Using packets as a safe way of obtaining a valid FileSysStartupMsg pointer

Ralph Babel <rbabel@babylon.pfm-mainz.de>*

24 February 1994

Many programs — among them Format, DiskCopy, DiskDoctor, DiskSalv, and DiskEd — need to obtain information about a disk partition's geometry data and the name, unit number, and flags required for accessing the underlying Exec device driver the disk is associated with. Even though dol_Startup may point to a FileSysStartupMsg structure, it is really a handler-private field that is used to hold integers and string pointers too, and interpreting it correctly without causing Enforcer hits or worse is therefore virtually impossible. The following standard provides a *simple* workaround for this problem.

dp_Type: ACTION_GET_DISK_FSSM (4201)
dp_Res1: APTR to a FileSysStartupMsg structure

Only filesystems with an underlying sector-oriented direct-access Exec device driver shall implement this packet. It is an error for any handler that does not operate on a trackdisk.device-like driver (e.g. EHandlers, dynamic RAM disks, and network filesystems) to respond with a nonzero dp_Res1 value to this packet. As applications may nevertheless attempt to interpret dol_Startup upon a 0/209 result in order to support old block-structured filesystems that predate this standard, new handlers that cannot return a valid FileSysStartupMsg pointer, but wish to indicate that there is no point in accessing their dol_Startup field in spite of dp_Res1 being zero, should return ERROR_OBJECT_WRONG_TYPE (212) in dp_Res2.

The FileSysStartupMsg structure so indicated is read-only, i.e. it must not be altered by the caller. All of its fields must remain constant and valid until invalidated explicitly by the associated ACTION_FREE_DISK_FSSM packet. fssm_Unit must not be a string pointer (as is possible with the 2.1 Mount command), and fssm_Device must be a constant NUL-terminated BCPL string.

*This standard is supported by Steve Beats, Martin J. Laubach <mjl@alison.co.at>, Bernhard Möllemann <zza@rz.uni-karlsruhe.de>, Heinz Wrobel <heinz@hwg.muc.de>, Franz-Josef Reichert <fjrei@kbsaar.saar.de>, Christian Balzer <cb@brewhq.swb.de>, Edgar Törnig <froese%donald@babylon.pfm-mainz.de>, Christopher A. Wichura <caw@miroc.chi.il.us>, Christopher A. Wolf <chris@alchemy.geo.cornell.edu>, and Neil Robinson <100016.2775@compuserve.com>. The packet numbers have been approved by Randell Jesup <jesup@cbmvax.cbm.commodore.com>.

fssm_Environ must be a valid BPTR to a DosEnvec structure and must not be zero. It must be safe to perform the following call with “io” being a pointer to an IOStdReq-sized I/O-request structure that has been initialized properly; the call itself, however, may well fail with a nonzero io_Error value:

```
OpenDevice(1 + (char *)BADDR(fssm->fssm_Device),
           fssm->fssm_Unit, io, fssm->fssm_Flags);
```

de_TableSize, de_BufMemType, de_MaxTransfer, de_Mask, de_DosType, and de_BootBlocks are constants. de_TableSize must be at least 10. The defaults for de_BufMemType, de_MaxTransfer, de_Mask, de_DosType, and de_BootBlocks are 3 (MEMF_CHIP | MEMF_PUBLIC), 0x7FFF FFFF, 0xFFFF FFFE, 0x444F 5300, and 2, respectively. de_DosType is the default DOS type to be used as Format()’s D3 parameter; it is not necessarily an indicator of the actual partition format, and its interpretation depends entirely on the filesystem.

The ten fields that follow de_TableSize directly describe a medium’s geometry and may be updated on the fly by the filesystem upon a change of medium so as to reflect the current disk’s geometry (e.g. with Commodore’s 150-rpm high-density drive and with CD-ROM drives). A consistent set of parameters can be obtained by means of Forbid()-locking. To avoid using an inappropriate geometry after a change of medium (a filesystem may return old or default data while no medium is present), an application should copy the geometry *after* the medium has been inserted, e.g. usually right before it intends to perform actual I/O. (For the CLI commands Format and DiskCopy, for instance, this would be after the user hit the Return key.)

Please note that de_SecOrg, de_PreAlloc, and de_Interleave are best left alone, as they are either undefined or not used in a way that is consistent across all existing filesystems. The contents of all other fields, i.e. de_NumBuffers, de_BootPri, de_Baud, and de_Control, are undefined from the caller’s point of view and must not be accessed.

The descriptor returned by ACTION_GET_DISK_FSSM must be “deallocated” eventually by using the following packet:

dp_Type: ACTION_FREE_DISK_FSSM (4202)
dp_Arg1: APTR to a FileSysStartupMsg structure
dp_Res1: DOSTRUE

The packet must be sent to the same MsgPort as the original ACTION_GET_-DISK_FSSM packet. dp_Arg1 must contain a pointer to a FileSysStartupMsg structure that has been obtained earlier through ACTION_GET_DISK_FSSM; ACTION_FREE_DISK_FSSM must not fail in this case.