```
-- RectanglesB.Mesa  Edited by Sandman on May 12, 1978  2:51 PM

DIRECTORY
  AltoDefs: FROM "altodefs" USING [PageSize],
  ImageDefs: FROM "imagedefs" USING [
    AddCleanupProcedure, AddFileRequest, AllReasons, CleanupItem, CleanupMask,
    CleanupProcedure, FileRequest],
  InlineDefs: FROM "inlinedefs" USING [BITAND],
  IODefs: FROM "iodefs" USING [CR, DEL, SP],
  MiscDefs: FROM "miscdefs" USING [Zero],
  RectangleDefs: FROM "rectangledefs" USING [
    backgtype, BitmapErrorCode, BitmapObject, blanklines, BMHandle, BMptr,
    DCB, DCBchainHead, DCBnil, DCBptr, FAptr, Fptr, leftmargin, Rectangle,
    ROptions, Rptr, xCoord, yCoord],
  RectanglesA: FROM "rectanglesa" USING [
    ComputeCharWidth, defaultcharwidths, defaultlineheight, defaultmapdata,
    defaultpfont, FixupRectangle],
  SDDefs: FROM "sddefs" USING [sAddFileRequest, SD],
  SegmentDefs: FROM "segmentdefs" USING [
    DefaultBase, DefaultPages, DefaultVersion, DeleteFileSegment, FileError,
    FileNameError, FileSegmentAddress, FileSegmentHandle, NewFile,
    NewFileSegment, OldFileOnly, Read, ReleaseFile, SwapIn, Unlock],
  StreamDefs: FROM "streamdefs" USING [
    DisplayHandle, GetDisplayStreamList, SetDisplayLine],
  SystemDefs: FROM "systemdefs" USING [
    AllocateHeapNode, AllocateResidentPages, FreeHeapNode, FreePages];

DEFINITIONS FROM RectangleDefs;

RectanglesB: PROGRAM [pagesformap, mapwordsperline: CARDINAL]
  IMPORTS ImageDefs, MiscDefs, RectanglesA, SystemDefs, SegmentDefs,
    StreamDefs
  EXPORTS RectangleDefs SHARES RectangleDefs, RectanglesA =
  BEGIN OPEN RectanglesA;

-- CHARACTER constants
  CR: CHARACTER = IODefs.CR;
  Space: CHARACTER = IODefs.SP;
  DEL: CHARACTER = IODefs.DEL;

-- GLOBAL PUBLIC Data (all PUBLIC for initialization guy ??)

  savedfirstDCB: DCBptr ← NIL;
  tempDCB: UNSPECIFIED;
  bitmaps: PUBLIC BMHandle ← NIL;
  defaultfont: PUBLIC Fptr ← NIL;       -- points to start of font
  defaultfontsegment: FileSegmentHandle ← NIL;
  SevenBitCharacter: TYPE = CHARACTER[0C..177C];
  FileSegmentHandle: TYPE = SegmentDefs.FileSegmentHandle;

-- GLOBAL Data

  wordsinpage: CARDINAL = AltoDefs.PageSize;

-- Bitmap Rectangle Routines

  CreateRectangle: PUBLIC PROCEDURE [
    bitmap: BMHandle, x0, width: xCoord, y0, height: yCoord] RETURNS[Rptr] =
    BEGIN
    rectangle: Rptr;
    rectangle ← SystemDefs.AllocateHeapNode[SIZE[Rectangle]];
    rectangle↑ ← Rectangle[NIL, FALSE,, bitmap, x0, width, 0, y0, height, 0];
    rectangle.options ← ROptions[FALSE,FALSE];
    rectangle.link ← bitmap.rectangles;
    bitmap.rectangles ← rectangle;
    FixupRectangle[rectangle];
    RETURN[rectangle];
    END;

  DestroyRectangle: PUBLIC PROCEDURE [rectangle: Rptr] =
    BEGIN
    prev: Rptr;
    bitmap: BMHandle ← rectangle.bitmap;
    IF bitmap.rectangles = rectangle THEN
      bitmap.rectangles ← rectangle.link
    ELSE
```

```
      BEGIN
      prev ← bitmap.rectangles;
      UNTIL rectangle = prev.link DO
        IF prev = NIL THEN ERROR;
        prev ← prev.link;
        ENDLOOP;
      prev.link ← rectangle.link;
      END;
    SystemDefs.FreeHeapNode[rectangle];
    END;

-- Bitmap Routines

  GetDefaultBitmap: PUBLIC PROCEDURE RETURNS [BMHandle] =
    BEGIN
    RETURN[defaultmapdata];
    END;

  EVEN: PROCEDURE[v: UNSPECIFIED] RETURNS [UNSPECIFIED] =
    BEGIN
    -- make an even value by rounding v up
    RETURN[v+InlineDefs.BITAND[v, 1]];
    END;

  CreateBitmap: PUBLIC PROCEDURE [pagesformap, wordsperline: CARDINAL] RETURNS[BMHandle] =
    BEGIN
    mapdata: BMHandle;
    dcb: DCBptr;
    mapdata ← SystemDefs.AllocateHeapNode[SIZE[BitmapObject]];
    mapdata↑ ←
      BitmapObject[NIL, NIL, NIL, NIL, 0, 0, 0, 0, 0, 0, 0,  high, white];
    -- NOTE: lots'a funnies because DCB's must be even
    -- and someone has to deallocate him eventually!!)
    dcb ← EVEN[mapdata.dcb ← SystemDefs.AllocateHeapNode[SIZE[DCB]+1]];
    dcb.next ← DCBnil;
    ReallocateBitmap[mapdata, pagesformap, wordsperline];
    mapdata.link ← bitmaps;
    bitmaps ← mapdata;
    RETURN[mapdata];
    END;

  DestroyBitmap: PUBLIC PROCEDURE [mapdata: BMHandle] RETURNS [POINTER] =
    BEGIN
    addr: POINTER;
    prev: BMHandle;
    IF mapdata.rectangles # NIL THEN
      SIGNAL BitmapError[mapdata, BitmapOperation];
    IF mapdata.addr # NIL THEN SystemDefs.FreePages[mapdata.addr];
    IF mapdata.dcb # NIL THEN SystemDefs.FreeHeapNode[mapdata.dcb];
    addr ← mapdata.addr;
    IF mapdata = bitmaps THEN bitmaps ← mapdata.link
    ELSE
      BEGIN
      prev ← bitmaps;
      UNTIL mapdata = prev.link DO
        IF prev = NIL THEN ERROR;
        prev ← prev.link;
        ENDLOOP;
      prev.link ← mapdata.link;
      END;
    SystemDefs.FreeHeapNode[mapdata];
    RETURN[addr];
    END;

  UpdateBitmap: PUBLIC PROCEDURE [mapdata: BMHandle] RETURNS [DCBptr] =
    BEGIN
    dcb: DCBptr = EVEN[mapdata.dcb];
    dcb.bitmap ← mapdata.addr;
    dcb.height ← mapdata.height/2;
    dcb.width ← mapdata.wordsperline;
    dcb.indenting  ← mapdata.indenting;
    dcb.resolution  ← mapdata.resolution;
    dcb.background  ← mapdata.background;
    RETURN[dcb];
    END;
```

```
-- MAIN BODY CODE

  -- make file request on second START
  mesapreopen: short ImageDefs.FileRequest ← ImageDefs.FileRequest [
    file: NIL, access: SegmentDefs.Read, link:,
    body: short[fill:, name: "MesaFont.Al."]];
  syspreopen: short ImageDefs.FileRequest ← ImageDefs.FileRequest [
    file: NIL, access: SegmentDefs.Read, link:,
    body: short[fill:, name: "SysFont.Al."]];

  IF SDDefs.SD[SDDefs.sAddFileRequest] # 0 THEN
    BEGIN
    ImageDefs.AddFileRequest[@mesapreopen];
    ImageDefs.AddFileRequest[@syspreopen];
    STOP;
    END;

  BEGIN OPEN SegmentDefs;
  IF mesapreopen.file = NIL THEN
    mesapreopen.file ← NewFile[mesapreopen.name, Read, DefaultVersion
      ! FileNameError, FileError => CONTINUE];
  IF syspreopen.file = NIL THEN
    syspreopen.file ← NewFile[syspreopen.name, Read, DefaultVersion
      ! FileNameError, FileError => CONTINUE];
  END;

  -- now really do it
  InitFontFile[];
  initbitmap[pagesformap, mapwordsperline];
  BEGIN OPEN ImageDefs;
  cleanup ← CleanupItem[link:, proc: CleanRecs,
    mask: AllReasons-CleanupMask[InLd]-CleanupMask[OutLd]];
  AddCleanupProcedure[@cleanup]
  END;


  END. of Rectangles
```