

```
-- file DisplayControl.Mesa
-- last edited by Sandman, May 12, 1978 3:42 PM
```

DIRECTORY

```
AltoFileDefs: FROM "altofiledefs" USING [FP],
ControlDefs: FROM "controldefs" USING [SD],
DirectoryDefs: FROM "directorydefs" USING [EnumerateDirectory],
DisplayDefs: FROM "displaydefs" USING [
  BlinkCursor, DisplayOff, DisplayOn, InitDisplay, SetFont, SetTypeScript],
FontDefs: FROM "fontdefs" USING [CreateFont, FontHandle],
ForgotDefs: FROM "forgotdefs",
ImageDefs: FROM "imagedefs" USING [
  AddCleanupProcedure, AddFileRequest, AllReasons, CleanupItem,
  CleanupProcedure, FileRequest],
ProcessDefs: FROM "processdefs" USING [Detach, Yield],
SDDefs: FROM "sddefs" USING [sAddFileRequest],
SegmentDefs: FROM "segmentdefs" USING [
  Append, DefaultBase, DefaultPages, DefaultVersion, DeleteFileSegment,
  FileHandle, FileSegmentHandle, InsertFile, NewFile, NewFileSegment, Read,
  ReleaseFile, Write],
StreamDefs: FROM "streamdefs" USING [
  CloseDiskStream, CreateByteStream, GetIndex, OpenDiskStream, SetIndex,
  StreamError, StreamHandle, StreamIndex, TruncateDiskStream],
StringDefs: FROM "stringdefs" USING [
  EquivalentString, EquivalentSubStrings, SubStringDescriptor],
SystemDisplay: FROM "systemdisplay" USING [typescript];
```

DisplayControl: PROGRAM

```
IMPORTS DirectoryDefs, DisplayDefs, FontDefs, ImageDefs, ProcessDefs, SegmentDefs, StreamDefs, String
**Defs, SystemDisplay
EXPORTS DisplayDefs, ForgotDefs =
```

BEGIN

```
mesafont: short ImageDefs.FileRequest ← [
  link: NIL, file:, access: SegmentDefs.Read,
  body: short[fill:, name: "MesaFont.a1."]];
sysfont: short ImageDefs.FileRequest ← [
  link: @mesafont, file:, access: SegmentDefs.Read,
  body: short[fill:, name: "SysFont.a1."]];
typescript: short ImageDefs.FileRequest ← [
  link: @sysfont, file:,
  access: SegmentDefs.Read+SegmentDefs.Write+SegmentDefs.Append,
  body: short[fill:, name: "Mesa.Typescript."]];

font: FontDefs.FontHandle;
fontseg: SegmentDefs.FileSegmentHandle;
initialize: BOOLEAN ← TRUE;
imaging: BOOLEAN;
```

```
cleanupitem: ImageDefs.CleanupItem ← [
  link:, mask: ImageDefs.AllReasons, proc: Cleanup];
```

```
Cleanup: ImageDefs.CleanupProcedure =
BEGIN OPEN ImageDefs;
file: SegmentDefs.FileHandle;
i: CARDINAL;
si: StreamDefs.StreamIndex;
ts: StreamDefs.StreamHandle;
SELECT why FROM
  Finish, Abort, Save =>
  BEGIN
  IF ~initialize AND SystemDisplay.typescript # NIL THEN
  BEGIN
  StreamDefs.TruncateDiskStream[SystemDisplay.typescript];
  SystemDisplay.typescript ← NIL;
  END;
  IF why = Save AND ~initialize THEN
  BEGIN
  DisplayDefs.DisplayOff[black];
  font.destroy[font];
  SegmentDefs.DeleteFileSegment[fontseg];
  END;
  IF why # Save THEN RETURN;
  imaging ← ControlDefs.SD[SDDefs.sAddFileRequest] # 0;
  mesafont.file ← NIL;
```

```

    IF imaging THEN ImageDefs.AddFileRequest[@mesafont];
    sysfont.file ← NIL;
    IF imaging THEN ImageDefs.AddFileRequest[@sysfont];
    typescript.file ← NIL;
    IF imaging THEN ImageDefs.AddFileRequest[@typescript];
  END;
Restore =>
  BEGIN OPEN SegmentDefs;
  IF (file←mesafont.file) = NIL THEN file ← sysfont.file
  ELSE ReleaseFile[sysfont.file];
  fontseg ← NewFileSegment[file,DefaultBase,DefaultPages,Read];
  font ← FontDefs.CreateFont[fontseg];
  IF initialize THEN
    BEGIN initialize ← FALSE; DisplayDefs.InitDisplay[24,14,20,font] END
  ELSE
    BEGIN DisplayDefs.SetFont[font]; DisplayDefs.DisplayOn[] END;
  IF (file←typescript.file) = NIL THEN
    file ← NewFile[typescript.name, Read+Write+Append, DefaultVersion];
  DisplayDefs.SetTypeScript[
    StreamDefs.CreateByteStream[file, Read+Write+Append]];
  END;
Checkpoint =>
  BEGIN
  IF SystemDisplay.typescript # NIL THEN
    StreamDefs.CloseDiskStream[SystemDisplay.typescript];
  DisplayDefs.DisplayOff[white];
  END;
Continue =>
  BEGIN
  DisplayDefs.DisplayOn[];
  IF (ts←SystemDisplay.typescript) # NIL THEN
    StreamDefs.OpenDiskStream[ts];
  END;
Restart =>
  BEGIN OPEN StreamDefs;
  DisplayDefs.DisplayOn[];
  IF (ts←SystemDisplay.typescript) # NIL THEN
    BEGIN
    OpenDiskStream[ts ! StreamError => IF error = StreamEnd THEN RESUME];
    ts.reset[ts]
    END;
  END;
InLd =>
  IF (ts←SystemDisplay.typescript) # NIL THEN
    StreamDefs.OpenDiskStream[ts];
OutLd =>
  BEGIN OPEN StreamDefs;
  IF (ts←SystemDisplay.typescript) = NIL THEN RETURN;
  si ← GetIndex[ts];
  ts.put[ts,15C];
  FOR i IN [0..9] DO ts.put[ts,'~'] ENDOLOOP;
  SetIndex[ts, si];
  CloseDiskStream[ts];
  END;
  ENDCASE;
RETURN
END;

Cursor: PROCEDURE =
  BEGIN
  timer: POINTER TO INTEGER ← LOOPHOLE[430B];
  blinktime: INTEGER;
  DO -- forever
    IF blinktime-timer↑ ~IN[0..13] THEN
      BEGIN [] ← DisplayDefs.BlinkCursor[]; blinktime ← timer↑+13 END;
    IF CursorProcess = NullProcess THEN EXIT;
    ProcessDefs.Yield[];
  ENDOLOOP;
  RETURN
  END;

NullProcess: PROCESS = LOOPHOLE[0];
CursorProcess: PUBLIC PROCESS ← NullProcess;

StartCursor: PUBLIC PROCEDURE =
  BEGIN

```

```
IF CursorProcess = NullProcess THEN
  ProcessDefs.Detach[CursorProcess ← FORK Cursor[]];
RETURN
END;

StopCursor: PUBLIC PROCEDURE =
BEGIN
  CursorProcess ← NullProcess;
RETURN
END;

-- file requests

ProcessFileRequests: PROCEDURE [rHead: POINTER TO ImageDefs.FileRequest] =
BEGIN OPEN AltoFileDefs;
  checkone: PROCEDURE [fp: POINTER TO FP, dname: STRING] RETURNS [BOOLEAN] =
  BEGIN
    ss: StringDefs.SubStringDescriptor ← [dname,0,dname.length];
    r: POINTER TO ImageDefs.FileRequest;
    prev: POINTER TO ImageDefs.FileRequest ← NIL;
    FOR r ← rHead, r.link UNTIL r = NIL DO
      IF (WITH r SELECT FROM
        long => StringDefs.EquivalentSubStrings[@ss,@name],
        short => StringDefs.EquivalentString[dname,name],
        ENDCASE => FALSE) THEN
        BEGIN
          IF r.file = NIL THEN r.file ← SegmentDefs.InsertFile[fp,r.access]
          ELSE r.file.fp ← fp↑;
          IF prev = NIL THEN rHead ← r.link
          ELSE prev.link ← r.link;
          END
        ELSE prev ← r;
        ENDLIST;
      RETURN[rHead = NIL]
    END;

  DirectoryDefs.EnumerateDirectory[checkone];
END;

ImageDefs.AddCleanupProcedure[@cleanupitem];
Cleanup[Save];
IF imaging THEN STOP ELSE ProcessFileRequests[@typescript];
Cleanup[Restore];
StartCursor[];

END...
```