td>
<td>Title of BITS Article</td>
<td>Date/Page</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>----------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Fone</td>
<td>Fone Tips</td>
<td>Mar. '94/p. 8</td>
</tr>
<tr>
<td>Graphical Monitoring Software</td>
<td>New Graphical Monitoring Software</td>
<td>July '94/p. 8</td>
</tr>
<tr>
<td>High Performance Storage System</td>
<td>High Performance Storage System Project Officially Under way</td>
<td>Apr. '94/p. 1</td>
</tr>
<tr>
<td>House Calls</td>
<td>CIC House Calls: Providing Customer Service Where It's Really Needed</td>
<td>Apr. '94/p. 3</td>
</tr>
<tr>
<td>Information Architecture</td>
<td>The Los Alamos Information Architecture</td>
<td>July '94/p. 1</td>
</tr>
<tr>
<td>JIT Contracts</td>
<td>Using JIT Contracts to Purchase Personal Computer Products</td>
<td>Feb. '94/p. 9</td>
</tr>
<tr>
<td>Kerberos</td>
<td>Kerberos: Life after SIMP</td>
<td>Aug. '94/p. 6</td>
</tr>
<tr>
<td>LAICS (Los Alamos Integrated</td>
<td>LAICS Update</td>
<td>Mar. '94/p. 1</td>
</tr>
<tr>
<td>Communications System)</td>
<td>LAICS implementation on the Horizon</td>
<td>Feb. '94/p. 4</td>
</tr>
<tr>
<td></td>
<td>LAICS Update: Interesting Facts about Your Phone Service</td>
<td>Oct. '94/p. 3</td>
</tr>
<tr>
<td>ldb</td>
<td>ldb Being Phased Out</td>
<td>Mar. '94/p. 7</td>
</tr>
<tr>
<td>Macintosh</td>
<td>The Power Macintosh</td>
<td>June '94/p. 12</td>
</tr>
<tr>
<td></td>
<td>A Look at the Macintosh System 7.5</td>
<td>Sept. '94/p. 13</td>
</tr>
<tr>
<td></td>
<td>Macintosh System 7.5 Follow-up</td>
<td>Oct. '94/p. 9</td>
</tr>
<tr>
<td>PAGES</td>
<td>PAGES: Expanded Services, Reduced Rates, and New Hours</td>
<td>Nov. '93/p. 5</td>
</tr>
<tr>
<td>PC</td>
<td>New IBM PC Products Available</td>
<td>Oct. '94/p. 15</td>
</tr>
<tr>
<td>PC/TCP</td>
<td>Version 2.31 of PC/TCP</td>
<td>Apr. '94/p. 1</td>
</tr>
<tr>
<td>PCMCIA</td>
<td>An Introduction to PCMCIA</td>
<td>Feb. '94/p. 11</td>
</tr>
<tr>
<td>Photoplotting</td>
<td>Photoplotting Services No Longer Available</td>
<td>Feb. '94/p. 1</td>
</tr>
<tr>
<td>Runaway [UNICOS]</td>
<td>Stop that Runaway!</td>
<td>May '94/p. 8</td>
</tr>
<tr>
<td>Software Disposal</td>
<td>Software Disposal</td>
<td>Mar. '94/p. 11</td>
</tr>
<tr>
<td>Software Distribution</td>
<td>Mac and PC Software Distribution at LANL</td>
<td>Oct. '94/p. 12</td>
</tr>
<tr>
<td>Supercomputing '94</td>
<td>Supercomputing '94</td>
<td>Mar. '94/p. 2</td>
</tr>
<tr>
<td>Survey</td>
<td>Desktop Software Site License Survey</td>
<td>Oct. '94/p. 16</td>
</tr>
<tr>
<td>Telephone Service Charges</td>
<td>Telephone Service Charges</td>
<td>Apr. '94/p. 11</td>
</tr>
<tr>
<td>TELNET</td>
<td>Using TELNET and FTP to Sign onto Alpha and Gamma</td>
<td>Feb. '94/p. 8</td>
</tr>
<tr>
<td>UNICOS</td>
<td>UNICOS CFT77 Compiler Listing Files</td>
<td>Dec. '93/p. 18</td>
</tr>
<tr>
<td>Validated ... for LANL Computing</td>
<td>Getting Validated and Registered for LANL Computing</td>
<td>June '94/p. 7</td>
</tr>
<tr>
<td>X Windows</td>
<td>Securing X Windows on UNIX Workstations</td>
<td>Apr. '94/p. 8</td>
</tr>
<tr>
<td></td>
<td>Did You Leave Your X Window Open</td>
<td>Feb. '94/p. 2</td>
</tr>
</tbody>
</table>
Since 1963, R&D Magazine has honored scientific innovators by selecting the top 100 technical products or advances of the past year for its prestigious R&D 100 Award. Winning innovations are selected on the basis of their technical importance and usefulness. Judges for the competition are the editors of R&D Magazine and a panel of outside experts. For more information about entry procedures and deadlines, call Judy Prono (CIC-1) at 5-8383. The schedule for kickoff meetings follows:

- November 1, 10:30-11:30 a.m.: IGPP Conference Room—TA-3/SM-100
- November 15, 10:30-11:30 a.m.: IPO Conference Room—TA-66/ATAC
- November 29, 10:30-11:30 a.m.: Study Center—Room 218

For those of you who use locally-developed UNICOS software, the ICN Consultants strongly urge you to reply to the survey that is being sent out to all users of the Los Alamos Integrated Computing Network (ICN). Please take a few minutes to complete and return the survey so that CIC Division will know what software warrants continued support. In the absence of user replies, the decisions as to what locally-developed software will or will not be supported will be left in the hands of CIC Division.

Rick Light has scheduled another “ICN2 Split and How It Will Affect You” seminar for November 10 at 10:00 a.m. in the CIC-Division Lecture Room (TA-3, SM-200, Room 210). However, because of the lack of attendees at the seminar held in September, the November 10 seminar will not be held if fewer than ten people appear. Instead, seminar notes and documentation will be distributed to those who indicate an interest and/or appear for the seminar. If you have an interest in attending this seminar or obtaining the seminar notes and documentation, please call Pat Vucenic at 667-9399 or send E-mail to pav@lanl.gov.