UNIVERSITY OF EDINBURGH
Edinburgh Regional Computing Centre

User Note 61

(January 1986)

Title:

EMAS-3: Naming Files

Author:	Roderick McLeod John Murison Keith Yarwood	Contact: Advisory service	Software Support Category: n/a

Synopsis

This Note describes the file system on EMAS-3. It is written for existing users of EMAS 2900.

Keywords

EMAS-3, file groups, files, filestore, hierarchy, NEWGROUP, USEGROUP

Edinburgh Regional Computing Centre

James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JZ. Telephone 031-667 1081

Introduction

This Note describes the EMAS-3 file system. With respect to EMAS 2900, the new file system extends the way in which files are named; this should assist users in organizing their work.

Outline of the changes

The file system development extends the way in which files can be named. It does not affect the contents of files, nor the ways in which you can use them. Using the new facilities you can:

- collect related files into groups. A group might cover a topic or project, or contain files organized in other ways that suit your style of working.
- specify that only a group of files rather than all your on-line files are to be examined and manipulated by commands such as FILES and PERMIT
- carry out operations on complete groups, as well as on individual files.
- use longer filenames, thus reducing the need for abbreviations.

Filenames on EMAS 2900

On EMAS 2900 a full filename is constructed from two elements, a username and a local name separated by a full stop:

username.localname

The username consists of 6 characters assigned to you by the System Manager when you are accredited to use EMAS.

The localname is chosen by you and consists of up to 11 alphanumeric characters of which the first must be a letter. When you refer to your own files you only need to use the localname. Within the username and the localname the case of letters has no significance, so

ERCC98.FILEA, ercc98.Filea and ERCC98.filea are equivalent

All files within EMAS 2900 have names constructed like this.

Filenames on EMAS-3

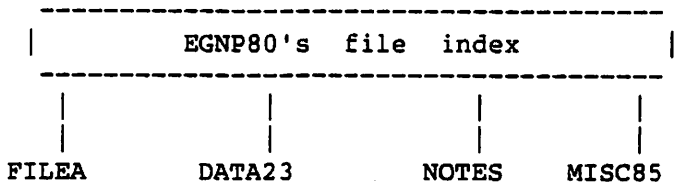
As on EMAS 2900 the username consists of 6 characters allocated by the System Manager when you are accredited to use the system. An important difference from EMAS 2900, however, is that the username must be separated from the rest of the filename by a colon rather than a full stop. Therefore

CONLIB.GENERAL becomes CONLIB:GENERAL

As on EMAS 2900 the username can be omitted when you are referring to your own files.

File Groups

The EMAS file system keeps information about a user's on-line files in his or her file index. Thus, for example:



This arrangement applies to EMAS-3, just as it does to EMAS 2900. On EMAS-3, however, it is also possible to group files together. A group is a named sub-index; its name is a common element in the names of files which are in the group.

For example, if you use EMAS for teaching and research you might choose to divide your files between two groups called TEACH and RESEARCH; and if your username is EGNP80 then the names of files used for teaching could be:

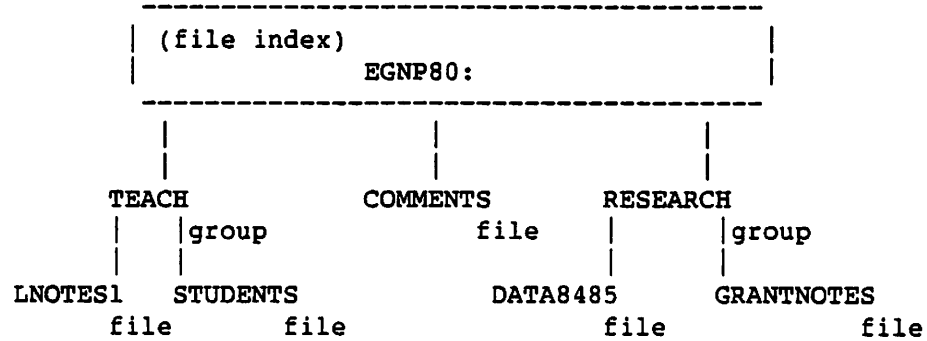
EGNP80:TEACH.LNOTES1
EGNP80:TEACH.STUDENTS

and your research files could include:

EGNP80:RESEARCH.DATA8485
EGNP80:RESEARCH.GRANTNOTES

You might also have a file of comments which is not in any group.

All this can be represented pictorially:



For obvious reasons, this type of filestore is sometimes described as 'hierarchical'.

It will seen from this that an EMAS-3 file which does not belong to a group is exactly equivalent to *any* on-line file on EMAS 2900. In fact, it is quite possible to work only with such files on EMAS-3, and ignore the existence of groups. However, they are likely to be in general use on EMAS-3 and it is therefore advisable to learn at least a little about them.

Apart from enabling related files to be classified together, a group can be given attributes, such as access permission and cherish status, which then apply to all the files in the group.

Naming Files and Groups

Files which do not belong to a group are equivalent to files on EMAS 2900. Apart from the change from a full stop to a colon they are named in exactly the same way as files on EMAS 2900:

username:localname

The localname consists of up to 11 alphanumeric characters of which the first must be a letter. The following are valid EMAS-3 filenames:

ERCC98:FILEABC
CONLIB:GRAPHICSLIB
SUBSYS:SYSTEM

The groups TEACH and RESEARCH in the earlier examples belong to EGNP80 directly and not to other groups. The format of the name of a group is identical to that of a file, consisting of up to 11 alphanumeric characters (of which the first must be a letter). Within a particular group the files and groups can have any localnames, except that you cannot have a file and a group with the same name at the same level within a given branch of the hierarchy. But you can have files with the same names in different groups. For example you might choose to keep a NOTES file in each of your groups. Using the example above their full filenames would be:

EGNP80:TEACH.NOTES
EGNP80:RESEARCH.NOTES

It is also possible to have groups within groups. If, for example, you teach on two courses you could use a group for each course and it might be logical to make these groups belong to the group TEACH. They would have names such as:

EGNP80:TEACH.PH3
EGNP80:TEACH.PH4

and files associated with these courses would have names such as:

EGNP80:TEACH.PH3.STUDENTS
EGNP80:TEACH.PH3.EXAMDATES
EGNP80:TEACH.PH4.COURSENOTES

The structure of groups within groups can continue for many levels. For example,
EGNP80:ALPHA.BRAVO.CHARLIE.DELTA.ECHO

is a valid, if somewhat cumbersome, EMAS-3 filename. The only limit is that a full filename cannot exceed 255 characters.

Summary of file naming rules

- The full form of a filename is:

  username:[groupname.]localname
  The field in square brackets can appear zero or more times.
  (The square brackets themselves do not appear in the name.)

- The case of any letters within filenames is not significant.

- The username consists of six characters assigned by the System Manager.

- The groupname fields, if any, each consist of up to 11 alphanumeric characters, of which the first must be a letter.

- The localname field consists of up to 11 alphanumeric characters, of which the first must be a letter.

Creating Groups

Groups are created explicitly using the command NEWGROUP, which takes one parameter, the name of the group to be created. For the examples above you would have to make calls:

Command: NEWGROUP TEACH
Command: NEWGROUP RESEARCH
Command: NEWGROUP TEACH.PH3
Command: NEWGROUP TEACH.PH4

An attempt to create a file or group in a non-existent group will produce a failure message. For example, if you type

Command: IMP TEST1, OBJ.TEST1

before creating a group OBJ then a failure will occur. Similarly, the last two NEWGROUP examples above would fail unless group TEACH had already been created.

A group is a file sub-index; it is *not* a file: creating a group does not create a separate file to contain a list of the group members, as happens on some other systems, for example UNIX. A group can exist even though it contains no files. Indeed, when it is first created it is empty.

It is useful to regard the file index as a 'super-group', which differs from other groups only in respect of its name: six alphanumeric characters, followed by a colon rather than a dot.

A group has the following attributes:

- an owner

- zero, one or more superior group names

- a localname - i.e. a name to distinguish it from other files and groups at the same level

- a CHERISH status (see below)

- an ARCHIVE status (see below)

- a set of access permissions, to its owner, to all other users and possibly to individual users or sets of users (see below)

- it may contain files

- it may contain groups

Using a Group as a Sub-Index

Normally when a file is referenced by its local name alone, the assumption is made that it does not belong to any group. However, it is possible to nominate a group to take the place, temporarily, of one's file index, and file and group names specified thereafter are automatically prefixed by the nominated group's name. The command to nominate a group in this way is USEGROUP. For example:

Command: USEGROUP TEACH

Thereafter,

Command: COPY MARKSA, MARKSB

would be equivalent to

Command: COPY TEACH.MARKSA, TEACH.MARKSB

before TEACH was selected. Note that *after* TEACH was selected, this command would refer to files TEACH.TEACH.MARKSA and TEACH.TEACH.MARKSB!

The prefixing with the nominated groupname takes place for all filenames which do not contain ":" and of which the first two characters are not "T#". Thus to reference a file not in the currently selected group you must give its full filename, except that ":" is a permitted abbreviation for 'username:' for the current process owner. Thus:

Command: COPY MARKSA, EGNP80:MARKS

is the form required to copy a file out of the currently selected group, which may be abbreviated to:

Command: COPY MARKSA, :MARKS

The name of the group currently selected is displayed by the command

Command: USEGROUP =

The command

Command: USEGROUP

is used to cancel the current group selection, so that each localname reverts to being prefixed implicitly by 'username:'.

The Archive Store

A file on archive keeps the full name which it had when it was on-line, and if an archived file is RESTOREd it will return to its original place in the hierarchy of on-line files. If you ARCHIVE a file called BONES.LEG, and subsequently destroy or rename the group BONES, then the file BONES.LEG would still exist in the archive store. The RESTORE command will attempt to create any intermediate groupnames required, if they have in the meantime been destroyed.

(In the EMAS-A service, a file will be archived *only* if an explicit request has been made by the user. The arrangement whereby unused, cherished files are moved to archive after a fixed time interval is being discontinued.)

The command RESTORE is not applicable to entire groups, only to individual files. The command has, however, been extended to enable you to RENAME a file while restoring it. This will make it possible to RESTORE it to a different place in the hierarchy. For example:

Command: RESTORE RESEARCH.MAR83, 19/09/85, RESEARCH.YEAR1983.MARCH

would restore file RESEARCH.MAR83, archived on 19/09/85, and rename it RESEARCH.YEAR1983.MARCH. This facility enables you to develop the structure of your files over the years as your pattern of work changes.

Operations on whole groups

Apart from providing a more flexible scheme for naming files, the new file system makes it possible to carry out operations on all the files in a group. In some cases these will be provided as extensions of existing Subsystem commands, and in others by new commands. The relevant Subsystem commands are:

- ANALYSE: when the parameter is a groupname, ANALYSE gives the same information as does the FILES command with a groupname as parameter, namely the filenames and groupnames belonging to the group (at the next level of the hierarchy only). If ",P" is given as the second parameter, the total size of all files within the group is given (down to the bottom level of each branch of the hierarchy).

- ARCHIVE, CHERISH, HAZARD and PERMIT operate essentially in the same way when the parameter is a groupname. The Archive/Cherish/Hazard status is propagated to each file contained in the group, down to the bottom level of each branch of the hierarchy. In addition, files added to the structure within the group acquire, on creation, the corresponding attribute which has been given to the group. When an archive is performed, and the group marked for archive becomes empty, the group is destroyed. It is also possible, of course, to ARCHIVE, CHERISH, HAZARD and PERMIT individual files within groups.

- COPY: if the parameter is a groupname, the files of the entire hierarchy belonging to the group are copied.

- DESTROY: if the parameter is a groupname, the group will be destroyed only if it is empty. This feature, and the command DESTROYGROUP, are provided to minimize the likelihood of unintended destruction of a possibly large amount of data, for example because of a typing error.

- DESTROYGROUP destroys the specified group, together with all the contained groups and files. *This command should be used with care.*

- FILES: the command FILES without a parameter prints the names of the files and groups at the currently selected ("USEGROUP") level of the hierarchy. It distinguishes between files and groups by appending a "." to the names of the latter. If you use the command with a group name, then you will obtain a list of files and groups at the first level within the nominated group.

- RENAME allows you to change the name of a file *or* group. In addition, you can move a file or group to a different part of your file hierarchy using this command. The parameters of RENAME are two file names *or* two group names. For example:

Command: RENAME :ACCOUNTS.Y8485, :ACCOUNTS.YEAR8485
(to rename a whole group)

Command: RENAME :TEACH.PH3.TIMETABLE, :TEACH.PH3.DATES
(for a single file)

Command: RENAME :TEACH.PH3.TIMETABLE, :GENERAL.TIMETABLE

A movement within the file hierarchy is implied in the last example.

In addition it is possible to use the "wild-carding" feature of the Parameter Acquisition Mechanism to achieve repeated invocation of a command for all files in a group. For example

Command: ANALYSE G.*

causes ANALYSE to be called once for each filename belonging to group G. The effect is of course different from that of

Command: ANALYSE G

Similarly note the difference between

Command: CHERISH G.*

which cherishes all present files belonging to group G, whereas

Command: CHERISH G

causes all present and future files in G to be cherished, because the cherish status is attached to the *group.*

Note that for the command FILES, the use of a "mask" as the first parameter does not lead to repeated invocation of the command by the Parameter Acquisition Mechanism. This is because the command FILES explicitly requests that repeated invocation be suppressed, and acquires the parameter "as is". (Details of this feature will be found in User Note 62, "EMAS-3: Parameter Acquisition Mechanism".)

Partitioned Files on EMAS-3

The contents of files are not relevant to the hierarchical file system. It follows that members of partitioned files are not recognized as part of the hierarchy. A whole partitioned file can have its place in the hierarchy, of course, just like any other file, but its subdivision into members is not 'visible' in the hierarchy. Partitioned files are available in EMAS-3 as a convenient facility for the internal organization and subdivision of files. They are not intended to provide facilities which emulate, or are compatible with, the organization of whole files into groups.

Partitioning should be seen as a way of subdividing or organizing the data within a file. It is not supported as an alternative to the filestore hierarchy. Facilities applicable to the hierarchy will not in general apply to members of partitioned files, and vice versa. Partitioned files will continue to serve the needs of some users, but in many cases it will be sensible to replace them with groups of files. The list below indicates the criteria that should be considered when choosing between partitioned

files and groups of files.

- File groups are more suitable when individual members are subjected to frequent changes.

- File groups must be used if different CHERISH status or access permissions are needed for individual members.

- File groups must be used if the files are to hold direct output from programs.

- Partitioned files are suitable when there are large numbers of members (more than 20) whose contents generally remain static. They are also more suitable if you have many very small members or when you need to access many members at the same time.

The commands GROUPTOPD and PDTOGROUP are provided. GROUPTOPD copies the contents of a group into a partitioned file:

Command: GROUPTOPD group, pdfile, compress

The first parameter is the name of a group, which must exist and be accessible for reading.

The second parameter is the name to be given to the partitioned file produced from the specified group. A file of this name must *not* exist already.

The third parameter is optional. It specifies what you want to happen to groups which are members of the group you are copying. If there are none then this parameter is ignored. This parameter should be set as follows:

n if you do not want sub-groups to be copied into the partitioned file.

y (the default) if you want each sub-group to be copied into a separate partitioned file and each of these partitioned files copied into the main partitioned file. This process will continue with sub-sub-groups etc. thus producing a partitioned file with a structure which is equivalent to that of the original group.

PDTOGROUP copies the contents of a partitioned file into an equivalent group.

Command: PDTOGROUP pdfile, group, expand

The first parameter is the name of a partitioned file, which must exist and be accessible for reading.

The second parameter is the name to be given to the group produced from the specified partitioned file. A group (or file) of this name must *not* exist already.

The third parameter is optional. It specifies what to do with any members which are, themselves, partitioned files. If none of the members are partitioned files then this parameter is ignored. This parameter should be set as follows:

n if you want to treat members which are partitioned files in the same way as all other members - that is, left as partitioned files in the group.

y (the default) if you want each of these members copied into a sub-group containing separate files for each of their members. (And so on as far as the nesting continues.) If this option is selected the resulting file group will have a structure which is logically the same as the original partitioned file.

File space limits

With the new hierarchic organization of the file store there is still an upper limit on the number of distinct files which you may keep on-line. As on EMAS 2900, this limit is set individually for each user, and it will normally be counted in tens or hundreds - not thousands. The larger capacity of the discs used by the EMAS-A service may allow the limit to be increased for some users, but the file store hierarchy does not, of itself, increase the limit or allow you to exceed your limit. Files within groups, and groups themselves, will be counted against the limit. The filestore hierarchy does *not*, therefore, allow an increase in the number of files which you may keep on-line.

Conclusion

The introduction of a hierarchical file store mechanism on EMAS-3 makes it possible to organize your files into groups which relate to the sections and subsections of your EMAS work. In many cases it supersedes the use of partitioned files and overcomes the serious limitations which they impose on the use of individual members.