VESOFT CONSULTANTS
506 N.Plymouth Blvd.
Los Angeles, CA 90004
U S A
(213) 465-7453

MPE/3000: EFFECTIVE USE OF MPE FILESET CONCEPT

By Eugene Volokh, Vladimir Volokh

Presentation to the HPGSUG
1980 International Meeting
Orlando, Florida, USA

ABSTRACT

This paper will discuss how one can use the concept of 'FILESETS' for general application systems development; it will describe the way in which MPE supports filesets, and discuss certain holes that are patched up by MPEX, a useful utility which greatly expands this concept.

WHAT IS A FILESET?

The MPE operating system supports the important concept of filesets. A fileset is a very convenient way to describe more than one file; e.g. just one file can be referred to as 'FILE.GROUP.ACCOUNT', but a name such as '@.SOURCE.AP' can refer to an entire SET of FILES -- in this instance all the files located in the group SOURCE of account AP; or, you can say 'AP@SRC', which means 'all the files that start with AP and end with SRC in the logon group and account'.

WHY A FILESET?

When an application program is written, it usually does not stand alone; it is logically linked to many other programs -- in short it is part of a SYSTEM of PROGRAMS. Whereas the system concept is a very good , it has its problems; very important one is that it is inherently hard to manipulate an entire system of programs. For instance, as you may well know, to store off a system of 50 programs I can use a FILESET in the STORE command, which is certainly much better than listing each one of the 50 filenames. This feature, allowed on the LISTF, STORE, and RESTORE commands is very handy. But, there are still many things to be desired -- can I list all my sources to the line printer? Can I find where in my 50-program system I refer to the item 'NUMBER-WIDGETS'? If I change my COPYLIB or make some other drastic change to my system, can I recompile it? Can I copy this system into another group, or RENAME it from AP@SRC to BU@SRC? With the current MPE facilities, the answer to this is unfortunately NO.

MPEX

These are the problems that confronted us in our application system development tasks. We found that after a substantial amount of alteration, our system needed a total recompilation; we found that when a new programmer arrived, we needed to give him a complete listing of all the sources in our system; we found that it was often necessary to find or change all occurences of a string in our system. We saw that MPE's fileset handling as it is implemented now is not adequate and not consistent. MPEX/3000 remedies this unfortunate inadequacy. It allows the fileset concept to be applied to several MPE commands/subsystems Thus, you can do a COBOL compile on a fileset; you can do a RELEASE/SECURE of an entire fileset; you can do a purge of an entire fileset; you can execute an EDITOR command upon an entire fileset; you can FCOPY/RENAME an entire fileset; you can list all the files of a certain filecode (PROG, EDITOR, etc.) that exist in the fileset.

MPEX even allows you to use the powerful SET OF SETS; i.e., you can perform an operation on several filesets at a time!!!

MPEX USAGE EXAMPLES

For instance, the following are some of the situations in which MPEX can prove to be an essential programmer's tool.

* A bulky 50-program system was written. It was then determined that a COPYLIB file must be changed, and thus the system must be recompiled. Ordinarily, you would need hundreds of MPE commands to do this. With MPEX, you can type one command and presto! a job is streamed that will compile and prep all of those programs, while you can continue productive work at your terminal.

* It was decided that a certain data base must be changed (e.g. with Alfredo Rego's ADAGER/3000). It would take about 3 hours to perform the actual structural changes; it would take days to modify the associated programs! With MPEX, you can get a complete listing of your system; you can find where an altered item/set is referenced; you can even automatically change the name of some item or set in all of your programs; and then, once all the modifications are completed, you can quickly and easily recompile the entire system.

* It is often very desirable to get a listing of the entire system of programs. Without MPEX, this task will be very time-consuming and error-prone. With MPEX, one MPEX EDIT command will create a listing of all those programs.

* It was decided that all the sources in the system are to be transferred from a crowded PUB group into the SOURCE group. Ordinarily, it will take many hours and many mistakes before this job is finished. With MPEX, one FCOPY or RENAME command will do the trick.

* Due to numerous system failures, many EDITOR K-files were generated. With MPEX, you can purge all those files in the entire system, in an account, or in a group; optionally, you can ensure that you will purge only the files you want to by asking MPEX to perform the PURGE with Y/N verification; i.e. for every file it will ask you whether you REALLY want to purge it.

* The system manager determines that disc space is very low because people are building huge, space-wasting data bases. He can do a LISTF of all the data-bases in the system and see which ones are necessary and which are not. Conversely, if an installation has decided to convert to IMAGE from KSAM, a system manager can insure that nobody is still using KSAM files by using MPEX's LISTF command to find all the KSAM files in the system.

* A set of source files must be secured against some programmers or must be released so everyone CAN look at them. Just run the MPEX RELEASE or SECURE command.

The list can go on and on.

MPEX supports many different commands. Following are some brief descriptions: (ALL THESE COMMANDS CAN BE EXECUTED ONLINE!!!)

COMMAND NAME	DESCRIPTION
COBOL	This command, similar in syntax to the MPE COBOL command, will compile and :PREP one fileset (the source-set) into another (the object-set), online or offline.
COBOLII	This command is identical in syntax to the COBOL command except that it uses the COBOLII compiler.
EDIT	The EDIT command will perform a specified EDITOR/3000 command upon a fileset.
FCOPY	This command will copy one fileset into another via FCOPY/3000. This command also has features that allow it to copy using SUPRSORT/ROBELLE, a product which provides faster file copying than FCOPY.
FORTRAN	This command is identical in syntax to the COBOL command except for its use of the FORTRAN compiler.
LISTF	MPEX provides an improved version of the :LISTF command which allows you to LISTF all the files with a certain file code (e.g. EDTCT, PROG, PRIV, USL, etc.) in the specified fileset.
PURGE	The PURGE command of MPEX lets you purge an entire set of files. This operation can be performed in a stream or ONLINE. It is suggested that the Y/N verification feature of MPEX (SEE BELOW) be used with this command.
QEDIT	This command will execute any QEDIT/ROBELLE command on a specified fileset. QEDIT is a ROBELLE Consulting product that is a fast, easy-to-use, and powerful replacement for the HP/3000 EDITOR.
RELEASE	This command lets you RELEASE (suspend security provisions) for a fileset.
RENAME	RENAME allows you to rename a fileset into another; this can be useful for renaming entire groups or accounts (RENAME @.@.OLD, @.@.NEW).
RPG	This command is like the COBOL and FORTRAN commands, except that the RPG compiler is invoked.
SECURE	SECURE is the opposite of the RELEASE command; it allows you to restore default security provisions to files that had previously been RELEASEd.

SPL	The SPL command is syntactically and functionally identical to the COBOL, FORTRAN, and RPG commands (except that SPL is used).
HELP	MPEX features an online HELP facility similar to that of MPE; it can be used for getting information on any MPEX commands and key features.
SP or SPOOK	All $STDLIST outputs of jobs streamed by MPEX stay in the MPE spooler, and can be looked at, printed out, and/or deleted via the spooler (SPOOK.PUB.SYS) subsystem. MPEX allows you to enter the spooler subsystem directly using this command.
ED or EDITOR	MPEX allows you to enter the EDITOR sysbsystem directly.
QE or QEDIT	MPEX allows you to run QEDIT.PUB.ROBELLE directly.
[:]mpe-command	Specifying any command other than the ones above directs MPEX to execute it as an MPE command; almost all MPE commands (e.g. :RUN, :PREP, :EDITOR, :BASIC, etc.) are allowed. As MPEX uses some MPE command names for its commands, some commands like PURGE, LISTF, and others will trigger their MPEX equivalents. To avoid ambiguity, prefix these commands with a ':' if you want them executed as an MPE command.

ADDITIONAL FEATURES OF MPEX/3000 COMMANDS.

Some other things you can do with MPEX commands:

Any command prefixed by a '!' is executed ONLINE as opposed to offline (in the background, the default). IT IS NOT SUGGESTED THAT COMPILES OF A LARGE NUMBER OF PROGRAMS BE DONE ONLINE!

A command prefixed by a '?' is executed ONLINE with Y/N verification; i.e., for every file selected the name of the file is printed, and you are asked to reply YES or NO. If you hit a carriage return or type anything other than a 'Y', that file will NOT be used. This is primarily useful with the PURGE command.

MPEX/3000 AS AN 'OPERATING SYSTEM' ON ITS OWN.

With all the above features, plus the capability to run, prep, and compile programs from MPEX, MPEX can be viewed as a self-contained operating system; it has been found to be possible and beneficial for programmers to 'live' in MPEX, running EDITOR or QEDIT when they must edit a program, compile single programs or sets of programs from MPEX, go into the spooler to retrieve the results of the job streams that MPEX streams off, etc. Moreover, if you choose not to use MPEX in this way, there are special hooks that can be installed that will allow you to run MPEX out of EDITOR rather than EDITOR out of MPEX; thus in EDITOR, you can say:

```
%COBOL AP@SRC, AP@OBJ, $NULL
```

and the command 'COBOL AP@SRC, AP@OBJ, $NULL' will be executed as if you were in MPEX.

AN EXAMPLE MPEX SESSION:

(all the lines that you type are marked '<<you>>').

```
:RUN MPEX.UTIL.SYS                                              <<you>>

This is MPEX/3000 Version 1.0 (VESOFT Consultants)
For help or in case of problems type HELP

%HELP COBOL                                                     <<you>>

[?]COBOL source-set,program-set,[list-set],[master],[new-set],
                    [usl-set],[prep-time-parms]

Compiles COBOL programs

%FILE COPYLIB=COPYLIB.SOURCE                                    <<you>>
%COBOL GL@SRC, GL@OBJ, XL'                                      <<you>>
 #J17
See the spooler after 'DONE' message
%
FROM/J17 USER.ACCT/ GL003SRC COMPILE FAILED!
FROM/J17 USER.ACCT/ GL012SRC PREPARE FAILED!
FROM/J17 USER.ACCT/ COBOL DONE!

SPOOK                                                           <<you>>
SPOOK B.00.03   (C) HEWLETT-PACKARD CO., 1976
> SHOW                                                          <<you>>
#FILE   #JOB    FNAME     STATE  OWNER
#034    #J17    $STDLIST READY   USER.ACCT
> TEXT 34       << look at spool file #034 >>                   <<you>>
> LIST 100/200  << list lines 100/200 >>                        <<you>>
...              << the Spooler lists the lines >>
> EXIT                                                          <<you>>
```

```
%EDIT GL@SRC, LIST ALL,OFFLINE << print all files to LP >> <<you>>
 #J19
See the spooler after 'DONE' message
%
FROM/J19 USER.ACCT/ EDIT DONE!

FCOPY GL@SRC, BU@SRC, NLW      << create online backup >>  <<you>>
 #J20
See the spooler after 'DONE' message

%EDITOR                                                   <<you>>
/TEXT GL003SRC                                            <<you>>
/<< this is the program which gave a compile failed; modify it >>
/KLEP                                                     <<you>>
/EXIT
%COBOL GL003SRC, GL003OBJ, *LP                            <<you>>
 #J21
See the spooler after 'DONE' message
%
FROM/J21 USER.ACCT/ COBOL DONE!
RUN GL003OBJ         << check it out >>                   <<you>>
...     << output from GL003OBJ >>
%?PURGE TEST@.JUNK+TRY@.JUNK                              <<you>>
TEST1.JUNK (y/n)? Y  << yes, purge (you reply) >>
TESTXX.JUNK (y/n)? N<< no, retain (default) >>
TRY.JUNK (y/n)? Y    << yes, purge >>
TRY2.JUNK (y/n)?     << CONTROL-Y pressed; operation stops >>

%SHOWTIME                                                 <<you>>
FRI, DEC 17, 1980,  2:33 PM
FROM/J20 USER.ACCT/ FCOPY DONE!

%LISTF @.@.PROD,PRIV              << typo! >>             <<you>>
UNKNOWN COMMAND NAME (CIERR 77)
%REDO                                                     <<you>>
LISTF @.@.PROD,PRIV
 D                                                        <<you>>
LISTF @.@.PROD,PRIV
                                                          <<you>>

ACCOUNT=   PROD         GROUP=   PUB

FILENAME  CODE  ----------LOGICAL RECORD----------  ----SPACE----
                  SIZE  TYP        EOF      LIMIT R/B  SECTORS #X MX

TEST      PRIV    212W  FB           5          5   1        1  1  1
TEST01    PRIV    333W  FB       29111      29111   1     5053  1  1
TEST02    PRIV    767W  FB       35957      35957   1     7671  1  1

%TELLOP PLEASE PURGE DATA BASE TEST.PUB.PROD - DISC WASTER.<<you>>
%EXIT

END OF PROGRAM
```