

-- BcdDefs.Mesa Edited by Satterthwaite on March 21, 1978 11:50 AM

DIRECTORY

```
AltoDefs: FROM "altodefs",
ControlDefs: FROM "controldefs",
TableDefs: FROM "tabledefs",
TimeDefs: FROM "timedefs";
```

BcdDefs: DEFINITIONS =
BEGIN

-- allocation codes for the binder

```
TableSelector: TYPE = TableDefs.TableSelector;
BinderNTables: CARDINAL = 12;
  treetype: TableSelector = 0; -- trees
  htype: TableSelector = 1; -- hash table
  sstype: TableSelector = 2; -- (packed) string table
  ctype: TableSelector = 3; -- config table
  mtype: TableSelector = 4; -- module table
  imptype: TableSelector = 5; -- import table
  exptype: TableSelector = 6; -- export table
  sgtype: TableSelector = 7; -- segment table
  filetype: TableSelector = 8; -- file table
  sttype: TableSelector = 9; -- semantic table
  cxtype: TableSelector = 10; -- context table
  nttype: TableSelector = 11; -- name table
```

TableLimit: CARDINAL = TableDefs.TableLimit;

-- version identification

```
VersionStamp: TYPE = RECORD [
  zapped: BOOLEAN,
  net: [0..128],
  host: AltoDefs.BYTE,
  time: TimeDefs.PackedTime];
```

-- BCD Header

VersionID: CARDINAL = 01268;

```
BCD: TYPE = RECORD [
  versionident: CARDINAL,
  version: VersionStamp,
  creator: VersionStamp,
  bound: VersionStamp,
  nPages: AltoDefs.PageCount,
  nConfigs, nModules: CARDINAL,
  nImports, nExports: CARDINAL,
  definitions: BOOLEAN,
  firstdummy: GFTIndex,
  nDummies: CARDINAL,
  ssOffset: CARDINAL, -- string table
  ssLimit: CARDINAL,
  ctOffset: CARDINAL, -- config table
  ctLimit: CTIndex,
  mtOffset: CARDINAL, -- module table
  mtLimit: MTIndex,
  impOffset: CARDINAL, -- import table
  impLimit: IMPIndex,
  expOffset: CARDINAL, -- export table
  expLimit: EXPIndex,
  sgOffset: CARDINAL, -- segment table
  sgLimit: SGIndex,
  ftOffset: CARDINAL, -- file table
  ftLimit: FTIndex,
  ntOffset: CARDINAL, -- name table
  ntLimit: NTIndex,
  source: NameRecord,
  sourceVersion: VersionStamp];
```

-- Portable Type

Portable: TYPE = {module, interface};

-- Name Table

```
PackedString: TYPE = MACHINE DEPENDENT RECORD [  
  SELECT OVERLAID * FROM  
    string => [string: StringBody],  
    size => [size: PACKED ARRAY [-3..-1) OF AltoDefs.BYTE],  
  ENDCASE];
```

```
NameString: TYPE = POINTER TO PackedString;
```

```
NameRecord: TYPE = RECORD [CARDINAL];
```

```
NullName: NameRecord = [1];
```

```
NTRRecord: TYPE = RECORD [  
  name: NameRecord,  
  item: Nameee];
```

```
Nameee: TYPE = RECORD [  
  SELECT type:* FROM  
    config => [cti: CTIndex],  
    module => [mti: MTIndex],  
    import => [impi: IMPIndex],  
    export => [expi: EXPIndex],  
  ENDCASE];
```

```
NTNull: NTIndex = LAST[NTIndex];
```

```
NTIndex: TYPE = POINTER [0..TableLimit) TO NTRRecord;
```

```
NTHandle: TYPE = POINTER TO NTRRecord;
```

-- Configuration Table

```
CTRecord: TYPE = RECORD [  
  name: NameRecord,  
  namedinstance: BOOLEAN,  
  file: FTIndex,  
  config: CTIndex,  
  control: MTIndex];
```

```
CTNull: CTIndex = LAST[CTIndex];
```

```
CTIndex: TYPE = POINTER [0..TableLimit) TO CTRecord;
```

```
CTHandle: TYPE = POINTER TO CTRecord;
```

-- Module Table

```
LinkLocation: TYPE = {frame, code};
```

```
MTRRecord: TYPE = RECORD [  
  name: NameRecord,  
  namedinstance: BOOLEAN,  
  initial: BOOLEAN,  
  file: FTIndex,  
  links: LinkLocation,  
  config: CTIndex,  
  code: CodeDesc,  
  sseg: SGIndex,  
  framesize: [0..ControlDefs.MaxFrameSize),  
  fsi: [0..ControlDefs.MaxAllocSlot],  
  gfi: GFTIndex,  
  ngfi: [1..ControlDefs.MaxNGfi],  
  frame: FrameFrag];
```

```
CodeDesc: TYPE = RECORD [  
  sgi: SGIndex,  
  packed: BOOLEAN,  
  linkspace: BOOLEAN,  
  offset, length: CARDINAL];
```

```
FrameFrag: TYPE = RECORD [  
  length: CARDINAL,  
  frag: ARRAY [0..0) OF ControlLink];
```

```
MTNull: MTIndex = LAST[MTIndex];
MTIndex: TYPE = POINTER [0..TableLimit) TO MTRRecord;
MTHandle: TYPE = POINTER TO MTRRecord;
```

-- Import Table

```
IMPRecord: TYPE = RECORD [
  name: NameRecord,
  port: Portable,
  namedinstance: BOOLEAN,
  file: FTIndex,
  gfi: GFTIndex,
  ngfi: [1..ControlDefs.MaxNGfi]];
IMPNull: IMPIndex = LAST[IMPIndex];
IMPIndex: TYPE = POINTER [0..TableLimit) TO IMPRecord;
IMPHandle: TYPE = POINTER TO IMPRecord;
```

-- Export Table

```
EXPRecord: TYPE = RECORD [
  name: NameRecord,
  size: CARDINAL,
  port: Portable,
  namedinstance: BOOLEAN,
  file: FTIndex,
  links: ARRAY [0..0) OF ControlLink];
EXPNull: EXPIndex = LAST[EXPIndex];
EXPIndex: TYPE = POINTER [0..TableLimit) TO EXPRecord;
EXPHandle: TYPE = POINTER TO EXPRecord;
```

-- Segment Table

```
SegClass: TYPE = {code, symbols};
SGRecord: TYPE = RECORD [
  class: SegClass,
  file: FTIndex,
  base: CARDINAL,
  pages, extraPages: [0..256)];
SGNull: SGIndex = LAST[SGIndex];
SGIndex: TYPE = POINTER [0..TableLimit) TO SGRecord;
SGHandle: TYPE = POINTER TO SGRecord;
```

-- File Table

```
FTRecord: TYPE = RECORD [
  name: NameRecord,
  version: VersionStamp];
FTNull: FTIndex = LAST[FTIndex];
FTSelf: FTIndex = LAST[FTIndex]-1;
FTIndex: TYPE = POINTER [0..TableLimit) TO FTRecord;
FTHandle: TYPE = POINTER TO FTRecord;
```

-- Control Link

```
GFTIndex: TYPE = ControlDefs.GFTIndex;
GFTNull: GFTIndex = ControlDefs.GFTNull;
EPIndex: TYPE = ControlDefs.EPIndex;
EPLimit: CARDINAL = LAST[EPIndex]+1;
ControlLinkTag: TYPE = ControlDefs.ControlLinkTag;
ControlLink: TYPE = ControlDefs.ControlLink;
NullLink: ControlLink = [procedure[0,0,frame]];
UnboundLink: ControlLink = [procedure[0,0,unbound]];
```

END.