```
-- ImageDefs.mesa  Modified by: Sandman, March 31, 1978  2:24 PM

DIRECTORY
AltoDefs: FROM "altodefs",
AltoFileDefs: FROM "altofiledefs",
BcdDefs: FROM "bcddefs",
ControlDefs: FROM "controldefs",
DiskDefs: FROM "diskdefs",
Mopcodes: FROM "mopcodes",
NovaOps: FROM "novaops",
SDDefs: FROM "sddefs",
SegmentDefs: FROM "segmentdefs",
StringDefs: FROM "stringdefs";

ImageDefs: DEFINITIONS SHARES DiskDefs =
PRIVATE BEGIN

FirstImageDataPage: AltoDefs.PageNumber = 2;
HeaderPages: CARDINAL = 1;

MapItem: TYPE = MACHINE DEPENDENT RECORD [
  page: [0..255],
  count: [0..127],
  body: SELECT tag:* FROM
    normal => NULL,
    change => [
      da: DiskDefs.DA,
      base: CARDINAL],
    ENDCASE];

VersionID: CARDINAL = 03168;  -- must match in Mesa.bcpl

ImagePrefix:
  TYPE = MACHINE DEPENDENT RECORD [
    versionident: CARDINAL,
    version, creator: BcdDefs.VersionStamp,
    options: WORD,
    leaderDA: AltoFileDefs.vDA,
    state: ControlDefs.StateVector,
    loadStateBase, initialLoadStateBase: CARDINAL,
    type: ImageType,
    fill: [0..37B],
    loadStatePages: AltoDefs.PageCount];

  ImageType: TYPE = {bootmesa, makeimage, checkfile, other};

  ImageHeader: TYPE = MACHINE DEPENDENT RECORD [
    prefix: ImagePrefix,
    map: ARRAY [0..0) OF MapItem];

  MapSpace: CARDINAL = AltoDefs.PageSize-SIZE[ImagePrefix]-1;

  FileRequest: PUBLIC TYPE = MACHINE DEPENDENT RECORD[
    link: POINTER TO FileRequest,
    file: SegmentDefs.FileHandle,
    access: SegmentDefs.AccessOptions,
    body: SELECT tag:* FROM
      short => [fill: [0..7777B], name: STRING],
      long => [fill: [0..7777B], name: StringDefs.SubStringDescriptor],
      ENDCASE];

  AddFileRequest: PUBLIC PROCEDURE [POINTER TO FileRequest] =
    MACHINE CODE BEGIN Mopcodes.zKFCB, SDDefs.sAddFileRequest END;

  MakeImage: PUBLIC PROCEDURE [name: STRING];
  MakeUnMergedImage: PUBLIC PROCEDURE [name: STRING];
  SwapOutDuringMakeImage: PUBLIC SIGNAL;
  SwapTrapDuringMakeImage: PUBLIC SIGNAL;
  SwapErrorDuringMakeImage: PUBLIC SIGNAL;
  NoRoomInImageMap: PUBLIC SIGNAL;
  InvalidImageName: PUBLIC SIGNAL;

  MakeCheckPoint: PUBLIC PROCEDURE [name: STRING];
  SwapOutDuringMakeCheck: PUBLIC SIGNAL;
  SwapTrapDuringMakeCheck: PUBLIC SIGNAL;
  SwapErrorDuringMakeCheck: PUBLIC SIGNAL;
```

```
  NoRoomInCheckMap: PUBLIC SIGNAL;

  RunImage: PUBLIC PROCEDURE [file: SegmentDefs.FileSegmentHandle];
  InvalidImage: PUBLIC SIGNAL;
  NoRoomForLoader: PUBLIC SIGNAL;

  ImageVersion: PUBLIC PROCEDURE RETURNS [BcdDefs.VersionStamp];

  CleanupItem: PUBLIC TYPE = RECORD [
    link: POINTER TO CleanupItem,
    mask: CARDINAL,
    proc: CleanupProcedure];

  CleanupProcedure: PUBLIC TYPE = PROCEDURE [why: CleanupReason];

  CleanupMask: PUBLIC ARRAY CleanupReason OF CARDINAL =
    [1, 2, 4, 10B, 20B, 40B, 100B, 200B, 400B];
  AllReasons: PUBLIC CARDINAL = 177777B;
  CleanupReason: PUBLIC TYPE = NovaOps.CleanupReason;
  NovaOpcode: TYPE = NovaOps.NovaOpcode;

  AddCleanupProcedure, RemoveCleanupProcedure: PUBLIC PROCEDURE [POINTER TO CleanupItem];
  UserCleanupProc: PUBLIC CleanupProcedure;

  StopMesa: PUBLIC PROCEDURE =
    MACHINE CODE BEGIN Mopcodes.zLI0+LOOPHOLE[NovaOpcode[Finish], CARDINAL]; Mopcodes.zSTOP END;
  AbortMesa: PUBLIC PROCEDURE =
    MACHINE CODE BEGIN Mopcodes.zLI0+LOOPHOLE[NovaOpcode[Abort], CARDINAL]; Mopcodes.zSTOP END;
  PuntMesa: PUBLIC PROCEDURE =
    MACHINE CODE BEGIN Mopcodes.zLI0+LOOPHOLE[NovaOpcode[Punt], CARDINAL]; Mopcodes.zSTOP END;


  END..
```