Inter-Office Memorandum

To	Mesa Users	Date	October 27, 1980
From	Brian Lewis, Ed Satterthwaite	Location	Palo Alto
Subject	**Mesa 6.0 Binder Update**	Organization	SDD/SS/Mesa

XEROX

Filed on: [Iris]<Mesa>Doc>Binder60.bravo (and .press)

This memo describes changes to the C/Mesa language and the Binder that have been made since the release of Mesa 5.0 (April 9, 1979).

Definitions of syntactic phrase classes used but not defined in this document come from the *Mesa Language Manual, Version 5.0*, Section 7.7.

Compatibility

Because of changes in BCD formats, you must rebind all your existing Mesa configurations after obtaining recompiled versions of their components. There is one potential incompatibility:

> Some of the quoted file names that appear in DIRECTORY clauses have different interpretations. Text inside angle brackets is no longer ignored; it is treated as the name of a local subdirectory (but we do not recommend using local subdirectories for Mesa programs at the present time).

A number of bugs have been fixed. As usual, the list of Binder-related change requests closed by Mesa 6.0 will appear separately as part of the Software Release Description. The most notable involve

> correctly checking the types of interfaces when positional notation is used,
>
> proper treatment of named imports or exports of a configuration,
>
> proper identification of object files that have been renamed.

Because of bug fixes, the Binder may reject previously acceptable configuration descriptions or may issue new warning messages. A number of error or warning messages now use symbolic names whenever the corresponding symbol tables are available.

New Language Features

Bracket Symbols

The bracket pair { } can be used in place of BEGIN END.

Syntax

CBody	::=	BEGIN CStatementSeries END
	|	{ CStatementSeries }

Multiple Control Modules

You can now specify an ordered list of control modules for any configuration.

Syntax

ControlClause	::=	CONTROL IdList
	|	empty
IdList	::=	identifier | IdList , identifier

When a (sub)configuration is started, either explicitly or as the result of a start trap (see the *Mesa Language Manual, Version 5.0*, Section 7.8.4), each of the modules named in the ControlClause is started in order.

Note that, if there are calls from a control module to one of its successors in the list, the order of starting will not necessarily follow the order of the ControlClause. In starting a configuration, any control modules that have already been started are skipped.

Operational Changes

The Binder is now available only as a `.bcd` file; you must have `Mesa.image` to run it. When compressing symbols, SymbolCompressor.bcd must be loaded first; see the description of the /x switch below.

User Interface

The Binder now reads commands only from the Alto Executive's command line; it no longer supports interactive input. At the start of the first binding, the message "Bind" is displayed in the cursor. If there are any warnings, "Warning" is displayed, and if there are errors "Errors" is shown. At the end of binding, the message "Type Key" is displayed in a flashing cursor if there are errors and you have requested the Binder to pause. Typing `Shift-Swat` aborts the executive's current command sequence; `Ctrl-Swat` invokes the Mesa Debugger; any other character causes normal exit from the Binder.

A summary of binder commands is written on the file `Binder.log` (formerly `Mesa.typescript`).

Commands

The Mesa 6.0 Binder allows you to control the names and contents of various output files when you invoke it; it accepts a series of commands, each of which usually has one of the following forms:

```
inputFile/switches
outputFile _ inputFile/switches
[key1: file1, ... keym: filem] _ inputFile/switches
```

(It is also possible to control the association between included modules and configurations and their file names; this is described below.) In the last form, `key` is one of `bcd`, `code`, or `symbols`.

The string `inputFile` names the file containing the source text of the configuration description, and its default extension is `.config`.

There is a *principal* output file, the name of which is determined as follows:

> If you use the first command form, it is `inputRoot.bcd`, where `inputRoot` is the string obtained by deleting any extension from `inputFile`.
>
> If you use the second form, it is `outputFile`, with default extension `.bcd`.
>
> If you use the third form and key_i is `bcd`, it is $file_i$, with default extension `.bcd`; otherwise, it is obtained as described for the first form.

If the Binder detects any errors, the principal output file is not written, and any existing file with the same name is deleted.

You may also request that the code and/or symbols of the constituent modules be copied to an output file, as follows:

> You request copying of code by specifying the `/c` switch *or* by using the third command form with keyword `code`. Code is copied to the principal output file unless you use the third form and key_i is `code`, in which case the code is copied to a file named $file_i$, with default extension `.code`.
>
> You request copying of symbols by specifying the `/s` or `/x` switch *or* by using the third command form with keyword `symbols`. Symbols are copied to the file `inputRoot.symbols` unless you use the third form and key_i is `symbols`, in which case they are copied to a file named $file_i$, with default extension `.symbols`. Compressed symbols are copied if the `/x` switch is specified.

Any warning or error messages are written on the file `outputRoot.errlog`, where `outputRoot` is the string obtained by deleting any extension from the name of the principal output file. If there are no errors or warnings, any existing error log with the same name is deleted at the end of the bind.

When more than one Binder command is given on the command line, the commands must be separated by semicolons. However, you may omit a semicolon between any two successive identifiers (file names or switches), or between a] and an identifier (but not between an identifier and a `/`). Note that any required semicolon in an Alto Executive command must be quoted.

Switches

The optional `switches` are a sequence of zero or more letters. Each letter is interpreted as a separate switch designator, and each may optionally be preceded by `-` or `~` to invert the sense of the switch.

The Binder recognizes the switches:

- `/c` - copy code
- `/s` - copy symbols
- `/x` - copy compressed symbols
- `/p` - pause after binding if there are errors or warnings
- `/r` - run the specified `.image` or `.run` file
- `/g` - (has no effect; retained for compatibility with Mesa 5)

The Mesa 5 Binder switch `/o` (output file) is no longer available.

In earlier versions of the Binder, copying symbols was an error if not all of the symbol files were available at bind time. The Binder now copies all symbols that it can find, leaves the symbol table references for the other modules in the original (unavailable) files, and issues a warning.

The switches `/c` and `/s` are interpreted differently in Mesa 6 than they were in Mesa 5. The following table outlines these changes.

Mesa 6	*Mesa 5*	*meaning*
`file/c`	`file/c`	code to file.bcd
`file/cs`	`file/c file/s`	code to file.bcd; symbols to file.symbols
`[symbols: file.bcd] _ file/c`	`file/cs`	code and symbols to file.bcd

Note that `file/cs` has quite a different meaning in Mesa 6 than before. Also, the common Mesa 5 command sequence `file/c file/s` will bind `file` *twice* in Mesa 6, the first time copying only code and the second time only symbols.

Compressed symbols are copied with the `/x` switch. In this mode, only the following symbols are included: all procedures, and the signals declared at the top level of modules. Symbols for their parameters and results are not copied. This option allows limited but often adequate debugging, and substantially reduces the size of the symbols file (typically by more than 50%). To copy compressed symbols, `SymbolCompressor.bcd` must be loaded ahead of the Binder. Thus

```
>Mesa.image SymbolCompressor.bcd Binder.bcd MySystem/x
```

will create `MySystem.bcd` and `MySystem.symbols` (compressed).

Global switches are set by a command with an empty file name. In the form `/switches`, each letter designates a different switch. The switches to copy code (`/c`), to copy symbols (`/s`), and to compress symbols (`/x`) may now be given as global switches, and hence apply to all source files thereafter. Unless a command to change the global switch settings comes first in the sequence of commands, it must be separated from the preceding command by an explicit semicolon.

Associating File Names with Modules and Configurations

The Binder now lets you control the association between file names and modules or subconfigurations when you call it. This is done by specifying a list of component identifier-file name pairs inside brackets after the input file name. For example, the command

```
MySystem[Test: UnpackedTest]
```

will bind `MySystem.config` using the previously bound configuration **Test** that is stored on the file `UnpackedTest.bcd`. A command that includes one of these optional component-file name lists will have one of the forms:

```
inputFile[id1: file1, ... idn: filen]/switches
outputFile _ inputFile[id1: file1, ... idn: filen]/switches
[key1: file1, ... keym: filem] _ inputFile[id1: file1, ...
                                           idn: filen]/switches
```

The module or configuration named by id_i in the configuration description will be read from the file $file_i$; `.bcd` is the default extension.

Distribution:
Mesa Users
Mesa Group
SD Support