

```
-- NubControl.Mesa
-- Edited by Sandman on October 6, 1977 4:31 PM
```

### DIRECTORY

```
AltoFileDefs: FROM "altofiledefs",
ControlDefs: FROM "controldefs",
CoreSwapDefs: FROM "coreswapdefs",
FrameDefs: FROM "framedefs",
ImageDefs: FROM "imagedefs",
IODefs: FROM "iodefs",
LoaderDefs: FROM "loaderdefs",
SegmentDefs: FROM "segmentdefs",
StreamDefs: FROM "streamdefs",
StringDefs: FROM "stringdefs",
Rectangles: FROM "rectangles",
Windows: FROM "windows",
TimeDefs: FROM "timedefs";
```

```
NubControl: PROGRAM
```

```
IMPORTS CoreSwapDefs, FrameDefs, ImageDefs, IODefs, LoaderDefs, SegmentDefs, StreamDefs, StringDefs,
**TimeDefs, Rectangles, Windows
=
```

### BEGIN

```
-- System Signals are converted to these to prevent NubCommand
-- from catching user generated signals
Delete: SIGNAL = CODE;          -- nee Rubout
StringTooLong: ERROR = CODE;   -- nee LineOverflow
BadFile: ERROR [badname: STRING] = CODE;          -- nee FileNameError
BadNumber: ERROR = CODE;       -- nee InvalidNumber
```

```
Program: TYPE = PROGRAM;
```

```
CallDebugger: PROCEDURE =
```

```
  BEGIN -- user's entry point to debugger
    state: ControlDefs.StateVector;

    state.stkptr ← 0;
    state.X ← defaultframe;
    CoreSwapDefs.CoreSwap[explicitcall, @state];
    RETURN
  END;
```

```
CommandChar: TYPE = {N,S,M,D,Q};
```

```
CommandString: ARRAY CommandChar OF STRING =
  ["New", "Start", "MakeImage", "Debug", "Quit"];
```

```
Command: PROCEDURE =
```

```
  BEGIN OPEN IODefs;
    c: CHARACTER;
    i: CommandChar;
    f: ControlDefs.GlobalFrameHandle;
    IF comcmRequest.file # NIL THEN NubHack[];
    SELECT c←ReadChar[] FROM
      'N,'n =>
        BEGIN
          WriteString[CommandString[N]];
          f ← LoadNew[getfilename[".bcd"]];
          IF f # ControlDefs.NULLFrame THEN defaultframe ← f;
        END;
      'S,'s =>
        BEGIN
          WriteString[CommandString[S]];
          f ← getgframe[];
          WriteFOI[];
          IF f.pc = ControlDefs.WordPC[0] THEN START LOOPHOLE[f, Program] ELSE RESTART F;
        END;
      'D,'d =>
        BEGIN
          WriteString[CommandString[D]];
          confirm[];
          CallDebugger[];
        END;
      'Q,'q =>
```

```

    BEGIN
    WriteString[CommandString[Q]];
    confirm[];
    ImageDefs.StopMesa[];
    END;

'M,'m =>
    BEGIN
    WriteString[CommandString[M]];
    ImageDefs.MakeImage[getfilename[".image"], FALSE];
    END;

CR,SP => NULL;

ENDCASE =>
    BEGIN
    WriteChar[c];
    WriteString[" Commands are:"];
    FOR i IN CommandChar DO
        WriteChar[' ']; WriteString[CommandString[i]]; ENDLOOP;
    END;
RETURN
END;

LoadNew: PROCEDURE [name: STRING] RETURNS [g: ControlDefs.GlobalFrameHandle] =
    BEGIN
    g ← FrameDefs.New[name | BadFile, LoaderDefs.VersionMismatch, UNWIND => NULL;
    ANY => ERROR BadFile[name]];
    IODefs.WriteString[" -- "];
    IODefs.WriteOctal[g];
    RETURN
    END;

defaultframe: ControlDefs.GlobalFrameHandle ← REGISTER[ControlDefs.Greg];

getgframe: PROCEDURE RETURNS [f: ControlDefs.GlobalFrameHandle] =
    BEGIN OPEN IODefs;
    WriteString[" Global frame: "];
    f ← ReadNumber[defaultframe,8
        ! LineOverflow => ERROR StringTooLong;
        Rubout => SIGNAL Delete;
        StringDefs.InvalidNumber => ERROR BadNumber];
    IF (f.accesslink # f) OR (FrameDefs.Class[f] # global) THEN
        BEGIN WriteString[" not a global frame!"]; SIGNAL CoreSwapDefs.CAbort END;
    defaultframe ← f;
    RETURN
    END;

idstring: STRING ← [40];

getfilename: PROCEDURE [defaulttextension: STRING] RETURNS [STRING] =
    BEGIN OPEN IODefs;
    WriteString[" Filename: "];
    ReadID[idstring
        ! Rubout => SIGNAL Delete;
        LineOverflow => ERROR StringTooLong];
    defaulttext[idstring, defaulttextension];
    RFTURN[idstring]
    END;

defaulttext: PROCEDURE [idstring, ext: STRING] =
    BEGIN
    i: CARDINAL;
    FOR i IN [0..idstring.length] DO
        IF idstring[i] = ' ' THEN EXIT;
        RFPFAT FINISHED =>
            StringDefs.AppendString[idstring, ext
                ! StringDefs.StringBoundsfault => ERROR StringTooLong];
        ENDLOOP;
    END;

confirm: PROCEDURE =
    BEGIN OPEN IODefs;
    WriteString[" [confirm] "];

```

```

DO
  SELECT ReadChar[] FROM
    CR => EXIT;
    DEL => SIGNAL Delete;
    ENDCASE => WriteChar['?'];
  ENDOLOOP;
WriteChar[CR];
RETURN
END;

WriteEOL: PROCEDURE =
  BEGIN OPEN IOdefs;
  IF ~NewLine[] THEN WriteChar[CR];
  RETURN
  END;

comcmRequest: short ImageDefs.FileRequest + [
  body: short[fill:, name: "Com.Cm."],
  file: NIL,
  access: SegmentDefs.Read,
  link: ];

NubHack: PROCEDURE =
  BEGIN OPEN IOdefs, StreamDefs;
  f: ControlDefs.GlobalFrameHandle;
  com: StreamHandle;
  name: STRING + [40];
  fa: AltoFileDefs.FA;
  GetToken: PROCEDURE [token: STRING] =
    BEGIN
      c: CHARACTER;
      token.length + 0;
      WHILE ~com.eof[com] DO
        SELECT (c + com.get[com]) FROM
          SP, CR => IF token.length # 0 THEN RETURN;
          ENDCASE => StringDefs.AppendChar[token, c];
        ENDOLOOP;
      RETURN
      END;
  Cleanup: PROCEDURE =
    BEGIN
      IF comcmRequest.file # NIL THEN
        BEGIN
          SegmentDefs.UnlockFile[comcmRequest.file];
          comcmRequest.file + NIL
        END;
      END;
      SegmentDefs.LockFile[comcmRequest.file];
      com + CreateByteStream[comcmRequest.file, Read];
      GetToken[name]; -- Skip Image Name
      GetToken[name];
      WHILE name.length # 0 DO
        ENABLE UNWIND => Cleanup[! ANY => NULL];
        WriteString[name];
        defaulttext[name, ".bcd"];
        StreamDefs.GetFA[com, @fa];
        com.destroy[com]; com + NIL;
        f + loadNew[name];
        WriteEOL[];
        IF f # ControlDefs.NULLFrame THEN START LOOPHOLE[(defaultframe + f), Program];
        WriteChar['>'];
        IF comcmRequest.file = NIL THEN RETURN;
        com + CreateByteStream[comcmRequest.file, Read];
        StreamDefs.JumpToFA[com, @fa ! StreamDefs.StreamError, SegmentDefs.InvalidFP => EXIT];
        GetToken[name];
        ENDOOP;
        com.destroy[com];
        Cleanup[];
      END;
  time: STRING + [18];

-- Main body

[] + StreamDefs.GetDefaultKey[]: -- Start keyboard process

```

```

ImageDefs.AddFileRequest[@comcmRequest];
START Rectangles[40, 30];
START Windows["Mesa.Typescript."];

STOP;

RESTART Rectangles;
RESTART Windows;
START IODefs.StreamIO[NIL, NIL];
[] ← LoaderDefs.Load8cd[NIL | ANY => CONTINUE];

TimeDefs.AppendDayTime[time, TimeDefs.UnpackDT[ImageDefs.ImageVersion[.time]];
time.length ← time.length - 3;
IODefs.WriteString["Alto/Mesa 3.0 "]; IODefs.WriteString[time];

DO OPEN IODefs; -- forever
  WriteEOL[]; WriteChar['>'];
  Command[
    ! Delete => BEGIN WriteString[" XXX"]; CONTINUE END;
    CoreSwapDefs.CAbort => CONTINUE;
    BadFile =>
      BEGIN
        WriteEOL[];
        WriteString["!File: "];
        WriteString[badname];
        CONTINUE
      END;
    BadNumber =>
      BEGIN WriteEOL[]; WriteString["!Number"]; CONTINUE END;
    StringTooLong =>
      BEGIN WriteEOL[]; WriteString["!String too long"]; CONTINUE END;
    CoreSwapDefs.CantSwap =>
      BEGIN
        WriteEOL[];
        WriteString["External Debugger not installed, type DEL to Abort "];
        UNTIL ReadChar[]=DEL DO WriteChar['?'] ENDLOOP;
        WriteString["...Aborted..."];
        CONTINUE
      END;
    LoaderDefs.VersionMismatch--[name: STRING] --=>
      BEGIN
        WriteEOL[];
        WriteString["!File: "];
        WriteString[name];
        WriteString[" referenced in different versions."];
        RESUME
      END;
    UNWIND =>CONTINUE
  ];
ENDLOOP;

END...

```