Inter-Office Memorandum

To	Mesa Users	Date	October 27, 1980
From	John Wick	Location	Palo Alto
Subject	**Mesa 6.0 Change Summary**	Organization	SDD/SS/Mesa

XEROX

Filed on: [Iris]<Mesa>Doc>Summary60.bravo (and .press)

This memo outlines changes made in Mesa since the last release (Mesa 5.0, April 9, 1979). The first section lists references which should be consulted for more detailed information. The second section contains an overview of the changes. The final section describes operational differences which you must know in order to use Mesa 6.0.

For most programs, you should be able to begin converting to Mesa 6.0 after reading only this memo; then consult the various updates when details of new features are required. If your application relies on the extended memory support previously provided by XMesa, you will probably need to read the System and XMesa updates before converting.

References

The following documents can be found on `<Mesa>Doc>`. (In Palo Alto, the official Mesa release directory is maintained on Iris (there is also a copy on Ivy); in El Segundo, it is on Isis. For other locations, consult your support group.) In addition, the file `Mesa60.press` is a compilation of this and other material (about 75 pages); hardcopy is available through your support group.

Mesa 6.0 Change Summary. `Summary60.press`
Mesa 6.0 Compiler Update. `Compiler60.press`
Mesa 6.0 Binder Update. `Binder60.press`
Mesa 6.0 System Update. `System60.press`
Mesa 6.0 XMesa Update. `XMesa60.press`
Mesa 6.0 Debugger Update. `Debugger60.press`
Mesa 6.0 Utilities Update. `Utilities60.press`
Mesa 6.0 Pup and Ftp Update. `PupFtp60.press`

The above memos describe changes since the last release. The `<Mesa>Doc>` directory also includes new versions of the following documents:

Mesa System Documentation. `System.press`
Mesa Debugger Documentation. `Debugger.press`
Mesa User's Handbook. `Muhb.press`
Mesa Pup Package Functional Specification. `Pup.press`

Mesa Ftp Functional Specification. `Ftp.press`
Integrated Mesa Environment. `CommandCentral.press` (in `Utilities60.press`)
Performance Mesaurement Tool. `PerformanceTool.press`
Control Transfer Counting Tool. `XferCounter.press`
Debugger: Extended Features. `XDF.press` (in `Debugger60.press`)

Because the language changes are upward compatible with Mesa 5.0, a new version of the *Mesa Language Manual* will not be issued with this release; consult the *Compiler Update* and the *Binder Update* for information on additions to Mesa and C/Mesa.

Highlights

The primary emphasis in this release has been on the following areas:

A number of significant new language features are included: extended defaults, floating point support, machine dependent records and enumerated types, dynamic storage allocation, sequences, and exported types. Less major revisions and extensions have been made in the following areas: syntactic and semantic glitches, keyword array constructors, packed array representation, successor and predecessor operations, the DIRECTORY statement, implicitly imported interfaces, type TEXT, loop control variables, extended NIL, the REJECT statement, and the ABORTED signal. The language changes are compatible with Mesa 5.0, and will not affect existing source code (there are some minor exceptions; see below). In addition, the Compiler's user interface and command line syntax have changed substantially. See the *Compiler Update* and Appendix A of the *Mesa User's Handbook* for details.

A Mesa source formatter is available with this release. It establishes a standard representation for source text, and will produce both plain and formatted `.bravo` and `.press` files. Complete documentation can be found in Appendix B of the *Mesa User's Handbook*.

The Binder has been updated to agree with the Compiler's treatment of the DIRECTORY statement and its user interface and command line syntax. In addition, it now supports multiple control modules in a single configuration. See the *Binder Update* and Appendix C of the *Mesa User's Handbook* for details.

Support for extended memory beyond 64K, formerly provided by XMesa, is now part of the standard system. The BasicMesa configuration has been abandoned in favor of facilities which dynamically delete non-essential components of the system. Support for subdirectories, fast directory scanning, floating point, Pup checksums, and 3K RAMs have been added. (Appendix G of the *Mesa User's Handbook* details extensions made to RunMesa.) Other than memory allocation facilities, changes to the System and Pup and Ftp interfaces are minor; see the updates for details.

The Debugger has a new user interface, a new interpreter, and a simple cut and paste editor. The Debugger's command language is essentially unchanged, but several extensions in window operations and source window facilities have been added. A description of the new **FileTool** is attached to the *Debugger Update*.

Several new commands have been added to the Lister, and the IncludeChecker has been extensively modified to support large configurations more efficiently. A new package, **CommandCentral**, is available for use with the Debugger; it serves as a small executive for controlling the editing, compilation, binding, and debugging of applications software. This package greatly speeds up the edit compile bind debug cycle when small changes are involved. Documentation is attached to the *Utilities Update*.

A new mode has been added to the Control Transfer Counting Tool which allows modules to be assigned to groups and information to be collected on cross group as well as cross module transfers. The performance measurement tools are documented separately as described in the first section of this memo.

Operational Changes

This section summarizes important operational differences which you must know in order to use Mesa 6.0*; do not begin to convert to Mesa 6.0 until you have read it*. More complete information is contained in the update memos listed in the first section of this document.

General

On large programs, performance of the Mesa 6.0 Compiler and Debugger on 64K Altos is considerably worse than in Mesa 5.0; *an extended memory machine with a minimum of 128K is recommended*.

Alto Operating System version 18 and Executive version 11 or later are required to run Mesa 6.0.

The new file creation date standard is now supported. The compiler inserts the creation date of the source file into the `.bcd` (as does the Binder), and the Debugger checks that the source and object file versions match. The IncludeChecker also makes use of these dates. *Therefore, you must use an editor and a file server that support the file date standard*. If you use dump files, be sure you have the latest version of Ftp.

Language

With two exceptions, Mesa 5.0 source files are compatible with Mesa 6.0. In file names in the DIRECTORY statement, names inside angle brackets are no longer ignored; they are treated as subdirectories of the Alto file system (use of this feature is not recommended). The following new reserved words have been added: FREE, PRED, PROC, REJECT, SEQUENCE, SUCC, UNCOUNTED, and ZONE.

Compiler

The order of evaluating the items in constructors (including argument lists) and the operands of infix operators (except AND and OR) has been changed, particularly in cases involving embedded procedure calls; it is no longer always left to right. In particular, expressions of the form word _ get[s]*256+get[s] are suspect, and probably incorrect.

Except during initialization, constructors for records containing MONITORLOCKs, CONDITIONs, and PORTs are not allowed (to prevent unintentionally overwriting these fields with their default values). Such records must be assigned to field by field.

For element sizes of four bits or less, the internal representation of packed arrays has changed. This is a potential problem in reading files containing packed arrays created by earlier versions of Mesa.

The Compiler no longer supports interactive command input; it reads commands only from the command line, and does not use the keyboard or display (limited feedback is via the cursor). The command language has been extended and switch processing has changed. The `/c` switch has been deleted; global switches must now be specified with a null file name, e.g. `Compile /p Defs Impl`.

Switches are restricted to a single letter. *Do not use complete switch names such as* `/pause` (each letter will be interpreted as a separate switch). Until you understand the full syntax of the command language, it is best to group all global switches at the beginning of the command line following a single slash.

The log is now written on `Compiler.log`, not `Mesa.typescript`; separate `.errlog` files are still produced. The error log (if any) is deleted if the compilation is successful; conversely, if the compilation fails, the `.bcd` (if any) is deleted.

The implementation of floating point has changed considerably; the IEEE standard format is now used, and the compiler generates calls directly to user-supplied microcode (this will produce undefined results if the proper microcode is not loaded). Calls to software floating point as in Mesa 5.0 can be generated with the `/-f` switch. *Do not use type* REAL *without first consulting with the supplier of your floating point package.*

Because of bug fixes, previously acceptable programs may no longer compile.

Binder

The Binder is now available only as a `.bcd` file; you must have `Mesa.image` to run it. Like the Compiler, the Binder takes commands only from the command line and does not use the keyboard or display; it writes its log file on `Binder.log`.

The meaning of switches used to copy code and symbols has changed substantially; read the *Binder Update* if you use these options. Compressed symbols have been compressed still further to include only procedure and signal names (without parameters or results); this substantially reduces the size of these `.symbol` files.

Because of bug fixes, previously acceptable configurations may no longer bind.

System

Mesa 6.0 is compatible with XMesa 5.0 microcode version 39 (but some new features are not available with the old microcode; e.g., extended memory BITBLT).

Features previously implemented by BasicMesa and XMesa are now a standard part of `Mesa.image`. **MakeImage** is no longer a part of the standard system; **ImageMaker** must be loaded or bound with the client configuration if it is needed.

The standard system now supports only command line input; the **MesaExec** can be loaded to provide interactive input. The command line switch `/b` can be used to convert the standard system to a basic one. The `/k` switch will disable the allocation of space for the Debugger’s bitmap on extended memory machines (see below).

The default maximum numbers of processes (75) and modules (384) have been increased. A version of the System called **SmallMesa** allows 33 processes and 256 modules.

Interface changes are described in the *System Update*.

Debugger

The Debugger has also been enhanced to take advantage of extended memory. If you have more than two banks (128K) of memory, see the installation section of the *Debugger Update* for an explanation of the options available; otherwise, the standard defaults will "do the right thing".

The Debugger now requires a strike font named `SysFont.strike` or `MesaFont.strike`; a version of Gacha10 is available on `<Mesa>MesaFont.strike`. Additional fonts are available on `[Maxc]<AltoFonts>`. (Strike fonts that include kerning are not supported.)

The selection scheme and the assignment of function keys and mouse buttons has changed. Clicking RED once selects a character, clicking twice selects a word, three times a line, etc.; the selection can be extended to the left or right with BLUE. The menu button is now YELLOW (formerly BLUE). FL4 is no longer the stuff key; use FR4 (Spare2) or Keyset2.

Scrollbars no longer occupy a dedicated part of the window, but instead come up on top of the left edge. To obtain a scroll bar, move left just past the edge of the window, then move right slightly, back into the window.

New source window menu commands have been added, and they have been factored into several menus. The Debugger's wisk window has been replaced by a more general `Split` menu command. There are now also `Normalize Insertion` and `Normalize Selection` commands.

The interpreter can now be used when in `Display Stack` and `Display Process` subcommand mode. Several commands now invoke the interpreter automatically (e.g., `Octal Read: @p, n: SIZE[R]`). The interpreter does procedure calls (the `Interpret Call` command has been deleted).

The constructs ABS, ERROR, LONG, LOOPHOLE, MAX, MIN, NIL, POINTER TO, PROC, PROCEDURE, SIGNAL, WORD, and open and half open intervals have been added to the interpreter's grammar. Type expressions following **%** must be enclosed in parentheses. The interpreter syntax **Expression?** replaces the `Interpret Expression` command.

Each breakpoint is now assigned a unique number used for displaying and clearing it. There are new commands for attaching conditions to break and tracepoints. Break/Tracepoints can no longer be set by typing a source line, and the `Break Module` and `Break Procedure` commands and corresponding `Trace` and `Clear` commands have been deleted; the menu commands must be used. `Clear All Entries/Xits` clears both break and tracepoints.

Tracepoints now automatically invoke the normal `Display Stack` command processor (with subcommand `p(arameters)`, `v(ariables)`, or `r(esults)` as appropriate). The `q(uit)` subcommand (not `b(reak)`) exits to the Debugger's command level; it no longer continues execution of the client.

If the source window is loaded with the `s(ource)` subcommand of `Display Stack`, the window will remember the appropriate context for setting breakpoints. For an exit break, the `s(ource)` subcommand now displays the declaration line of the procedure.

The Debugger no longer ignores case, and the case commands have been deleted; *identifiers must be typed with their correct capitalization.*

Distribution:
Mesa Users
Mesa Group
SDSupport