

Inter-Office Memorandum

To	IFS Administrators	Date	October 3, 1982
From	Ed Taft	Location	Palo Alto
Subject	IFS Operation (version 1.37) <i>Last edited 18 January 1985 by Ron Weaver</i>	Organization	PARC/CSL

XEROX

Filed on: <IFS>Operation1 & 2.bravo, <IFS>Operation.press

This memo describes operating procedures for the Interim File System software. It assumes that you are familiar with standard Alto software in general and with IFS from the user's point of view (<IFS>HowToUse.bravo).

A short summary of revisions to this document may be found at the end. Also, section 14 contains a summary of known bugs in the current release of the software and how to avoid them.

The current release of IFS should be run under OS version 20 or later. For correct error reporting, you must have current versions of Swat and Sys.errors.

1. Hardware requirements and organization

IFS requires a standard Alto with a Trident disk controller and one to eight Trident T-80 or T-300 disk drives in any combination. A T-80 disk pack holds 36,675 pages of 1024 words each, while a T-300 holds 139,365 pages. Due to a software limitation, only 130,986 pages of a T-300 are accessible.

The software deals with a *primary* file system consisting of one or more disk packs. A multiple-pack IFS behaves logically as a single file system, and all packs must be present and on-line in order to access any files. Files created by IFS are not accessible to other Trident-based programs (e.g., TFU, FTP) or vice versa.

Additionally, the software can deal with one or more *secondary* file systems that may be mounted and dismounted while IFS is running. An example of a secondary file system is a disk pack used for on-line, incremental backup.

The number of disk drives needed to support a given size file system depends both on how backup is to be accomplished and what degree of redundancy is desired in case of disk drive failure. Backup procedures are discussed in a later section.

Any Alto (Alto-I or Alto-II) with a Trident disk controller will run the IFS software; however, for performance reasons, it is strongly recommended that an Alto-II with at least 128K words of memory be used. See section 13.2 for details. Beginning with of IFS 1.36, we have abandoned the practice of packaging the code to minimize working set in a 64K configuration. Certain common operations such as FTP Store now have a working set greater than 64K, so a server with only 64K of memory is likely to thrash very badly even under light load.

There is an Alto-II hardware modification that is recommended for machines that are to be IFSs (or other servers requiring high reliability). This modification corrects a design bug in the memory single-error correction. The memory chips are so reliable that the single-error correction is hardly ever invoked; but servers that run for months at a time eventually exercise even low-probability bugs. The modification is described in [Maxc2]<IFS>MemECMod.press.

If possible, the hardware should be installed in a clean, air-conditioned environment with little human traffic. In particular, do *not* install a Dover or other xerographic printer in the same room; the pollution given off by such equipment (toner and fuser oil) is extremely hazardous to disk drives and is likely to lead to severe reliability problems including damage to disk heads and packs.

There exists a disk controller for the Shugart 4000-series disks. This controller is compatible with the Trident controller, so the IFS software should be able to operate with Shugart disks.

The IFS software runs only on Altos, not on other machines emulating Altos; this is because IFS employs special Alto-specific microcode.

2. Obtaining IFS software

The software is presently distributed from the <IFS> directory on Maxc2. It consists of the following files:

IFS.Run, the program itself.

IFS.Syms, for debugging by means of Swat.

IFS.Errors, mapping error numbers to strings.

IFSScavenger.Run, the IFS Scavenger.

IFSScavenger.Syms, symbols for the IFS Scavenger.

Versions of these files also exist on [Indigo]<IFS>, but in general these are development versions of the software that have *not* been released. You should use only the versions on Maxc2 unless instructed otherwise.

These files should be installed on a fairly clean Alto disk, along with standard Alto subsystems such as FTP. Additionally, the following programs may be useful (obtained from the <Alto> directory):

TFU.Run, for formatting and testing Trident disk packs.

Triex.Run, for checking out the Trident disk hardware.

CopyDisk.Run, for copying disks.

There exists a command file, [Maxc2]<IFS>IFSOpDisk.cm, that constructs an IFS Operation disk with all the necessary files. You should first do an Install, go through the long installation dialogue, and erase the disk. Then retrieve IFSOpDisk.cm and execute it. This command file gets all files from Maxc2; you might want to edit it to get standard subsystems and the like from a local file server instead, if you are sure the file server has up-to-date versions.

All announcements of new IFS releases are made to the distribution list IFSAdministrators^.PA. If you operate an IFS, you should make sure you are included in this list (add yourself using the Maintain program, or send a message to Owners-IFSAdministrators^.PA). Such announcements include the version number of the system (e.g., 1.03); this is the first number printed out in the IFS herald when you connect using FTP or Chat. Be sure to obtain all three of the IFS.* files listed above.

It is important that you use the version of IFSScavenger appropriate to the version of IFS you are running. The IFSScavenger has intimate knowledge of the file system format, which sometimes changes in minor ways from one release to the next.

3. Testing disk hardware and disk packs

A file system should be built on a set of disk packs that have been thoroughly tested using the `Certify` command in the latest release of TFU. This procedure is essential for reliable operation. The procedure for each pack is as follows.

Mount the pack on some drive, *d*. Then issue the command:

```
TFU Drive d | Certify
```

TFU will display the message `Confirm wiping the pack on drive d`, to which you should respond `OK`.

TFU will now initialize the headers on the pack, then execute 10 passes of writing and reading random data on the entire surface of the pack. Any bad spots that it finds are recorded permanently on the pack in a place that IFS and other Trident software can find. IFS will simply avoid using pages known to contain bad spots. The running time for this test is 27 minutes for a T-80 pack and 97 minutes for a T-300 pack. If possible, it is recommended that the test be run for more than 10 passes. This is accomplished by the command `TFU Drive d | Certify n | BadSpots`, where *n* is the desired number of passes. The running time is approximately 3 minutes per pass on a T-80 and 9 minutes per pass on a T-300.

Before IFS is run in a new installation, it is a good idea to exercise the hardware thoroughly. The TFU program has an `Exercise` command that exercises the hardware thoroughly in a manner similar to IFS. The basic procedure is as follows.

1. Mount scratch packs on all Trident disk drives (including backup and spare drives).
2. If the packs haven't already been certified, use `TFU Certify` to initialize the headers on all packs, as described above.
3. For each drive, issue the command `TFU Drive d | Erase`. TFU will display the message `Confirm wiping the pack on drive d`, to which you should respond `OK`.
4. For each T-300 drive (if any), issue the command `TFU Drive 40d | Erase`, and continue as in step 3. This initializes empty file systems on the second half of each T-300 disk.
5. Issue the command `TFU Exercise n`.

This begins a lengthy exercise procedure that creates, reads, writes, and copies files on all packs. *n* is the number of passes to execute. Each pass takes about 20 minutes per T-80 and 60 minutes per T-300 being exercised. The display will flicker and tear while this test is running; do not be alarmed. TFU writes a log file `TFU.Log` on the Diablo disk (which should be quite empty), at the end of which should appear the message `There were 0 errors` when TFU is done.

For further information on the operation of TFU, consult the Trident software documentation in the Alto Subsystems Manual or in `<AltoDocs>TFS.tty`.

4. Initializing a file system

IFS is invoked by the command

```
IFS/switches
```

where the switches control the operation of the system in various ways. For a binary switch, preceding it with `'` negates its action. Switches defined at present are:

- `/A` Allocator debug (every call to the storage allocator invokes some special consistency checking; this debugging aid slows down system operation slightly.)
- `/C` Create a new file system (see below)

- `/D` Debug mode (various non-fatal errors call Swat rather than just continuing on).
- `/F` Each occurrence of this switch extends the directory file map by 100 words. This is necessary only if the directory has become badly fragmented. The standard file map's capacity is 40 runs; each occurrence of `/F` increases it by 50 runs.
- `/L` Retain the Leaf server capability even if the server itself has not been enabled (see section 9.4).
- `/M` Enable the 'miscellaneous' servers (name, time, and boot). They are ordinarily enabled and disabled by means of the privileged Change System-parameters command (described later), but `/M` or `/ M` will override the effects of that command for one invocation of IFS. (See section 9.)
- `/S` Create a buffer (spyBuffer) for use by the Swat Spy facility, for performance measurements.
- `/V` Verify the structure of the directory B-Tree when the system is started (see section 5). The default state of this switch is true.
- `/X` Use extended memory, if present, to hold executable code overlays (rather than swapping them from disk into bank zero when needed). This improves performance substantially. The default state of this switch is true.
- `/n` (n is a number between 1 and 4) Use at most $n*64k$ of memory, even if the Alto has more than that. This is useful primarily for debugging purposes.

When IFS is started with the `/C` switch, it enters a dialogue in which you must supply various file system parameters. The dialogue takes the following form.

Do you really want to create a file system?

Answer `y`'.

Number of disk units:

Type in the number of disk units to be included in the file system, terminated by Return. Maximum is 5. However, the file system can be extended to up to 7 drives using the "Extend" command. See Section 10.

Logical unit 0 = Disk drive:

Type the physical unit number of the drive that is to be logical unit 0 in the file system. This question is repeated for each logical unit in the file system.

File system name:

Enter the name of the file system, followed by Return. This name is displayed as the first part of the herald generated by the file server and Executive, and should be something like `Parc IFS`'.

Directory size (pages):

This determines the number of disk pages to preallocate for the directory. IFS will suggest a number (based on the number of disk units) which you may confirm by typing Return; or you may type some other number followed by Return. To be conservative, you should specify 1000 times the number of disk packs you ever expect to include in the primary file system. (See section 10.)

Ok? [Confirm]

Answer 'y' if you want to go ahead, or 'n' if you made a mistake and wish to repeat the dialogue.

IFS now initializes the file system, an operation that takes about 2 minutes per T-80 and 7 minutes per T-300 in the system. When the screen turns black and the cursor changes from an hourglass to IFS', initialization is complete.

The file system initially has only three directories defined: System, Default-User, and Mail. The password for System is IFS. You should connect to the IFS using Chat, login as System (password IFS), create some other users (by means of the Create command, described below), and change System's password for the sake of security.

Before putting the system into service, there are various system parameters you must set; these are described in later sections. They include:

clock correction and server limit (section 7);

switches to enable or disable Press printing and the Boot, Name, Time, Leaf, CopyDisk, and LookupFile servers (sections 7 and 9);

switches, default registry, and group names for Grapevine authentication and access control (section 6);

mail system parameters (section 8);

backup system parameters (section 11).

5. Normal operation

The system is normally started simply by invoking IFS with no switches. All file system packs must be mounted and on-line, and *all Read-only switches must be turned off* (including those on backup and spare drives). It is unimportant which packs are mounted on which drives, since the software reads each pack to discover what the system configuration is. (However, there must be no other primary IFS packs on-line. After copying an IFS pack with CopyDisk, you should be careful to remove the copy from the system.)

If IFS fails to start up properly, it will call Swat with an appropriate error message. This will occur, for example, if all necessary disk drives are not on-line. While IFS is starting up, the cursor contains an hourglass; this changes to IFS' when startup is complete and the system is in operation.

During startup, IFS also runs a brief test of the Control RAM and S-registers, failure of which will cause IFS not to function even though a great deal of other Alto software works correctly. IFS will call Swat with an appropriate message if this test fails.

When IFS is started, it verifies the consistency of the directory B-Tree (unless inhibited by /V'). Inconsistencies can result from crashes at inopportune moments when the B-Tree is in the process of being modified, so the startup-time check is valuable in determining whether it is appropriate to run the IFS Scavenger. The time required for the check is proportional to the number of files in the system; it has been observed to take 2 minutes for 20,000 files.

If an inconsistency is detected, IFS will call Swat with an appropriate message. The message 'Record count disagrees' is relatively benign and it is reasonably safe to proceed from this (with control-P). If you do so, it is likely that one or more files in the file system will become inaccessible and will remain so until the next time the IFS Scavenger is run. Other possible errors include 'Records out of order' and 'Malformed B-Tree record'; these are more serious and proceeding is *not* recommended.

Immediately after IFS starts, it will perform some other initialization operations requiring a lot of disk activity, including obtaining and installing the network directory (name server data base) and boot files. IFS can service user requests during this time (5 minutes or more), but performance will

be noticeably poorer than normal.

While IFS is running, the entire screen is black except for the cursor. The position of the cursor is an indication of disk activity. The horizontal position indicates the disk unit most recently accessed (unit zero at the extreme left, unit seven at the right), and the vertical position is the cylinder at which the heads are currently positioned (zero at the top, 814 at the bottom). The cursor blinks each time a page is transferred to or from a user file by the file server.

You can tell whether IFS is running normally by pressing the space bar. If the Alto screen flashes, everything is in order; if not, the system has failed in some way.

There are two ways to stop the system. The normal (and cleaner) way is to connect to IFS using Chat, log in, Enable, and issue the Halt command. The system will then refuse to admit further users, will wait for all present users (including you) to log out or disconnect, and will return control to the Alto Executive.

The system may also be stopped by typing Shift-Swat on the IFS Alto keyboard (the Swat key must be pressed firmly). This aborts all active connections and returns control to the Executive immediately; however, it may leave partially-written files lying around so it is not recommended for normal use.

6. Directory and user group management

This section assumes that you thoroughly understand IFS's use of Grapevine for authentication and access control, described in the *How to use IFS'* document. Additionally, before beginning to create IFS directories and groups, you should read *Access Controls'* file [Maxc]<IFS>AccessControls.press, for a description of the policies and procedures required to ensure proper information security.

6.1. Setting up user directories

The IFS Executive (accessed via Chat) has several privileged commands that are available only to users with the 'wheel' capability, and only after enabling this capability with the Enable command. When you are so enabled, the Executive's prompt is '!' rather than '@'. While you are enabled, IFS will not log you out automatically after three minutes of inactivity as it does normally.

The following privileged commands are defined.

```
! Create (directory) directory-name [new] (password) password
!! sub-commands
```

Creates a directory with the supplied name and password. Capitalization of the name should be precisely as the user wants to see it. Capitalization of the password is unimportant.

If Grapevine authentication is enabled, *directory-name* may or may not be a fully-qualified R-Name of the form *simpleName.registry*. For a local user whose registry is the same as the IFS's default registry, it is best to omit the *.registry* when creating the directory, and just use the *simpleName*. However, for a user whose registry is different from the IFS's default registry, the directory name must consist of the full R-Name, else Grapevine authentication won't work properly.

In general, names for files-only directories should *not* be qualified by registry (and shouldn't be registered in Grapevine either).

For a directory that corresponds to a Grapevine R-Name, the *password* you supply is irrelevant, since IFS consults Grapevine for authentication. For a non-Grapevine name, if you specify an empty password, then the password cannot be used to gain access to the

directory. This is useless for a login directory; but it is useful for a files-only directory to which access is granted solely on the basis of group membership rather than on users knowing the directory's password.

The *sub-commands* are used to change various parameters from their default values. These include all the sub-commands of the Change Directory-Parameters command, described in How to use IFS', and additionally include:

!! Files-only (owner) *user-name*

Declares the directory to be a files-only (i.e., non-login) directory. The *user-name* is the person responsible for this directory; that user is permitted to connect to the directory without giving a password. *User-name* must include a *.registry* part if the user's registry is different from the IFS's default registry, but may be omitted if it is the same as the IFS's default registry.

Files Only directory "Owner" properties: In a Files-Only directory on an IFS, "Owner" means the name of the directory or the name of the person who is presently connected. It allows the name of the user in the "Owner" field to connect without using the password of the directory. Even though the "Owner" has create permission, he must still connect to store files on the directory. However, an access control group which has create permission does NOT have to be connected in order to write new files to the directory.

!! Disk-limit *number*

Specifies the maximum number of disk pages that may be used in this directory.

!! Wheel

Declares the user to have the 'wheel' capability, which permits issuing privileged commands and bypasses all access checking and disk limits.

!! Mail

Creates a mailbox, thereby enabling IFS to receive Laurel mail for this user. (More information about mail is presented in section 8.)

!! Group Membership (in groups) *groups*

!! Group Ownership (of groups) *groups*

!! No Group Membership (in groups) *groups*

!! No Group Ownership (of groups) *groups*

Establishes the user's membership in or ownership of user groups. (The Group Membership sub-command duplicates the function of the top-level Change Group-Membership command.) Use of these commands is meaningful only for non-Grapevine members of non-Grapevine groups.

Note that in an IFS that does not use Grapevine for access control, any user able to log into the IFS is automatically a member of World, regardless of the setting of the directory's group membership. In an IFS that does use Grapevine for access control, a user whose name is registered in Grapevine is a member of World or not according to whether he is a member of the IFS's 'World' group, usually USRegistries^.Internet. A user whose name is *not* registered in Grapevine but who *is* able to log into the IFS (because an IFS directory by that name exists) is a member of World only if so specified by use of the Group Membership command.

If *directory-name* already exists, you are not asked for a password but rather are sent directly into subcommand mode. This permits modifying parameters for an existing directory. In

this context, the following additional subcommands are of interest:

!! Password *password*

!! Not Files-only

!! Not Wheel

!! Not Mail

Note that to make a non-Grapevine directory unusable for login purposes, you should issue the Password sub-command with an empty *password*. The top-level Change Password command cannot be used for this purpose.

The Create command is terminated by typing two Returns in response to the '!!' subcommand prompt. You may cancel the entire command by typing control-C.

The default values of all parameters for new directories are copied from a dummy directory called Default-User. When a file system is created, the default values are Not Files-only, Not Wheel, Not Mail, and Disk-limit 1000. To change the defaults, use the Create command to modify the parameters for the directory Default-User. Not Mail' is the default for files-only directories, even if Default-User specifies Mail'.

! Destroy (directory) *directory-name* [Confirm]

Destroys the specified directory. This operation includes deleting all the files contained within it, and destroying the associated mailbox if there is one.

! Change Directory-Parameters (of directory) *directory-name*

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

These commands work as described in the 'How to Use' document with the addition that while you are enabled, you may access any directory, not just your own. Also, while you are enabled, Change Directory-Parameters has all the sub-commands described above under Create.

6.2. Setting up Grapevine authentication and groups

The commands in this section are used to control IFS's use of Grapevine for authentication and access control.

! Change System-Parameters

!! *sub-commands*

Permits you to issue sub-commands to change one or more system operating parameters. Each sub-command takes effect immediately. Sub-command mode is terminated when you type CR in response to the '!!' prompt.

!! Default-Registry *registry*

Sets the default Grapevine registry to be *registry* e.g., PA, ES, etc. This should be the registry to which the majority of the local users of the IFS belong. Names of directories belonging to these users need not be (and generally should not be) qualified by the registry name; but names of directories belonging to users in other registries must be qualified by registry name.

You should set a default registry even if you are not using Grapevine for authentication and access control. This is to permit local users to present

their user names either with or without registry qualification.

Note: if the mail server is enabled (section 8), *registry* must be the same as the mail server registry specified by the Registry sub-command of the Mail command.

!! Enable Grapevine Authentication !! Disable Grapevine Authentication

Enables and disables IFS's use of Grapevine for authentication. When this is enabled, any user whose name is registered in Grapevine and who is able to supply the correct (Grapevine) password can log into IFS. When this is disabled, only users who have personal directories on the IFS can log in, and they must present the password maintained by that IFS.

Before enabling Grapevine authentication, you must specify a default registry as described above.

!! Enable Grapevine Groups !! Disable Grapevine Groups

Enables and disables IFS's use of Grapevine for access control (group membership checking). When this is enabled, users' access to files is controlled by membership in Grapevine groups, as discussed in How to use IFS'. When this is disabled, user group membership is maintained entirely within the IFS, and is controlled by the Change Group-Membership or Change Directory-Parameters commands.

Before enabling Grapevine group checking, you must specify a default registry as described above, and you must associate Grapevine group names with IFS group numbers as described below. It is probably not sensible to enable Grapevine group checking without also enabling Grapevine authentication.

!! Group (name of group) *group-number* (is) *group-name*

Associates the Grapevine *group-name* with the IFS *group-number*, which is a number in the range 0 to 61. The *group-name* must be a fully-qualified name acceptable to Grapevine as either a real group (e.g., CSL^.PA) or a pseudo-group (e.g., *.PA).

Caution: if you use this command to associate a name with a group number that already has a name, the meanings of all existing protections that refer to that group will be changed. For example, suppose group 3 is originally associated with the name CSL^.PA, and some files' protections are set to permit reading by members of group CSL^.PA. If you then change group 3 to associate it with the name PARC^.PA, those files will become readable by all members of PARC^.PA.

It is permissible for a group to have a name that is not registered in Grapevine; however, such a group can only contain members whose names are also not registered in Grapevine, and their membership must be managed by use of the IFS Change Group-Membership and Change Directory-Parameters commands. Consequently, this mode of use is really suitable only for IFSs in which Grapevine group checking is disabled.

To eliminate a group's name (i.e., make the group number no longer have a name), use the Group command as described above, but specify an empty *group-name*. That is, when IFS prompts you for the *group-name* and suggests the existing one as a default, erase the existing name using CTRL-W and then strike CR.

!! World (group is) *group-name*

Defines the membership of the 'World' group for this IFS. An appropriate value of *group-name* for most IFSs at Xerox sites in the U.S. is USRegistries^.Internet'. If this is unspecified or empty, any user able to log into the IFS is considered a member of 'World'.

Note: this restricted definition of 'World' is enforced only if Grapevine group membership checking is enabled.

6.3. Converting an existing IFS to use Grapevine

This section outlines the procedure for converting an existing IFS to use Grapevine for authentication and access control. The general idea is to extract the existing authentication information and group structure from the IFS and then capture this state in the Grapevine data base. Some of the information here is also relevant when starting up a new IFS.

6.3.1. Use the Accountant program to obtain an accounts listing and a group membership summary, as described in section 12.

6.3.2. Decide what the default registry is to be. This will usually be obvious: the Grapevine registry of the majority of the local users of the IFS. Set this by means of the Default-Registry sub-command of Change System-Parameters.

6.3.3. For each login (non-files-only) directory in the IFS, make sure that *directoryName.defaultRegistry* is registered as an individual in Grapevine, and that the Grapevine entry indeed represents the same user as the IFS directory's owner. For most local users this will already be the case, assuming that Grapevine is in regular use locally for mail service. (Note: you need not take any action on the built-in directory names Default-User, Mail, and System.)

In the case of a login directory belonging to a user in another registry, you may take one of three actions:

1. Most commonly, such a directory exists solely to give that user access to files in other directories on the same IFS, and the user does not store any files at all on his own directory. In this case, simply destroy the directory and instruct the user to use his own full R-Name when logging in. If the user is a member of any IFS user groups, you should make sure that the user's full R-Name appears in the corresponding Grapevine groups (below).
2. If the remote user actually uses his own directory for storage of files on the IFS, you should change the directory name to include the registry qualification. For example, if the IFS's default registry is PA, and user Jones.ES has a directory whose name is simply Jones' (or JohnJones' or something), then the directory's name should be changed to Jones.ES'. Unfortunately, the only way to do this is to create a new directory, move the files from the old directory to the new (using FTP or Brownie), and destroy the old directory.
3. This alternative is not recommended, but is sometimes workable in desperate situations. If the result of appending the IFS's default registry to the directory name yields an R-Name that is *not* registered in Grapevine, then you can leave the directory as-is. The disadvantages of this alternative are that the user cannot gain access to files by virtue of his R-Name being in Grapevine groups, and the directory name may conflict with a name registered in Grapevine in the future. Note: you may need to explicitly specify the directory's membership in the World group, as discussed above in the description of the Group Membership sub-command of Create.

Since all registered users are able to log in under their full R-Names, Guest' accounts are no longer needed and should be abolished. This will result in a substantial improvement in information security. Accounts that permit automatic access by programs using compiled-in credentials should likewise be abolished; procedures for dealing with such situations are described in the Access

Controls' memo.

6.3.4. In general, the names of files-only directories should *not* be registered in Grapevine. Ordinarily, users gain access to a files-only directory by virtue of membership in user groups referenced by that directory's connect protection, not by presenting the directory's password. However, so long as the result of appending the IFS's default registry to the directory name yields an R-Name that is *not* registered as an individual in Grapevine, the directory may be left as it is; IFS will use local information to authenticate users' attempts to connect to it. (If you are so unfortunate as to encounter a files-only directory whose name *is* registered in Grapevine, then either users must use the Grapevine password to connect or you must change the name of the directory.)

6.3.5. Issue the Enable Grapevine Authentication sub-command of Change System-Parameters. IFS will now use Grapevine for user authentication, while continuing to use local user group information for access control.

6.3.6. You will need the cooperation of the Grapevine registry's maintainer to perform this step. Compare the IFS group membership information obtained in step 6.3.2 with the existing Grapevine groups, using the Maintain program. In most cases you will find that some existing Grapevine group is substantially the same as the IFS group, and that any discrepancies are most likely mistakes or oversights. (This is particularly true of organization and project groups.) In such cases, adjust the Grapevine group membership as necessary and use it for the IFS group. For the remaining IFS groups, create new Grapevine groups with the same memberships as the IFS groups. Remember that each name in a Grapevine group must include a registry name, *even when the registry name is the same as the IFS's default registry.*

The groups used for IFS access control should be chosen with some care. The simplest groups that will do the job should be used. Grapevine takes a long time to check for membership in extremely large or complex groups (e.g., SDD[^].ES, which is both large and complex); this has the potential for causing severe performance bottlenecks. In this connection, it should be mentioned that pseudo-groups of the form **.registry*' (e.g., *.PA, *.ES) can be checked extremely rapidly, in contrast with groups such as AllPA[^].PA which require lengthy enumerations.

It is OK for a group to refer to other groups, but the *simpleNames* of those other groups must end in [^], else they won't be expanded by Grapevine when checking for group membership. (However, it is not required that the top-level group name contain [^].)

Now use the Group sub-command of Change System-Parameters to associate the chosen Grapevine group names with the in-use IFS group numbers. Again, each group name must be a full R-Name, including registry.

You should do this step carefully to ensure that you have captured in Grapevine the existing IFS group structure, because step 6.3.8 will invalidate and destroy the IFS group structure.

6.3.7. Decide on a definition of the 'World' group. This controls access to all files (both existing and new) whose protections specify 'World' which is to say, all files accessible to Guest in IFSs that have had Guest accounts until now.

'World' is intended to be an all-encompassing group that permits limited access to public files by all authentic users. The main reason for restricting 'World' membership is to satisfy U.S. technology export regulations. Several non-U.S. Grapevine registries are now being established, and foreign access to U.S. information is expected to be conducted through more formal and controlled channels. (This topic is discussed in detail in the Access Controls' memo.)

Therefore, the definition of 'World' appropriate for most IFSs in the U.S. is a special group called USRegistries[^].Internet, which includes all members of all U.S. registries. Certain IFSs may choose to adopt a more restricted group for 'World'. Note that a user who is not a member of an IFS's 'World' group can still be a member of other of that IFS's groups and can access files whose protections mention those groups.

Use the World sub-command of Change System-Parameters to set the name of the IFS's 'World' group.

6.3.8. Issue the Enable Grapevine Groups sub-command of Change System-Parameters. The conversion to Grapevine is complete.

6.3.9. You may now delete all user directories that have disk limits of zero. Such directories presumably exist solely to give the users access to other directories on the IFS; since authentication and access control are now done using Grapevine, these directories are no longer needed.

Exception: directories that have special capabilities (Mail' or Wheel') should *not* be deleted, as these capabilities are remembered only by the IFS and not by Grapevine.

It is worth mentioning the measures IFS takes to prevent Grapevine from becoming a bottleneck and to enable continued service if all Grapevine servers become unavailable. When a user first logs in or first attempts to access a file in a way that requires membership in some group, IFS interacts with Grapevine in whatever ways are necessary and remembers the result. This means that subsequent logins or file accesses involving the same group membership do not require any interaction with Grapevine.

The Grapevine information that IFS remembers locally is invalidated after 12 hours; this prevents IFS from getting more than 12 hours out of date with respect to Grapevine. Furthermore, IFS remembers only *successful* user accesses, not unsuccessful ones. This means that when a new Grapevine R-Name is created for a user, or a user is added to a Grapevine group used for access control by IFS, the changes have an immediate effect on the user's access to IFS. However, when an R-Name or group member is removed, IFS may continue to allow access on the basis of the obsolete information for up to 12 hours.

Finally, if Grapevine becomes inaccessible for an extended period, IFS will continue to use its remembered Grapevine information indefinitely.

7. Other privileged commands

! Disable

Leaves enabled mode (the Executive's prompt reverts to @').

! Halt

Stops IFS and returns control to the Alto Executive as soon as all present users (including you) log out or disconnect. If any Leaf connections are active, there may be a delay of several minutes before the connections are broken and IFS halts.

! Show Printing-requests (for user) *user*

! Cancel (printing requests) (for user) *user*

If you type just RETURN in place of *user*, these commands will run through all printing requests for all users.

! Change System-Parameters

!! *sub-commands*

Permits you to issue sub-commands to change one or more system operating parameters. Each sub-command takes effect immediately. Sub-command mode is terminated when you type CR in response to the !!' prompt.

The following sub-commands have permanent effects that survive restarts of IFS. (The permanent information is kept in file <System>Info!1 in the IFS's primary file system.)

!! Clock-Correction *correction*

Sets the software clock correction, which is specified as a sign (+ or) followed by a decimal number. This causes the Alto clock to run faster

(+) or slower () than its nominal rate by that number of seconds per day. Alto clocks are quite stable but not particularly accurate, and software correction is desirable in a server that runs continuously for long periods of time.

The amount by which the clock should be corrected may be determined by comparison with an accurate reference over a period of several days (using the IFS Executive's 'DayTime' command), or by use of an accurate frequency counter to measure the Alto system clock (slot 5 pin 63 on an Alto-I, slot 13 pin 63 on an Alto-II) and computing the correction by the formula

$$c = 86400 * (1 - f / 5880000)$$

where f is the frequency in Hz.

!! Server-Limit n

Limits the number of simultaneous server processes (FTP, Chat, and Mail combined) to n . At present, n may be as high as 6 on an Alto with 64K of memory, 8 with 128K, and 10 with 192K or more; the default Server-Limit is one less than the absolute maximum. See section 13.2 for information on setting this parameter properly.

!! Enable Press-printing	!! Disable Press-printing
!! Enable CopyDisk-server	!! Disable CopyDisk-server
!! Enable Boot-server	!! Disable Boot-server
!! Enable Name-server	!! Disable Name-server
!! Enable Time-server	!! Disable Time-server
!! Enable LookupFile-server	!! Disable LookupFile-server

Enables and disables the Press printing facility and various servers (the FTP server is always enabled, and the mail system is controlled by subcommands to the Mail command). When a file system is created, all the servers are disabled.

!! Enable New-boot-files	!! Disable New-boot-files
--------------------------	---------------------------

This controls the operation of the automatic boot file maintenance mechanisms, and is relevant only if the boot server is enabled. If New-boot-files is enabled, IFS will obtain and maintain copies of all boot files available from any other boot servers on the same Ethernet. If New-boot-files is disabled, IFS will maintain only those boot files that it already has. When a file system is created, New-boot-files is enabled.

!! Enable Leaf-server	!! Disable Leaf-server
-----------------------	------------------------

Enables and disables the Leaf page-level access server (see section 9.4). The Enable command does not take effect until IFS is next restarted, unless IFS was last restarted with the /L switch. Similarly, the Disable command does not free resources consumed by the Leaf server until IFS is next restarted. When a file system is created, the Leaf server is disabled.

The following sub-commands have one-time-only effects.

!! Disable Logins

Disallows further user access to the system. This is useful during debugging and while reloading the file system from backup.

!! Enable Logins

Cancels the effect of Disable Logins.

!! Reset-Time

Causes IFS to reset its clock from a time server on the directly-connected Ethernet. This operation is performed once automatically, immediately after IFS restarts.

8. Mail system

IFS contains a mail server that is compatible with Laurel and the Grapevine message system. IFS can keep in-boxes for users of that system and can also forward mail to mail servers in other Grapevine servers, IFSs, and Maxc.

If your IFS is in the Xerox Research Internet and you wish to operate a mail server, you should first consult the network support organization <NetSupport.WBST> to obtain useful advice. A mail server must be assigned a *registry name* that is distinct from the name of the IFS Alto itself. An IFS can be either a self-contained registry or an adjunct to a Grapevine registry.

To enable a user to receive mail, issue the Mail subcommand of the Create command, as described previously. Of course, if you turn on the Mail capability for Default-User, then all new user accounts you create subsequently will have Mail capability automatically.

When mail is received from a user mail program such as Laurel, it is queued briefly in files named <Mail>New>Mail!*'. A process called MailJob then wakes up and distributes the messages to individual in-boxes, which are files named <Mail>Box>user-name!1'. When a user retrieves the mail from his in-box, the in-box file is reset to empty. If you disable the mail capability for a user who has mail pending in his in-box, the in-box file will be deleted next time it is read.

The MailJob process also forwards mail to other mail server hosts. Specifically, messages addressed to *user.registry*, where *registry* is the name of some other mail server, will get forwarded to that server by the Mailer process. While being forwarded, such messages are queued in files named <Mail>Fwd>registry'. If *registry* maps to multiple addresses, as does *GV*' and other Grapevine registries, then the Mailer will try each of those addresses, closest first, until it succeeds in delivering the messages.

When a file system is first created, the mail system is disabled. You should set the enable switches and configure the mail system using the commands below.

8.1. Mail system commands

The mail system is controlled by a command processor which is entered via the privileged Mail command to the IFS Executive. The commands that change parameters take effect immediately and survive restarts of IFS.

- | | |
|----------------------------|-----------------------------|
| * Enable (mail) System | * Disable (mail) System |
| * Enable (mail) Forwarding | * Disable (mail) Forwarding |

Enables and disables the mail system. The first command turns on and off the mail system as a whole; the second command enables or disables forwarding of mail to other mail servers.

- * Dead-letter (recipient name) *name*

Specifies the name of the in-box to which notification of mail system problems (e.g., undeliverable messages with no return address) should be directed. *Name* may be the name of an in-box on this IFS or (if forwarding is enabled) the fully-qualified

name (*user.registry*') of a recipient on some other server. If the IFS has access to a Grapevine server, *DeadLetter.MS*' is a good value for *name*.

* Distribution-lists (directory name) *name*

Specifies the name of the directory that holds distribution lists (extension *.dl*') that are to be remotely accessible via the mail server. *Name* must *not* be enclosed by '<' and '>'; that is, if you keep distribution lists as *<System>DLs>*.dl*' , you should specify *Distribution-lists System>DLs*' .

This mechanism is used to keep distribution lists only for non-Grapevine registries. For Grapevine registries, distribution lists (groups) are maintained exclusively by Grapevine, even if IFS keeps some of the mailboxes for that registry.

If *name* is empty, the distribution list retrieval mechanism is disabled.

* Grapevine (server name) *name*

Specifies the name of a Grapevine server (ordinarily *GV*'). Mail to a user at a remote registry is forwarded to *name*. Mail to a user whose registry maps to this IFS or matches the registry system parameter, but for which no local mailbox exists, is forwarded to *name*. If *name* is empty, mail for a remote user is forwarded to its registry, and in the second case is returned as undeliverable.

* Registry (we are part of) *name*

Specifies the name of a Grapevine registry to which this IFS belongs. Mail to a user whose registry matches *name* is treated as if the recipient's mailbox is local. If no local mailbox exists and a Grapevine name has been specified, then the mail is forwarded there. If *name* is empty then this IFS's registry name is determined by looking up its machine address (mail server socket 7). Note: if Grapevine authentication is enabled, *name* must be the same as the default registry established by the *Default-Registry* sub-command of *Change System-Parameters*.

* Status

Gives the current status of the mail system, and a summary of mail server operating statistics covering the interval since the file system was created or last reloaded. Statistics are kept on five items; for each item three things are recorded: *Samples*' , the total number of times a statistic for that item was recorded; *Avg*' , the average value of the statistic; and *Histogram*' , an eight slot, logarithmically scaled histogram of the values.

Length (characters)

This is the length of a message received by the mail server, including header text. A message with multiple recipients is counted once in this statistic.

Recipients

This is the number of recipients (*To:*' plus *cc:*') of a message received by the mail server.

Sort delay (sec)

This is the time between receiving a message and appending copies to local in-boxes or queueing copies for forwarding. It is recorded once for each local recipient and once for each remote mail site.

Fwd delay (sec)

This is the time between queueing a message for a remote mail server and successfully delivering it to that server. The statistic is recorded once for each message forwarded, even if the message is addressed to multiple recipients at the given remote server.

Retrieve delay (min)

This is the time between the mail system's appending a copy of a message to a local in-box and the owner's retrieving it.

If any messages have been discarded by the mail system, the number of occurrences will be displayed. A message is discarded when it can't be delivered to a recipient and it can't be returned to its sender and it can't be delivered to the dead letter destination. This is an indication of serious trouble.

*** Reset (mail statistics)**

Resets the mail statistics, which are kept continuously and ordinarily survive restarts of IFS.