## Inter-Office Memorandum

| | | | |
|---|---|---|---|
| To | Mesa Users | Date | October 27, 1980 |
| From | John Wick | Location | Palo Alto |
| Subject | **Mesa 6.0 System Update** | Organization | SDD/SS/Mesa |

XEROX

Filed on: [Iris]<Mesa>Doc>System60.bravo (and .press)

This memo outlines changes made in the Mesa system interfaces since the last release (Mesa 5.0, April 9, 1979). A number of internal changes made in the system and the microcode are also discussed.

This memo is intended as a quick guide to conversion, not a detailed specification of the changes. Names in square brackets refer to sections of the *Mesa System Documentation* or to other publicly available reference documents (e.g., the *Alto Operating System Reference Manual*).

### External Interfaces

Major changes include integrated support for Alto extended memory and elimination of BasicMesa. The System also now exports versions of the following Development Software interfaces: **Ascii**, **Format**, **Inline**, **Process**, **Runtime**, **Storage**, **String**, **System**, and **Time**. (Note that implementation of these interfaces may not be complete.) Other changes are relatively minor.

*AllocDefs*

Private types and operations have been removed. **AllocInfo**, **MakeDataSegment**, and **MakeSwappedIn** are now defined in **SegmentDefs**, and temporarily duplicated here for compatibility. [**Segment Package**]

*AltoDefs*

**MaxVMPage** has been increased to support up to a million words of memory; **MaxMDSPage** and **PagesPerMDS** have been added. [**Segment Package**]

*AltoDisplay*

**MaxBitsPerLine** has been changed to 608 (it was 606). **Cursor**, **CursorBits**, and **CursorHandle** define the location and format of the cursor. **Coordinate**, **CursorXY**, and **MouseXY** define the location and format of the cursor and mouse coordinates. [**Display Package**]

*AltoFileDefs*

Support for the **DiskShape** and **PartitionName** properties of the directory's leader page has been added. The definition of a file serial number (**SN**) has been changed to isolate the flag bits (**directory**, **random**, and **nolog**) into a separate structure (**SNBits**). [*Alto Operating System Reference Manual*]

### *AltoHardware*

This new interface defines most structures of the Alto hardware, including the processor, display, keyboard, mouse, keyset, printer interface, disk, and Ethernet. [*Alto Hardware Manual*]

### *Ascii*

This new interface defines the ASCII control character codes; for compatibility, these continue to be defined in **IODefs**. [**StreamIO Package**]

### *BasicMesa*

The facilities of **BasicMesa** have been replaced by procedures in the standard system and a command line switch (`/b`) which can be used to destroy the display and keyboard packages (see **DisplayDefs** and **StreamDefs**). **MakeImage** is no longer a standard part of `Mesa.image`, and must be loaded separately. [**Section 3**]

### *BitBltDefs*

The extended memory option now supports use of the normal and alternate bank registers, whose values are supplied in the unused word of the **BBTable** (this option is *not* supported under XMesa 5.0 microcode (version 39)). **AlignedBBTable** (and **BBTableSpace**) can be used to properly align **BITBLT** argument records. [*Alto Hardware Manual*]

### *CharIO*

This new interface provides many of the functions of **IODefs**, but each operation takes a **StreamDefs.StreamHandle** as its first parameter, allowing formatted input and output to any standard stream. [**StreamIO Package**]

### *DirectoryDefs*

This interface has been changed slightly to speed up directory searches (by about a factor of 3). In addition, support for subdirectories was added (see *Alto Operating System Reference Manual*). The following items have changed (note that a NIL **DiskHandle** does *not* imply the system directory):

```
EnumerateEntries: PROCEDURE [
  dir: DiskHandle,
  proc: PROCEDURE [CARDINAL, StreamScan.Handle, DEptr] RETURNS [BOOLEAN],
  inspectFree: POINTER TO READONLY BOOLEAN,
  lengthFilter: CARDINAL _ 0] RETURNS [index: CARDINAL];
```

The procedure **proc** is called for each directory entry; free entries are passed only if **inspectFree^** is TRUE. If the **lengthFilter** is non-zero, only entries with a filename length equal to **lengthFilter** characters will be passed to **proc**.

The following procedure inserts an entry into the directory; unlike **Lookup**, it does not create a file. If the file already exists, TRUE is returned (and **fp^** is undisturbed).

```
Insert: PROCEDURE [
  dir: DiskHandle, fp: POINTER TO AltoFileDefs.FP, name: STRING]
  RETURNS [old: BOOLEAN];
```

**ParseFileName** replaces **ExpandFileName**; it strips the leading directory information from **name**, puts the result in **filename** (appending a period if necessary), and returns a stream (with access **dirAccess**) open on the directory in which the file should be looked up.

```
ParseFileName: PROCEDURE [
  name, filename: STRING, dirAccess: SegmentDefs.AccessOptions]
  RETURNS [StreamDefs.DiskHandle];
```

The following procedures set and return the directory used for looking up files which do not specify a directory name (initially set to `"<SysDir."`).

**SetWorkingDir: PROCEDURE [dir: SegmentDefs.FileHandle];**

**GetWorkingDir: PROCEDURE RETURNS [dir: SegmentDefs.FileHandle];**

Finally, the signal **BadDirectory** no longer takes a string parameter. [**Directory Package**, *Alto Operating System Reference Manual*]

*DisplayDefs*

**DestroyDisplay** can be used to delete the display package; it turns off the display, deallocates the bitmap, destroys the font, and UNNEWs all the display modules. [**Display Package**]

*DoubleDefs*

This interface is no longer implemented or supported, since **LONG** data types are now a standard part of the language and runtime support.

*FrameDefs*

**Validate(Global)Frame** and **Invalid(Global)Frame** now take (return) **UNSPECIFIED**. The procedure

**LoadConfig: PROCEDURE [name: STRING] RETURNS [PROGRAM];**

loads a configuration without starting it and returns its control module or control module list (or NIL if there is no control module). Note that this will not handle configurations whose control modules take parameters. [**Modules**]

*FSPDefs*

The error **ZoneTooLarge** is now raised by **Make(New)Zone** and **AddTo(New)Zone** when an attempt is made to make a zone of more than 32K words. [**Storage Management**]

*ImageDefs*

**MakeImage** takes an optional second parameter (**merge: BOOLEAN _ TRUE**); **MakeUnMergedImage** has been temporarily retained for compatibility. The **ImageMaker** package is no longer a part of the standard system; it must be loaded or bound with the client configuration. [**Image Files**]

*InlineDefs*

The **LongCOPY** operation for use with long pointers is now implemented by the extended memory microcode. The types **BytePair** and **BcplLongNumber** have been added; the procedures **MesaToBcplLongNumber** and **BcplToMesaLongNumber** implement conversion between Mesa and BCPL long numbers. [**Miscellaneous**]

*IODefs*

The procedure **WriteSubString** has been added. [**StreamIO Package**]

*MiscDefs*

The **ByteBlt** procedure has been added. [**Miscellaneous**]

*MiscOps*

**ReleaseDebuggerBitmap** can be used to free the storage normally allocated (on extended memory machines) for the Debugger's bitmap. [**Section 3**]

*OsStaticDefs*

The type of **ClockSecond** has been changed to use **InlineDefs.BcplLongNumber**. [*Alto Operating System Reference Manual*]

*ProcessDefs*

**Aborted** has been redefined to be equal to the predeclared error ABORTED. A **Pause** procedure has been added which delays execution of its caller by the specified number of ticks. **Detach** and **GetCurrent** now take (return) a **PROCESS** instead of an **UNSPECIFIED**. [**Processes and Monitors**]

*SegmentDefs*

The majority of changes in **SegmentDefs** are due to the incorporation of XMesa and extended memory support into the standard system. Clients of XMesa should see the XMesa update document.

The definition of **SegmentObjects** has changed to allow for a twelve bit page number. The **read** bit in **FileSegmentObjects** has been deleted (read access is always assumed) and **MaxSegLocks** has been reduced to fifteen (**MaxSegLocks** and **MaxFileLocks** replace **MaxLocks**).

A type field has been added to **DataSegmentObject**s with predefined values **UnknownDS**, **FrameDS**, **TableDS**, **HeapDS**, **SystemDS**, **BitmapDS**, **StreamBufferDS**, and **PupBufferDS**. These types are interpreted by the Debugger's `Coremap` command.

For clients of low level memory allocation, the definition of **AllocInfo** has changed and the constants **HardUp**, **HardDown**, **EasyUp**, and **EasyDown** have been defined. The procedures **MakeDataSegment** and **MakeSwappedIn** have been moved here from **AllocDefs**.

The access options **ReadWrite**, **WriteAppend** and **ReadWriteAppend** have been added. **NewFile** and **InsertFile** now default the **access** and **version** parameters.

The following two procedures have been added to provide access to file times:

```
GetFileTimes: PROCEDURE [file: FileHandle]
  RETURNS [read, write, create: TimeDefs.PackedTime];

SetFileTimes: PROCEDURE [
  file: FileHandle,
  read, write, create: TimeDefs.PackedTime _ TimeDefs.DefaultTime];
```

**GetFileTimes** does not modify any of the file's times. In **SetFileTimes**, if any of the times are defaulted, the current time is used. [**Segment Package**]

*StreamDefs*

**DestroyKeyHandler** can be used to delete the standard keyboard handler; it destroys the keyboard process and UNNEWs all the keyboard modules. [**Keyboard**]

The type **StreamPosition**, defined as a **LONG CARDINAL**, can be used in place of a **StreamIndex**. The operations **GetPosition**, **SetPosition**, and **ModifyPosition** are similar to the corresponding index operations; **IndexToPosition** and **PositionToIndex** perform conversions between positions and indicies. The access options **ReadWrite**, **WriteAppend**, and **ReadWriteAppend** have been added, as has the signal **FileNameError**. [**Disk Streams**]

### StreamScan

This new interface allows overlapped disk input when reading from a stream. It is a transliteration of the same code from the Alto Operating System (version 17). The following are defined in StreamScan.mesa:

```
Descriptor: TYPE = RECORD [
  da: AltoFileDefs.vDA,
  pageNumber: CARDINAL,
  numChars: CARDINAL,
  -- private fields];

Handle: TYPE = POINTER TO READONLY Descriptor;

Init: PROCEDURE [
  stream: StreamDefs.StreamHandle, bufTable: POINTER, nBufs: CARDINAL]
  RETURNS [Handle];

GetBuffer: PUBLIC PROCEDURE [ssd: Handle] RETURNS [POINTER];

Finish: PROCEDURE [ssd: Handle];
```

**Init** sets up a scan stream from a disk stream. In addition to the stream, the client supplies a vector of pointers to 256 word areas useable as disk buffers (**bufTable**). The number of buffers supplied is **nBufs**. At least one buffer must be supplied (the normal stream buffer is also used). Each call to **GetBuffer** will return a pointer to the next sequential page of the file and returns the previous buffer page to the buffer pool (first call returns data page 0; file page 1). The public fields of the **Handle** are correct for the page returned by the most recent call to **GetBuffer**. **GetBuffer** returns **NIL** when there are no more pages to be read. A call to **Finish** terminates the scan. No other stream operations should be performed between **Init** and **Finish**. [**Disk Streams Package**]

### StringDefs

**CompareStrings** lexically compares two strings and returns -1, 0, or 1 if the first is less than, equal to, or greater than the second; an optional parameter may be supplied to ignore case differences. All procedures in this interface now handle NIL string parameters. [**String Package**]

### SystemDefs

**CopyString** allocates storage from the system heap and copies its argument into it, optionally making the new string longer. **ExpandString** performes a similar function, allocating a new string (and freeing the old one) if necessary. **Even** and **Quad** can be used to align pointers on double and quad word boundaries. [**Storage Management**]

### TimeDefs

The type **HardwareTime** has been replaced by **InlineDefs.BcplLongNumber**. Default values have been added to **UnpackDT**, **PackDT**, and **AppendDayTime**. **ReadClock** returns the current value of the Alto's realtime clock (part of which can be found at location **RealTimeClock**). [**Time Package**]

### TrapDefs

**StackError** no longer takes a parameter; **UnboundProcedure** now takes an **UNSPECIFIED**. The following signals have been added (not all of which can be generated by Alto/Mesa): **ZeroDivisor**, **DivideCheck**, **UnimplementedInst**, **WakeupError**, **PageFault**, **WriteProtectFault**, and **HardwareError**. [**Traps**]

*XMesa Extended Memory Support*

Functions formerly provided by XMesa are now integrated with the standard system. A 3K RAM or Mesa microcode in ROM1 is required to support the extended memory option. [**Segment Package**]

### Internal Interfaces

The following changes are internal to the implementation and do not affect public interfaces; they may affect performance and/or space requirements, however. Note that Mesa 6.0 continues to support version 39 of the XMesa microcode available with Mesa 5.0; obviously, certain new features listed below are not available if your ROM contains the old microcode (e.g., Long BitBlt).

*3K RAM Support*

Mesa now supports the Alto 3K RAM option (available only on extended memory machines).

*Debugger Bitmap*

If the extended memory option is present, the system allocates part of the client's memory for use by the debugger for its display bitmap; this improves the debugger's response times considerably. The debugger bitmap may be deallocated by the procedure **MiscOps.ReleaseDebuggerBitmap** or the command line switch /k (in the former case, the call must be made before the debugger is first entered).

*Long Copy, Long BitBlt*

These opcodes now include support for extended memory.

*Misc Opcodes*

Misc opcodes (except for **RCLK**) now provide a general escape to user microcode in the RAM if Mesa is running on a 2K ROM or 3K RAM machine; they produce undefined results otherwise. Alpha bytes for the currently implemented MISC functions are defined in **MiscAlpha**.

*Overflow Microcode*

RunMesa has been upgraded to include microcode support for Pup checksums, IEEE floating point, and HBlt (used by Griffin). This microcode is loaded with the XMesa overflow microcode on Altos with the 2K ROM (with version 41 microcode) or 3K RAM option. Users who have been loading microcode for these functions need no longer do so. This change affects Alto IIs only.

*Range Checking*

The bounds check instruction (**BNDCK**) is now implemented correctly.

Distribution:
Mesa Users
Mesa Group
SDSupport