

# XEROX PARC CSL MEMORANDUM

December 2, 1974

To: Maxc Users  
From: Ed Taft  
Subject: File Backup and Archiving

This is an update to two previous memos: "File Backup and Archiving" (June 21) and "Archiving" (September 20). New material in this memo includes:

- 1) A revision to backup policies (we now do all backups to disk packs rather than tapes).
- 2) A major change in the syntax of the "Archive" command (as provided in Exec version 1.52 just released by BBN).
- 3) Some suggestions on usage of the archive system.

## *Terminology*

On Maxc, we perform both *backup* and *archiving*. Though both these operations have the effect of copying on-line files to secondary storage, backup and archiving are otherwise completely different.

We perform *backup* of all on-line files at a frequent enough interval to be able to recover files lost or damaged as a result of hardware or software errors or gross blunders on the part of system personnel. Hence, the information saved during backup is such as is needed to restore the file system to precisely the state it was in at the time of that backup. Convenience of access to backed-up files is of no importance, since ideally we should never need to perform such access.

*Archiving*, on the other hand, is a mechanism for providing another level of addressable file storage, beyond the on-line Tenex file system. Files may selectively be sent to the archive for any of a number of reasons; e.g. to permanently preserve a valuable working version of a program, or to free up disk space occupied by files not referenced for a long time. Important attributes of an archive system are permanence, reliability, and ease of interrogation and retrieval.

## *Backup Procedures*

File backup is done to auxiliary disk packs in the following manner: Once a month, the entire file system is dumped. Then, every weekday, an "incremental dump" is performed to back up all files written since the previous dump (full or incremental). Hence, in case

of a catastrophe, it would be possible to restore the file system to its exact state at the time of the most recent incremental dump.

Since disk packs are fairly expensive, we have only a limited supply of backup packs, and we must recycle them regularly. Our present strategy is as follows: After each monthly full dump is completed, the packs are taken to Building 31 for safekeeping (in case of fire, flood, earthquake, or similar event) and the packs from the previous dump are returned to the Maxc room for re-use the next month. Incremental dump packs are recycled as we run out of packs (but never before the incremental dumps on them have been superceded by a newer full dump). At present, we anticipate having enough packs to keep approximately 1-1/2 months' worth of incremental dumps, though of course this depends on the file creation rate (which is presently on the order of 10,000 pages per day!)

In the past, some users have discovered that I am only marginally receptive to requests for retrieval of files from backup tapes. Further, my reaction tends to vary according to my own subjective feelings about how reasonable the request is or how dumb the mistake that caused the file to get lost. This will continue to be the case (perhaps even more so) concerning requests for retrieval of files from the *backup* system. Such requests are a nuisance because the backup system is not organized for easy interrogation or retrieval of individual files. Furthermore, obtaining a file from the previous full dump would require getting the packs brought back up from Building 31.

### *The Archive System*

Since files sent to the archive are kept "forever", using disk packs for archiving is out of the question; we have to use tape. The procedures currently used for operation of the archive system are as follows.

There are two forms of archiving: voluntary and involuntary. A user may voluntarily request that a file be archived. The option exists for specifying whether the file is to be retained or deleted from on-line storage after archiving.

Once a week (currently on Monday), someone (currently Chuck Geschke or myself) "runs" the archive system to process all archive requests that have accumulated in the past week, write the appropriate files onto tape, and perform necessary deletions and bookkeeping.

When a file is archived, an entry is made in an archive directory containing some of the attributes of the file and information concerning the date and time and the tape on which it was archived. An "Interrogate" command exists for obtaining information about archived files in a manner similar to the use of the "Directory" command for on-line files.

A request may be made to retrieve any archived file to which you have read access. Such requests accumulate until someone (currently Chuck Geschke or myself) processes them. We currently do this every weekday, so you should expect a one-day turnaround for retrieval of archived files. "Emergency" requests for faster service (or for retrieval of files from the *backup* system) are handled by Chuck Geschke and myself on an individual basis and with a decided lack of enthusiasm. Unfortunately, there is no way for you to do retrievals yourself because the retrieval process requires performing some privileged operations such as restoring file attributes and marking the archive directory.

The most obvious question for the user is: what files should I archive? If there were a simple, universally acceptable answer to that question, we would simply implement the corresponding decision routine and the problem would be solved. Of course, that is not the case. Since the answer is complex, the most this memo can do is lay out the relevant parameters of the archival facility so that users can make the most appropriate use of the system.

There are several classes of files that it is appropriate to archive. It is expected that people will use the archive system to save things like on-line theses and other documents, old but possibly useful versions of programs, etc., that really aren't being used at all but might be useful at some time in the future. In this case, it is generally appropriate to save only source files, since generated files may be recreated from these sources.

Another important use of the archive system is to take "snapshots" of working versions of systems, for future reference or to fall back to in case future versions prove troublesome. For example, whenever a new version of Tenex has been working reliably for several weeks after major changes, I archive all the source files (without deleting them, of course).

It has been the experience at other Tenex sites that the archive has quickly turned into a trash bin, due to people using the "Archive" command as a substitute for the "Delete" command. Users should be aware of some of the characteristics of the archive system so as to avoid inundating it. Archiving a file requires that it be written on two different archive tapes. Each tape can hold approximately 9,000 Tenex pages and takes 40 minutes to an hour to write. Since 9,000 pages is less than the average *daily* turnover for files on Maxc, it can be seen that the archive system will become unworkable unless users are very selective about the files they archive.

If voluntary archiving fails to keep a minimum (roughly 12,000 pages) of free space in the primary file system, we will be forced to implement involuntary archiving criteria. For example, all files not referenced within 60 days might be forcibly archived. Involuntary archiving has the obvious disadvantage that the system's selection of files to archive cannot be made as intelligently as that of the user to

whom the files belong. Therefore, we hope to postpone the introduction of involuntary archiving schemes as long as possible.

The remainder of this memo describes the user facilities available for dealing with the archive system. This documentation is kept on-line as file <DOC>ARCHIVE-SYSTEM.DOC.

### *The Archive Command*

The Tenex Exec has a command called "Archive File" that allows the user to specify that files be archived. The basic form of the command is simply:

`@ARCHIVE FILE list of file names`

where the file names may include asterisks.

This form of the command does not change the status of the file in any way except to mark it for later archiving. Next time the archive system is run, all files marked in this way are written on two different archive tapes and then deleted from the disk. You will be notified via SNDMSG when this process has been completed.

The "Archive File" command has several subcommands, which may be accessed in the usual manner by terminating the primary command with a comma followed by a carriage return. The subcommands modify the action of the "Archive File" command in the following ways:

`@@DON'T DELETE`

Indicates that the file(s) should not be deleted after being written on the archive tapes.

`@@DON'T ARCHIVE`

Indicates that the file(s) should never be archived. This subcommand will be relevant only at such time as forced archiving is instituted, and indiscriminate use of this subcommand will be frowned upon.

There are also other subcommands of no interest to users because they specify default actions ("Deferred", "Delete") or actions that are not yet implemented ("Immediate").

Another command of interest is the "Archive Status" command, given in the form:

`@ARCHIVE STATUS list of file names`

This command reports on the archival status of the specified file(s). Possible states are:

ARCHIVE AND DELETION PENDING -- An "Archive File" command (without subcommands) has been given for this file, and it is to be archived and deleted in the next archival run.

ARCHIVE WITHOUT DELETION PENDING -- An "Archive File" command with the "Don't Delete" subcommand has been given for this file.

ARCHIVE NOT ALLOWED -- An "Archive File" command with the "Don't Archive" subcommand has been given for this file.

NONE -- The file has no pending "Archive File" request.

Note that this command is to be used to get the status of on-line files that have *not* yet been written on the archive tapes. A file that has been written on the archive tapes is considered to be "previously archived", and information about the file is obtained via the "Interrogate" command (described below).

The command:

@ARCHIVE RESET list of file names

resets the archival status of the specified files to "NONE"; i.e., it cancels any previous "Archive File" command that has not yet been processed. This command has no effect on previously archived files (ones that have already been written on archive tapes), and is not to be confused with the "Archive Undelete" command to be described later.

#### *The Interrogate Command*

For every file ever archived from a given directory, an entry is made in a special "archive directory". (This is contained in a file called ]ARCHIVE-DIRECTORY[.;1 which has been made as difficult as possible for you to clobber accidentally.) The "Interrogate" command is used to request information about archived files, in a manner exactly analogous to use of the "Directory" command for on-line files.

The basic form of the command is:

@INTERROGATE filename

where "filename" may be either a single file or a file group specification (containing asterisks), and the default is \*.\* in your connected directory. This command generates a list of all archived files whose names match the given file specification, along with the numbers of the tapes on which they are archived.

Additionally, if only a single file was specified, the system then prints out:

Do you want it retrieved? (Y or N)

to which you should give the appropriate response. If you answer "Y", a request will be entered into the system for retrieval of that file. The next time the archive system is run by system personnel, the file will be read from tape and restored to the disk, and you will be notified via SNDMSG when this is completed.

"Interrogate" also has a number of subcommands, invoked by terminating the filename with a comma (no carriage return). The subcommands are similar to the "Directory" subcommands and are mostly self-explanatory:

```
@@ALL FILES
@@DELETED FILES ONLY
@@UNDELETED FILES ONLY
@@VERBOSE
@@EVERYTHING
@@DATES
@@TIMES AND DATES
@@PROTECTION
@@BEFORE DATE
@@SINCE DATE
@@DOUBLE SPACE
@@NO HEADING
@@REVERSE ORDER
@@OUTPUT TO FILE
@@LPT
```

#### *Other Commands*

The commands "Archive Delete" and "Archive Undelete" allow manipulation of the information about previously archived files in a manner similar to the way in which "Delete" and "Undelete" work on normal, on-line files.

@ARCHIVE DELETE multiple file designator

causes the named file(s) to have their entries deleted from your *archive* directory. The effect is immediate; no operator action is required, and no information on the archive tapes is actually destroyed. This command is not to be confused with the "Delete" subcommand of "Archive File".

@ARCHIVE UNDELETE multiple file designator

restores the information about the named archived file(s) by undoing the work of a previous "Archive Delete". As with "Archive Delete", the effect is immediate, with no operation action required.

### *Access to Archived Files*

It is important to understand that archived files are always considered to belong to the directories from which they were archived, no matter who requests their archiving or retrieval. While it is necessary to have write access to a file in order to request that it be archived, it is possible to request retrieval of an archived file even if you do not have write access to that file or to the directory from which it was archived. The file is always restored to the directory from which it was archived, and all its original attributes are restored with it, including protection. Hence if you had only read access to the file before it was archived, you will have only read access after it is retrieved.

### *Files-Only Directories*

When a file is archived from an ordinary (Login) directory, that user is notified directly via SNDMSG, as explained above. However, when a file is archived from a files-only directory, the user to be notified is determined by an entry in the file <SYSTEM>ARCHIVE-FILES-ONLY.TXT. We have designated an "owner" for each files-only directory that presently exists on Maxc, i.e. somebody whom we believe to be principally responsible for it. If there are any errors in this list, we will be glad to rectify them.

Note that this list is used only for notification of *archiving*. When a file is *retrieved*, the user who requested the retrieval is the one notified.

### *Future Plans*

The "Interrogate" command is not really an Exec command but is rather implemented by an "ephemeral" subsystem. This explains any peculiarities you may notice in command typein and editing conventions. We will attempt to smooth out such problems in the near future.

Present plans call for the interim facility (24 hour turn-around, and magnetic tape) to be replaced by a large (~billion byte) archival file system accessible over the Ethernet by both Maxc and Altos. Retrieval time from this system will be on the order of milliseconds and so it can be expected to alleviate many of the problems inherent in the present interim system.