Deletion Protection System
(GCE Undelete Facility)

Overview:

How many times have you had requests of the form "I just accidentally deleted ****. What can I do?"?

One of the key problems in most computer sites is that people accidentally delete files, which then must be retrieved from backups, often with serious lost time and information repercussions. In MSDOS systems, where there is one user only, it is possible when this happens to pick up the pieces of erroneously deleted files and to thus "Undelete" them. In multi-user systems like VMS or Unix, disk space is shared and quickly reused, so that the pieces of these files are almost never intact. A deleted file is thus generally lost irretrievably once it is deleted. To avoid this, one must somehow change what the deletion operation means to the system, since these deletions may come as a result of programs other than DCL issuing the commands. Also, one must take care that deleting files doesn't result in allowing anyone on the system to go "dumpster diving" in them.

A related problem occurs in some sites needing permanent records (e.g., essentially all government sites, where regulations require that files like mail be kept for historical archives). In these sites, the necessary cleanup to allow any disk space to be freed is cumbersome due to these retention needs. A system which can allow deletion to occur normally but route deleted files first to a backup can vastly reduce the technical and administrative burdens of space management without visible impact on the users.

The Undeletion Facility is a system to deal with these problems. It is built into the VMS file system and temporarily preserves files that are deleted so that a RELIABLE "Undelete" is possible. In addition it is able to route files being permanently deleted to backing store before this is done if the site needs this. The system is intelligent enough to include or exempt various classes of files and to impose minimal overhead in providing its protections. Because it becomes part of the file system, it can preserve the files and preserve their protection, changing the meaning of deletion without affecting system operation. Because it can respond to low-space conditions, it can also deal sensibly with situations where deletion is needed to gain space, and can operate in a totally automated way. Thus system management duties no longer need be tied up by retrieving just-deleted files that turn out to have been important, and record keeping can be made to conform to official requirements if such exist.

User Manual:

The Undeletion Facility allows undeletion of VMS files that were accidentally deleted, within a period of time, and without generating security holes in one's file system.

It operates by interception code which gains control at any delete request on a controlled volume and which activates a delete daemon to perform the actual "deletion" operations.

There are 3 major modes the system has: 1. *Rename* the files deleted somewhere 2. *Copy* the files deleted somewhere 3. *Run a site command file* to copy, rename, compress, etc. the files.

The daemon can do 1 and 2 directly, and will check a logical name to see what filenames (or parts thereof) may be deleted immediately (e.g., *.OBJ*) so that saving easily-recreated files can be avoided. The site command file option (using a kept subprocess to avoid constant process creation overhead) allows ultimate flexibility where the customer so desires. A different policy can be selected per disk if desired.

The system is capable of detecting out-of-space conditions and running a site command script at that time if this is needed. This procedure is just spawned from the daemon if this is enabled. A mode control allows a site to decide whether deletion where no room exists for the new file or not.

A cleanup process runs every hour or so and deletes older files.

An *undelete* command exists which can restore a file (or any wildcarded filenames) to their original locations (only) so long as the file has not been removed entirely. There is also provision for allowing an EXPUNGE verb (identical to the DELETE verb exactly except that a different copy of the image is used) which will perform immediate deletions where this is needed, not subject to undeletion.

Finally, a hook in the cleanup daemon will allow deleted files to be recorded "somewhere". Where a HSM package is in use, this may consist of moving files to a nearline site and leaving the headers around tagged for automatic retrieval.

File ownership is left intact by the delete/undelete cycle where possible. (Note that a system defined command file may fail to do this, but that is the site's responsibility.)

The facility offers also a space monitor such that a site script can be run if any file extend or create will exceed disk free space. This can be used to run the cleanup operation if desired, so that deleted files remain "in the trashcan" until space is needed, and the trashcan is then automatically emptied of anything older than a minimal period.

Installation

Deletion Protection System (DPS) is installed with VMSINSTAL. You must select a directory into which DPS files will be placed. This can be anywhere, but it is good practice not to mix DEC and non-DEC files, so it is suggested that sys$system not be used. This directory must be created before installation. As a protection against inadvertent use of a mis-spelled directory this is required.

Once the software is installed, you must run the JT_SETUP_DEL script which is furnished. This is done via the command

$ @GCY$SYS:JT_SETUP_DEL

and presents you with a menu which looks like this:

```
              DPS SETUP 12:03:45

   --> *Set area to hold DPS database files
        Set start intercept driver unit number (now 0)
       *Set area for DPS executable images
        Done this menu, process disk selection
        Remove a disk from an existing DPS configuration
        Set images which are exempt from DPS (e.g. defraggers)
        Set area for scratch
        Set area to save deleted files
        Set mode of deletion handling (current value: 0)
        Enable volume space monitoring
        Quit, do nothing
```

If you have rebooted since running VMSINSTAL, you will have to reselect the area to hold DPS files. If not (as in the screen above), the selection of this area has already been done (thus it is tagged with *) and need not be repeated. The area selected at VMSINSTAL should be used for DPS images, since they will be there already. The area used to hold DPS databases is initially this same area, but may be reassigned.

The "Set start intercept driver unit number" is used where more than one deletion daemon will exist in the system. You run this script once per daemon, and each disk you select to assign to that daemon uses one intercept driver unit. Therefore you must set the start intercept unit to something nonzero and greater than all units currently set to be used should you select multiple daemons. If you configure, for example, three disks for the first delete daemon, that will use intercept units 0, 1, and 2. Therefore start the next time at unit 3. This sort of counting is simple and therefore is up to the user. (If a script tries to start using an intercept unit that is already in use, the attempt to use it the second time will be ignored.)

DPS stores files in a "wastebasket" area which can be anywhere on your system. It need not be accessible to anyone without privilege (since the undelete image and DPS itself have such) but must be able to hold deleted files during the period between deletion and actual removal. (This area can be kept on a virtual disk if desired to limit its impact on the rest of the system.)

(In the case of rename mode, this name is just a directory specification which must be there on any disks controlled).

This area gets files being deleted and saves them so that they can be deleted later. (Note that to do immediate deletion, a process may define logical GCY$DELNOW to be "YES" in its process table and the deletions will be permitted.)

The menu item

```
        Set area to save deleted files
```

allows selection of this area. If you are going to use rename mode be sure it does NOT contain a device: specification, just a directory which had better be there. Otherwise it should contain a device specification as well as a directory.

The

```
        Set area for scratch
```

is needed if you wish to define a scratch area to be used for possible file access over net or for compression. It will define the logical area GCY$SCRATCH. If files are saved via command file, the command file can be set up to compress the deleted files and later decompress them; the FILDEL.COM and FILUNDEL.COM scripts will then need to be edited to do this. The scratch area also is needed for some files created when making space on a volume by the default scripts.

Space Monitoring

With any "wastebasket" facility, a disk may fill up, and it will be desired to be able to handle this automatically in some fashion. The DPS system includes an optional space monitor which runs the command file GCY$SYS:MAKSPC.COM to make room on a disk whenever a file create or extend needs more space than exists on the volume. This command file may simply run the deleted-file-purge command to expunge all files deleted less than, say, 5 minutes ago on the disk in question, or may perform other maintenance type operations. Some example files are furnished with the system. By using such a facility, the undelete system becomes a "set it and forget it" utility in which files are kept around as long as possible and deleted when the room is needed for other things. The menu item allows you to select this functionality or disable it. When selected, the item is flagged with * on screen.

Mode Setting

There are several modes for handling file deletion. When you select the item

```
        Set mode of deletion handling (current value: 0)
```

you enter a new menu which allows you to select how files being deleted are to be treated. This menu looks like this:

```
             DPS MODES SETUP 11:57:40

  --> * Run .COM file FILDEL.COM to process deletions
        Rename deleted files on volume to wastebasket
        Copy deleted files to wastebasket area
        Don't delete any files after processing
        Delete file if no room for saving (else do not delete)
        Run GCY$CM:DELBAK.COM before wastebasket purges
        Set files to ignore (allowing normal deletion of them)
        Done this menu
```

```
Quit this menu, leave major modes alone
```

Major Modes

The first three selections are mutually exclusive. You select, when a file is deleted, to either:

1. Run a .COM file (in a process that stays around so this doesn't need the overhead of a new process startup) (FILDEL.COM) to move a copy of the file somewhere before deletion, or
2. Rename the file being deleted somewhere on the disk and fake successful deletion, or
3. Copy the file somewhere, within the daemon and using callable Convert.

Note that option 3 might lose some access control information.

The undelete command uses the same methods, using FILUNDEL.COM to undo the delete if method 1 above was chosen (all .COM files are in area GCY$SYS).

The basic tradeoffs are space and speed. The .COM file method is far the most flexible, and methods 1 and 3 are not too different in speed, while the rename method is fastest, but does not clear space from a disk. Not however that DPS also can respond to a low-space situation by running a command file GCY$SYS:MAKSPC.COM, so that where space is low, the wastebasket cleanup utility can be run.

OUT OF SPACE Action

The fourth option,

```
Don't delete any files after processing
```

is designed for use with mode 1 (.COM file) processing. It arranges that the files are not deleted, but success is faked, when the .COM file exits. (Normally the deletion proceeds at that point.)

This can be used to allow a .COM file to create or append to a "work list" file of files to be deleted, and let another detached process perform the actual deletion processing of these files separately. Where this is done, typically files would be renamed to a scratch area, then copied somewhere and deleted from the scratch area as the separate process got to them. The delete operation by a user would then complete faster, though the file would not actually disappear (nor the file ID become invalid) for some period afterwards.

The item "Delete file if no room for saving (else do not delete)" lets you select what to do where the wastebasket area is too small to hold a file. Basically the original file can either be deleted or not deleted; this allows control over this action. Default is to delete the file (as would have happened with no DPS). Note that the FILDEL script supplied will perform space checks and run the MAKSPC script itself, so that if that method is used, the issue of deleting a file or not will usually be moot. It is also generally moot if separate daemons are used for the save area disk from the other disks, or if the rename method is used.

File Preservation

The option that asks "Run GCY$CM:DELBAK.COM before wastebasket purges" allows you to create a command file which is run before any purges are done. If there is a desire to keep deleted files in a longterm store somehow, this hook is present to allow this to be done. Construction of the command file is up to the site.

IGNORE-FILES List

There may be some file types that should not be protected from delete. At some sites, *.OBJ, *.LIS, and *.MAP are examples, and others may be found. By specifying parts of filenames to exclude from protection, DPS can be set not to keep such files around for undeletion.

The form of the list varies a bit depending on the mode of deletion processing. Where rename or "callable-convert copy" modes are in use, the strings are used in STR$MATCH_WILD and so must have * as wildcard fields. Thus, to exclude *.OBJ* one would use a string containing just *.OBJ* (which would allow FOO.OBJ;4, MUMBLE.OBJ;32, or BAR.OBJ_SAVE:5 to be deleted at once).

If one is using command procedure handling, the strings are handled with F$LOCATE, so any * characters are stripped before use. To exclude *.OBJ* in that case one could just use the string ".OBJ". Note that more elaborate tests can readily be inserted in the command file as a per site test. Thus the complexity of the tests can be whatever your site needs. Often, though, the built in tests will suffice. Allowing some files to be promptly deleted is a convenience feature designed to speed up operation by not doing useless work.

The ignore-files list may contain many such file specifiers, separated by commas.

Disk Selection

DPS can be separately tailored for each disk on the system or set up for any collection of disks. Once mode and function setup is done, you select the main menu "Done..." item and proceed to pick which disks the daemon will apply to. As many (or few) disks as you like can be controlled. This means that if you want to leave, for example, your system disk uncontrolled by DPS but just use it on a user disk, you can do that.

The disk select menu shows mounted disks first, then all other disks in the system, and can be scrolled to show different ones.

The menu looks something like this:

```
   DPS Configuration   Disk Selection
 Use arrows to move to selection. Use RETURN to select.

 End disk selection
 _ARISIA$DKA700:
 _ARISIA$DKB0:
 _ARISIA$DKB300:
 _ARISIA$DCA0:
 _ARISIA$DCA2:
 _ARISIA$DCA3:
 _ARISIA$DCA4:
 _ARISIA$DCA5:
 _ARISIA$VDB0:
 _ARISIA$VDB1:
 _ARISIA$DKB200:
 _ARISIA$DKB700:
 _ARISIA$DCA1:
 _ARISIA$DCA6:
 _ARISIA$DCA7:
 _ARISIA$FQA0:
 _ARISIA$FQA1:
 _ARISIA$FQA2:
 Type H for help.      Currently on item 1 of 151
```

Move the cursor (indicated by reverse video) to the desired disk and select it. When done selecting disks, select the "End ..." item and the setup script will produce two files,

SYS$MANAGER:DPS_STARTUP.COM

which should be run from SYSTARTUP

and

SYS$MANAGER:DPS_LOGIN.COM

which should be run from SYLOGIN.COM to define UNDELETE verbs and so on as needed.

Use of UnDelete command:

The undelete command is of form

$undelete <filestring>

and accepts wildcards which include any part of the device name or filename of the original file; it puts the file back where it came from, provided the file still exists in the wastebasket area. The repeating batch job that the startup command file generates runs sys$manager:jtpurge.com which purges the wastebasket daily. If the purging should happen oftener or less often, edit this file to adjust the times. With the default, files more than a day old will be gone, but accidental deletions less long ago can be undone.

EXPUNGE command

There are times when you will want to delete some files immediately and irretrievably. The EXPUNGE command is provided for this. It simply does an ASSIGN/USER "YES" GCY$DELNOW and then runs the DELETE utility. The presence of this logical allows DPS to know that this particular deletion is to be done without special action. If, for example, you create a very large scratch file sometime, you may want to delete it without having it fill the wastebasket area. The EXPUNGE command is designed to make this simple, but explicit.