**Inter-Office Memorandum**

| | | | |
|---|---|---|---|
| To | IFS Users | Date | April 24, 1982 |
| From | Ed Taft and David Boggs | Location | Palo Alto |
| Subject | How to Use IFS (version 1.36) | Organization | PARC/CSL |

# XEROX

Filed on: [Maxc2]<IFS>HowToUse.bravo, .press

This memo describes how to use the IFS (Interim File System) servers.  This is the complete user-level documentation.  The *Alto User's Handbook* has some introductory material and summarizes commonly-used procedures, so new users are advised to look there first.

Information in this memo applies to all IFSs except where otherwise noted.  To obtain a directory on your local IFS, or to find out about local operating practices, consult your IFS administrator.

This edition describes IFS version 1.36.  User-visible changes since version 1.34 are as follows:

> IFS can be configured to use Grapevine for authentication and access control.  This substantially changes the facilities for dealing with file protections; see section 3 for details.

> The Show System-Parameters command has been added.

## 1. How to access IFS

At present, the file services provided by IFS are limited to a fairly basic set.  The normal mode of access from Altos is through FTP.  The basic operations (Store, Retrieve, List, Delete, and Rename) are invoked through the Alto FTP program, or through other programs that use the FTP protocol such as the Mesa FileTool.

You should consult the FTP documentation in the *Alto User's Handbook*, the Alto Subsystems manual, or [Maxc2]<AltoDocs>FTP.tty, for general information on the use of FTP.  IFS can also be reached from Maxc by means of the PUPFTP subsystem.

File naming conventions on IFS are a mixture of Maxc and Alto conventions.  The general form of an IFS file name is:

> <directory>name!version

All printing characters except  *' are legal in the name.  The complete file name may be up to 99 characters long (longer than either Maxc or Alto permit).

All IFS files have version numbers (in the range 1 to 65535) which are defaulted in the usual way, as follows:

| | |
|---|---|
| Retrieve | highest existing version |
| Store | next higher version |
| Delete | lowest existing version |
| List | all versions |

Versions other than the default one may be referred to explicitly (by specifying the version number) or by the notations !L' (lowest existing version), !H' (highest existing version), or !N' (next higher version).

There is presently no facility for automatic deletion of non-current versions; you must delete them manually. The Delete command, described below, has an option that facilitates this.

 *' expansion is supported during Retrieve, List, and Delete commands. The expansion is similar to that provided by the Alto Executive; that is, each *' matches zero or more real characters in a file name.

You may find it convenient to organize your files into *sub-directories* by giving them names such as <Taft>Memos>HowToUse.Bravo'. Then all files belonging to a particular sub-directory may be accessed by a specification such as <Taft>Memos>*', and you may direct your attention to a particular sub-directory by establishing a default such as Directory Taft>Memos'.

The appearance of sub-directories is entirely an artifact of naming conventions; the file system actually keeps track of the complete file name for each file. <' and >' are ordinary characters, and *' matches them the same as any other character. For this reason, if you use sub-directories at all, you will find them most useful if you employ them in a consistent fashion.


## 2. Access via Chat

The current definition of the File Transfer Protocol (the means by which FTP communicates with a file server) limits itself to the basic set of operations mentioned previously. It lacks the means for expressing a number of other essential operations. Improved file access protocols are a topic of current research.

In the meantime, rather than attempting to extend FTP, we have provided an Executive in IFS which you can access by means of Chat (or the bottom *Telnet* window in FTP). This Executive is patterned after the one in Maxc, but has a very limited command repertoire.

Typein and editing conventions are the ones familiar to most users. BS or CTRL-A erases the preceding character, CTRL-W deletes a word, and DEL deletes an entire command or sub-command. Deleted characters are not actually erased from the Alto screen because Chat does not provide such a capability. Most commands must be terminated by RETURN. CTRL-C may be used to abort any command. If you are using any sort of display terminal, typeout will stop at the end of every page (as on Maxc) and IFS will wait for you to type any character before continuing. If you type ahead, this feature is disabled.

The commands of interest to most users are the following:

> @ Login (user) *user-name* (password) *password*
>
>> Logs you into IFS. This is necessary before issuing most other commands. Ordinarily, Chat will do this for you automatically.
>
> @ Logout
> @ Quit
>
>> Logs you out and closes the connection.
>
> @ Connect (to directory) *directory-name* (password) *password*
>
>> Sets your default directory to be *directory-name*, and gives you owner-like access to it. The *password* may be omitted if *directory-name* is your own directory or one to which you have connect privileges.
>
> @ Directory (default) *directory-name*

Sets your default directory to be *directory-name*, but without changing your access rights (and therefore without requiring a password). All subsequent commands dealing with files will behave as if *<directory-name>*' appeared at the beginning of each file name argument that doesn't name a directory explicitly (i.e., that doesn't begin with <'). *Directory-name* may include sub-directories (e.g., <Jones>Memos>').

When you issue the Directory command, IFS first displays your current default directory. You may either edit this field (by first backspacing at least one character) or replace it simply by typing the replacement. If you erase the entire field (with CTRL-W), the default directory reverts to your current connected directory.

If the first character of *directory-name* is >', IFS prefixes the name of your current connected directory. That is, if you are currently connected to directory Jones, the command Directory >Memos>' is equivalent to the command Directory <Jones>Memos>'. Also, the outermost <' and >' are optional. Note that the foregoing descriptions also apply to the Directory command in the FTP server.

## @ DskStat

Prints the number of used pages and the maximum allowed in the connected directory, followed by the number of free pages in the system. One IFS page is 1024 words or 2048 characters, which is equivalent to four Alto pages or approximately one Maxc page.

## @ List (files) *file-designators*

Lists the names of all files matching *file-designators*, which is a list of up to 10 file names (separated by spaces), any of which may contain *'s to denote multiple files. The files matching each *file-designator* are listed in alphabetical order on the basis of the entire file name (including directories and sub-directories, if any). To save space, directory and sub-directory names are printed only when they change, above the list of files to which they apply.

If you terminate the last *file-designator* with a comma followed by RETURN (rather than just RETURN), IFS enters a sub-command mode in which you may specify additional information to be printed about each file:

| | |
|---|---|
| @@ Type | file type and byte size |
| @@ Size | size in pages |
| @@ Length | length in bytes |
| @@ Creation | date of file creation |
| @@ Write | date of last write |
| @@ Read | date of last read |
| @@ Backup | date of last backup |
| @@ Times | times as well as dates |
| @@ Author | creator of file |
| @@ Protection | file protection |
| @@ Verbose | same as Type Size Write Read Author |
| @@ Everything | |

Sub-command mode is terminated when you type just RETURN in response to the @@' prompt. The columns of printout will be aligned properly only if you are running Chat with a fixed-pitch font such as Gacha12 or Gacha10.

## @ Delete *file-designator*

Deletes all files matching *file-designators*, which is a list of up to 10 file names (separated by spaces), any of which may contain *'s to denote multiple files. The version number defaults to the lowest existing version; to delete all versions, you must end each *file-designator* with !*'. IFS prints out each file name, followed by [Confirm]'. You should respond with Y' or RETURN to delete the file, or with N'

or DEL to leave it alone.

If you terminate the last *file-designator* with a comma followed by RETURN, IFS enters a sub-command mode in which you may request the following additional actions:

> @@ Confirm (all deletes automatically)
>
> > IFS will not ask you to confirm deleting each file but will just go ahead and do it.
>
> @@ Keep (# of versions) *number*
>
> > IFS will retain the *number* most recent versions of each file and delete all remaining versions.  That is, to delete all but the most recent version of each file, specify  Keep 1'.

On IFS (unlike Maxc), files are deleted immediately; there is no Undelete command.  To delete a file, you must have write access to it.

@ Rename *existing-filename* (to be) *new-filename*

> Changes the name of *existing-filename* to be *new-filename*.  It is permissible to change any part of the file name, so it is possible to move a file from one directory or subdirectory to another by renaming it.  If you terminate *new-filename* with ESC rather than RETURN, the body of *old-filename* (with directory and version stripped off) will be appended to *new-filename*.
>
> The Rename operation requires that you have write access to the file and create access to the directory into which the file is being renamed.
>
> It is permissible to rename a file to itself in order to change its capitalization.  Note that a new version of a file always inherits the capitalization of the previous version; renaming a file to itself (i.e., with the same version number) is the only way to defeat this.

@ Change Attributes (of files) *file-designator*
@@ *sub-commands*

> Changes certain attributes of all files matching *file-designators*, which is a list of up to 10 file names (separated by spaces), any of which may contain  *'s to denote multiple files.  You specify the changes to be made by means of one or more of the following sub-commands:
>
> > @@ Read (access permitted to) *groups*
> > @@ Write (access permitted to) *groups*
> > @@ Append (access permitted to) *groups*
> > @@ Reset (all existing access)
> >
> > > Changes file protections, as described in section 3.
> >
> > @@ Type *type*
> > @@ Byte-size *number*
> >
> > > Changes the type and byte size used for subsequent FTP retrievals of the files.  *Type* may be  Text',  Binary', or  Unspecified'.  You should change these attributes only if you thoroughly understand their purpose and interpretation (see the FTP documentation in *Alto User's Handbook*).
> >
> > @@ Backup
> > @@ No Backup

Enables or disables the automatic maintenance of backup copies of the files. Ordinarily, all files are backed up. Specifying No Backup' is appropriate only for very large files that are themselves duplicates of files stored elsewhere; you should use No Backup' only after consulting with your IFS's administrator.

All requested changes take effect when you type RETURN immediately after the @@' prompt.

@ Print (files) *file-designator*
@ Press (files) *file-designator*

Requests that all Press files matching *file-designator* be sent to your default printing server ( Print' and Press' are synonyms). *File-designator* is a list of up to 10 file names (separated by spaces), any of which may contain *'s to denote multiple files. IFS prints out the name of each file followed by [Confirm]'; you should respond with Y' or RETURN to print the file, or with N' or DEL to skip over it.

If you terminate the last *file-designator* with a comma followed by RETURN, IFS enters a sub-command mode in which you may specify the following parameters:

@@ Copies *number*

Specifies the number of copies of each Press document to print.

@@ Duplex

Specifies that the document is to be printed on both sides of the paper (if the printing server is capable of doing so).

@@ Password *password*

Causes the document not to be printed immediately but rather to be held by the printing server until you type *password* on the printing server's terminal. Only certain servers are capable of password-protected printing. Contact your local support staff for instructions on using this feature.

@@ Printed-by *name*

Causes *name* to appear in the Printed-by' field on the cover page of the printed output. (Ordinarily, your user name is printed.)

@@ Server *server-name*

Specifies the name of the printing server to which the Press files are to be transmitted. This may be either a registered name or an internetwork address of the form *net#host#*' (don't leave off the trailing #').

You terminate sub-command mode by typing RETURN in response to the @@' prompt. In the absence of any sub-commands, IFS will cause one copy of each Press file to be printed on your default printing server. You may establish or change your default printing server by means of a sub-command of the Change Directory-Parameters command, as follows:

@ Change Directory-Parameters (of directory) *directory*
@@ Printing-Server *host-name*

where *directory* is the name of your directory, i.e., your user name. If you have not

established your default printing server, IFS will require you to issue a Server sub-command every time you request printing.

Actual transmission of the Press files to the printing server is performed by a background process, so you need not remain connected to IFS while the printing is taking place.  If the printing server is down at the time, IFS will queue the files for later delivery.  If the Press files cannot be delivered within eight hours, however, the printing request is discarded without a trace.

Printing request may be examined and canceled with the following commands:

@ Show Printing-requests

displays all printing requests you have issued that have not been completed.

@ Cancel (printing requests)

displays all outstanding printing requests you have issued, and for each one asks you whether or not you wish to cancel it (answer  Y’ or  N’).

Note that *only* Press-format files can be printed; IFS checks that every file is a Press file and will refuse to print any file that is not.

@ Change Password
@ Change Protection
@ Change Directory-Parameters
@ Show Directory-Parameters
@ Change Group-Membership
@ Show Group-Membership

See section 3.

@ Systat

Shows who is presently using IFS, what service they are accessing (FTP, Telnet, CopyDisk, or Mail), and the name or inter-network address of the machine they are coming from.

@ DayTime

Displays the current date and time.

@ Statistics
@ Show System-Parameters

Prints out various operating statistics and parameters that are generally of interest only to IFS administrators.  The Show System-Parameters command may be used to find out what services are enabled on the IFS, whether or not Grapevine is used for authentication and access control, etc.

## 3. Passwords, protections, and Grapevine

IFS has facilities for controlling access to a file server as a whole and to individual directories and files within the file server.  A file server that is in communication with a Grapevine registration server can use Grapevine for authentication and access control.  A file server that does not have access to Grapevine (or whose administrator has chosen not to enable the Grapevine facilities) maintains its own local data base for these purposes.

Since the IFS’s behavior as seen by users is quite different in the two cases, we first describe the

case in which the IFS is using Grapevine for authentication and access control; then we present the additional facilities required to manipulate the IFS's local data base if Grapevine is not being used.

This description assumes that you understand the organization of Grapevine names and groups and are familiar with the use of the Maintain program for managing them.  For an introduction, read section 3.7 and appendix B.2 of the *Laurel Manual*, filed as <Laurel>Laurel.press on most file servers.  Complete instructions for using the Maintain program are included in the *Maintain Reference Guide*, file <Laurel>Maintain.press.

### 3.1. User names and passwords

Ordinarily, you log into an IFS with the user name and password by which you are known to Grapevine.  If you are a Laurel user, this is the same as the name and password you use with Laurel.  If you change your Grapevine password using the Maintain program, IFS will recognize your new password automatically.

It is always correct to use your full registered name (R-Name), which is in the form *simpleName.registry* for example,  Jones.PA' or  Smith.ES'.  Additionally, each IFS has a default registry which is the Grapevine registry of most of its local users; for example, the default registry for the Ivy and Indigo file servers is  PA'.  Users whose R-Names are in the default registry need not specify the registry name.

You probably have your own directory on at least one local IFS, for storage of personal files.  This directory's name is generally the same as the *simpleName* portion of your R-Name.  Once you have logged in, you are given control of the directory whose name either matches your full R-Name or can be extended to match your full name by appending a period and the IFS's default registry name.  For example, on an IFS whose default registry is PA, user Jones.PA gets control of the directory whose name is  Jones', but user Jones.ES does not; however, on the same file server there can also be a directory named  Jones.ES' which user Jones.ES controls and Jones.PA does not.

You probably don't have your own directory on any IFSs besides the ones belonging to your local organization.  However, you may still log in to any IFS that is using Grapevine for authentication; in such a situation, you won't have the right to store personal files on that IFS, but you may access files in other directories if their file protections permit.

You will probably find it convenient to install your full R-Name on your Alto disk (or in your Pilot User.cm, or whatever) instead of just your *simpleName*.  This enables you to log into remote IFSs with the same user name you use locally.

### 3.2. User groups

Your access to files and directories is permitted or denied on the basis of your membership in *user groups*.  Most authentic users are members of a special user group called  World' (discussed in more detail below).  You are a member of another user group called  Owner' with respect to files in your own directory, and temporarily to files in any other directory to which you connect (using the Connect command in FTP or Chat).

Additionally, you may be a member of one or more other user groups whose names and members are registered with Grapevine.  Such user groups generally correspond either to organizations or to specific projects.  A group's name is assigned by an administrator responsible for the registry; the name is an R-Name whose *simpleName* portion typically ends in  ^', as in CSL^.PA or IFSAdministrators^.PA.  A group's membership is controlled by the group's owners; ownership of a group can be as restrictive or as open as the administrator deems appropriate.

You are probably already familiar with Grapevine groups: many of them are used as distribution lists for Laurel messages.  For example, the group CSL^.PA is a list of members of the PARC Computer Science Laboratory.  If CSL^.PA is specified as a recipient of a message, then all members of the group receive a copy.  Similarly, if an IFS file's protection specifies that the file is

readable by CSL^.PA, then all members of the group are permitted to read that file.  The same group may be used for both purposes.

There is a fixed set of groups that may be used in specifying IFS file protections; this set is determined on a system-wide basis by the IFS's administrator, and there are at most 62 such groups.  All the acceptable group names are displayed by the Show System-Parameters command, and are also displayed as alternatives if you type  ?' when IFS is expecting you to type a group name.

You may examine groups used for IFS access control just the same as groups used as distribution lists, namely by running the Maintain program in Laurel.  Similarly, you may request to be added to a group by contacting the group's owner; one way to do this is to send a message to  Owners-*x*', where *x* is the full name of the group (e.g.,  Owners-CSL^.PA').

There is not actually a Grapevine group called  World'.  Rather,  World' is a shorthand name for a special group; for most IFSs in the United States, this group is USRegistries^.internet, which includes all registered Xerox employees in the U.S. and Canada.  Foreign individuals are excluded from  World' in order to satisfy U.S. technology export regulations; foreign access to U.S. information is expected to be conducted through more formal and controlled channels.  The definition of  World' is controlled on a system-wide basis by the IFS's administrator; it is displayed by the Show System-Parameters command.

### 3.3. File protections

A *file protection* specifies, for each individual file, what types of access are permitted to which groups.  There are three types of file access: *read*, *write*, and *append*.  If you have read access to a file, you are permitted to read (i.e., retrieve) its contents.  Similarly, write access permits you to overwrite, delete, or rename the file, and append access permits you to append to an existing file, even if you don't have write access.  Append access is not presently implemented.

The standard default file protection permits read, write, and append access to the Owner and read access to the World (i.e., to members of the IFS's  World' group, described above).  Hence if the file is in your own directory or the directory to which you are connected, you may do anything to it; otherwise you may only read it (assuming you are a member of the  World' group).  But, for example, if the file protection also permits write access by group CSL^.PA, and you are a member of group CSL^.PA, then you may overwrite (or delete or rename) the file, even if it is not in your directory or the directory to which you are connected.  Note that the read, write, and append access types are independent.  It is therefore possible, though perhaps not particularly useful, for a file protection to permit writing but prohibit reading by some user group.

In addition to the protection associated with each file, there are some protections associated with a directory as a whole.  The first is the *default file protection* for files in that directory.  When a file is created, its protection is assigned in one of two ways.  If there is an existing version of the same file, then the new file inherits its protection.  More precisely, when version *n* of a file is created, it inherits the protection of the highest-numbered existing version less than *n*, if there is one.  Otherwise, the protection assigned is the default file protection of the directory in which the file is being created.

There are two additional types of access to the directory: *create* and *connect*.  If you have create access to a directory, then you are permitted to create new files in that directory.  If you have connect access to a directory, you are permitted to connect to that directory without giving its password.  (Once connected to a directory, you have the same privileges as that directory's owner.)  As with file protections, these types of access are granted or denied individually to Owner, World, and each user group.  The standard directory protection permits create and connect access only to the owner.

Each files-only directory has an *owner*.  The owner of a files-only directory is permitted to connect to that directory without giving a password, regardless of the connect protection of the directory.  This feature avoids the need to define one-member user groups in order to grant owner access to files-only directories managed by a single person.

*3.4. Commands dealing with protections*

The Chat Executive contains several commands by means of which you may manipulate protections of files and directories.

@ Change Protection (of files) *file-designators*
@@ *sub-commands*

>Changes the protection of all files matching *file-designators*, which is a list of up to 10 file names (separated by spaces), any of which may contain *'s to denote multiple files. You specify the changes to be made by means of one or more of the following sub-commands (which are a subset of the sub-commands of the Change Attributes command):
>
>>@@ Read (access permitted to) *groups*
>>@@ Write (access permitted to) *groups*
>>@@ Append (access permitted to) *groups*
>
>where *groups* is a list of up to 10 instances of Owner', World', or group names (separated by spaces) to which the specific access type is to be granted. None' may be used in place of *groups* to specify that access is to be denied to all groups. You may precede a sub-command by the word No' to specify individual groups to which access is to be denied. The changes take effect when you type RETURN immediately after the @@' prompt.
>
>Normally, the changes that you specify by means of these sub-commands are *incremental*. That is, the only access/group combinations that are changed are the ones you mention explicitly, while all the remaining ones are unchanged. However, there is an additional sub-command,
>
>>@@ Reset (all existing access)
>
>that denies all types of access to all groups. In this case, the entire file protection is changed to permit only those access/group combinations that you enable explicitly.
>
>You may change the protection of any file to which you presently have write access, and of any file in your own directory or one to which you are connected regardless of its protection. That is, you can change the protection of any file of your own even if its present protection does not permit read, write, or append access by you.

@ List ...

>The Protection sub-command to the List command (described previously) displays a file's protection thus:
>
>>R: *groups*; W: *groups*; A: *groups*
>
>For example:
>
>>R: Owner World; W: CSL^.PA Owner; A: None

@ Change Directory-Parameters (of directory) *directory-name*
@@ *sub-commands*

>Changes the information associated with the directory as a whole in the manner specified by the *sub-commands*. The directory must be either your own or one to which you are connected.
>
>You may change the default file protection by means of the Read, Write, and Append sub-commands in the same manner as in the Change Protection command. Additionally, you may change the create and connect access using the sub-

commands:

> @@ Create (access permitted to) *groups*
> @@ Connect (access permitted to) *groups*

The No' prefix may be applied to these as well as to the others.

The Reset sub-command requires an additional keyword to specify what it is that you wish to reset:

> @@ Reset Default-File-Protection
> @@ Reset Create-Protection
> @@ Reset Connect-Protection

You may change your default printing server by means of the sub-command:

> @@ Printing-Server *host-name*

The changes are not actually made until you type the confirming RETURN in response to the @@' prompt.

@ Show Directory-Parameters (of directory) *directory-name*

> Displays all information about *directory-name*, and additionally prints some other parameters, such as the disk limit and the owner of a files-only directory, that may be changed only by an IFS administrator. This command also displays some information about user group membership; this reflects information that IFS has determined by querying Grapevine, and is not necessarily accurate or up-to-date.

## 3.5. Non-Grapevine users and groups

In an IFS that does not use Grapevine for authentication and access control, the IFS must maintain local information about user names, passwords, and group memberships. Additionally, even in an IFS that does use Grapevine, there may be some non-Grapevine user names and groups. (In particular, the names and passwords of files-only directories are typically not registered with Grapevine but are maintained solely by the IFS.) The commands described in this section are used to manipulate the local authentication and access control data base.

Note that IFS uses the same data structures to keep temporary copies of Grapevine information as it uses to keep permanent local information. That is, the commands described below can be used to manipulate Grapevine as well as non-Grapevine user names and group memberships. However, using these commands to examine Grapevine R-Names will in general yield information that is incomplete or out-of-date; and attempts to make changes to the local data base will not have permanent effects. You must use Maintain to get correct information or make permanent changes involving Grapevine R-Names.

The password of a non-Grapevine directory may be changed by the following command:

> @ Change Password (of directory) *directory-name* (old password) *password* (new password) *password*

> > Changes the password of the specified directory, which must be either your own or the one to which you are presently connected. (Contrary to normal practice, the new password is displayed as you type it; this is so that if you make a typing mistake you will be able to see it.)

An IFS administrator can assign you to be the *owner* of one or more non-Grapevine user groups. If you are the owner of a group, you are permitted to change and examine the membership of that group, using the following commands:

@ Change Group-Membership (of group) *group*
@@ *sub-commands*

The sub-commands are one or more of the following:

@@ Add *user-name*
@@ Remove *user-name*

These cause the specified users to be added to or removed from the group.  The sub-commands take effect immediately.  You exit sub-command mode by typing RETURN immediately after the  @@' sub-command prompt.

@ Show Group-Membership (of group) *group*

Displays the list of users who are members of the specified group.  This command takes a long time to execute, because it has to read the directory parameters of every user in the system.

In addition to having names, groups have numbers in the range 0 to 61.  The association between group numbers and names is displayed by the Show System-Parameters command.  A group can be used for protection purposes even if it doesn't have a name; all commands that accept group names also accept group numbers.

## 4. Other servers

### *4.1. Mail server*

IFS optionally makes available a mail server compatible with the Laurel message system interface and the Grapevine transport mechanism.  Most registries are now served by Grapevine servers.  However, there are a few registries that still use an IFS as its sole mail server; also, an IFS can serve as an adjunct to a Grapevine registry by keeping some of its mailboxes.

It is largely invisible to a Laurel user whether one's mailbox is kept by Grapevine or by an IFS.

### *4.2. CopyDisk server*

IFS contains a CopyDisk server compatible with the CopyDisk program available from the NetExec.  When CopyDisk prompts you for a disk name, you can specify a  disk' on IFS by typing, for example:  [Ivy]<BasicDisks>NonProg.bfs'.

We expect that this server will be primarily used to distribute copies of the basic Alto disks, eliminating the need for physical disk packs which often get mislaid.  No doubt other applications will evolve with time.  By convention, files in CopyDisk format have extension  .bfs', which stands for Basic File System.

CopyDisk files can be quite large.  A single Diablo 31 disk takes 1275 IFS pages more than a typical user's entire disk allocation.  CopyDisk does not copy free pages, so that number is the worst case for a completely full disk; none the less, it is easy to generate gigantic files that use up your disk allocation.

### *4.3. Leaf server*

IFS optionally makes available a server for the  Leaf' page-level access protocol, which permits random access to parts of IFS files as opposed to the transfer of entire files.  There are several user programs that take advantage of this capability, though these programs are still experimental and not widely available.

Not all IFSs run Leaf servers.  An IFS that is running Leaf identifies itself by an  L' suffix on the IFS version number in the herald displayed at connection time for example,  1.36L'.


**5. File backup**

Reliability of file storage is accomplished by two facilities.  First, we have a Scavenger capable of reconstructing the IFS directory from redundant information kept in the file system.  We expect to be able to recover from most file system crashes in this manner, with no loss of user files.

Second, we have an automatic backup system that periodically copies files to a backup disk pack. The backup system runs between 2:00 and 5:00 a.m. every day (users accessing IFS during that time may notice some significant degradation in performance).  During each backup run, all files not previously backed up or last backed up more than 30 days ago are copied.

This backup system serves two purposes.  First, if the file system fails catastrophically in a way that the Scavenger can't recover from, we will be able to reconstruct the file system from backup, with at most one day's files lost.  Second, files accidentally deleted or overwritten by users will usually be recoverable if the loss is noticed within 30 days.  (The recovery procedure is not particularly convenient, so please don't depend on it as a regular service.)


**6. Present limitations and future plans**

IFS now provides facilities sufficient to make it a useful service.  It is unlikely that any further major development will be undertaken (recent history notwithstanding).  IFS has already far exceeded its intended  interim' specifications, and will ultimately be replaced by better facilities.

A major problem is that of performance of the file system.  An IFS is nothing more than an Alto with some large disks connected to it.  There is insufficient capacity (particularly main memory) in the IFS Alto to support more than a small number of simultaneous users.

We are presently imposing a relatively small limit (somewhere between 4 and 10) on the number of concurrent connections FTP, Mail, CopyDisk, and Chat users combined.  When this limit is reached, the system will refuse to accept additional service requests.  To prevent idle users from tying up these precious slots, the IFS will break connections after a relatively brief period of inactivity.

We would be pleased to receive reasonable suggestions for changes or improvements in the set of facilities provided by IFS.  However, please be conscious of the limited manpower available for implementing such improvements.


**Acknowledgments**

Implementation of IFS would have been impossible without the assistance and cooperation of several individuals who have contributed considerable effort in support of this project.  Peter Deutsch provided the Overlay, VMem, and ISF packages and implemented a number of improvements needed by IFS.  Ed McCreight made available his B-Tree package, which is used for maintaining user directories, and likewise contributed IFS-related improvements.  Bob Sproull and Roger Bates sank considerable energy into the Trident disk hardware, microcode, and software to make it work reliably.  Steve Butterfield initially implemented the Mail and Leaf facilities and made some important internal improvements.  Ted Wobber contributed improvements to the Press printing facility and presently maintains the Leaf server.