

To: Lisa Software
From: Rich Page, Ken Friedenbach and Ralph Sahn
Date: July 15, 1982
Subj: Pascal Development System Release 10.0

CAUTION!!!

This release is intended for use with O.S. release 5.2. If you use the Diskcopy utility or the Z(ero command in the Sysmgr from this release to upgrade a Profile, you will NOT be able to use that Profile with any older revisions of the Monitor.

Overview

Major changes in this release are as follows:

- 1) Approximately a two fold performance increase for Profiles.
- 2) The disk driver for the Profile now supports the following:
 - a) Block headers (20 bytes) compatible with the Lisa O.S. Disks without headers are still supported.
 - b) Interleave of 9:1 (for disks with headers).
- 3) The Monitor file system has been modified to relax the 77 file limit to 786 files per volume.
- 4) The Monitor can now be booted from the lower Twiggy drive (Use Command-G at boot time).
- 5) The Formatter utility now does a certification test to catch bad media. Media is currently not being tested by the vendor.
- 6) Filer and Sysmgr have been modified to support the changes in the Monitor file system and disk drivers.

CAUTION: The Z(ero command in the new Filer now asks a new question. This may affect MON.STARTUP files and other execute files which attempt to initialize MEMORY: (#4:).

- 6) The Sysmgr and Monitor have been modified to ignore illegal values from the clock.
- 7) The Compiler and Linker have been modified to provide better support for interfaces in IU library files.

Disks

This release consists of a Twiggy Boot Disk (with System Files) and an Apple II Boot Disk.

Twiggy Boot Disk: (with Release System Files)

REL10.0:

Boot Files:	MON.LOADER	10	13-Jul-82	6	512	Datfile
	CONFIG.DATA	1	7-Jun-82	16	512	Datfile
	BOOTFILES.DATA	1	2-Jun-82	17	512	Datfile
	LISABUG.OBJ	60	13-Jul-82	18	512	Datfile
	LISABUG2.OBJ	23	19-May-82	78	512	Datfile
	DRIVERS.OBJ	15	21-Jun-82	101	512	Datfile
	MS.OBJ	4	12-Jul-82	116	512	Datfile
	UARTDRVR.OBJ	2	23-Apr-82	120	512	Datfile
	LOADER.OBJ	25	21-May-82	122	512	Datfile
	TWGDRVR4.OBJ	4	13-Jul-82	147	512	Datfile
	MONITOR.OBJ	35	15-Jul-82	151	512	Datfile
	MONITOR.SYMBOLS	11	15-Jul-82	186	512	Datfile

System Files:	MON.MISCINFO	1	12-Jun-80	197	192	Datfile
	MON.MISCINFO	1	12-Jun-80	197	192	Datfile
	MON.STARTUP	4	8-Oct-81	198	512	Textfile
	MONSTART1.OBJ	3	4-Jun-81	202	512	Datfile
	FORMATTER.OBJ	13	13-Jul-82	205	512	Datfile
	SYSMGR.OBJ	35	12-Jul-82	218	512	Datfile
	DISKCOPY.OBJ	14	11-Jul-82	253	512	Datfile
	FILER.OBJ	123	16-Jul-82	267	512	Datfile
	DUMPOBJ.OBJ	47	11-Jul-82	390	512	Datfile
	COMPILER.OBJ	213	11-Jul-82	437	512	Datfile
	CODE.OBJ	145	11-Jul-82	650	512	Datfile
	CHANGESEG.OBJ	4	11-Jul-82	795	512	Datfile
	GXREF.OBJ	15	11-Jul-82	799	512	Datfile
	CODESIZE.OBJ	15	11-Jul-82	814	512	Datfile
	SEGMAP.OBJ	5	11-Jul-82	829	512	Datfile
	TO.OS.OBJ	3	11-Jul-82	834	512	Datfile
	CHANGETRAP.OBJ	4	11-Jul-82	837	512	Datfile
	IULINKER.OBJ	70	14-Jul-82	841	512	Datfile
	OBJIOLIB.OBJ	102	14-Jul-82	911	512	Datfile
	IUMANAGER.OBJ	28	14-Jul-82	1013	512	Datfile
	PACKSEG.OBJ	10	14-Jul-82	1041	512	Datfile
	INTRINSIC.LIB	8	14-Jul-82	1051	512	Datfile
	IUPASLIB.OBJ	15	14-Feb-82	1059	512	Datfile
	< UNUSED >	508		1074		

34/34 files<listed/in-dir>, 1074 blocks used, 508 unused, 508 in largest

Apple II Boot Disk:

RELI0BT:

SYSTEM.APPLE	32	14-Aug-80	6	512	Datafile
SYSTEM.PASCAL	40	13-Nov-80	38	512	Datafile
SYSTEM.MISCINFO	1	4-May-79	78	512	Datafile
CONFIG.DATA	1	7-Jun-82	79	512	Datafile
SYSTEM.STARTUP	14	11-Jun-82	80	512	Codefile
BOOTFILES.DATA	1	15-Jul-82	94	512	Datafile
LISABUG.OBJ	60	13-Jul-82	95	512	Datafile
LISABUG2.OBJ	23	19-May-82	155	512	Datafile
DRIVERS.OBJ	15	21-Jun-82	178	512	Datafile
MS.OBJ	4	12-Jul-82	193	512	Datafile
UARTDRVR.OBJ	2	23-Apr-82	197	512	Datafile
LOADER.OBJ	25	21-May-82	199	512	Datafile
TWGDRVR4.OBJ	4	13-Jul-82	224	512	Datafile
MONITOR.OBJ	35	15-Jul-82	228	512	Datafile
MONITOR.SYMBOLS	11	15-Jul-82	263	512	Datafile
WITHTWIGGY.DATA	1	15-Jul-82	274	512	Datafile
< UNUSED >	5		275		

16/16 files<listed/in-dir>, 275 blocks used, 5 unused, 5 in largest

Installation

1. Boot your machine with your current boot disk and filer.
2. If you wish to save your current *Intrinsic.lib, save it under a different name.
3. Transfer the System Files from the Twiggy boot diskette (or female Apple II diskettes) to the root volume on your working device.
4. Boot your machine with the new boot disk.
5. If you saved an old *Intrinsic.lib, use the IUManager to install the files *OBJIOLIB.OBJ and *IUPASLIB.OBJ. Then use the Filer to replace *Intrinsic.lib.

Note: The Apple II Boot Disk is setup for machines without Twiggies. If you wish to boot from the Apple II and still be able to use twiggies, replace the file BOOTFILES.DATA with the file WITHTWIGGY.DATA.

Diskcopy

This utility allows a user to copy between two Monitor devices (hard disks only). The source device may be either an old Monitor device (no headers) or a new Monitor device. If the target device is a Profile it will have headers when the copy is complete.

Caution: The DISKCOPY utility writes headers without concern

for any previous headers that may be present. For a Profile that contains both O.S. and Monitor Volumes the DISKCOPY utility will have the effect of destroying the O.S. Volume.

Note: The Z(ero command in Sysmgr provides a much faster way of formatting a Profile when there is no information to be saved. (see below for more details)

To upgrade a Profile using the DISKCOPY utility you will need a spare Profile and a 4-port card. Using the spare Profile do the following:

X(ecute DISKCOPY

Reply with source device number. (&3 for built-in port, etc.)
Reply with target device number. (The spare Profile !)

DISKCOPY will echo source and target device numbers.

Type space to continue or <esc> if you made an error.

Type <cr> to prompt for number of blocks to transfer.

DISKCOPY will copy source to target (about 17 mins).

If there are no errors DISKCOPY will ask if you want to copy the target back to the source. Reply with Y.

DISKCOPY will echo new source and target numbers.

Type space to continue or <esc> if you made an error.

DISKCOPY will copy source to target (about 13 mins).

If there are no errors then your original Profile will now have headers and be interleaved 9:1.

Note: The Diskcopy utility uses a buffer of about 250k bytes and each group of four dots represents a read, write, read,read and verify sequence. If any system failure or power failure should occur at most you will lose the target device of the last copy operation. If the crash happened during the first copy simply start the process over from scratch.

If the crash happened during the second copy use Diskcopy to copy from the spare profile (now the source) to your Profile (now the target). After the first copy is complete you are finished (Reply with N to the copy the target back to the source question).

Filer

The Filer now supports the building of directories which are either small (limit of 77 filenames) or large (limit of 786 filenames).

The Z(ero command now asks an additional question. After asking for the number of blocks on the volume the Filer now asks if a large directory should be built. A response of Y or y will cause a large directory to be built.

The Volume Manager will automatically create a large directory for any volume with 2048 or more blocks.

Sysmgr

The Z(ero command no longer has any options. It will zero both Mount Tables and Volume Tables. In addition, for a ProFile, the Z(ero command will write headers on all the blocks (8 min.).

The T(ime and S(et time commands have been modified to ignore bad values returned by the Clock hardware.

The Sysmgr has a new command:

The V(olume command can be used to set the level of sound generated by the speaker (ie. chr(7) is the Bell or Control-G).

Compiler

The definition of the {\$U+} compiler option has been modified to make interfaces in I.U. Libraries easier to use.

{\$U-} still means get all interfaces from unlinked files as specified by the {\$U filename} option.

{\$U+} (the default) now means attempt to get interfaces from the IU Library files named in *Intrinsic.lib, provided the unit being used does not occur in the same IU Library file as the unit being compiled.

```
Example:      Unit Foo; Intrinsic;
              uses {$U A.OBJ} A,
                 {$U B.OBJ} B,

              {$U C.OBJ} C,
              {$U D.OBJ} D;
```

Suppose that Units A and B are linked into IULibOne and that Units C, D, and Foo are linked into IULibTwo. Assuming the \$U+ option is in effect, the compiler will attempt to get the interfaces for Units A and B from the file IULibOne. (If IULibOne does not

contain interfaces then the compiler will go to the unlinked files A.OBJ and B.OBJ). However, since Foo is in the same library as C and D, the compiler will not attempt to use interfaces from IULibTwo.

.linker

The linker now places two new records in IU Library files. One is an Interface Location record which is used by the Compiler to quickly access Unit Interfaces. The other is a Segment Location record which can be used to support the "look around" strategy of the O.S. in finding intrinsic segments (of the same release but with minor variations, such as packed for Lisa Office System or unpacked for the Development System). Dumpobj, PackSeg, and the IUManager have been modified to support these changes.

IUManager

The IUManager now copies the (optional) Code Block at the end of the input file to the output file. This is intended for the version of the O.S. which will not have LisaBug. The ChangeTrap Utility can be used to install a new IU trap handler.

T0.0S

The T0.0S utility is not needed, unless you wish to use Monitor 10.0 to build software to run on O.S. 5.0.x (or earlier). Use of the T0.0S utility will prevent the compiler from being able to access Unit interfaces indirectly through *Intrinsic.lib.