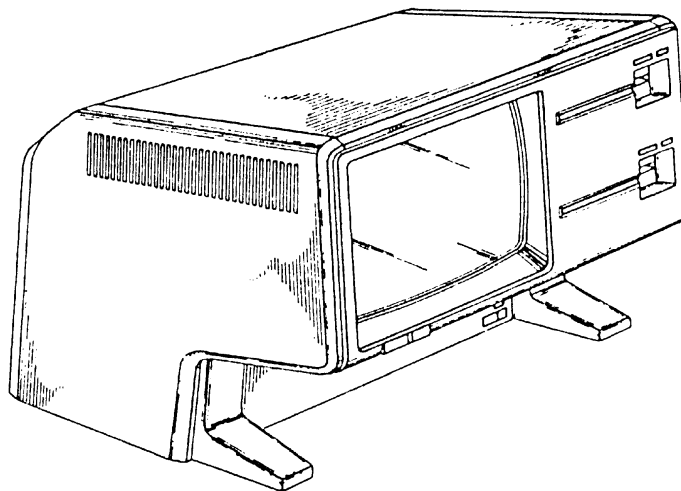


Apple Lisa

Desktop Library 3.0

Pascal Interfaces



This is a listing of the Lisa desktop library interfaces. These interfaces provided the foundation for the Lisa's desktop and windowing metaphor (collectively called "Lisa Technology").

Apple later based the Macintosh Toolbox on these libraries and in some cases used nearly an identical programming interface (e.g. MENUS).

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNIT TABLE"
=====
```

```
000001 *****
000002 *
000003 *           APPLE LISA DESKTOP 3.0 LIBRARIES UNIT TABLE
000004 *
000005 *****
000006
000007 UnitName Unit# File# Type   DataSize
000008 -----
000009 PASLIB      1     3  Intrin  000000
000010 UNITSTD    2     4  Intrin  000002
000011 GRAFUTIL   3     4  Intrin  000000
000012 UNITHZ     4     4  Shared  000024
000013 STORAGE    5     4  Intrin  000008
000014 QUICKDRA   6     4  Intrin  0000CE
000015 HWINT      7     4  Intrin  000000
000016 FEDEC      8     4  Intrin  000000
000017 FONTMGR    9     4  Shared  000054
000018 EVENTS    10    4  Shared  000210
000019 WINDOWS   11    4  Shared  0010F2
000020 FOLDERS    12    4  Shared  00000A
000021 MENUS     13    4  Shared  00009C
000022 FLDUT     14    4  Intrin  000000
000023 WMLSTD    15    4  Intrin  000018
000024 WMLCRS    16    4  Intrin  0003FE
000025 WMLGROW   17    4  Intrin  00000E
000026 WMLSB     18    4  Intrin  000038
000027 INTERNAT  19    4  Intrin  000000
000028 FPLIB     20    5  Intrin  000006
000029 ALERTMGR  21    4  Intrin  000300
000030 MATHLIB   22    5  Intrin  000016
000031 UNITFF    23    4  Shared  00004A
000032 UNITCS    24    4  Shared  000644
000033 UNITFILE  25    4  Shared  000156
000034 UNITFMT   26    4  Shared  0001F2
000035 BGENV     27    8  Intrin  000058
000036 MATMGR    28    8  Intrin  00188E
000037 PARSER    29    8  Intrin  0006FC
000038 COMPUTE   30    8  Intrin  000084
000039 STUBS     31    8  Intrin  000000
000040 UUNIVERS  32    4  Intrin  000222
000041 FEGLOBAL  33    8  Intrin  000398
000042 COMGR     34    8  Intrin  00033C
000043 USTDTERM  35    9  Intrin  0002CE
000044 UQPGRAPH  36    9  Intrin  000056
000045 TEEEXEC   37    8  Intrin  000004
000046 UQPPRINT  38    9  Intrin  000188
000047 LCFEXEC   39    8  Intrin  0001A6
000048 TEENV     40    6  Intrin  00080E
000049 TM        41    6  Intrin  000BCA
000050 FMGRUTIL  42    4  Intrin  000000
000051 PMM       43    4  Shared  000042
000052 PMDECL    44    4  Intrin  000000
000053 UNITFIGA  45    4  Shared  00002A
000054 SYSLLOCK  46    4  Intrin  000000
000055 UCLASCAL  47    3  Intrin  00004E
000056 UTKUNIVE  49    11  Intrin  0001CA
000057 UOBJECT   50    10  Intrin  000164
000058 UTEXT     51    11  Intrin  00027C
000059 UDRAW     52    10  Intrin  0001B2
000060 UABC      53    10  Intrin  0009F2
000061 UDIALOG   54    11  Intrin  000564
000062 LCUT      55    6  Intrin  00002C
```

Apple Lisa Computer Technical Information

000063	IOPRIMIT	56	12	Intrin	00057E
000064	SHELLCOM	57	12	Shared	00070C
000065	PROGCOMM	58	12	Intrin	000006
000066	RECOVERY	59	6	Intrin	000000
000067	LOWLEVEL	60	6	Intrin	000126
000068	DBDECL1	61	6	Intrin	000000
000069	POOLER	62	6	Intrin	000052
000070	DBENV	63	4	Intrin	000000
000071	HEAP	64	6	Intrin	000000
000072	VLTREE	65	6	Intrin	000000
000073	CZCOMPAC	66	6	Intrin	00001C
000074	LABSCAN	68	6	Intrin	000000
000075	SCHEMA	69	6	Intrin	000000
000076	SCAN	70	6	Intrin	000000
000077	FIELDEDI	71	4	Intrin	0000CA
000078	SCRAP	73	4	Shared	000250
000079	FILERCOM	75	4	Shared	0002E8
000080	PRPUBLIC	76	7	Intrin	000000
000081	PRSTDINF	77	4	Shared	0005D0
000082	PRSTDPRO	78	7	Intrin	000002
000083	PRFILEPR	79	7	Intrin	000000
000084	PRBUF	80	7	Intrin	000010
000085	PRSPool	81	7	Intrin	000346
000086	QUEUES	82	7	Intrin	000000
000087	PREVENTS	83	7	Intrin	000000
000088	PRDLGMR	84	7	Intrin	0000BA
000089	PRMGR	85	7	Intrin	000026
000090	UVT100	86	9	Intrin	00001C
000091	USOROC	87	9	Intrin	00001C
000092	STDUNIT	88	12	Intrin	0005B0
000093	IUMAN	90	1	Intrin	0000BC
000094	OBJIO	91	1	Intrin	00011C
000095	FILEIO	92	1	Intrin	000000
000096	GRAPHS	94	1	Intrin	000008
000097	TREES	95	1	Intrin	000000
000098	REFS	96	1	Intrin	000004
000099	PARTS	97	1	Intrin	000004
000100	LISTS	98	1	Intrin	000000
000101	MEMMAN	99	1	Intrin	000000
000102	PASDEFS	100	1	Intrin	00025E
000103	MPASLIB	101	2	Intrin	000000
000104	BLKIOINT	102	3	Intrin	0000DE
000105	BLOCKIO	103	3	Shared	0005C4
000106	PASHEAP	104	3	Intrin	000022
000107					
000108	*****				
000109					
000110			THE END		
000111					
000112	*****				
000113					

End of File -- Lines: 113 Characters: 3628

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 ALERTMGR.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : ALERTMGR
000004 *
000005 *****
000006
000007 USES {$U+} ALERTMGR;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libhw/HWINT} HwInt,
000016         {$U libsm/UNITSTD} UnitStd,
000017         {$U libsm/UNITHZ} UnitHz,
000018         {$U libqd/STORAGE} Storage,
000019         {$U libqd/QUICKDRAW} QuickDraw,
000020         {$U libfm/FONTMGR} FontMgr,
000021         {$U libfm/FMGRUTIL} FMgrUtil,
000022         {$U libos/SYSCALL} SysCall,
000023         {$U libos/PSYSCALL} PSysCall,
000024         {$U libpm/PMDECL} PMDecl,
000025         {$U libpm/PMM} PMM,
000026         {$U libwm/EVENTS} Events,
000027         {$U libwm/FOLDERS} Folders,
000028         {$U libwm/MENUS} Menus,
000029         {$U libwm/WINDOWS} Windows,
000030         {$U libsb/WMLSTD} WmlStd,
000031         {$U libsb/WMLCRS} WmlCrs,
000032         {$U libdb/DBENV} dbenv;
000033         {$setc alSymbols := FSymOK}
000034         {$setc alDebug := FDbgOK}
000035         {$setc dbgAlWould := alDebug}
000036         {$setc dbgAlrt := false}
000037         {$setc dbgAlEvt := false}
000038         {$setc dbgAlRdErr := false}
000039         {$setc dbgAlOpErr := false}
000040
000041     CONST
000042         maxButn      = 10;
000043         noButn       = 11;
000044
000045     TYPE
000046         TButn        = 0..maxButn;
000047         TAlertFile   = ^PAlertRec;
000048         PAlertRec    = ^TAlertRec;
000049         TAlertRec    = RECORD
000050             falerts: longint;
000051             fnumAlerts: integer;
000052             frefNum: integer;
000053             fPos: integer;
000054             END;
000055         TParamAlert  = String[40];
000056         TArgAlert    = 0..5;
000057         TCountAlert  = 7..9;
000058         TPstr255     = ^Str255;
000059         TAlertKind   = (getProc, drawProc, askProc, waitProc, stopProc, noteProc,
000060             cautionProc, cautionOKproc, cautionCancelProc,
000061             cautionStopProc, cautionNoteProc, cautionInsistProc);
```

Apple Lisa Computer Technical Information

```
000062
000063   VAR
000064     alertError: integer;
000065     refuseDeactivate: boolean;
000066     traceWouldAlert: boolean;
000067     preventReentry: boolean;
000068     BackGrInProgress: boolean;
000069     {$ifc dbgALrdErr}
000070     fakeALError: integer;
000071     {$sendc}
000072
000073   FUNCTION AskAlert(alertFile: TAlertFile; alertNumber: integer): integer;
000074
000075   PROCEDURE ArgAlert(n: TArgAlert; s: Str255);
000076
000077   FUNCTION BackgroundAlert(alertFile: TAlertFile; alertNumber: integer;
000078     alertType: TAlertKind): integer;
000079
000080   PROCEDURE BeepAlert(volume: integer);
000081
000082   FUNCTION ButnPushed(first, last: TButn; VAR pushed: TButn;
000083     pt: Point): boolean;
000084
000085   FUNCTION CautionAlert(alertFile: TAlertFile; alertNumber: integer): boolean;
000086
000087   FUNCTION CalcButWidth(topButn, botButn: TButn; VAR height: integer): integer;
000088
000089   PROCEDURE CountAlert(whichCounter: TCountAlert; countValue: integer);
000090
000091   PROCEDURE DrawAlert(alertFile: TAlertFile; alertNumber: integer;
000092     marginRect: Rect);
000093
000094   PROCEDURE DTAlert(alertFile: TAlertFile; osDT: longint;
000095     VAR userString: TParamAlert);
000096
000097   PROCEDURE EndWaitAlert;
000098
000099   PROCEDURE GetAlert(alertFile: TAlertFile; alertNumber: integer;
000100     pStr: TPstr255);
000101
000102   PROCEDURE GetButn(d: TButn; pStr: TPstr255);
000103
000104   PROCEDURE HideButn(d: TButn);
000105
000106   PROCEDURE InitAlerts(cacheSize, cacheBytes: integer; hz: THz; idleProcPtr,
000107     errProcPtr: procPtr);
000108
000109   PROCEDURE initMP(myHeap: Thz; toolPreFix: TParamAlert;
000110     ptrToAlertProc: procPtr; phraseversion: integer;
000111     myMenus: rMenuPtr; VAR alertFile: TAlertFile;
000112     VAR error: integer);
000113
000114   PROCEDURE LdSgAl;
000115
000116   PROCEDURE LocateAlert(top: integer);
000117
000118   PROCEDURE LockAlert(alertFile: TAlertFile; alertNumber: integer);
000119
000120   PROCEDURE NoteAlert(alertFile: TAlertFile; alertNumber: integer);
000121
000122   FUNCTION NthAlert(ordinal: INTEGER; alertFile: TAlertFile; VAR alertNumber,
000123     numStages: integer; VAR alertKind: TAlertKind): BOOLEAN;
000124
000125   PROCEDURE OpenPhraseFile(VAR refNum: integer; path: pathName);
000126
000127   PROCEDURE ParamAlert(cite1, cite2, cite3: Str255);
```

Apple Lisa Computer Technical Information

```
000128
000129     PROCEDURE PushButn(d: TButn; f: boolean);
000130
000131     FUNCTION ReadAlerts(refNum, version: integer): TAlertFile;
000132
000133     PROCEDURE ReshowButn(d: TButn);
000134
000135     PROCEDURE SetButWidths(width: integer);
000136
000137     PROCEDURE ShowButn(d: TButn; h, v: integer; fDefault: boolean);
000138
000139     PROCEDURE StopAlert(alertFile: TAlertFile; alertNumber: integer);
000140
000141     PROCEDURE UnlockAlerts;
000142
000143     PROCEDURE WaitAlert(alertFile: TAlertFile; alertNumber: integer);
000144
000145     PROCEDURE WouldAlert(VAR menu: menuInfo; itemIndex: integer);
000146
000147 *****
000148 *
000149 *             THAT'S ALL FOLKS ...
000150 *
000151 *****
000152
```

End of File -- Lines: 152 Characters: 4546

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 BLKIOINT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : BLKIOINT
000004 *
000005 *****
000006
000007 USES {$U+} BLKIOINT;
000008
000009
000010 INTRINSIC; {$%+}
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libos/syscall.obj } syscall,
000016         {$U libpl/UClascal.obj} UClascal;
000017
000018         {$SETC MONITOR := false}
000019         {$SETC DEBUG   := false}
000020         {$SETC FRANGECK := TRUE }
000021         {$SETC fSYMBOL  := TRUE }
000022
000023     CONST
000024         MMaxInt      = $7FFF;
000025         MaxUnit      = 20;
000026         MaxDir       = 77;
000027         MaxDir1     = 78;
000028         VIDLeng      = 7;
000029         TIDLeng      = 15;
000030         MaxSeg       = 15;
000031         FBlkSize     = 512;
000032         DirBlk       = 2;
000033         NameLen      = 23;
000034         FillrLn      = 11;
000035         SysUnit      = 5;
000036         DirSize      = 2048;
000037         DirNBlocks   = 4;
000038         unitScConsole = 1;
000039         unitScKeyboard = 2;
000040         unitWiConsole = 4;
000041         unitWiKeyboard = 5;
000042         unitScPrinter = 6;
000043         unitWiPrinter = 7;
000044         unitOSFile   = 10;
000045
000046     TYPE
000047         {$IFC MONITOR}
000048         IORsltWd      = (INoError, IBadBlock, IBadUnit, IBadMode, ITimeOut,
000049             ILostUnit, ILostFile, IBadTitle, INoRoom, INoUnit,
000050             INoFile, IDupFile, INotClosed, INotOpen, IBadFormat,
000051             IStrgOvfl, IWrProt, IDevError, INoDirSpace);
000052         {$ELSEC}
000053         IORsltWd      = (INoError);
000054         {$ENDC}
000055         DateRec       = PACKED RECORD
000056             Year: 0..100;
000057             Day: 0..31;
000058             Month: 0..12;
000059             END;
000060         UnitNum       = 0..MaxUnit;
000061         VID           = STRING[VIDLENG];
```

Apple Lisa Computer Technical Information

```

000062     DirRange      = 0..MaxDir;
000063     dirrang1         = 0..MaxDir1;
000064     TitleID          = STRING[TIDLENG];
000065     FullID            = STRING[NAMELEN];
000066     FileKind          = (UNTYPEDFILE, XDSKFILE, CODEFILE, TEXTFILE, INFOFILE,
000067     DATAFILE, GRAFFILE, FOTOFILE, SECUREDIR);
000068     DirEntry          = PACKED RECORD
000069         DFIRSTBLK: INTEGER;
000070         DLASTBLK: INTEGER;
000071         STATUS: BOOLEAN;
000072         FILLER1: 0..1024;
000073         CASE DFKind: FILEKIND OF
000074             SECUREDIR, UNTYPEDFILE:
000075             (DVID: VID;
000076              DEOVLK: INTEGER;
000077              DNUMFILES: INTEGER;
000078              DLOADTIME: INTEGER;
000079              DLASTBOOT: DATEREC);
000080             XDSKFILE, CODEFILE, TEXTFILE, INFOFILE, DATAFILE,
000081             GRAFFILE, FOTOFILE:
000082             (DTID: TitleID;
000083              DLASTBYTE: integer;
000084              CASE BOOLEAN OF
000085                  TRUE:
000086                      (DORefnum: integer; );
000087                  FALSE:
000088                      (DACCESS: DATEREC; ); )
000089         END;
000090     DirP               = ^Directory;
000091     Directory          = ARRAY [DirRange] OF DirEntry;
000092     CloseType          = (CNormal, CLock, CPurge, CCrunch);
000093     CharArray          = PACKED ARRAY [0..FBlkSize] OF CHAR;
000094     WindowP            = ^Window;
000095     Window             = CharArray;
000096     FileState          = (FJandW, FNeedChar, FGotChar);
000097     FIBP               = ^FIB;
000098     FVIDrec            = RECORD
000099         CASE BOOLEAN OF
000100             TRUE:
000101                 (FTrefnum: integer);
000102             FALSE:
000103                 (FVID: VID; );
000104         END;
000105     FIB                = RECORD
000106         FWindow: WindowP;
000107         FEOLN, FEOF: BOOLEAN;
000108         FIsOS: BOOLEAN;
000109         FState: FileState;
000110         FRecSize: INTEGER;
000111         CASE FIsOpen: BOOLEAN OF
000112             TRUE:
000113                 (FIsBlkd: BOOLEAN;
000114                  FNewFile: BOOLEAN;
000115                  FUnit: UnitNum;
000116                  FVIDstuff: FVIDrec;
000117                  FMaxBlk, FNxtBlk, FReptCnt: INTEGER;
000118                  HiByt3: 0..100;
000119                  FModified: BOOLEAN;
000120                  FHeader: DIRENTRY;
000121                  HiByt4: 0..100;
000122                  CASE FSoftBuf: BOOLEAN OF
000123                      TRUE:
000124                          (FMaxByte, FNxtByte: INTEGER;
000125                           HiByt5: BOOLEAN;
000126                           FBufChngd: BOOLEAN;
000127                           FBuffer: CharArray))

```


Apple Lisa Computer Technical Information

```

000128             END;
000129     {$IFC MONITOR}
000130     UTBLENTRY     = RECORD
000131                 UVID: VID;
000132                 LOGGED: BOOLEAN;
000133                 lockdir: boolean;
000134                 CASE UISBLKD: BOOLEAN OF
000135                     TRUE:
000136                         (UEOVBLK: INTEGER)
000137             END;
000138     {$ENDC}
000139     envirRec      = RECORD
000140                 A6_init: longint;
000141                 PC_2setup: longint;
000142                 PC_reinit: longint;
000143                 SP_init: longint;
000144             END;
000145     longaddr      = ^longint;
000146     Tstr255       = string[255];
000147     TTerminal     = SUBCLASS OF NIL
000148                 FUNCTION CREATE: TTerminal; ABSTRACT;
000149                 PROCEDURE VWrite(VAR str: Tstr255); ABSTRACT;
000150                 FUNCTION VRead: CHAR; ABSTRACT;
000151                 PROCEDURE VGoToxy(x, y: INTEGER); ABSTRACT;
000152                 FUNCTION VKeyPress: BOOLEAN; ABSTRACT;
000153                 FUNCTION VAbortKey: BOOLEAN; ABSTRACT;
000154                 PROCEDURE VScreenCtr(contrfun: INTEGER); ABSTRACT;
000155                 FUNCTION OpenPrinter: INTEGER; ABSTRACT;
000156                 FUNCTION PrWrite(VAR str: Tstr255): INTEGER; ABSTRACT;
000157                 FUNCTION ClosePrinter: INTEGER; ABSTRACT;
000158                 PROCEDURE CleanUp; ABSTRACT;
000159                 PROCEDURE SetupIOFile;
000160                 PROCEDURE InWindow(yesno: BOOLEAN);
000161             END;
000162
000163     VAR
000164     IORslt:       INTEGER;
000165     PMyid:        longint;
000166     PAbortKeyF:   boolean;
000167     PResProgram:  boolean;
000168     PBeginEnv:    envirRec;
000169     PTrLisaChar:  Boolean;
000170     PConsWindow:  BOOLEAN;
000171     PInWindow:    INTEGER;
000172     PInputfwindow: CHAR;
000173     PInputfile:   FILE;
000174     POutfwindow:  CHAR;
000175     POutputfile:  FILE;
000176     PasTerm:      TTerminal;
000177     TermMptr:     ^longint;
000178     Termdata:     RECORD
000179                 classptr: longint;
000180             END;
000181     haltaddress:  LONGINT;
000182     fpcb:         INTEGER;
000183     {$IFC MONITOR}
000184     gdirectory:  PACKED ARRAY [1..dirsize] OF CHAR;
000185     {$ENDC}
000186
000187     FUNCTION %_FIORESULT: INTEGER;
000188
000189     PROCEDURE %_Setiorslt(ior: integer);
000190
000191     PROCEDURE %_Backupinput(VAR f: text);
000192
000193     FUNCTION %%zenviron: longint;

```

Apple Lisa Computer Technical Information

```
000194
000195 *****
000196 *
000197 *           THAT'S ALL FOLKS ...
000198 *
000199 *****
000200
```

End of File -- Lines: 200 Characters: 7086

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 BLOCKIO.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : BLOCKIO
000004 *
000005 *****
000006
000007 USES {$U+} BLOCKIO;
000008
000009
000010 INTRINSIC SHARED; {$%+}
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libos/syscall.obj} syscall,
000016         {$U libos/psyscall.obj} psyscall,
000017         {$U libpl/blkiointr.obj} blkiointr;
000018
000019     CONST
000020         CclearScreen = 1;
000021         CclearEScreen = 2;
000022         CclearELine = 3;
000023         CgoHome = 11;
000024         CleftArrow = 12;
000025         CrightArrow = 13;
000026         CupArrow = 14;
000027         CdownArrow = 15;
000028
000029     TYPE
000030         consoledest = (alscreen, mainscreen, xsorocA, xsorocB, folder, spare1,
000031             spare2, spare3);
000032         dsProcCode = (dsResProg, dsSoftPwbtn, dsPrintDev, dsSetGPrefix,
000033             dsEnbDisk, dsGetDiskEnbF);
000034         dsProcParam = RECORD
000035             CASE ProcCode: dsProcCode OF
000036                 dsResProg:
000037                     (RProcessId: longint);
000038                 dsSoftPwbtn:
000039                     (SPButton: boolean);
000040                 dsPrintDev:
000041                     (PrDevice: e_name);
000042                 dsSetGPrefix:
000043                     (errnum: INTEGER;
000044                     prefix: pathname);
000045                 dsEnbDisk:
000046                     (toEnbDisk: boolean);
000047                 dsGetDiskEnbF:
000048                     (diskEnbF: BOOLEAN);
000049             END;
000050         pdsProcCode = (pdsSomething);
000051         pdsProcParam = RECORD
000052             CASE ProcCode: pdsProcCode OF
000053                 pdsSomething:
000054                     (something: longint);
000055             END;
000056
000057     PROCEDURE %_FINIT(VAR F: FIB; WINDOW: WINDOWP; recbytes: INTEGER);
000058
000059     PROCEDURE %_FOPEN(VAR F: FIB; VAR FTITLE: pathname; FOPENOLD: BOOLEAN;
000060         JUNK: longint);
000061
```

Apple Lisa Computer Technical Information

```
000062 PROCEDURE %_FCLOSE(VAR F: FIB; FTYPE: integer);
000063
000064 FUNCTION %_FBLOCKIO(VAR F: FIB; A: longint; NBLOCKS, RBLOCK: INTEGER;
000065 DOREAD: BOOLEAN): INTEGER;
000066
000067 PROCEDURE %_FGET(VAR F: FIB);
000068
000069 PROCEDURE %_FPUT(VAR F: FIB);
000070
000071 PROCEDURE %_WRITELN(VAR F: FIB);
000072
000073 PROCEDURE %_WRITECHAR(VAR F: FIB; charaddr: longaddr; count: integer);
000074
000075 PROCEDURE %_READLN(VAR F: FIB);
000076
000077 FUNCTION %_READCHAR(VAR F: FIB): CHAR;
000078
000079 PROCEDURE %_FSEEK(VAR F: FIB; RECORDNUM: integer);
000080
000081 PROCEDURE %_FGOTOXY(X, Y: INTEGER);
000082
000083 FUNCTION PAbortFlag: boolean;
000084
000085 PROCEDURE GetGPrefix(VAR prefix: pathname);
000086
000087 PROCEDURE ScreenCtr(contrfun: integer);
000088
000089 PROCEDURE GetPrDevice(VAR PrDevice: e_name);
000090
000091 PROCEDURE PTranLisaChar(toTranslate: boolean);
000092
000093 FUNCTION PaslibVersion: INTEGER;
000094
000095 PROCEDURE BlockIOInit;
000096
000097 PROCEDURE BlockIodisinit;
000098
000099 PROCEDURE lockPaslib(VAR errnum: integer);
000100
000101 PROCEDURE lockPasiolib(VAR errnum: integer);
000102
000103 PROCEDURE moveconsole(VAR errnum: integer; applconsole: consoledest);
000104
000105 PROCEDURE ExecReset(VAR errnum: INTEGER; VAR execfile: pathname;
000106 stopexec: BOOLEAN);
000107
000108 FUNCTION ExecFlag: BOOLEAN;
000109
000110 PROCEDURE OutputRedirect(VAR errnum: INTEGER; VAR outfile: pathname;
000111 stopoutput: BOOLEAN);
000112
000113 FUNCTION OutputRFlag: BOOLEAN;
000114
000115 PROCEDURE DSPaslibCall(VAR ProcParam: dsProcParam);
000116
000117 PROCEDURE %_PDSPaslibCall(VAR ProcParam: pdsProcParam);
000118
000119 PROCEDURE %_blockiosetup(firstttime: boolean; VAR regis: envirRec);
000120
000121 FUNCTION %_CkResident(VAR regis: envirRec): boolean;
000122
000123 FUNCTION %_pkeypress(VAR f: fib): boolean;
000124
000125 PROCEDURE %_funitiocall(unitno: integer; bufaddr: longint; nbytes: integer;
000126 blocknum: integer; mode: integer; unitcode: integer);
000127
```

Apple Lisa Computer Technical Information

```
000128     FUNCTION %_funitbusy(unitno: integer): boolean;
000129
000130     *****
000131     *
000132     *                               THAT'S ALL FOLKS ...
000133     *
000134     *****
000135
```

End of File -- Lines: 135 Characters: 3909

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 CZCOMPAC.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : CZCOMPAC
000004 *
000005 *****
000006
000007 USES { $U+ } CZCOMPAC;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $IFC OSBuilt }
000016             { $U libsm/unitstd.obj } unitstd,
000017             { $U libdb/dbenv.obj } dbenv,
000018             { $U libdb/dbdecl1.obj } dbdecl1,
000019             { $U libos/syscall.obj } syscall,
000020             { $U libos/psyscall.obj } psyscall,
000021             { $U libdb/lowlevel.obj } lowlevel,
000022             { $U libdb/pooler.obj } pooler,
000023             { $U libdb/heap.obj } heap;
000024     { $ELSEC }
000025         { $U obj:dbenv.obj } dbenv,
000026         { $U obj:dbdecl1.obj } dbdecl1,
000027         { $U obj:syscall.obj } syscall,
000028         { $U obj:lowlevel.obj } lowlevel,
000029         { $U obj:pooler.obj } pooler,
000030         { $U obj:heap.obj } heap;
000031     { $ENDC }
000032
000033     { $SETC debug := false }
000034
000035     Type
000036
000037
000038     states_de_compact = (no_try, first_try);
000039
000040     VAR
000041         pheap_segment: longint;
000042         heap_segment: integer;
000043         heapsize: longint;
000044         czone: zonedesc;
000045         compact_state: states_de_compact;
000046         { $IFC PRERELEASE }
000047         debug_compact: boolean;
000048         { $ENDC }
000049
000050     PROCEDURE heap_compact;
000051
000052     FUNCTION init_compact_state: states_de_compact;
000053
000054     FUNCTION retry_compact(VAR error_code: integer;
000055                             VAR local_state: states_de_compact): boolean;
000056
000057 *****
000058 *
000059 *             THAT'S ALL FOLKS ...
000060 *
000061 *****
```

Apple Lisa Computer Technical Information

000062

End of File -- Lines: 62 Characters: 1552

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 DBDECL1.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : DBDECL1
000004 *
000005 *****
000006
000007 USES { $U+ } DBDECL1;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013     { $SETC OSBUILT := TRUE }
000014
000015     USES
000016         { $IFC OSBUILT }
000017             { $U libsm/unitstd.obj } unitstd,
000018             { $U libdb/dbenv.obj } dbenv,
000019             { $U libos/syscall.obj } syscall,
000020             { $U libos/psyscall.obj } psyscall,
000021             { $U libin/intrlibp.obj } international;
000022     { $ELSEC }
000023         { $U OBJ:dbenv.obj } dbenv,
000024         { $U INTRLIB.obj } international;
000025     { $ENDC }
000026
000027     { $SETC PRERELEASE := FALSE }
000028     { $SETC DOLLARD := NOT fSymOk }
000029     { $SETC doTraceDB := TRUE }
000030     { $setc fTraceDB := doTraceDB AND fTRACE }
000031
000032     Const
000033
000034
000035         lt           = 0;
000036         gt           = 1;
000037         eq           = 2;
000038         wordsize     = 2;
000039         elemsize     = 16;
000040         halfelemsize = 8;
000041         maxrsize     = 1000;
000042         ascending    = 0;
000043         descending    = 1;
000044         ftypnull     = 0;
000045         ftypint1     = 1;
000046         ftypint2     = 2;
000047         ftypint4     = 3;
000048         ftypint8     = 4;
000049         ftypfptr     = 5;
000050         ftyptick     = 6;
000051         ftyppack2    = 7;
000052         ftypstr      = 8;
000053         ftypvstr     = 9;
000054         ftypvnull    = 10;
000055         ftypsvstr    = 11;
000056         ftypsvnull   = 12;
000057         ftypes       = 12;
000058         nullint1     = - 128;
000059         nullint2     = - 32768;
000060         nullint4     = - 2147483648;
000061         fnilval      = - 1;
```


Apple Lisa Computer Technical Information

```
000062      dbleft      = 0;
000063      dbright     = 1;
000064      dbcentered  = 2;
000065      dbwordwrap  = 3;
000066      dbnullok    = 4;
000067      dbdescend   = 5;
000068      dbisvisible = 6;
000069      none        = 0;
000070      ss          = 1;
000071      chk        = 2;
000072      zip        = 3;
000073      ph         = 4;
000074      dol        = 5;
000075      num        = 6;
000076      tim        = 7;
000077      dt         = 8;
000078      lets       = 9;
000079      tick       = 10;
000080      OEMtick    = 11;
000081      OEMFlets   = 12;
000082      OEMSVlets = 13;
000083      OEMFGen    = 14;
000084      OEMVGen    = 15;
000085      OEMSVGen   = 16;
000086      ecstrBnd  = 40;
000087
000088      TYPE
000089      short      = - 32767..32767;
000090      ptrinteger = ^integer;
000091      ptrlongint = ^longint;
000092      fileptr    = longint;
000093      ptrfileptr = ^fileptr;
000094      ticket     = RECORD
000095                  high: integer;
000096                  low: longint;
000097      END;
000098      ptrticket  = ^ticket;
000099      fstring    = PACKED ARRAY [0..0] OF char;
000100      ptrfstring = ^fstring;
000101      fint8      = RECORD
000102                  high: longint;
000103                  low: longint;
000104      END;
000105      ptrfint8   = ^fint8;
000106      vfld       = RECORD
000107                  size: integer;
000108                  offset: integer;
000109      END;
000110      ptrvfld    = ^vfld;
000111      svfld      = PACKED RECORD
000112                  offset: 0..1023;
000113                  size: 0..63;
000114      END;
000115      ptrsvfld   = ^svfld;
000116      bitrange   = 0..15;
000117      elemset    = SET OF bitrange;
000118      bits       = ARRAY [0..0] OF elemset;
000119      ptrbits    = ^bits;
000120      fieldrange = ftypnull..ftypes;
000121      ptr_field  = RECORD
000122                  CASE fieldrange OF
000123                      ftypnull, ftypint1, ftyppack2:
000124                          (ptr: ptrdata);
000125                      ftypint2:
000126                          (pint2: ptrinteger);
000127                      ftypint4:
```

Apple Lisa Computer Technical Information

```
000128             (pint4: ptrlongint);
000129             ftypint8:
000130             (pint8: ptrfint8);
000131             ftypfptr:
000132             (pfptr: ptrfileptr);
000133             ftyptick:
000134             (ptick: ptrticket);
000135             ftypstr:
000136             (pstr: ptrfstring);
000137             ftypvstr, ftypvnull:
000138             (pvfld: ptrvfld);
000139             ftypsvstr, ftypsvnull:
000140             (psvfld: ptrsvfld)
000141             END;
000142 bitoffset      = RECORD
000143             nbits: bitrange;
000144             sbit: bitrange;
000145             END;
000146 flddesc       = RECORD
000147             offset: integer;
000148             direction: byte;
000149             ftype: byte;
000150             CASE boolean OF
000151             false:
000152                 (size: integer);
000153             true:
000154                 (bo: bitoffset)
000155             END;
000156 recptr        = ARRAY [0..0] OF flddesc;
000157 ptrrecptr     = ^recptr;
000158 V0header      = RECORD
000159             space: integer;
000160             version: integer;
000161             nfields: short;
000162             nkeys: short;
000163             nsort: short;
000164             keybytes: short;
000165             recsize: integer;
000166             flags: elemset;
000167             END;
000168 header        = RECORD
000169             space: integer;
000170             version: integer;
000171             nfields: short;
000172             nkeys: short;
000173             nsort: short;
000174             keybytes: short;
000175             recsize: integer;
000176             flags: elemset;
000177             MarketCode: integer;
000178             END;
000179 checkinfo     = ARRAY [1..5] OF integer;
000180 ecstr         = string[ecstrBnd];
000181 ptrdbfield    = ^dbfield;
000182 dbfield       = RECORD
000183             fldsize: integer;
000184             name: short;
000185             display: short;
000186             highval: short;
000187             lowval: short;
000188             defval: short;
000189             editcheck: checkinfo;
000190             fldtype: integer;
000191             options: elemset;
000192             reptype: short;
000193             offset: integer;
```

Apple Lisa Computer Technical Information

```
000194             CASE boolean OF
000195                 false:
000196                     (size: integer);
000197                 true:
000198                     (bo: bitoffset);
000199             END;
000200     monthstr     = string[9];
000201     longstr      = string[255];
000202
000203     PROCEDURE move1g(p1, p2: ptrdata; nbytes: integer);
000204
000205     PROCEDURE moverg(p1, p2: ptrdata; nbytes: integer);
000206
000207     PROCEDURE movelf(p1, p2: ptrdata; nbytes: integer);
000208
000209     PROCEDURE moverf(p1, p2: ptrdata; nbytes: integer);
000210
000211     FUNCTION extract(pbyte: ptrdata; nbit, sbit: integer): integer;
000212
000213     PROCEDURE deposit(value: integer; pbyte: ptrdata; nbit, sbit: integer);
000214
000215     FUNCTION mult424(m4: fileptr; m2: integer): fileptr;
000216
000217     FUNCTION comp44(s1, s2: fileptr): integer;
000218
000219     FUNCTION comphand(VAR s1, s2: fint8): integer;
000220
000221     FUNCTION div424(m4: fileptr; m2: integer; VAR r2: integer): fileptr;
000222
000223     PROCEDURE setset(pset: ptrdata; lastelement, nelements: integer);
000224
000225     PROCEDURE clearset(pset: ptrdata; elements: integer);
000226
000227     PROCEDURE clearbit(pset: ptrdata; bit: integer);
000228
000229     FUNCTION inpset(pset: ptrdata; bit: integer): boolean;
000230
000231     PROCEDURE setbit(pset: ptrdata; bit: integer);
000232
000233     FUNCTION pintersect(pset1, pset2: ptrbits; setelements: integer): boolean;
000234
000235     FUNCTION compare(desc1, desc2: ptrrecptr; nfields: integer; prec1,
000236                   prec2: ptrdata): integer;
000237
000238     PROCEDURE extractkey(pdesc1, pdesc2: ptrrecptr; nflds: integer;
000239                   pentry: ptrdata; pwhere: ptrdata; vflds: integer);
000240
000241     FUNCTION findchanges(pdesc1: ptrrecptr; nfields: integer; pbefore,
000242                   pafter: ptrdata; pset: ptrbits): boolean;
000243
000244     PROCEDURE GetRepType(VAR result: integer; DataType: INTEGER;
000245                   CheckArray: CheckInfo; VAR RepType: Short;
000246                   VAR Size: INTEGER);
000247
000248     PROCEDURE ValidRepType(VAR result: integer; DataType: INTEGER;
000249                   CheckArray: CheckInfo; RepType: Short;
000250                   Size: INTEGER);
000251
000252     *****
000253     *
000254     *             THAT'S ALL FOLKS ...
000255     *
000256     *****
000257
```

End of File -- Lines: 257 Characters: 8002

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 DBENV.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : DBENV
000004 *
000005 *****
000006
000007 USES {$U+} DBENV;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     TYPE
000015     byte           = - 128..127;
000016     data           = ARRAY [0..0] OF byte;
000017     ptrdata        = ^data;
000018     hnddata        = ^ptrdata;
000019     intdata        = ARRAY [0..0] OF integer;
000020     ptrintdata     = ^intdata;
000021     hndintdata     = ^ptrintdata;
000022
000023 *****
000024 *
000025 *                 THAT'S ALL FOLKS ...
000026 *
000027 *****
000028
```

End of File -- Lines: 28 Characters: 684

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 EVENTS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : EVENTS
000004 *
000005 *****
000006
000007 USES { $U+ } EVENTS;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U libos/SysCall    } SysCall,
000016         { $U libos/PSysCall  } PSysCall,
000017         { $U libhw/HWInt     } HWInt,
000018         { $U libsm/UnitStd   } UnitStd,
000019         { $U libsm/UnitHz    } UnitHz,
000020         { $U libqd/Storage   } Storage,
000021         { $U libpm/PmDecl    } PmDecl,
000022         { $U libpm/PmM       } PmM,
000023         { $U libqd/QuickDraw } QuickDraw,
000024         { $U libfm/FontMgr   } FontMgr;
000025
000026         { $SetC wmOS        := TRUE   }
000027         { $SetC wmDebug    := FALSE  }
000028         { $SetC wmSymbols  := fSymOk }
000029         { $SetC wmJournal  := TRUE   }
000030         { $SetC wmMonkey   := FALSE  }
000031         { $SetC wmScrn2File := TRUE   }
000032         { $SETC doTraceWM  := TRUE   }
000033         { $SetC fTraceWM   := doTraceWM AND fTRACE }
000034
000035     CONST
000036         nilEvent      = 0;
000037         buttonDown    = 1;
000038         buttonUp      = 2;
000039         keyDown       = 3;
000040         folderActivate = 4;
000041         folderDeactivate = 5;
000042         folderUpdate  = 6;
000043         folderMoved   = 7;
000044         filerEvent    = 8;
000045         abortEvent    = 9;
000046         diedEvent     = 10;
000047         private1      = 11;
000048         private2      = 12;
000049         private3      = 13;
000050         diskEvent     = 14;
000051         catalogEvent  = 15;
000052         disk1InCode   = 1;
000053         disk1OutCode  = 2;
000054         disk2InCode   = 3;
000055         disk2OutCode  = 4;
000056         paraCode      = 5;
000057         buttonCode    = 6;
000058         mouseCode     = 7;
000059         powerCode     = 8;
000060         microInCode   = 11;
000061         micro1InCode  = 12;
```

Apple Lisa Computer Technical Information

```
000062     micro2InCode = 13;
000063     micro3InCode = 14;
000064     microOutCode = 15;
000065     micro1Outcode = 16;
000066     micro2OutCode = 17;
000067     micro3OutCode = 18;
000068     optionCode   = 104;
000069     shiftCode    = 126;
000070     commandCode  = 127;
000071     nilUserData  = 0;
000072     nilProcess   = 0;
000073     whyNot       = 0;
000074     whyClick     = 300;
000075     whyClose     = 301;
000076     whyDisk      = 302;
000077     whyDied      = 303;
000078     whyFind      = 304;
000079
000080     TYPE
000081     KeyCode      = 0..127;
000082     KeySet       = SET OF KeyCode;
000083     ProcessId    = LongInt;
000084     WindowPtr    = GrafPtr;
000085     EventHandle  = ^EventPtr;
000086     EventKind    = INTEGER;
000087     EventPtr     = ^EventRecord;
000088     EventRecord  = RECORD
000089         who: WindowPtr;
000090         what: EventKind;
000091         where: Point;
000092         when: LongInt;
000093         why: INTEGER;
000094         shiftKey: BOOLEAN;
000095         alphaKey: BOOLEAN;
000096         codeKey: BOOLEAN;
000097         appleKey: BOOLEAN;
000098         mouseKey: BOOLEAN;
000099         repeatKey: BOOLEAN;
000100         keyCap: KeyCode;
000101         ascii: CHAR;
000102         toProcess: ProcessId;
000103         fromProcess: ProcessId;
000104         fromFolder: WindowPtr;
000105         userData: LongInt;
000106     END;
000107
000108     VAR
000109     wmHeap:      Thz;
000110     deskPort:   GrafPtr;
000111     menuFolder: WindowPtr;
000112     alertFolder: WindowPtr;
000113     dialogFolder: WindowPtr;
000114     filerFolder: WindowPtr;
000115     scrapFolder: WindowPtr;
000116     activeFolder: WindowPtr;
000117     keyWindow:  WindowPtr;
000118     filerProcess: ProcessId;
000119     scrapProcess: ProcessId;
000120     activeProcess: ProcessId;
000121     keyProcess: ProcessId;
000122     clickDelay: LongInt;
000123     caretOnTime: LongInt;
000124     caretOffTime: LongInt;
000125     eventDebug: BOOLEAN;
000126     trapKeys:   KeySet;
000127     keyTrapProc: ProcPtr;
```

Apple Lisa Computer Technical Information

```
000128     trapPrCs:      ProcessId;
000129     _CA_StartFlag:  BOOLEAN;
000130     queueLocked:   BOOLEAN;
000131     recordFlag:    BOOLEAN;
000132     playFlag:      BOOLEAN;
000133     monkeyFlag:    BOOLEAN;
000134     monkeyWorld:   BOOLEAN;
000135     wmspare1:      LONGINT;
000136     wmspare2:      LONGINT;
000137     wmspare3:      LONGINT;
000138     wmspare4:      LONGINT;
000139     wmspare5:      LONGINT;
000140
000141     FUNCTION Abort: BOOLEAN;
000142
000143     FUNCTION Button: BOOLEAN;
000144
000145     PROCEDURE CheckEvents(updateOK: BOOLEAN);
000146
000147     PROCEDURE DeleteEvent(evHandle: EventHandle);
000148
000149     FUNCTION ElapsTime: LongInt;
000150
000151     FUNCTION EventAvail: BOOLEAN;
000152
000153     FUNCTION FirstEvent: EventHandle;
000154
000155     PROCEDURE GetEvent(VAR event: EventRecord);
000156
000157     PROCEDURE GetEvFrom(fromProcess: ProcessID; VAR event: EventRecord);
000158
000159     PROCEDURE GetKeys(VAR keys: KeySet);
000160
000161     PROCEDURE GetMouse(VAR pt: Point);
000162
000163     FUNCTION ImActive: BOOLEAN;
000164
000165     PROCEDURE InitEvents;
000166
000167     PROCEDURE LetOthersRun;
000168
000169     FUNCTION NextEvent(evHandle: EventHandle): EventHandle;
000170
000171     FUNCTION PeekEvent(VAR event: EventRecord): BOOLEAN;
000172
000173     PROCEDURE PrintScreen;
000174
000175     PROCEDURE PushEvent(VAR event: EventRecord);
000176
000177     PROCEDURE SendEvent(VAR event: EventRecord; toPrCs: ProcessId);
000178
000179     FUNCTION StillDown: BOOLEAN;
000180
000181     FUNCTION Time: LongInt;
000182
000183     PROCEDURE WakeUpHead;
000184     { $IFC wmJournal }
000185
000186     PROCEDURE StartPlayback(fileName: PathName; journalMode: INTEGER);
000187
000188     PROCEDURE StartRecording(fileName: PathName; journalMode: INTEGER);
000189
000190     PROCEDURE StopPlayback;
000191
000192     PROCEDURE StopRecording;
000193     { $ENDC }
```

Apple Lisa Computer Technical Information

```
000194
000195 *****
000196 *
000197 *           THAT'S ALL FOLKS ...
000198 *
000199 *****
000200
```

End of File -- Lines: 200 Characters: 5104

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FEDEC.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FEDEC
000004 *
000005 *****
000006
000007 USES {$U+} FEDEC;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/unitstd.obj} UnitStd,
000016         {$U libsm/unithz.obj} UnitHz,
000017         {$U libqd/QuickDraw} QuickDraw,
000018         {$U libfm/FontMgr} FontMgr,
000019         {$U libqd/Storage} Storage,
000020         {$U libdb/dbenv} dbenv;
000021         {$SETC doTraceFE := TRUE}
000022         {$SetC fTraceFE := doTraceFE AND fTRACE}
000023
000024     CONST
000025         fldFull      = 1;
000026         fldProtected = 2;
000027         nullSelect   = 3;
000028         invScrapType = 4;
000029         invOpType    = 5;
000030         noRoomForRuns = 6;
000031         invalidEdit  = 7;
000032         FEallocationFailed = 8;
000033         FENotUndoable = 9;
000034         left         = 1;
000035         right        = 2;
000036         center       = 3;
000037         setNormal    = 1;
000038         setUnderline = 2;
000039         setBold      = 3;
000040         setItalics   = 4;
000041         setOutline   = 5;
000042         setShadow    = 6;
000043         setFont      = 7;
000044         setToModern   = 8;
000045         setToClassic = 9;
000046         noOp         = 0;
000047         cutOp        = 1;
000048         copyOp       = 2;
000049         pasteOp      = 3;
000050         chInputOp    = 4;
000051         clearOp      = 5;
000052         formatOp     = 6;
000053         valueToFormulaOp = 7;
000054         formulaToValueOp = 8;
000055         MaxFieldLngh = 1000;
000056
000057     TYPE
000058         Tstr          = string[50];
000059         Ttriple       = ^Tstr;
000060         interval      = RECORD
000061             lpFst: integer;
```

Apple Lisa Computer Technical Information

```
000062         lpLim: integer;
000063         END;
000064     TtySel     = (tySelPt, tySelWd, tySelPar);
000065     TtyHiligh  = (tyHiNil, tyHiInvert, tyHiCaret, tyHiDimInvert,
000066                 tyHiDimCaret);
000067     txSel     = RECORD
000068         int: interval;
000069         tySel: TtySel;
000070         tyHiligh: TtyHiligh;
000071     END;
000072     hndRuns   = ^ptrRuns;
000073     ptrRuns   = ^runs;
000074     run       = RECORD
000075         lpFst: integer;
000076         font: integer;
000077         face: Style;
000078     END;
000079     runs      = ARRAY [0..0] OF run;
000080     field     = RECORD
000081         coords: Rect;
000082         maxLen: integer;
000083         growLen: integer;
000084         curLen: integer;
000085         align: QDByte;
000086         drawPad: QDByte;
000087         curValue: hndData;
000088         maxFmts: integer;
000089         growFmts: integer;
000090         curFmts: integer;
000091         fmtInfo: hndRuns;
000092         protect: boolean;
000093     END;
000094     ptrField  = ^field;
000095     hndField  = ^ptrField;
000096     fieldState = RECORD
000097         select: txSel;
000098         anchor: txSel;
000099         selectOn: boolean;
000100         visCaret: boolean;
000101         selectRect: Rect;
000102         bsPtr: integer;
000103         space: interval;
000104         valid: boolean;
000105         validLp: integer;
000106         newSel: boolean;
000107         changed: boolean;
000108     END;
000109     ptrFState = ^fieldState;
000110     hndFState = ^ptrFState;
000111     undoInfo  = RECORD
000112         curField: hndField;
000113         curFS: hndFState;
000114         oldSelFld: hndField;
000115         oldSelFS: hndFState;
000116         select: txSel;
000117         endselect: txSel;
000118         anchor: txSel;
000119         endanchor: txSel;
000120         oldValFld: hndField;
000121         oldValFS: hndFState;
000122         curLen: integer;
000123         curValue: hndData;
000124         curFmts: integer;
000125         fmtInfo: hndRuns;
000126         lastOp: integer;
000127     END;
```

Apple Lisa Computer Technical Information

```
000128 ptrUndoInfo = ^undoInfo;
000129 hndUndoInfo = ^ptrUndoInfo;
000130 HndScrap = ^ptrScrap;
000131 PtrScrap = ^ScrapRec;
000132 ScrapRec = RECORD
000133     Scrapfld: Field;
000134     isWord: boolean;
000135     END;
000136
000137 *****
000138 *
000139 *           THAT'S ALL FOLKS ...
000140 *
000141 *****
000142
```

End of File -- Lines: 142 Characters: 4377

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FIELDEDIT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FIELDEDI
000004 *
000005 *****
000006
000007 USES { $U+ } FIELDEDI;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U libsm/unitStd } UnitStd,
000016         { $U libsm/unitHz } UnitHz,
000017         { $U libqd/Storage } Storage,
000018         { $U libqd/QuickDraw } QuickDraw,
000019         { $U libfm/FontMgr } FontMgr,
000020         { $U libos/SysCall } SysCall,
000021         { $U libwm/Events } Events,
000022         { $U libwm/Folders } Folders,
000023         { $U libsb/WMLStd } WMLStd,
000024         { $U libsb/WMLsB } WMLsB,
000025         { $U libdb/dbenv } dbenv,
000026         { $U libfe/FEDec } FEDec,
000027         { $U libsu/Scrap } Scrap,
000028         { $U libsu/UnitFmt } UnitFmt,
000029         { $U libpm/PmDecl } PmDecl,
000030         { $U libpr/PrStdInfo } PrStdInfo,
000031         { $U libsu/UnitCs } UnitCs,
000032         { $U libfe/Fldut } FldUT;
000033         { $SETC FLDDEBUG := FDBGOK }
000034         { $SETC FLDSYMBOLS := FSYMOK }
000035
000036     CONST
000037         MaxFontNumber = 24;
000038         { $IFC NOT FLDDEBUG }
000039         fldTest      = false;
000040         { $ENDC }
000041
000042     VAR
000043         { $IFC FLDDEBUG }
000044         fldTest:      boolean;
000045         { $ENDC }
000046         ToModern, ToClassic: ARRAY [0..MaxFontNumber] OF - 128..127;
000047         ScrapHnd:     HndScrap;
000048         fldHeap:      THz;
000049         fmtSize:      integer;
000050         uInfo:        undoInfo;
000051         SubFont:      Boolean;
000052         SubDev:       Integer;
000053
000054     PROCEDURE FEInit(hz: THz; passtrpl: Ttriple; VAR errnum: integer);
000055
000056     PROCEDURE FETerminate;
000057     { $IFC FLDDEBUG }
000058
000059     PROCEDURE SetFldTest(testOn: boolean);
000060
000061     PROCEDURE DumpFldInfo(hf: hndField; hfs: hndFState);
```

Apple Lisa Computer Technical Information

```
000062    {$ENDC}
000063
000064    PROCEDURE LdFEseg;
000065
000066    PROCEDURE UseSubstituteFont(Substitute: Boolean; DevtoSub: Integer);
000067
000068    PROCEDURE SetFCoords(hf: hndField; hfs: hndFState; c: Rect);
000069
000070    PROCEDURE SetFAlign(hf: hndField; hfs: hndFState; a: byte);
000071
000072    PROCEDURE SetFPad(hf: hndField; hfs: hndFState; pad: byte);
000073
000074    PROCEDURE SetFFProtect(hf: hndField; hfs: hndFState; p: boolean);
000075
000076    PROCEDURE InitFState(hf: hndField; hfs: hndFState);
000077
000078    FUNCTION CreateField(c: Rect; iLen, gLen: integer; a: byte; pad: byte; iFmts,
000079                      gFmts: integer; fontNum: integer; p: boolean): hndField;
000080
000081    PROCEDURE RemoveField(hf: hndField);
000082
000083    FUNCTION FmtFromLp(hf: hndField; hfs: hndFState; lp: integer): integer;
000084
000085    PROCEDURE FmtFromInt(hf: hndField; hfs: HndFState; Int: interval;
000086                       VAR Plain: boolean; VAR result: run);
000087
000088    PROCEDURE SetRunInterval(hf: hndField; hfs: hndFState; fontNum: integer;
000089                          format: integer; int: interval;
000090                          VAR errNum: integer);
000091
000092    PROCEDURE SetRunFormat(hf: hndField; hfs: hndFState; fontNum: integer;
000093                        format: integer; VAR errNum: integer);
000094
000095    PROCEDURE UpdateRun(hf: hndField; hfs: hndFState; fontNum: integer;
000096                     format: integer; VAR errNum: integer);
000097
000098    PROCEDURE VertCoords(hf: hndField; VAR ascent, descent: integer);
000099
000100    PROCEDURE MoveField(hf: hndField; hfs: hndFState; dx: Point);
000101
000102    PROCEDURE DrawField(hf: hndField; hfs: hndFState; showSelect,
000103                    erase: boolean);
000104
000105    PROCEDURE DrawFldAt(hf: hndField; hfs: hndFState; dx: Point; showSelect,
000106                    erase: boolean);
000107
000108    PROCEDURE SetSel(hf: hndField; hfs: hndFState; lpFst: integer;
000109                  lpLim: integer; selType: TtySel);
000110
000111    PROCEDURE SelectAll(hf: hndField; hfs: hndFState; VAR t: integer);
000112
000113    PROCEDURE ExitField(hf: hndField; hfs: hndFState);
000114
000115    PROCEDURE CompactField(hf: hndField);
000116
000117    PROCEDURE ErrorSelect(hf: hndField; hfs: hndFState; lpFst: integer;
000118                      lpLim: integer; VAR t: integer);
000119
000120    PROCEDURE HilightSel(hf: hndField; hfs: hndFState; fshow: boolean);
000121
000122    PROCEDURE ChangeSelHilight(hf: hndField; hfs: hndFState; toDim: boolean);
000123
000124    FUNCTION LpFromX(x: integer; hf: hndField; hfs: hndFState;
000125                  seltype: ttysel): integer;
000126
000127    PROCEDURE LwFromLp(orig: integer; hf: hndField; hfs: hndFState;
```

Apple Lisa Computer Technical Information

```
000128             VAR lWord: interval; VAR FoundWord: boolean);
000129
000130 FUNCTION CompressedLp(lp: integer; hf: hndField; hfs: hndFState;
000131             VAR valid: boolean): integer;
000132
000133 FUNCTION ExpandedLp(lp: integer; hf: hndField; hfs: hndFState): integer;
000134
000135 PROCEDURE RepSelect(lp: integer; hf: hndField; hfs: hndFState);
000136
000137 PROCEDURE Select(dxy: Point; hf: hndField; hfs: hndFState; VAR n: Rect;
000138             VAR t: integer);
000139
000140 PROCEDURE RepWordSelect(int: interval; hf: hndField; hfs: hndFState);
000141
000142 PROCEDURE SelectWord(dxy: Point; hf: hndField; hfs: hndFState; VAR n: rect;
000143             VAR t: integer);
000144
000145 PROCEDURE RepGrowSel(chlp: integer; hf: hndField; hfs: hndFState);
000146
000147 PROCEDURE GrowSel(dxy: Point; hf: hndField; hfs: hndFState; VAR n: Rect);
000148
000149 PROCEDURE CaretErased(hf: hndField; hfs: hndFState);
000150
000151 PROCEDURE StartBlink(VAR t: integer);
000152
000153 PROCEDURE BlinkCaret(hf: hndField; hfs: hndFState; VAR t: integer);
000154
000155 PROCEDURE CaretOn(hf: hndField; hfs: hndFState);
000156
000157 PROCEDURE InsCh(ch: char; hf: hndField; hfs: hndFState; VAR errNum: integer);
000158
000159 PROCEDURE Backspace(hf: hndField; hfs: hndFState; VAR errNum: integer);
000160
000161 PROCEDURE ForwardSpace(hf: hndField; hfs: hndFState; VAR errNum: integer);
000162
000163 PROCEDURE BackWord(hf: hndField; hfs: hndFState; VAR errNum: integer);
000164
000165 PROCEDURE ForwardWord(hf: hndField; hfs: hndFState; VAR errNum: integer);
000166
000167 PROCEDURE DrawScrap(VAR errnum: integer);
000168
000169 PROCEDURE CutCopyField(hf: hndField; hfs: hndFState; fCut: boolean;
000170             fIntoScrap: boolean; VAR errNum: integer);
000171
000172 PROCEDURE PasteField(hf: hndField; hfs: hndFState; VAR errNum: integer;
000173             pasteFmt: boolean; PasteLimit: integer);
000174
000175 PROCEDURE UndoEdit(hf: hndField; hfs: hndFState; VAR errNum: integer);
000176
000177 PROCEDURE ClearField(hf: hndField; hfs: hndFState; VAR errNum: integer);
000178
000179 FUNCTION qualifies(lp: integer; hf: hndfield; hfs: hndfstate): boolean;
000180
000181 *****
000182 *
000183 *             THAT'S ALL FOLKS ...
000184 *
000185 *****
000186
```

End of File -- Lines: 186 Characters: 5951

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FILEIO.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FILEIO
000004 *
000005 *****
000006
000007 USES { $U+ } FILEIO;
000008
000009
000010 { $IFC isIntrin } INTRINSIC; { $ENDC }
000011
000012 INTERFACE
000013
000014     USES
000015         { $IFC SrcOnOS }
000016         ( *$U OBJ/PASDEFS.OBJ * ) PasDefs,
000017         ( *$U OBJ/MEMMAN.OBJ * ) Memman;
000018     { $ELSEC }
000019         ( *$U OBJ/PASDEFS.OBJ * ) PasDefs,
000020         ( *$U OBJ/MEMMAN.OBJ * ) Memman;
000021     { $ENDC }
000022
000023     CONST
000024
000025
000026         MaxBuffTop    = 32767;
000027         BuffBlock     = 512;
000028         Null          = '';
000029         CtrlC         = 3;
000030         Left          = 8;
000031         Down          = 10;
000032         Up            = 11;
000033         Right         = 12;
000034         CR            = 13;
000035         Esc           = 27;
000036
000037     TYPE
000038         Byte           = 0..255;
000039         SetOfChar     = SET OF Char;
000040         Ptr            = ^Integer;
000041         FileByte      = - 128..127;
000042         Buffer          = ARRAY [0..MaxBuffTop] OF FileByte;
000043         BuffPtr        = ^Buffer;
000044         FileHandle     = ^FileDesc;
000045         FileDesc       = RECORD
000046             Info: FILE;
000047             OldFile: boolean;
000048             NewLastBlk: boolean;
000049             Buff: BuffPtr;
000050             MaxBlks, CurBlks: Integer;
000051             BlockNr0: Integer;
000052             BuffTop, BuffIdx: Integer;
000053             BuffChd, ReadEOF: boolean;
000054             LastBlk: Integer;
000055         END;
000056
000057     PROCEDURE InitFile(VAR FilePtr: FileHandle; nBlocks: Integer);
000058
000059     FUNCTION IAnd(I, J: Integer): Integer;
000060
000061     FUNCTION IOr(I, J: Integer): Integer;
```

Apple Lisa Computer Technical Information

```
000062
000063     FUNCTION IXOr(I, J: Integer): Integer;
000064
000065     FUNCTION INot(I: Integer): Integer;
000066
000067     FUNCTION PeekB(P: Ptr; SkipB: Integer): Byte;
000068
000069     PROCEDURE PokeB(P: Ptr; SkipB: Integer; B: Byte);
000070
000071     PROCEDURE ToHex(P: Ptr; SkipN, NrN: Integer; VAR S: LString);
000072
000073     PROCEDURE WriteHex(VAR F: Text; P: Ptr; SkipN, NrN: Integer);
000074
000075     PROCEDURE OpenFile(VAR FilePtr: FileHandle; FileName: LString;
000076                       NewFile: boolean);
000077
000078     PROCEDURE ZeroFileEnd(FilePtr: FileHandle);
000079
000080     PROCEDURE CloseFile(FilePtr: FileHandle; Save: boolean);
000081
000082     PROCEDURE GetFilePtr(FilePtr: FileHandle; VAR BytePtr: LongInt);
000083
000084     PROCEDURE SetFilePtr(FilePtr: FileHandle; BytePtr: LongInt);
000085
000086     PROCEDURE SkipBytes(FilePtr: FileHandle; NrBytes: LongInt);
000087
000088     FUNCTION FileEOF(FilePtr: FileHandle): boolean;
000089
000090     PROCEDURE CopySeq(InFile, OutFile: FileHandle; NrBytes: LongInt);
000091
000092     PROCEDURE GetSeq(FilePtr: FileHandle; Stuff: Ptr; NrBytes: LongInt);
000093
000094     PROCEDURE GetByte(FilePtr: FileHandle; VAR B: Byte);
000095
000096     PROCEDURE GetWord(FilePtr: FileHandle; VAR W: Integer);
000097
000098     PROCEDURE GetLong(FilePtr: FileHandle; VAR L: LongInt);
000099
000100     PROCEDURE PutSeq(FilePtr: FileHandle; Stuff: Ptr; NrBytes: LongInt);
000101
000102     PROCEDURE PutByte(FilePtr: FileHandle; B: Byte);
000103
000104     PROCEDURE PutWord(FilePtr: FileHandle; W: Integer);
000105
000106     PROCEDURE PutLong(FilePtr: FileHandle; L: LongInt);
000107
000108     *****
000109     *
000110     *           THAT'S ALL FOLKS ...
000111     *
000112     *****
000113
```

End of File -- Lines: 113 Characters: 3042

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FILERCOM.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FILERCOM
000004 *
000005 *****
000006
000007 USES {$U+} FILERCOM;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libos/SysCall      } SysCall,
000016         {$U libos/PSysCall    } PSysCall,
000017         {$U libsm/UnitStd     } UnitStd,
000018         {$U libsm/UnitHz      } UnitHz,
000019         {$U libqd/Storage     } Storage,
000020         {$U libqd/QuickDraw   } QuickDraw,
000021         {$U libqd/GrafUtil    } GrafUtil,
000022         {$U libsb/WmlStd      } WmlStd,
000023         {$U libsb/WmlSb       } WmlSb,
000024         {$U libfm/FontMgr     } FontMgr,
000025         {$U libwm/Events      } Events,
000026         {$U libwm/windows     } Windows,
000027         {$U libwm/folders     } Folders,
000028         {$U libsu/Scrap       } Scrap;
000029         {$SETC fcDebug = fDbgOk }
000030         {$SETC fcSymbols = fSymOk }
000031
000032     CONST
000033         scrapRef      = 8;
000034         cmAppInfo     = 1268;
000035         cmIconInfo    = 1269;
000036         nameSeparator = '<';
000037         fceNoErrors   = 0;
000038         fceAborted    = 4033;
000039         fceBadEventType = 4025;
000040         fceBadReason  = 4026;
000041         fceCantRead   = 4027;
000042         fceCantWrite  = 4028;
000043         fceInUse      = 4029;
000044         fceNoMemory   = 4030;
000045         fceOutOfDiskSpace = 4031;
000046         fceBadLDSN    = 4032;
000047         fceBadPassword = 4033;
000048         fceDuplicateName = 890;
000049         fceInvalidParam = 971;
000050         fceNameNotFound = 972;
000051         cmdClose      = 1001;
000052         cmdClosAll    = 1002;
000053         deskTool      = 0;
000054         writeTool     = 1;
000055         graphTool     = 2;
000056         matrixTool    = 3;
000057         drawTool      = 4;
000058         listTool      = 5;
000059         ballsTool     = 6;
000060         manualTool    = 7;
000061         projectTool   = 8;
```

Apple Lisa Computer Technical Information

```
000062      termEmTool      = 10;
000063      configTool      = 11;
000064      calcTool        = 12;
000065      clockTool       = 13;
000066      MwriteTool      = 14;
000067      MpaintTool      = 15;
000068      MdrawTool       = 16;
000069      maxKnownTool    = 16;
000070      iconWidth       = 48;
000071      iconHt          = 32;
000072      nilKind         = 0;
000073      fileKind        = 1;
000074      deskKind        = 2;
000075      folderkind      = 3;
000076      docKind         = 4;
000077      docPad          = 5;
000078      trashKind       = 6;
000079      printKind       = 7;
000080      calcKind        = 8;
000081      trayKind        = 9;
000082      computerKind    = 10;
000083      comp2Kind       = 11;
000084      inBox1Kind      = 12;
000085      inBox2Kind      = 13;
000086      outBox1Kind     = 14;
000087      outBox2Kind     = 15;
000088      folderPad       = 16;
000089      clipBdKind     = 17;
000090      clockKind       = 19;
000091      letterKind      = 20;
000092      letterPad       = 21;
000093      toolKind        = 24;
000094      diskKind        = 25;
000095      disk1Kind       = 26;
000096      disk2Kind       = 27;
000097      drawerKind      = 28;
000098      profileKind     = 29;
000099      priamKind       = 30;
000100      slotKind        = 31;
000101      slot1Kind       = 32;
000102      slot2Kind       = 33;
000103      slot3Kind       = 34;
000104      MdocKind        = 35;
000105      MdiskKind       = 36;
000106      lastKind        = 36;
000107
000108      TYPE
000109      FilingCmd      = LONGINT;
000110      FilerOp        = (fcClose, fcCopy, fcDfClose, fcNone, fcPrint, fcPut,
000111                      fcResume, fcShred, fcSuspend, fcTerminate,
000112                      fcNameToPrefix, fcPrefixToName, fcDupIcon, fcMakeTool);
000113      FReply         = (dfClosed, dfNotClosed, docClosd, docNotClosed, docXfered,
000114                      docNotXfered, InitFailed);
000115      FReason        = (allOK, badData, cantConvert, cantRead, cantWrite,
000116                      dirtyDoc, internalError, needToConvert, newerDoc,
000117                      noDiskSpace, noMemory, noMoreDocs, okButNoMore,
000118                      docPutBack, aUserAbort, wrongPassword);
000119      FilerExt       = RECORD
000120                      theFlrOp: FilerOp;
000121                      theErr: INTEGER;
000122                      theOffset: INTEGER;
000123                      theDf: INTEGER;
000124                      thePassword: E_Name;
000125                      thePrefix: Pathname;
000126                      theResult: Pathname;
000127      END;
```

Apple Lisa Computer Technical Information

```
000128     FCopyOp      = (fcDocCopy, fcDocMove, fcDocBackup);
000129     ObjectKind    = nilKind..lastKind;
000130     TIconRef      = RECORD
000131                 kind: ObjectKind;
000132                 toolID: LONGINT;
000133                 userName: STRING[80];
000134                 password: E_Name;
000135                 diskName: Pathname;
000136             END;
000137     PIconRef      = ^TIconRef;
000138     HIconRef      = ^PIconRef;
000139     TAppInfo      = RECORD
000140                 window: WindowPtr;
000141                 proc: INTEGER;
000142             END;
000143     PAppInfo      = ^TAppInfo;
000144     HAppInfo      = ^PAppInfo;
000145     hFilerExt     = ^pFilerExt;
000146     pFilerExt     = ^FilerExt;
000147     ReplyPtr      = ^Reply;
000148     Reply         = RECORD
000149                 theReply: FReply;
000150                 theReason: FReason;
000151             END;
000152
000153     VAR
000154     iconData:     ARRAY [ObjectKind] OF CHAR;
000155     iconMask:     ARRAY [ObjectKind] OF CHAR;
000156     iconOverlay:  ARRAY [deskTool..maxKnownTool] OF CHAR;
000157     iconBoxes:   ARRAY [ObjectKind] OF Rect;
000158     tinyData:    ARRAY [ObjectKind] OF CHAR;
000159     tinyMask:    ARRAY [ObjectKind] OF CHAR;
000160     tinyOverlay: ARRAY [deskTool..maxKnownTool] OF CHAR;
000161
000162     PROCEDURE CopyDoc(VAR error: INTEGER; fromPrefix, toPrefix: Pathname;
000163                     useLdsn: INTEGER; theOp: FCopyOp; VAR docSize: LONGINT);
000164
000165     PROCEDURE DoFilingCmd(whichCmd: FilingCmd);
000166
000167     PROCEDURE GetAddParms(VAR error: INTEGER; theEvent: EventRecord;
000168                          VAR theFilerExt: FilerExt);
000169
000170     PROCEDURE TellFiler(VAR error: INTEGER; what: FReply; why: FReason;
000171                        myFolder: WindowPtr);
000172
000173     PROCEDURE NameToPrefix(VAR err, offset: INTEGER; myFolder: WindowPtr;
000174                          name: Pathname; VAR prefix: Pathname);
000175
000176     PROCEDURE PrefixToName(VAR err, offset: INTEGER; myFolder: WindowPtr;
000177                          prefix: Pathname; VAR name: Pathname);
000178
000179     PROCEDURE StartIconRef(VAR err: INTEGER; myWindow: WindowPtr);
000180
000181     PROCEDURE AddIconRef(iconInfo: TIconRef);
000182
000183     PROCEDURE EndIconRef;
000184
000185     PROCEDURE DrawRefScrap(VAR err: INTEGER);
000186
000187     PROCEDURE ReadIconRefs(VAR err: INTEGER; VAR proc: ProcessID;
000188                          VAR window: WindowPtr; VAR identity: TiconRef;
000189                          IDProc: TProc);
000190
000191     PROCEDURE CopyDiskfile(VAR err: INTEGER; source, destination: Pathname;
000192                          bufrAdrs, bufrSize: LONGINT; theOp: FCopyOp;
000193                          VAR osErr: INTEGER);
```

Apple Lisa Computer Technical Information

```
000194
000195     PROCEDURE InitFC;
000196
000197 *****
000198 *
000199 *                THAT'S ALL FOLKS ...
000200 *
000201 *****
000202
```

End of File -- Lines: 202 Characters: 6379

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FLDUT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FLDUT
000004 *
000005 *****
000006
000007 USES {$U+} FLDUT;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/unitstd.obj} UnitStd,
000016         {$U libsm/unithz.obj} UnitHz,
000017         {$U libqd/QuickDraw} QuickDraw,
000018         {$U libfm/FontMgr} FontMgr,
000019         {$U libqd/Storage} Storage,
000020         {$U libos/SysCall} SysCall,
000021         {$U libwm/Events} Events,
000022         {$U libdb/dbenv} dbenv,
000023         {$U libfe/FEDec} FEDec,
000024         {$U libpm/pmdecl} pmdecl,
000025         {$U libpr/PrStdInfo} PrStdInfo,
000026         {$U libsu/unitfmt.obj} Unitfmt,
000027         {$U libsu/unitcs.obj} UnitCs,
000028         {$U libsu/scrap.obj} Scrap;
000029         {$SETC FLDUTDBG := FDBGOK}
000030
000031     TYPE
000032         PFldUT      = ^TFldUT;
000033         TFldUT      = RECORD
000034             cs: Tcs;
000035             UTfield: hndfield;
000036         END;
000037
000038     FUNCTION CreateFldUt(StreamHeap: THz): TB;
000039
000040     PROCEDURE SeqLpdFld(Lpd: TLpd; VAR achad: Tachad);
000041     {$IFC FLDUTDBG}
000042
000043     PROCEDURE PxFldUT(hcs: thcs);
000044     {$ENDC}
000045
000046 *****
000047 *
000048 *                 THAT'S ALL FOLKS ...
000049 *
000050 *****
000051
```

End of File -- Lines: 51 Characters: 1275

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FMGRUTIL.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FMGRUTIL
000004 *
000005 *****
000006
000007 USES {$U+} FMGRUTIL;
000008
000009
000010 {$SETC fmOS := TRUE}
000011
000012 INTRINSIC;
000013
000014 INTERFACE
000015
000016     USES
000017         {$U libsm/UnitStd} UnitStd,
000018         {$U libsm/UnitHz} UnitHz,
000019         {$U libos/SysCall} SysCall,
000020         {$U libqd/QuickDraw} QuickDraw,
000021         {$U libfm/FontMgr} FontMgr,
000022         {$U libqd/Storage} Storage;
000023         {$SETC fmdebug := fdbgok}
000024         {$SETC fmSymbols := fsymok}
000025
000026     FUNCTION GetIcon(fam: Tfam; cc: TCc; VAR rSrc: Rect; VAR bmSrc: BitMap;
000027                     VAR cError: TC): TF;
000028
000029     PROCEDURE DrawIcon(fam: Tfam; rDst: Rect; cc: TCc; cmode: TC;
000030                       VAR cError: TC);
000031
000032     PROCEDURE GetPattern(fam: Tfam; ccPat: TCc; VAR pat: Pattern;
000033                          VAR cError: TC);
000034
000035     PROCEDURE GetCursor(fam: Tfam; ccData, ccMask: TCc; VAR curse: Cursor;
000036                          VAR cError: TC);
000037
000038 *****
000039 *
000040 *             THAT'S ALL FOLKS ...
000041 *
000042 *****
000043
```

End of File -- Lines: 43 Characters: 1176

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FOLDERS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FOLDERS
000004 *
000005 *****
000006
000007 USES {$U+} FOLDERS;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libhw/HWInt      } HWInt,
000016         {$U libsm/UnitStd   } UnitStd,
000017         {$U libsm/UnitHz    } UnitHz,
000018         {$U libqd/Storage   } Storage,
000019         {$U libqd/QuickDraw } QuickDraw,
000020         {$U libfm/FontMgr   } FontMgr,
000021         {$U libos/SysCall   } SysCall,
000022         {$U libwm/Events    } Events,
000023         {$IFC wmos }
000024         {$U libpm/PmDecl    } PmDecl,
000025         {$U libpm/Pmm      } Pmm,
000026         {$ENDC }
000027         {$U libwm/Windows  } Windows;
000028
000029     CONST
000030         maxWindWidth  = 720;
000031         maxWindHeight = 364;
000032         minWindWidth  = 50;
000033         minWindHeight = 14;
000034         noIcon        = 255;
000035
000036     TYPE
000037         WindowInfo = RECORD
000038             visible: BOOLEAN;
000039             hasTab:  BOOLEAN;
000040             okMove:  BOOLEAN;
000041             okGrow:  BOOLEAN;
000042             okFront: BOOLEAN;
000043             okClose: BOOLEAN;
000044             okActivate: BOOLEAN;
000045         END;
000046
000047     PROCEDURE BeginUpdate(folder: WindowPtr);
000048
000049     PROCEDURE DialogHeight(height: INTEGER; fUpdate: BOOLEAN);
000050
000051     PROCEDURE DisposeFolder(folder: WindowPtr);
000052
000053     PROCEDURE EndUpdate(folder: WindowPtr);
000054
000055     PROCEDURE FlushInput(window: WindowPtr);
000056
000057     PROCEDURE FlushWindow(window: WindowPtr);
000058
000059     PROCEDURE FlushProcess(process: ProcessID);
000060
000061     PROCEDURE FolderSize(folder: WindowPtr; width, height: INTEGER);
```

Apple Lisa Computer Technical Information

```
000062             fUpdate: BOOLEAN);
000063
000064 FUNCTION GetFldrRefCon(folder: WindowPtr): LongInt;
000065
000066 PROCEDURE GetFldrTitle(folder: WindowPtr; VAR title: Str255);
000067
000068 PROCEDURE GetWindInfo(window: WindowPtr; VAR info: WindowInfo);
000069
000070 PROCEDURE GiveControl(event: EventRecord);
000071
000072 PROCEDURE GiveWindow(window: WindowPtr; toProcess: ProcessID);
000073
000074 PROCEDURE HeDied(deadProcess: ProcessID);
000075
000076 PROCEDURE HideFolder(folder: WindowPtr);
000077
000078 PROCEDURE HiLiteTab(folder: WindowPtr; fHiLite: BOOLEAN);
000079
000080 PROCEDURE ImDying;
000081
000082 PROCEDURE InitWM;
000083
000084 PROCEDURE InvalidateRgn(folder: WindowPtr; rgn: RgnHandle);
000085
000086 PROCEDURE InvalRect(badRect: Rect);
000087
000088 PROCEDURE InvalRgn(badRgn: RgnHandle);
000089
000090 PROCEDURE LocateDialog(top, height: integer);
000091
000092 PROCEDURE MakeFldrActive(folder: WindowPtr; reasonWhy: LongInt);
000093
000094 PROCEDURE MakeTopActive;
000095
000096 PROCEDURE MoveFolder(folder: WindowPtr; hGlobal, vGlobal: INTEGER);
000097
000098 FUNCTION NewFolder(folderRect: Rect; title: Str255; visible: BOOLEAN;
000099                 behind: WindowPtr; refCon: LongInt; process: ProcessID;
000100                 whichIcon: INTEGER; overlay: INTEGER): WindowPtr;
000101
000102 PROCEDURE OpenWM;
000103
000104 PROCEDURE SetFldrTitle(folder: WindowPtr; title: Str255);
000105
000106 PROCEDURE SetFldrRefCon(folder: WindowPtr; data: LongInt);
000107
000108 PROCEDURE SetWindInfo(window: WindowPtr; VAR info: WindowInfo);
000109
000110 PROCEDURE ShowFolder(folder: WindowPtr);
000111
000112 PROCEDURE TakeControl(event: EventRecord; keepMenus, keepDialog: BOOLEAN);
000113
000114 PROCEDURE TakeWindow(window: WindowPtr);
000115
000116 PROCEDURE ValidRect(goodRect: Rect);
000117
000118 PROCEDURE ValidRgn(goodRgn: RgnHandle);
000119
000120 PROCEDURE WMClosePicture;
000121
000122 PROCEDURE WMKillPicture(window: WindowPtr);
000123
000124 PROCEDURE WMKillProcess(process: ProcessID);
000125
000126 PROCEDURE WMOpenPicture(window: WindowPtr);
000127
```


Apple Lisa Computer Technical Information

```
000128     PROCEDURE WMStartDoc(window: WindowPtr);
000129
000130 *****
000131 *
000132 *                THAT'S ALL FOLKS ...
000133 *
000134 *****
000135
```

End of File -- Lines: 135 Characters: 3537

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FONTMGR.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FONTMGR
000004 *
000005 *****
000006
000007 USES {$U+} FONTMGR;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libqd/QuickDraw} QuickDraw,
000016         {$U libsm/UnitStd} UnitStd,
000017         {$U libsm/UnitHz } UnitHz,
000018         {$U libos/SysCall} SysCall,
000019         {$U libos/PSysCall} PSysCall;
000020         {$SETC fmdebug := FALSE}
000021         {$SETC fmSymbols := fsmok}
000022         {$setC fTraceFM := fTrace}
000023
000024     CONST
000025         ccMax          = 255;
000026         ccBS           = 8;
000027         ccHT           = 9;
000028         ccLF           = 10;
000029         ccVT           = 11;
000030         ccFF           = 12;
000031         ccCR           = 13;
000032         ccESC          = 27;
000033         ccPara         = 166;
000034         ccSec          = 164;
000035         ccCopy         = 169;
000036         ccReg          = 168;
000037         ccTM           = 170;
000038         ccDeg          = 161;
000039         ccDag          = 160;
000040         ccBul          = 165;
000041         ccPound        = 163;
000042         ccElip         = 201;
000043         ccSysApple     = 202;
000044         ccLisaLogo     = 0;
000045         ccCheck        = 142;
000046         ccApple        = 143;
000047         ccLreg         = 0;
000048         ccLalt         = 1;
000049         ccLdash        = 2;
000050         ccRreg         = 3;
000051         ccRalt         = 4;
000052         ccRdash        = 5;
000053         ccHreg         = 6;
000054         ccHalt         = 7;
000055         ccHdash        = 8;
000056         ccVreg         = 9;
000057         ccValt         = 10;
000058         ccVdash        = 11;
000059         ccG90f         = 12;
000060         ccG90c         = 13;
000061         ccG45f         = 14;
```

Apple Lisa Computer Technical Information

```
000062      ccG45c      = 15;
000063      ccWhite     = 16;
000064      ccXltGray  = 17;
000065      ccLtGray   = 18;
000066      ccMedGray  = 19;
000067      ccGray     = 20;
000068      ccDkGray   = 21;
000069      ccBlack    = 22;
000070      ccNil      = 255;
000071      ccWhiteMask = 255;
000072      sysText    = 0;
000073      wmfont     = 1;
000074      sysPat     = 2;
000075      sysCursor  = 3;
000076      tile12    = 4;
000077      tile18    = 5;
000078      tile24    = 6;
000079      p15Tile   = 7;
000080      p12Tile   = 8;
000081      p10Tile   = 9;
000082      cent12    = 10;
000083      cent18    = 11;
000084      cent24    = 12;
000085      p12cent   = 13;
000086      p10cent   = 14;
000087      elite     = 15;
000088      courier   = 16;
000089      boldface  = 17;
000090      calcFont  = 18;
000091      p20Tile   = 19;
000092      marker    = 20;
000093      tile7     = 21;
000094      fiIconFont = 22;
000095      sysLogo   = 23;
000096      devScreen = 0;
000097      famLst    = 23;
000098      substitution = - 3024;
000099      BigCache  = 98000;
000100      HighCache = 45000;
000101      LowCache  = 30000;
000102      {$IFC NOT fmdebug}
000103      fdebug    = FALSE;
000104      fdebug1   = FALSE;
000105      fdebug2   = FALSE;
000106      fdebug3   = FALSE;
000107      fdebug4   = FALSE;
000108      fdebug5   = FALSE;
000109      fdebug6   = FALSE;
000110      fdebug7   = FALSE;
000111      fdebug8   = FALSE;
000112      fdebug9   = FALSE;
000113      fdebug10  = FALSE;
000114      fdebug11  = FALSE;
000115      fdebug12  = FALSE;
000116      fdebug13  = FALSE;
000117      fdebug14  = FALSE;
000118      fdebug15  = FALSE;
000119      fdebug16  = FALSE;
000120      fdebug17  = FALSE;
000121      fdebug18  = FALSE;
000122      fdebug19  = FALSE;
000123      fdebug20  = FALSE;
000124      fdebug21  = FALSE;
000125      fdebug22  = FALSE;
000126      fdebug23  = FALSE;
000127      fdebug24  = FALSE;
```

Apple Lisa Computer Technical Information

```
000128      fdebug25      = FALSE;
000129      fdebug26      = FALSE;
000130      fdebug27      = FALSE;
000131      fdebug28      = FALSE;
000132      fdebug29      = FALSE;
000133      fdebug30      = FALSE;
000134      fdebug31      = FALSE;
000135      fdebug32      = FALSE;
000136      fdebug33      = FALSE;
000137      fdebug34      = FALSE;
000138      fdebug35      = FALSE;
000139      fdebug36      = FALSE;
000140      fdebug37      = FALSE;
000141      fdebug38      = FALSE;
000142      fdebug39      = FALSE;
000143      fdebug40      = FALSE;
000144      {$ENDC}
000145
000146      TYPE
000147      TCc          = 0..255;
000148      TStr30       = STRING[30];
000149      TFam         = TC;
000150      TDev         = - 128..127;
000151      TEdev        = (edevScreen, edevDotMx, edevHDMx, edevTyper, edevLaser,
000152                  edev1, edev2);
000153      TEwhcl       = (ewhcl20, ewhcl15, ewhcl12, ewhcl10, ewhclPS, ewhclMF,
000154                  ewhclMP, ewhclLCS);
000155      TEskcl       = (eskclMPPS, eskclMPFP, eskclMFR, eskclMFI, eskclApple,
000156                  eskclFQume, eskclPQume);
000157      TAdev        = PACKED RECORD
000158                  CASE BOOLEAN OF
000159                      TRUE:
000160                          (dev: TDev);
000161                      FALSE:
000162                          (fOffline: TF;
000163                          CASE edev: TEdev OF
000164                              edevScreen, edevLaser:
000165                                  ();
000166                              edevDotMx, edevHDMx:
000167                                  (fHiRes: TF;
000168                                  fPortrait: TF);
000169                              edevTyper:
000170                                  (fSpoke: TF;
000171                                  ewhcl: TEwhcl);
000172                              edev1, edev2:
000173                                  ());
000174                      END;
000175      TFntid       = PACKED RECORD
000176                  fam: TFam;
000177                  seteface: Style;
000178                  END;
000179      Tlfntid      = PACKED RECORD
000180                  fam: TFam;
000181                  seteface: Style;
000182                  CASE BOOLEAN OF
000183                      TRUE:
000184                          (dev: TDev);
000185                      FALSE:
000186                          (fOffline: TF;
000187                          CASE edev: TEdev OF
000188                              edevScreen, edevLaser:
000189                                  ();
000190                              edevDotMx, edevHDMx:
000191                                  (fHiRes: TF;
000192                                  fPortrait: TF);
000193                              edevTyper:
```

Apple Lisa Computer Technical Information

```
000194             (fSpoke: TF;
000195             ewhcl: TEwhcl);
000196             edev1, edev2:
000197             ());
000198             END;
000199     Twid         = 0..255;
000200     TMpccwid     = PACKED ARRAY [TCc] OF Twid;
000201     TPmpccwid   = ^TMpccwid;
000202     THmpccwid   = ^TPmpccwid;
000203     TPcharStyle = ^TCharStyle;
000204     TCharstyle  = PACKED RECORD
000205             hks: TH;
000206             hmpccwid: THmpccwid;
000207             bBold: TB;
000208             bItalic: TB;
000209             bUnderline: TB;
000210             bShadow: TB;
000211             cExtra: TC;
000212             END;
000213     TMpefaceb   = ARRAY [StyleItem] OF TB;
000214     TModifier   = RECORD
000215             seteface: Style;
000216             mpefaceb: TMpefaceb;
000217             END;
000218
000219     VAR
000220     {$IFC fmdebug}
000221     fdebug:     TF;
000222     fdebug1:    TF;
000223     fdebug2:    TF;
000224     fdebug3:    TF;
000225     fdebug4:    TF;
000226     fdebug5:    TF;
000227     fdebug6:    TF;
000228     fdebug7:    TF;
000229     fdebug8:    TF;
000230     fdebug9:    TF;
000231     fdebug10:   TF;
000232     fdebug11:   TF;
000233     fdebug12:   TF;
000234     fdebug13:   TF;
000235     fdebug14:   TF;
000236     fdebug15:   TF;
000237     fdebug16:   TF;
000238     fdebug17:   TF;
000239     fdebug18:   TF;
000240     fdebug19:   TF;
000241     fdebug20:   TF;
000242     fdebug21:   TF;
000243     fdebug22:   TF;
000244     fdebug23:   TF;
000245     fdebug24:   TF;
000246     fdebug25:   TF;
000247     fdebug26:   TF;
000248     fdebug27:   TF;
000249     fdebug28:   TF;
000250     fdebug29:   TF;
000251     fdebug30:   TF;
000252     fdebug31:   TF;
000253     fdebug32:   TF;
000254     fdebug33:   TF;
000255     fdebug34:   TF;
000256     fdebug35:   TF;
000257     fdebug36:   TF;
000258     fdebug37:   TF;
000259     fdebug38:   TF;
```

Apple Lisa Computer Technical Information

```
000260      fdebug39:      TF;
000261      fdebug40:      TF;
000262      {$ENDC}
000263      InitialCacheSize: TL;
000264      MinCache:      TL;
000265      cRefnumLib:    TC;
000266
000267      PROCEDURE FMinit(VAR cError: TC);
000268
000269      PROCEDURE FMOpen(VAR cError: TC);
000270
000271      PROCEDURE FMCloseLib(VAR cError: TC);
000272
000273      PROCEDURE FMCacheSize(lCacheSize: TL; VAR cError: TC);
000274
000275      FUNCTION FMAddFont(str30: TStr30; modifier: TModifier; dev: TDev;
000276                      VAR cError: TC): TFam;
000277
000278      PROCEDURE FMDelfont(lfntid: TLfntid);
000279
000280      FUNCTION FMSwapFont(VAR lfntid: TLfntid; VAR cError: TC): TPcharstyle;
000281
000282      PROCEDURE FMLockFont(VAR lfntid: TLfntid; flock: TF; VAR cError: TC);
000283
000284      FUNCTION FMFontMetrics(VAR lfntid: TLfntid; VAR finfo: FontInfo;
000285                          VAR cError: TC): TF;
000286
000287      FUNCTION FMFontWidths(VAR lfntid: TLfntid; VAR hmpccwid: THmpccwid;
000288                          VAR dExtra: TB; VAR cError: TC): TF;
000289
000290      PROCEDURE FMMapFont(VAR lfntid: TLfntid; VAR cError: TC);
000291      {$IFC fmdebug}
000292
000293      PROCEDURE FMDebug(fdbg: TF);
000294
000295      PROCEDURE FMDumpHeurTable;
000296      {$ENDC}
000297
000298      *****
000299      *
000300      *              THAT'S ALL FOLKS ...
000301      *
000302      *****
000303
```

End of File -- Lines: 303 Characters: 8457

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 FPLIB.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FPLIB
000004 *
000005 *****
000006
000007 USES {$U+} FPLIB;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     {$C Copyright 1983, 1984, Apple Computer Inc. }
000015
000016     CONST
000017         SIGDIGLEN      = 20;
000018         DECSTRLEN      = 255;
000019
000020     TYPE
000021         Single         = real;
000022         Double         = ARRAY [0..3] OF integer;
000023         Comp           = ARRAY [0..3] OF integer;
000024         Extended      = ARRAY [0..4] OF integer;
000025         SigDig         = string[SIGDIGLEN];
000026         DecStr         = string[DECSTRLEN];
000027         Decimal        = RECORD
000028             sgn: 0..1;
000029             exp: integer;
000030             sig: SigDig
000031         END;
000032         Environ        = integer;
000033         RoundDir       = (TONEAREST, UPWARD, DOWNWARD, TOWARDZERO);
000034         RelOp          = (GT, LT, GL, EQ, GE, LE, GEL, UNORD);
000035         Exception      = (INVALID, UNDERFLOW, OVERFLOW, DIVBYZERO, INEXACT);
000036         NumClass       = (SNAN, QNAN, INFINITE, ZERO, NORMAL, DENORMAL);
000037         DecForm        = RECORD
000038             style: (FLOATDECIMAL, FIXEDDECIMAL);
000039             digits: integer
000040         END;
000041
000042     PROCEDURE AddS(x: Single; VAR y: Extended);
000043
000044     PROCEDURE AddD(x: Double; VAR y: Extended);
000045
000046     PROCEDURE AddC(x: Comp; VAR y: Extended);
000047
000048     PROCEDURE AddX(x: Extended; VAR y: Extended);
000049
000050     PROCEDURE SubS(x: Single; VAR y: Extended);
000051
000052     PROCEDURE SubD(x: Double; VAR y: Extended);
000053
000054     PROCEDURE SubC(x: Comp; VAR y: Extended);
000055
000056     PROCEDURE SubX(x: Extended; VAR y: Extended);
000057
000058     PROCEDURE MulS(x: Single; VAR y: Extended);
000059
000060     PROCEDURE MulD(x: Double; VAR y: Extended);
000061
```

Apple Lisa Computer Technical Information

```
000062    PROCEDURE MulC(x: Comp; VAR y: Extended);
000063
000064    PROCEDURE MulX(x: Extended; VAR y: Extended);
000065
000066    PROCEDURE DivS(x: Single; VAR y: Extended);
000067
000068    PROCEDURE DivD(x: Double; VAR y: Extended);
000069
000070    PROCEDURE DivC(x: Comp; VAR y: Extended);
000071
000072    PROCEDURE DivX(x: Extended; VAR y: Extended);
000073
000074    FUNCTION CmpX(x: Extended; r: RelOp; y: Extended): boolean;
000075
000076    FUNCTION RelX(x, y: Extended): RelOp;
000077
000078    PROCEDURE I2X(x: integer; VAR y: Extended);
000079
000080    PROCEDURE S2X(x: Single; VAR y: Extended);
000081
000082    PROCEDURE D2X(x: Double; VAR y: Extended);
000083
000084    PROCEDURE C2X(x: Comp; VAR y: Extended);
000085
000086    PROCEDURE X2X(x: Extended; VAR y: Extended);
000087
000088    PROCEDURE X2I(x: Extended; VAR y: integer);
000089
000090    PROCEDURE X2S(x: Extended; VAR y: Single);
000091
000092    PROCEDURE X2D(x: Extended; VAR y: Double);
000093
000094    PROCEDURE X2C(x: Extended; VAR y: Comp);
000095
000096    PROCEDURE L2X(x: longint; VAR y: Extended);
000097
000098    PROCEDURE X2L(x: Extended; VAR y: longint);
000099
000100    PROCEDURE S2Dec(f: DecForm; x: Single; VAR y: Decimal);
000101
000102    PROCEDURE D2Dec(f: DecForm; x: Double; VAR y: Decimal);
000103
000104    PROCEDURE C2Dec(f: DecForm; x: Comp; VAR y: Decimal);
000105
000106    PROCEDURE X2Dec(f: DecForm; x: Extended; VAR y: Decimal);
000107
000108    PROCEDURE Dec2S(x: Decimal; VAR y: Single);
000109
000110    PROCEDURE Dec2D(x: Decimal; VAR y: Double);
000111
000112    PROCEDURE Dec2C(x: Decimal; VAR y: Comp);
000113
000114    PROCEDURE Dec2X(x: Decimal; VAR y: Extended);
000115
000116    PROCEDURE Str2Dec(s: DecStr; VAR index: integer; VAR d: Decimal;
000117                VAR ValidPrefix: boolean);
000118
000119    PROCEDURE Dec2Str(f: DecForm; d: Decimal; VAR s: DecStr);
000120
000121    PROCEDURE S2Str(f: DecForm; x: Single; VAR y: DecStr);
000122
000123    PROCEDURE D2Str(f: DecForm; x: Double; VAR y: DecStr);
000124
000125    PROCEDURE C2Str(f: DecForm; x: Comp; VAR y: DecStr);
000126
000127    PROCEDURE X2Str(f: DecForm; x: Extended; VAR y: DecStr);
```


Apple Lisa Computer Technical Information

```
000128
000129   PROCEDURE Str2S(x: DecStr; VAR y: Single);
000130
000131   PROCEDURE Str2D(x: DecStr; VAR y: Double);
000132
000133   PROCEDURE Str2C(x: DecStr; VAR y: Comp);
000134
000135   PROCEDURE Str2X(x: DecStr; VAR y: Extended);
000136
000137   PROCEDURE RemX(x: Extended; VAR y: Extended; VAR quo: integer);
000138
000139   PROCEDURE SqrtX(VAR x: Extended);
000140
000141   PROCEDURE RintX(VAR x: Extended);
000142
000143   PROCEDURE NegX(VAR x: Extended);
000144
000145   PROCEDURE AbsX(VAR x: Extended);
000146
000147   PROCEDURE CpySgnX(VAR x: Extended; y: Extended);
000148
000149   PROCEDURE NextS(VAR x: Single; y: Single);
000150
000151   PROCEDURE NextD(VAR x: Double; y: Double);
000152
000153   PROCEDURE NextX(VAR x: Extended; y: Extended);
000154
000155   FUNCTION ClassS(x: Single; VAR sgn: integer): NumClass;
000156
000157   FUNCTION ClassD(x: Double; VAR sgn: integer): NumClass;
000158
000159   FUNCTION ClassC(x: Comp; VAR sgn: integer): NumClass;
000160
000161   FUNCTION ClassX(x: Extended; VAR sgn: integer): NumClass;
000162
000163   PROCEDURE ScalbX(n: integer; VAR y: Extended);
000164
000165   PROCEDURE LogbX(VAR x: Extended);
000166
000167   PROCEDURE SetRnd(r: RoundDir);
000168
000169   PROCEDURE SetEnv(e: Environ);
000170
000171   FUNCTION GetRnd: RoundDir;
000172
000173   PROCEDURE GetEnv(VAR e: Environ);
000174
000175   FUNCTION TestXcp(x: Exception): boolean;
000176
000177   PROCEDURE SetXcp(x: Exception; OnOff: boolean);
000178
000179   FUNCTION TestHlt(x: Exception): boolean;
000180
000181   PROCEDURE SetHlt(x: Exception; OnOff: boolean);
000182
000183   PROCEDURE ProcEntry(VAR e: Environ);
000184
000185   PROCEDURE ProcExit(e: Environ);
000186
000187   PROCEDURE Log2X(VAR x: Extended);
000188
000189   PROCEDURE LnX(VAR x: Extended);
000190
000191   PROCEDURE Ln1X(VAR x: Extended);
000192
000193   PROCEDURE Exp2X(VAR x: Extended);
```

Apple Lisa Computer Technical Information

```
000194
000195     PROCEDURE ExpX(VAR x: Extended);
000196
000197     PROCEDURE ExplX(VAR x: Extended);
000198
000199     PROCEDURE XpwrI(i: integer; VAR x: Extended);
000200
000201     PROCEDURE XpwrY(y: Extended; VAR x: Extended);
000202
000203     PROCEDURE Compound(r, n: Extended; VAR x: Extended);
000204
000205     PROCEDURE Annuity(r, n: Extended; VAR x: Extended);
000206
000207     PROCEDURE AtanX(VAR x: Extended);
000208
000209     PROCEDURE SinX(VAR x: Extended);
000210
000211     PROCEDURE CosX(VAR x: Extended);
000212
000213     PROCEDURE TanX(VAR x: Extended);
000214
000215     PROCEDURE RandomX(VAR x: Extended);
000216
000217     FUNCTION GetHltAddress: longint;
000218
000219     PROCEDURE SetHltAddress(HltAddress: longint);
000220
000221     PROCEDURE InitFPLib;
000222
000223     FUNCTION SANE_Environ: longint;
000224
000225 *****
000226 *
000227 *             THAT'S ALL FOLKS ...
000228 *
000229 *****
000230
```

End of File -- Lines: 230 Characters: 5565

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 GRAFUTIL.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : GRAFUTIL
000004 *
000005 *****
000006
000007 USES { $U+ } GRAFUTIL;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES                               { $U libqd/QuickDraw } QuickDraw;
000015
000016     TYPE
000017         Fixed           = LongInt;
000018         Int64Bit       = RECORD
000019             hiLong: LongInt;
000020             loLong: LongInt;
000021         END;
000022
000023     FUNCTION BitAnd(long1, long2: LongInt): LongInt;
000024
000025     FUNCTION BitOr(long1, long2: LongInt): LongInt;
000026
000027     FUNCTION BitXor(long1, long2: LongInt): LongInt;
000028
000029     FUNCTION BitNot(long: LongInt): LongInt;
000030
000031     FUNCTION BitShift(long: LongInt; count: INTEGER): LongInt;
000032
000033     FUNCTION BitTst(bytePtr: QDPtr; bitNum: LongInt): BOOLEAN;
000034
000035     PROCEDURE BitSet(bytePtr: QDPtr; bitNum: LongInt);
000036
000037     PROCEDURE BitClr(bytePtr: QDPtr; bitNum: LongInt);
000038
000039     PROCEDURE LongMul(a, b: LongInt; VAR dst: Int64Bit);
000040
000041     FUNCTION FixMul(a, b: Fixed): Fixed;
000042
000043     FUNCTION FixRatio( numer, denom: INTEGER): Fixed;
000044
000045     FUNCTION HiWord(x: Fixed): INTEGER;
000046
000047     FUNCTION LoWord(x: Fixed): INTEGER;
000048
000049     FUNCTION FixRound(x: Fixed): INTEGER;
000050
000051 *****
000052 *
000053 *             THAT'S ALL FOLKS ...
000054 *
000055 *****
000056
```

End of File -- Lines: 56 Characters: 1355

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 GRAPHS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : GRAPHS
000004 *
000005 *****
000006
000007 USES {$U+} GRAPHS;
000008
000009
000010 {$IFC isIntrin} INTRINSIC; {$ENDC}
000011
000012 INTERFACE
000013
000014     USES
000015     {
000016     {$IFC SrcOnOS }
000017     (*$U OBJ/PASDEFS.OBJ *) PasDefs,
000018     (*$U OBJ/MEMMAN.OBJ *) MemMan,
000019     (*$U OBJ/LISTS.OBJ *) Lists;
000020     {$ELSEC }
000021     (*$U OBJ: PASDEFS.OBJ *) PasDefs,
000022     (*$U OBJ: MEMMAN.OBJ *) MemMan,
000023     (*$U OBJ: LISTS.OBJ *) Lists;
000024     {$ENDC }
000025
000026     TYPE
000027
000028     Vertex      = integer;
000029     Arc          = integer;
000030     VertexRec   = RECORD
000031         firstIn, firstOut: Arc;
000032     END;
000033     ArcRec      = RECORD
000034         initial, terminal: Vertex;
000035         nextIn, nextOut: Arc;
000036     END;
000037     VtxRowPtr   = ^VtxRow;
000038     ArcRowPtr   = ^ArcRow;
000039     VtxRow      = ARRAY [ - 128..127 ] OF VertexRec;
000040     ArcRow      = ARRAY [ - 128..127 ] OF ArcRec;
000041     VtxArrPtr   = ^VtxArr;
000042     ArcArrPtr   = ^ArcArr;
000043     VtxArr      = ARRAY [ - 128..127 ] OF VtxRowPtr;
000044     ArcArr      = ARRAY [ - 128..127 ] OF ArcRowPtr;
000045     GRHandle    = ^GraphStore;
000046     GraphStore  = RECORD
000047         memVertices, memArcs: MMHandle;
000048         vertices: VtxArrPtr;
000049         arcs: ArcArrPtr;
000050     END;
000051     GraphHandle = ^GraphObject;
000052     GraphObject = RECORD
000053         gstore: GRHandle;
000054         inL, outL: ListHandle;
000055     END;
000056
000057     PROCEDURE AddArc(graph: GraphHandle; init, term: Vertex; VAR newA: Arc;
000058         FUNCTION GetMore(n, row: integer): boolean);
000059
000060     PROCEDURE AddVertex(graph: GraphHandle; VAR newV: Vertex; FUNCTION
000061         GetMore(n, row: integer): boolean);
```

Apple Lisa Computer Technical Information

```
000062
000063 PROCEDURE DeleteArc(graph: GraphHandle; oldArc: Arc);
000064
000065 PROCEDURE DeleteVertex(graph: GraphHandle; oldVertex: Vertex);
000066
000067 FUNCTION FindArc(graph: GraphHandle; init, term: Vertex): Arc;
000068
000069 FUNCTION OppositeVertex(graph: GraphHandle; a: Arc;
000070 oneVertex: Vertex): Vertex;
000071
000072 FUNCTION InitVertex(graph: GraphHandle; a: Arc): Vertex;
000073
000074 FUNCTION TermVertex(graph: GraphHandle; a: Arc): Vertex;
000075
000076 FUNCTION IsSource(graph: GraphHandle; v: Vertex): boolean;
000077
000078 FUNCTION IsSink(graph: GraphHandle; v: Vertex): boolean;
000079
000080 FUNCTION IsIsolated(graph: GraphHandle; v: Vertex): boolean;
000081
000082 PROCEDURE EachVertex(graph: GraphHandle; PROCEDURE
000083 Visit(v: Vertex));
000084
000085 PROCEDURE EachArc(graph: GraphHandle; PROCEDURE
000086 Visit(a: Arc));
000087
000088 PROCEDURE EachInArc(graph: GraphHandle; v: Vertex; PROCEDURE
000089 Visit(inArc: Arc));
000090
000091 PROCEDURE EachOutArc(graph: GraphHandle; v: Vertex; PROCEDURE
000092 Visit(outArc: Arc));
000093
000094 PROCEDURE EachIncidentArc(graph: GraphHandle; v: Vertex; PROCEDURE
000095 Visit(incidentArc: Arc));
000096
000097 PROCEDURE EachPredecessorVertex(graph: GraphHandle; v: Vertex; PROCEDURE
000098 Visit(predecessor: Vertex));
000099
000100 PROCEDURE EachSuccessorVertex(graph: GraphHandle; v: Vertex; PROCEDURE
000101 Visit(successor: Vertex));
000102
000103 PROCEDURE EachAdjacentVertex(graph: GraphHandle; v: Vertex; PROCEDURE
000104 Visit(adjacent: Vertex));
000105
000106 PROCEDURE InitGLists;
000107
000108 PROCEDURE InitGraph(VAR graph: GraphHandle; nVertices, nArcs: longint;
000109 FUNCTION MoreVerts(n, row: integer): boolean; FUNCTION
000110 MoreArcs(n, row: integer): boolean);
000111
000112 PROCEDURE DFSDir(g: GraphHandle; start: Vertex; PROCEDURE
000113 First(vi, a, vt: Vertex); PROCEDURE
000114 Again(vi, a, vt: Vertex); PROCEDURE
000115 Done(v: Vertex));
000116
000117 PROCEDURE BFSUnDir(g: GraphHandle; start: Vertex; FUNCTION
000118 Test(v: Vertex; VAR cont: boolean): boolean);
000119
000120 *****
000121 *
000122 * THAT'S ALL FOLKS ...
000123 *
000124 *****
000125
```

End of File -- Lines: 125 Characters: 4154

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 HEAP.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : HEAP
000004 *
000005 *****
000006
000007 USES {$U+} HEAP;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     CONST
000015         noerror         = 0;
000016         nomem           = 3411;
000017         lhead           = 2;
000018         uhead           = 10;
000019         free_overhead   = 12;
000020         allocover       = 8;
000021
000022     TYPE
000023         _ptr             = ^integer;
000024         ptrptr           = ^_ptr;
000025         heapptr          = ^heapblk;
000026         heapblk          = RECORD
000027             CASE boolean OF
000028                 true:
000029                     (size: integer;
000030                      prep: heapptr;
000031                      next: heapptr);
000032                 false:
000033                     (blk: ARRAY [0..0] OF - 127..127)
000034             END;
000035         zonedesc         = RECORD
000036             avail: heapptr;
000037             heapmin, heapmax: _ptr;
000038             lastdiff: integer;
000039             availsz: integer;
000040             END;
000041         ptrzonedesc      = ^zonedesc;
000042
000043     PROCEDURE init_heap(pzone: ptrzonedesc; initsize: integer; where: _ptr);
000044
000045     FUNCTION grow_heap(pzone: ptrzonedesc; growincr: integer): integer;
000046
000047     FUNCTION nnew(pzone: ptrzonedesc; reqsize: integer;
000048                 pdataptr: ptrptr): integer;
000049
000050     PROCEDURE ndispose(pzone: ptrzonedesc; pdata: _ptr);
000051
000052 *****
000053 *
000054 *             THAT'S ALL FOLKS ...
000055 *
000056 *****
000057
```

End of File -- Lines: 57 Characters: 1595

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 HWINT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : HWINT
000004 *
000005 *****
000006
000007 USES {$U+} HWINT;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     TYPE
000015     Ascii           = Char;
000016     Pixels          = Integer;
000017     ManyPixels     = LongInt;
000018     CursorHeight   = Integer;
000019     CursorPtr       = ^Integer;
000020     LogicalAddress  = LongInt;
000021     DateArray       = RECORD
000022                     year: Integer;
000023                     day: Integer;
000024                     hour: Integer;
000025                     minute: Integer;
000026                     second: Integer;
000027     END;
000028     Frames          = LongInt;
000029     Seconds         = LongInt;
000030     MilliSeconds    = LongInt;
000031     MicroSeconds    = LongInt;
000032     AlarmNumber     = Integer;
000033     SpeakerVolume   = Integer;
000034     ScreenContrast  = Integer;
000035     VisibleScrn     = (PriScrn, AltScrn);
000036     KeybdQIndex     = 1..1000;
000037     KeybdId         = Integer;
000038     KeyCap          = 0..127;
000039     KeyCapSet       = SET OF KeyCap;
000040     KeyEvent        = PACKED RECORD
000041                     key: KeyCap;
000042                     ascii: Char;
000043                     state: Integer;
000044                     mouseX: Pixels;
000045                     mouseY: Pixels;
000046                     time: MilliSeconds;
000047     END;
000048
000049     PROCEDURE DriverInit;
000050
000051     PROCEDURE DiskDriver(routine: LogicalAddress);
000052
000053     PROCEDURE TwiggyDriver(routine: LogicalAddress);
000054
000055     PROCEDURE DiskSync(busy: Boolean);
000056
000057     PROCEDURE NMISync;
000058
000059     PROCEDURE COPSSync;
000060
000061     PROCEDURE Poll;
```

Apple Lisa Computer Technical Information

```
000062
000063     PROCEDURE MouseLocation(VAR x: Pixels; VAR y: Pixels);
000064
000065     PROCEDURE MouseUpdates(delay: MilliSeconds);
000066
000067     PROCEDURE MouseScaling(scale: Boolean);
000068
000069     PROCEDURE MouseThresh(threshold: Pixels);
000070
000071     FUNCTION MouseOdometer: ManyPixels;
000072
000073     PROCEDURE CursorLocation(x: Pixels; y: Pixels);
000074
000075     PROCEDURE CursorTracking(track: Boolean);
000076
000077     PROCEDURE CursorImage(hotX: Pixels; hotY: Pixels; height: CursorHeight;
000078                           data: CursorPtr; mask: CursorPtr);
000079
000080     PROCEDURE CursorHide;
000081
000082     PROCEDURE CursorShield(left: Pixels; top: Pixels; right: Pixels;
000083                           bottom: Pixels);
000084
000085     PROCEDURE CursorDisplay;
000086
000087     PROCEDURE CursorObscure;
000088
000089     PROCEDURE CursorInit;
000090
000091     PROCEDURE CursorReInit;
000092
000093     PROCEDURE BusyImage(hotX: Pixels; hotY: Pixels; height: CursorHeight;
000094                       data: CursorPtr; mask: CursorPtr);
000095
000096     PROCEDURE BusyDelay(delay: MilliSeconds);
000097
000098     FUNCTION FrameCounter: Frames;
000099
000100     PROCEDURE ScreenSize(VAR x: Pixels; VAR y: Pixels);
000101
000102     FUNCTION ScreenAddr: LogicalAddress;
000103
000104     FUNCTION AltScreenAddr: LogicalAddress;
000105
000106     FUNCTION ScreenKeybd: VisibleScreen;
000107
000108     PROCEDURE SetScreenKeybd(screen: VisibleScreen);
000109
000110     FUNCTION Contrast: ScreenContrast;
000111
000112     PROCEDURE SetContrast(contrast: ScreenContrast);
000113
000114     PROCEDURE RampContrast(contrast: ScreenContrast);
000115
000116     FUNCTION DimContrast: ScreenContrast;
000117
000118     PROCEDURE SetDimContrast(contrast: ScreenContrast);
000119
000120     FUNCTION FadeDelay: MilliSeconds;
000121
000122     PROCEDURE SetFadeDelay(delay: MilliSeconds);
000123
000124     PROCEDURE PowerDown;
000125
000126     PROCEDURE PowerCycle(delay: Seconds);
000127
```


Apple Lisa Computer Technical Information

```
000128 FUNCTION Volume: SpeakerVolume;
000129
000130 PROCEDURE SetVolume(volume: SpeakerVolume);
000131
000132 PROCEDURE Noise(waveLength: MicroSeconds);
000133
000134 PROCEDURE Silence;
000135
000136 PROCEDURE Beep(waveLength: MicroSeconds; duration: MilliSeconds);
000137
000138 FUNCTION Keyboard: KeybdId;
000139
000140 FUNCTION Legends: KeybdId;
000141
000142 PROCEDURE SetLegends(id: KeybdId);
000143
000144 FUNCTION KeyIsDown(key: KeyCap): Boolean;
000145
000146 PROCEDURE KeyMap(VAR keys: KeyCapSet);
000147
000148 FUNCTION KeybdPeek(repeats: Boolean; index: KeybdQIndex;
000149 VAR event: KeyEvent): Boolean;
000150
000151 FUNCTION AltKeyPeek(repeats: Boolean; index: KeybdQIndex;
000152 VAR event: KeyEvent): Boolean;
000153
000154 FUNCTION KeybdEvent(repeats: Boolean; wait: Boolean;
000155 VAR event: KeyEvent): Boolean;
000156
000157 FUNCTION AltKeyEvent(repeats: Boolean; wait: Boolean;
000158 VAR event: KeyEvent): Boolean;
000159
000160 PROCEDURE RepeatRate(VAR initial: MilliSeconds;
000161 VAR subsequent: MilliSeconds);
000162
000163 PROCEDURE SetRepeatRate(initial: MilliSeconds; subsequent: MilliSeconds);
000164
000165 PROCEDURE KeyPushed(key: KeyCap);
000166
000167 FUNCTION NMIKey: KeyCap;
000168
000169 PROCEDURE SetNMIKey(key: KeyCap);
000170
000171 FUNCTION ToggleKey: KeyCap;
000172
000173 PROCEDURE SetToggleKey(key: KeyCap);
000174
000175 FUNCTION KeyToAscii(key: KeyCap; State: Integer): Ascii;
000176
000177 FUNCTION MicroTimer: MicroSeconds;
000178
000179 FUNCTION Timer: MilliSeconds;
000180
000181 PROCEDURE AlarmAssign(VAR alarm: AlarmNumber; routine: LogicalAddress);
000182
000183 PROCEDURE AlarmReturn(alarm: AlarmNumber);
000184
000185 PROCEDURE AlarmAbsolute(alarm: AlarmNumber; time: MilliSeconds);
000186
000187 PROCEDURE AlarmRelative(alarm: AlarmNumber; delay: MilliSeconds);
000188
000189 PROCEDURE AlarmOff(alarm: AlarmNumber);
000190
000191 PROCEDURE DateTime(VAR date: DateArray);
000192
000193 PROCEDURE SetDateTime(date: DateArray);
```

Apple Lisa Computer Technical Information

```
000194
000195     PROCEDURE DateToTime(date: DateArray; VAR time: Seconds);
000196
000197     FUNCTION TimeStamp: Seconds;
000198
000199     PROCEDURE SetTimeStamp(time: Seconds);
000200
000201     PROCEDURE TimeToDate(time: Seconds; VAR date: DateArray);
000202
000203 *****
000204 *
000205 *                THAT'S ALL FOLKS ...
000206 *
000207 *****
000208
```

End of File -- Lines: 208 Characters: 5154

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 INTERNAT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : INTERNAT
000004 *
000005 *****
000006
000007 USES {$U+} INTERNAT;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/UnitStd } UnitStd;
000016
000017     CONST
000018         MaximumCharsMatching = 10;
000019         MCodeNil             = 0;
000020         MCodeUS              = 1;
000021         MCodeUK              = 2;
000022         MCodeGermany        = 3;
000023         MCodeFrance         = 4;
000024         MCodeItaly          = 5;
000025         MCodeSweden         = 6;
000026         MCodeSpain          = 7;
000027
000028     TYPE
000029         KindOfCompare = (CompAbsolute, CompStrong, CompWeak, CompVeryWeak);
000030         CompChrP      = ^CompChr;
000031         CompChr       = PACKED ARRAY [0..0] OF CHAR;
000032         CompStrP      = ^CompStr;
000033         CompStr       = STRING[255];
000034
000035     PROCEDURE InitInternational(VAR errnum: INTEGER);
000036
000037     PROCEDURE UpStrShift(sP: CompStrP);
000038
000039     PROCEDURE UpChrShift(pc: CompChrP; length: INTEGER);
000040
000041     PROCEDURE DownStrShift(sP: CompStrP);
000042
000043     PROCEDURE DownChrShift(pc: CompChrP; length: INTEGER);
000044
000045     FUNCTION CompStrMagnitude(string1, string2: CompStrP;
000046                             veryWeak: BOOLEAN): INTEGER;
000047
000048     FUNCTION CompChrMagnitude(char1, char2: CompChrP; length1, length2: INTEGER;
000049                             veryWeak: BOOLEAN): INTEGER;
000050
000051     FUNCTION CompareChar(ch1, ch2: CHAR; kind: KindOfCompare): BOOLEAN;
000052
000053     PROCEDURE CharsMatching(ch: CHAR; kind: KindOfCompare;
000054                             matchingChars: CompChrP;
000055                             VAR nOfMatchigChars: INTEGER);
000056
000057     FUNCTION CompStrIdentity(string1, string2: CompStrP;
000058                             kind: KindOfCompare): BOOLEAN;
000059
000060     FUNCTION CompChrIdentity(char1, char2: CompChrP; length1, length2: INTEGER;
000061                             kind: KindOfCompare): BOOLEAN;
```

Apple Lisa Computer Technical Information

```
000062
000063 *****
000064 *
000065 *           THAT'S ALL FOLKS ...
000066 *
000067 *****
000068
```

End of File -- Lines: 68 Characters: 1915

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 IOPRIMIT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : IOPRIMIT
000004 *
000005 *****
000006
000007 USES {$U+} IOPRIMIT;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$u SULib} StdUnit;
000016
000017     CONST
000018         IOMaxStr      = 132;
000019         IOMaxStP1    = 133;
000020         IOBlkSize    = 6;
000021         IOByteSize   = 3072;
000022         IOBytSizeM1  = 3071;
000023         IOIncMax     = 5;
000024         IOPvtErr     = 32000;
000025         IOOpnErr     = 32001;
000026         IOIncErr     = 32002;
000027         IOBfrErr     = 32003;
000028         IOMem1Err    = 32004;
000029         IOMem2Err    = 32005;
000030
000031     TYPE
000032         IOModes      = (IORead, IOWrite);
000033         IOCloseKind  = (IONormal, IOLock, IOPurge, IOCrunch);
000034         IODevType    = (IOBlkDev, IOKbDev, IOConsDev, IOPrDev);
000035         IOStrP       = ^IOStr;
000036         IOStr        = String[IOMaxStr];
000037         IOBufrP      = ^IOBufr;
000038         IOBufr       = PACKED ARRAY [0..IOBytSizeM1] OF Char;
000039         IOPBufP      = ^IOPBufr;
000040         IOPBufr      = RECORD
000041             Next: IOPBufP;
000042             Bufr: IOBufrP;
000043         END;
000044         IOFCBP       = ^IOFCB;
000045         IOFCB        = RECORD
000046             Filename: Sustr;
000047             DevType: IODevType;
000048             F: FILE;
000049             LineNbr: Integer;
000050             Bufr: IOBufrP;
000051             BufrCp: Integer;
000052             PvtBufr: IOPBufP;
000053             PvtFCB: Boolean;
000054             BlkNbr: Integer;
000055             CurrLine: String[IOMaxStP1];
000056             CurrLen: Integer;
000057             Next, Prev, Avail: IOFCBP;
000058             PtrToFCBPtr: ^IOFCBP;
000059             Pushed: Boolean;
000060             BufrSaved: Boolean;
000061             CASE Mode: IOModes OF
```

Apple Lisa Computer Technical Information

```
000062             IORead:
000063                 (LastBlk: Integer;
000064                   LastByte: Integer;
000065                   Eol: Boolean);
000066             IOWrite:
000067                 (EndPage: Integer);
000068             END;
000069     IOTextP      = ^Text;
000070
000071     VAR
000072     IOCurrFCB:   IOFCBP;
000073     IOIncDepth:  Integer;
000074     IOIncBufPr:  IOBufPr;
000075     IOIncFCBs:   ARRAY [1..IOIncMax] OF IOFCBP;
000076     IOInTotal:   LongInt;
000077     IOOutTotal:  LongInt;
000078     IOEndfile:   Char;
000079     IONewline:   Char;
000080     IOKeyboard:  Text;
000081     IOObjExt:    Boolean;
000082     IONoExt:     Boolean;
000083     IONoDLEs:   Boolean;
000084
000085     PROCEDURE InitIO;
000086
000087     PROCEDURE EndIO;
000088
000089     FUNCTION OpenF(Fname: SStr; VAR FCB: IOFCBP; Mode: IOModes;
000090                  BufPr: IOBufPr): Integer;
000091
000092     FUNCTION CreateF(Fname: SStr; VAR FCB: IOFCBP; BufPr: IOBufPr): Integer;
000093
000094     FUNCTION SysOpenF(Fname: SStr; VAR FCB: IOFCBP; Mode: IOModes;
000095                     BufPr: IOBufPr): Integer;
000096
000097     PROCEDURE CloseF(FCB: IOFCBP; CloseKind: IOCloseKind);
000098
000099     FUNCTION PushInput(Fname: SStr): Integer;
000100
000101     FUNCTION PopInput: Boolean;
000102
000103     PROCEDURE NextPage(FCB: IOFCBP);
000104
000105     PROCEDURE FillBufPr(FCB: IOFCBP);
000106
000107     PROCEDURE SeekBlock(FCB: IOFCBP; Block: Integer);
000108
000109     PROCEDURE FilePosition(FCB: IOFCBP; VAR Block, Byte: Integer);
000110
000111     FUNCTION GetcF(FCB: IOFCBP; VAR c: Char): Char;
000112
000113     FUNCTION GetLine(FCB: IOFCBP; Line: IOStrPr): Boolean;
000114
000115     PROCEDURE PutcF(FCB: IOFCBP; c: Char);
000116
000117     PROCEDURE PutLinePr(FCB: IOFCBP; Line: IOStrPr);
000118
000119     PROCEDURE PutLineS(FCB: IOFCBP; Line: IOStr);
000120
000121     PROCEDURE PutStrPr(FCB: IOFCBP; S: IOStrPr; Width: Integer);
000122
000123     PROCEDURE PutStrS(FCB: IOFCBP; S: IOStr; Width: Integer);
000124
000125     FUNCTION PutIntPr(N: LongInt; Width: Integer): IOStrPr;
000126
000127     FUNCTION IOError(IOStatus: Integer; Msg: SStr): Boolean;
```

Apple Lisa Computer Technical Information

```
000128
000129     FUNCTION SysReset(F: IOTextP; FN: SUStr): Integer;
000130
000131     *****
000132     *
000133     *                               THAT'S ALL FOLKS ...
000134     *
000135     *****
000136
```

End of File -- Lines: 136 Characters: 3872

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 IUMAN.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : IUMAN
000004 *
000005 *****
000006
000007 USES { $U+ } IUMAN;
000008
000009
000010 { $IFC isIntrin } INTRINSIC; { $ENDC }
000011
000012 INTERFACE
000013
000014     USES
000015     { $IFC SrcOnOS }
000016     ( *$U OBJ/STDUNIT.OBJ * ) StdUnit,
000017     ( *$U OBJ/PASDEFS.OBJ * ) PasDefs,
000018     ( *$U OBJ/MEMMAN.OBJ * ) MemMan,
000019     ( *$U OBJ/FILEIO.OBJ * ) FileIO,
000020     ( *$U OBJ/OBJIO.OBJ * ) ObjIO;
000021     { $ELSEC }
000022     ( *$U OBJ:STDUNIT.OBJ * ) StdUnit,
000023     ( *$U OBJ:PASDEFS.OBJ * ) PasDefs,
000024     ( *$U OBJ:MEMMAN.OBJ * ) MemMan,
000025     ( *$U OBJ:FILEIO.OBJ * ) FileIO,
000026     ( *$U OBJ:OBJIO.OBJ * ) ObjIO;
000027     { $ENDC }
000028
000029     CONST
000030
000031
000032     ConfigMax      = '127.127.127.127';
000033
000034     TYPE
000035     pSegLocVariant = ^iSegLocVariant;
000036     pOldULVariant  = ^iOldULVariant;
000037     pUnitLVariant  = ^iUnitLVariant;
000038     { $IFC isSAndE }
000039     pSoftSystem    = ^iSoftSystem;
000040     pSftSysRec     = ^sftSysRec;
000041     SftSysRec      = RECORD
000042         SDep: isftSysVariant;
000043         nxt: pSftSysRec;
000044     END;
000045     { $ENDC }
000046     IUSegArr       = ARRAY [1..TMAXSEGS] OF pSegLocVariant;
000047     IUUnitArr      = ARRAY [1..TMAXUNITS] OF pUnitLVariant;
000048     IUStrArr       = ARRAY [1..TMAXFILES] OF PLString;
000049     PIUStrArr      = ^IUStrArr;
000050     { $IFC isSAndE }
000051     IUSysArr       = ARRAY [1..TMAXSYSS] OF pSoftSystem;
000052     IUSysDep       = ARRAY [1..TMAXSYSS] OF pSftSysRec;
000053     { $ENDC }
000054
000055     VAR
000056     IUMaxSegs, IUMaxUnits, IUMaxFNames, IUMaxSys: integer;
000057     iuLibSegs:    ^IUSegArr;
000058     iuLibUnts:    ^IUUnitArr;
000059     iuLibFNam:    PIUStrArr;
000060     { $IFC isSAndE }
000061     iuLibSys:     ^IUSysArr;
```


Apple Lisa Computer Technical Information

```
000062      iuLibSNam:    PIUStrArr;
000063      iuLibSDep:    ^IUSysDep;
000064      {$ENDC}
000065      iuLibFile:    ObjHandle;
000066      iuLibName:    LString;
000067      OldFormat:    boolean;
000068      ThisIULib:    integer;
000069      sysConfig, sysMinConfig, sysMaxConfig: Longint;
000070
000071      PROCEDURE InitIUMan;
000072
000073      PROCEDURE EachIUSeg(PROCEDURE visit(s: integer));
000074
000075      PROCEDURE EachIUUnit(PROCEDURE visit(u: integer));
000076
000077      PROCEDURE EachIUFile(PROCEDURE visit(f: integer));
000078      {$IFC issAndE}
000079
000080      PROCEDURE EachIUSys(PROCEDURE visit(s: integer));
000081      {$ENDC}
000082
000083      PROCEDURE VersToStr(VAR s: LString; VAR vers: Longint);
000084
000085      PROCEDURE WriteVers(VAR f: text; vers: Longint; len: integer);
000086
000087      PROCEDURE StrToVers(s: LString; VAR vers: Longint);
000088
000089      PROCEDURE ReadVers(VAR vers: Longint; VAR stat: PromptState; DefL: Longint);
000090
000091      PROCEDURE VerifyVers(minfile: Longint; maxfile: Longint; msg: LString);
000092
000093      FUNCTION ReadIULib(allocDelta, unitsOnly, filter: boolean;
000094                      SSNam: PLString): boolean;
000095
000096      PROCEDURE WriteIULib(VCtrl: ObjFileType; SysC: Longint; ModL: integer;
000097                          TimeS: Longint; VAR iuOutFile: ObjHandle;
000098                          RootIdx: integer);
000099
000100      PROCEDURE WriteIUVers(VCtrl: ObjFileType; SysC: Longint; ModL: integer;
000101                          TimeS: Longint; VAR iuOutFile: ObjHandle;
000102                          RootIdx: integer; EmitOSBlk: boolean);
000103
000104      FUNCTION IUInstall(VAR FName: LString; FNum: integer; VAR vFile: ObjHandle;
000105                      VAR PName: LString; PITSeg: integer): boolean;
000106
000107      PROCEDURE CopyInter(inF, outF: ObjHandle; VAR outBlock: ObjBlock; NBytes,
000108                      bufP: Longint);
000109
000110      *****
000111      *
000112      *           THAT'S ALL FOLKS ...
000113      *
000114      *****
000115
```

End of File -- Lines: 115 Characters: 3471

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 LABSCAN.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : LABSCAN
000004 *
000005 *****
000006
000007 USES {$U+} LABSCAN;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$IFC OSBUILT }
000016             {$U libsm/unitstd.obj} unitstd,
000017             {$U libdb/dbenv.obj } dbenv,
000018             {$U libin/INTRLIBp.obj } international,
000019             {$U libdb/dbdecl1.obj } dbdecl1,
000020             {$U libos/syscall.obj } syscall,
000021             {$U libos/psyscall.obj } psyscall,
000022             {$U libdb/lowlevel.obj } lowlevel,
000023             {$U libdb/pooler.obj } pooler,
000024             {$U libdb/heap.obj } heap,
000025             {$U libdb/czcompact.obj } czCompact,
000026             {$U libdb/vltree.obj } vltree,
000027             {$U libdb/scan.obj } scan;
000028         {$ELSEC}
000029             {$U OBJ:dbenv.obj } dbenv,
000030             {$U INTRLIB.obj } international,
000031             {$U OBJ:dbdecl1.obj } dbdecl1,
000032             {$U OBJ:syscall.obj } syscall,
000033             {$U OBJ:lowlevel.obj } lowlevel,
000034             {$U OBJ:pooler.obj } pooler,
000035             {$U OBJ:heap.obj } heap,
000036             {$U OBJ:czcompact.obj } czCompact,
000037             {$U OBJ:vltree.obj } vltree,
000038             {$U OBJ:scan.obj } scan;
000039         {$ENDC}
000040
000041     {$SETC debug := false }
000042
000043     {$IFC PRERELEASE }
000044
000045     var
000046
000047
000048         labdebug:    boolean;
000049     {$ENDC }
000050
000051     PROCEDURE lfetch(VAR ddresult: integer; scanid: integer; which: integer;
000052                     nsearch: integer; plabrec: ptrlabelentry);
000053
000054     PROCEDURE linsert(VAR ddresult: integer; scanid: integer;
000055                      plabrec: ptrlabelentry; size: integer;
000056                      newticket: boolean);
000057
000058     PROCEDURE ldelete(VAR ddresult: integer; scanid: integer; which: integer;
000059                      nsearch: integer; plabrec: ptrlabelentry);
000060
000061 *****
```

Apple Lisa Computer Technical Information

```
000062 *  
000063 *          THAT'S ALL FOLKS ...  
000064 *  
000065 *****  
000066
```

End of File -- Lines: 66 Characters: 1901

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 LCUT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : LCUT
000004 *
000005 *****
000006
000007 USES { $U+ } LCUT;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         ( *$U libsm/UnitStd * ) UnitStd,
000016         ( *$U libsm/UnitHz * ) UnitHz,
000017         ( *$U libqd/Storage * ) Storage,
000018         { $U libqd/QuickDraw } QuickDraw,
000019         { $U libfm/FontMgr   } FontMgr,
000020         ( *$U libdb/dbenv * ) dbenv,
000021         ( *$U libfe/FEdec * ) FEdec,
000022         ( *$U libfe/fld * ) fieldedit,
000023         ( *$U libos/SysCall * ) Syscall,
000024         ( *$U libte/teenv * ) teenv,
000025         ( *$U libpr/PMDecl * ) PMDecl,
000026         ( *$U libpr/PrStdInfo * ) PrStdInfo,
000027         ( *$U libwm/events * ) events,
000028         ( *$U libsu/scrap * ) scrap,
000029         ( *$U libsu/unitFile * ) unitFile,
000030         ( *$U libsu/unitFmt * ) unitFmt,
000031         ( *$U libsu/unitCs * ) UnitCs,
000032         ( *$U libsu/unitFF * ) unitFF;
000033         ( *$SETC LCUTdebug := not teProduction * )
000034
000035     CONST
000036         LCUTAlloc      = 0;
000037         LCUTNoData     = 1;
000038         LCUTTabProb    = 2;
000039         LCUTWidProb    = 3;
000040         LCUTNoHeap     = 4;
000041         ( *$IFC LCUTdebug * )
000042         TraceLCUT      = false
000043         ( *$ENDC * )
000044
000045 *****
000046 *
000047 *                 THAT'S ALL FOLKS ...
000048 *
000049 *****
000050
```

End of File -- Lines: 50 Characters: 1328

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 LISTS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : LISTS
000004 *
000005 *****
000006
000007 USES {$U+} LISTS;
000008
000009
000010 {$IFC isIntrin} INTRINSIC; {$ENDC}
000011
000012 INTERFACE
000013
000014     USES
000015         {$IFC SrcOnOS }
000016             (*$U OBJ/PASDEFS.OBJ *) PasDefs,
000017             (*$U OBJ/MEMMAN.OBJ *) MemMan;
000018         {$ELSEC }
000019             (*$U OBJ/PASDEFS.OBJ *) PasDefs,
000020             (*$U OBJ/MEMMAN.OBJ *) MemMan;
000021         {$ENDC }
000022
000023     TYPE
000024
000025
000026         Head           = integer;
000027         Item           = integer;
000028         ListOpsPtr     = ^ListOpsRec;
000029         EnvrPtr        = ^integer;
000030         ListOpsRec     = RECORD
000031             setFirst: ProcPtr;
000032             getFirst: ProcPtr;
000033             setNext: ProcPtr;
000034             getNext: ProcPtr;
000035             before: ProcPtr;
000036         END;
000037         ListHandle     = ^ListStore;
000038         ListStore      = RECORD
000039             ops: ListOpsPtr;
000040             lstore: longint;
000041         END;
000042
000043     PROCEDURE AddItem(list: ListHandle; LHead: Head; NewItem: Item);
000044
000045     PROCEDURE DeleteItem(list: ListHandle; LHead: Head; OldItem: Item);
000046
000047     PROCEDURE EachItem(list: ListHandle; LHead: Head; PROCEDURE
000048         Visit(LItem: Item));
000049
000050     FUNCTION FirstItem(list: ListHandle; LHead: Head): Item;
000051
000052     PROCEDURE InitList(list: ListHandle; NewList: Head);
000053
000054     PROCEDURE InitLStore(VAR list: ListHandle; LO: ListOpsPtr; LS: longint);
000055
000056     PROCEDURE InsertItem(list: ListHandle; LHead: Head; NewItem: Item);
000057
000058     FUNCTION ListEmpty(list: ListHandle; LHead: Head): boolean;
000059
000060     FUNCTION ListSize(list: ListHandle; LHead: Head): integer;
000061
```

Apple Lisa Computer Technical Information

```
000062     PROCEDURE ReverseList(list: ListHandle; LHead: Head);
000063
000064     *****
000065     *
000066     *                               THAT'S ALL FOLKS ...
000067     *
000068     *****
000069
```

End of File -- Lines: 69 Characters: 1923

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 LOWLEVEL.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : LOWLEVEL
000004 *
000005 *****
000006
000007 USES { $U+ } LOWLEVEL;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $IFC OSBuilt }
000016         { $U libsm/unitstd.obj } unitstd,
000017         { $U libdb/dbenv.obj } dbenv,
000018         { $U libdb/dbdecl1.obj } dbdecl1,
000019         { $U libos/syscall.obj } syscall,
000020         { $U libos/psyscall.obj } psyscall;
000021     { $ELSEC }
000022         { $U obj:dbenv.obj } dbenv,
000023         { $U obj:dbdecl1.obj } dbdecl1,
000024         { $U obj:syscall.obj } syscall;
000025     { $ENDC }
000026
000027     Type
000028
000029
000030         ptrpathname = ^pathname;
000031
000032     VAR
000033         prefix_length: integer;
000034         prefix:        pathname;
000035         passwd_length: integer;
000036         password:      e_name;
000037
000038     PROCEDURE setprefix(newprefix: ptrpathname);
000039
000040     PROCEDURE setpasswd(VAR newpasswd: e_name);
000041
000042     PROCEDURE prefix_name(fname: ptrpathname);
000043
000044 *****
000045 *
000046 *             THAT'S ALL FOLKS ...
000047 *
000048 *****
000049
```

End of File -- Lines: 49 Characters: 1120

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 MATHLIB.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : MATHLIB
000004 *
000005 *****
000006
000007 USES { $U+ } MATHLIB;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     { $C Copyright 1983, 1984, Apple Computer Inc. }
000015
000016     USES
000017         { $U- } { $U LibFP/NewFPLib } FPLib;
000018
000019     CONST
000020         RandModulus = 2147483647;
000021         LsigDigLen  = 30;
000022
000023     TYPE
000024         RoundPrecision = (ExtPrecision, DblPrecision, RealPrecision);
000025         FP_Type        = (TFP_byte, TFP_integer, TFP_longint, TFP_Comp, TFP_real,
000026             TFP_Double, TFP_Extended);
000027         Free_Format    = RECORD
000028             MaxSig: integer;
000029             Sig_FForm, Trail_Point, Int_EForm, Plus_EForm: boolean;
000030         END;
000031         P_QR_Record    = longint;
000032         LongSigDig     = string[LsigDigLen];
000033         LongDecimal    = RECORD
000034             sgn: 0..1;
000035             exp: integer;
000036             sig: LongSigDig;
000037         END;
000038
000039     PROCEDURE ASinX(VAR x: Extended);
000040
000041     PROCEDURE ACosX(VAR x: Extended);
000042
000043     PROCEDURE SinhX(VAR x: Extended);
000044
000045     PROCEDURE CoshX(VAR x: Extended);
000046
000047     PROCEDURE TanhX(VAR x: Extended);
000048
000049     PROCEDURE Abs2X(x, y: Extended; VAR z: Extended);
000050
000051     PROCEDURE ATan2X(x, y: Extended; VAR z: Extended);
000052
000053     FUNCTION NextRandom(lastrandom: longint): longint;
000054
000055     PROCEDURE ClearXcps;
000056
000057     PROCEDURE ClearHlts;
000058
000059     PROCEDURE SetPrecision(p: RoundPrecision);
000060
000061     FUNCTION GetPrecision: RoundPrecision;
```


Apple Lisa Computer Technical Information

```
000062
000063 PROCEDURE Math_Sort(first, last: integer; FUNCTION
000064 sorted(i, j: integer): boolean; PROCEDURE
000065 swap(i, j: integer)VAR error: boolean);
000066
000067 FUNCTION SignOfX(x: Extended): boolean;
000068
000069 FUNCTION FP_New(n: longint): longint;
000070
000071 PROCEDURE FP_Size(x: Extended; VAR sgn: integer; VAR class: NumClass;
000072 VAR size: FP_Type);
000073
000074 PROCEDURE FP_Free_Ascii(x: Extended; width: integer; form: Free_Format;
000075 VAR s: Decstr);
000076
000077 PROCEDURE Fin_Npv(first, last, net: integer; rate: Extended;
000078 VAR Npv: Extended; PROCEDURE
000079 payment(i: integer; VAR pmt: Extended));
000080
000081 PROCEDURE Fin_Return(first, last: integer; negperiod, posperiod: integer;
000082 negrate, posrate: Extended; VAR ncs: integer;
000083 VAR ret: Extended; PROCEDURE
000084 payment(i: integer; VAR pmt: Extended));
000085
000086 PROCEDURE Mat_Mult(n, p, m: integer; overlap: boolean; VAR error: boolean;
000087 PROCEDURE afetch(i, j: integer;
000088 VAR aij: Extended)PROCEDURE
000089 xfetch(i, j: integer; VAR xij: Extended)PROCEDURE
000090 bstore(i, j: integer; bij: Extended));
000091
000092 PROCEDURE QR_Factor(n, p: integer; pivot: boolean; VAR QR: P_QR_Record;
000093 PROCEDURE afetch(i, j: integer; VAR aij: Extended));
000094
000095 PROCEDURE QR_Condition(QR: P_QR_Record; VAR cond: Extended);
000096
000097 PROCEDURE QR_Determinant(QR: P_QR_Record; VAR det: Extended);
000098
000099 PROCEDURE QR_Solve(m: integer; QR: P_QR_Record; VAR error: boolean; PROCEDURE
000100 bfetch(i, j: integer; VAR bij: Extended); PROCEDURE
000101 xstore(i, j: integer; xij: Extended));
000102
000103 PROCEDURE QR_Residual(n, p: integer; m: integer; PROCEDURE
000104 afetch(i, j: integer; VAR aij: Extended); PROCEDURE
000105 bfetch(i, j: integer; VAR bij: Extended); PROCEDURE
000106 xfetch(i, j: integer; VAR xij: Extended); PROCEDURE
000107 rstore(i, j: integer; rij: Extended));
000108
000109 PROCEDURE QR_Improve(m: integer; QR: P_QR_Record; VAR error: boolean;
000110 PROCEDURE afetch(i, j: integer; VAR aij: Extended);
000111 PROCEDURE bfetch(i, j: integer; VAR bij: Extended);
000112 PROCEDURE xfetch(i, j: integer; VAR xij: Extended);
000113 PROCEDURE xstore(i, j: integer; xij: Extended));
000114
000115 PROCEDURE QR_Transolve(m: integer; QR: P_QR_Record; VAR error: boolean;
000116 PROCEDURE bfetch(i, j: integer; VAR bij: Extended);
000117 PROCEDURE xstore(i, j: integer; xij: Extended));
000118
000119 PROCEDURE QR_TransDeterminant(QR: P_QR_Record; VAR det: Extended);
000120
000121 PROCEDURE X2LDec(f: DecForm; x: Extended; VAR y: LongDecimal);
000122
000123 PROCEDURE LDec2X(prec: RoundPrecision; x: LongDecimal; VAR y: Extended);
000124
000125 PROCEDURE Math_Solve(est1, est2: Extended; VAR result: Extended; PROCEDURE
000126 f(x: Extended; VAR fx: Extended));
000127
```

Apple Lisa Computer Technical Information

```
000128 *****
000129 *
000130 *           THAT'S ALL FOLKS ...
000131 *
000132 *****
000133
```

End of File -- Lines: 133 Characters: 4707

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 MEMMAN.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : MEMMAN
000004 *
000005 *****
000006
000007 USES {$U+} MEMMAN;
000008
000009
000010 {$IFC isIntrin} INTRINSIC; {$ENDC}
000011
000012 INTERFACE
000013
000014     USES
000015         {$IFC SrcOnOS }
000016             (*$U OBJ/PASDEFS.OBJ *) PasDefs;
000017         {$ELSEC }
000018             (*$U OBJ: PASDEFS.OBJ *) PasDefs;
000019         {$ENDC }
000020
000021     CONST
000022
000023
000024         NullItem      = - 32768;
000025
000026     TYPE
000027         MMStatus      = (MMAlloc, MMFree);
000028         MMStore       = ^MMStatArr;
000029         PMMStatRow   = ^MMStatRow;
000030         MMStatArr    = ARRAY [ - 128..127] OF PMMStatRow;
000031         MMStatRow    = PACKED ARRAY [ - 128..127] OF MMStatus;
000032         MMHandle     = ^MMRec;
000033         MMRec        = RECORD
000034             maxElements, curElements: longint;
000035             FirstFree: longint;
000036             mstore: MMStore;
000037         END;
000038         MMbyte       = - 128..127;
000039         MMIntBytes   = RECORD
000040             CASE boolean OF
000041                 true:
000042                     (int: integer);
000043                 false:
000044                     (hi, lo: MMbyte)
000045             END;
000046
000047     PROCEDURE InitMem(VAR mem: MMHandle; numElements: longint; FUNCTION
000048         GetMore(n, row: integer): boolean);
000049
000050     FUNCTION MMAllocate(mem: MMHandle; FUNCTION
000051         GetMore(n, row: integer): boolean): integer;
000052
000053     PROCEDURE Reserve(mem: MMHandle; i: integer; FUNCTION
000054         GetMore(n, row: integer): boolean);
000055
000056     PROCEDURE Free(mem: MMHandle; i: integer);
000057
000058     FUNCTION IsFree(mem: MMHandle; i: integer): boolean;
000059
000060     PROCEDURE EachAllocated(mem: MMHandle; PROCEDURE
000061         visit(i: integer));
```

Apple Lisa Computer Technical Information

```
000062
000063     PROCEDURE HeapAlloc(VAR p: MemPtr; n: longint; m: longint);
000064
000065     FUNCTION MMRows(n: longint): integer;
000066
000067     FUNCTION NullMore(n, row: integer): boolean;
000068
000069 *****
000070 *
000071 *             THAT'S ALL FOLKS ...
000072 *
000073 *****
000074
```

End of File -- Lines: 74 Characters: 2079

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 MENUS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : MENUS
000004 *
000005 *****
000006
000007 USES {$U+} MENUS;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libhw/HWInt      } HWInt,
000016         {$U libsm/UnitStd   } UnitStd,
000017         {$U libsm/UnitHz    } UnitHz,
000018         {$U libqd/Storage   } Storage,
000019         {$U libqd/QuickDraw } QuickDraw,
000020         {$U libfm/FontMgr   } FontMgr,
000021         {$U libos/SysCall   } SysCall,
000022         {$U libwm/Events    } Events,
000023         {$U libwm/Windows   } Windows,
000024         {$U libwm/Folders   } Folders;
000025
000026     CONST
000027         maxItem      = 31;
000028         noMark       = 0;
000029         vertSpace    = 15;
000030         {$IFC wmos }
000031         checkMark    = ccCheck;
000032         appleMark    = ccApple;
000033         {$ELSE}
000034         checkMark    = 142;
000035         appleMark    = 143;
000036         {$ENDC}
000037
000038     TYPE
000039         MenuPtr      = ^MenuInfo;
000040         MenuInfo     = RECORD
000041             menuId: INTEGER;
000042             menuWidth: INTEGER;
000043             menuHeight: INTEGER;
000044             DrawProc: ProcPtr;
000045             ChooseProc: ProcPtr;
000046             enableFlags: PACKED ARRAY [0..maxItem] OF BOOLEAN;
000047             menuData: Handle;
000048             END;
000049         rMenuInfo     = ARRAY [0..0] OF MenuInfo;
000050         rMenuPtr      = ^rMenuInfo;
000051         ItemPtr       = ^Str255;
000052
000053     VAR
000054         menuDelay:    INTEGER;
000055         desktopMenu:  ARRAY [0..1] OF MenuInfo;
000056
000057     PROCEDURE CalcMenuSize(VAR menu: MenuInfo);
000058
000059     PROCEDURE CheckItem(VAR menu: MenuInfo; item: INTEGER; checked: BOOLEAN);
000060
000061     PROCEDURE ChooseTxtItem(VAR menu: MenuInfo; menuRect: Rect; hitPt: Point;
```

Apple Lisa Computer Technical Information

```
000062             VAR whichItem: INTEGER);
000063
000064     PROCEDURE ClearMenuBar;
000065
000066     PROCEDURE DeleteMenu(menuId: INTEGER);
000067
000068     PROCEDURE DisableItem(VAR menu: MenuInfo; item: INTEGER);
000069
000070     PROCEDURE DisableMenu(VAR menu: MenuInfo);
000071
000072     PROCEDURE DrawMenuBar;
000073
000074     PROCEDURE DrawTxtMenu(VAR menu: MenuInfo; menuRect: Rect);
000075
000076     PROCEDURE EnableItem(VAR menu: MenuInfo; item: INTEGER);
000077
000078     PROCEDURE EnableMenu(VAR menu: MenuInfo);
000079
000080     PROCEDURE GetItem(VAR menu: MenuInfo; item: INTEGER; itemString: ItemPtr);
000081
000082     PROCEDURE HiLiteMenu(menuId: INTEGER);
000083
000084     PROCEDURE InitMenus;
000085
000086     PROCEDURE InsertMenu(VAR menu: MenuInfo; beforeId: INTEGER);
000087
000088     PROCEDURE MarkItem(VAR menu: MenuInfo; item: INTEGER; whichMark: INTEGER);
000089
000090     PROCEDURE MenuSelect(startPt: Point; VAR whichMenu, whichItem: INTEGER);
000091
000092     PROCEDURE MenuKey(ch: CHAR; VAR whichMenu, whichItem: INTEGER);
000093
000094     PROCEDURE ReadMenu(VAR menuFile: TEXT; VAR menu: MenuInfo);
000095
000096     PROCEDURE SetItem(VAR menu: MenuInfo; item: INTEGER; itemString: ItemPtr);
000097
000098     *****
000099     *
000100     *             THAT'S ALL FOLKS ...
000101     *
000102     *****
000103
```

End of File -- Lines: 103 Characters: 2851

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 OBJIO.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : OBJIO
000004 *
000005 *****
000006
000007 USES {$U+} OBJIO;
000008
000009
000010 {$IFC isIntrin} INTRINSIC; {$ENDC}
000011
000012 INTERFACE
000013
000014     USES
000015         {$IFC SrcOnOS }
000016             (*$U OBJ/PASDEFS.OBJ *) PasDefs,
000017             (*$U OBJ/FILEIO.OBJ *) FileIO;
000018         {$ELSEC }
000019             (*$U OBJ:PASDEFS.OBJ *) PasDefs,
000020             (*$U OBJ:FILEIO.OBJ *) FileIO;
000021         {$ENDC }
000022     CONST
000023
000024
000025         TMAXSEGS      = 4096;
000026         TMAXUNITS     = 4096;
000027         TMAXFILES     = 1024;
000028         TMAXSYSS      = 127;
000029         DLTASEGS      = 128;
000030         DLTAUNITS     = 128;
000031         DLTAFILES     = 32;
000032         DLTASYSS      = 16;
000033
000034     TYPE
000035         BlockType     = (ModuleName, EndBlock, EntryPoint, External, StartAddress,
000036             CodeBlock, Relocation, CommonReloc, CommonDef,
000037             ShortExternal, DInitDef, DInitData, DInitRef, DInitDRef,
000038             QuickLoad, OldExecutable, LibModule, LibEntry, UnitBlock,
000039             InterfLoc, PhysicalExec, Executable, VersionCtrl,
000040             SegmentTable, UnitTable, SegLocation, OldUntLoc,
000041             StringBlock, MACExecutable, PackedCode, PackTable,
000042             DInitCRef,
000043         {$IFC isSAndE}
000044             SoftSystem,
000045         {$ENDC}
000046             UnitLocation, OSData, DebugSymbols, DebugEntry,
000047             DebugCommon, EOFMark, UnknownBlock);
000048
000049         VariantType   = (NoVariant, RefVariant, ShortRef, ModVariant, Comments,
000050             SegVariant, UnitVariant, IntfLocVariant, SegLocVariant,
000051             UnitLocVariant, OldULVariant, StringVariant,
000052             SftSysVariant, JumpTVariant, JTSegVariant, ObjectCode,
000053             ProcHeap, OldJumpTV, OldJTSegV, MACJumpTV, Jump4TV,
000054             UnknownVariant);
000055
000056         FileAddr      = longint;
000057         MemAddr       = longint;
000058         SegAddr       = longint;
000059
000060         ObjFileType   = (OldObjFile, IUDirectory, IULibrary, MainProg, IntrinUnit,
000061             RegUnit, RegModule, DbgFile, PITFile);
```

Apple Lisa Computer Technical Information

```
000062
000063     VersNum      = - 1..127;
000064     VersVal      = RECORD
000065                 CASE boolean OF
000066                 true:
000067                     (VersLong: longint);
000068                 false:
000069                     (System, Release, Version, InterFLevel: VersNum)
000070                 END;
000071
000072     ModVal        = RECORD
000073                 CASE boolean OF
000074                 true:
000075                     (ModInt: integer);
000076                 false:
000077                     (ImplLevel, FixLevel: VersNum)
000078                 END;
000079
000080     iRefVariant   = SegAddr;
000081     iShortRef     = integer;
000082     iModVariant   = integer;
000083
000084     iSegVariant   = RECORD
000085                 SegName: NameString;
000086                 SegNumber: integer;
000087                 Version1: longint;
000088                 Version2: longint;
000089                 END;
000090
000091     iUnitVariant  = RECORD
000092                 UnitName: NameString;
000093                 UnitNumber: integer;
000094                 UnitType: integer;
000095                 END;
000096
000097     iIntfLocVariant = RECORD
000098                 UnitName: NameString;
000099                 IfLoc: FileAddr;
000100                 END;
000101
000102     iSegLocVariant = RECORD
000103                 SegName: NameString;
000104                 SegNumber: integer;
000105                 UseDescPtr: longint;
000106                 UseCount: integer;
000107                 SysNumber: integer;
000108                 FileNumber: integer;
000109                 FileLocation: FileAddr;
000110                 SizePacked: integer;
000111                 SizeUnpacked: integer;
000112                 END;
000113
000114     iOldULVariant = RECORD
000115                 UnitName: NameString;
000116                 UnitNumber: integer;
000117                 FileNumber, UnitType: FileByte;
000118                 DataSize: longint;
000119                 END;
000120
000121     iUnitLVariant = RECORD
000122                 UnitName: NameString;
000123                 UnitNumber: integer;
000124                 ShrdDataPtr: longint;
000125                 UseCount, SysNumber, FileNumber, DataPtrIdx,
000126                 UnitType: integer;
000127                 DataSize: longint;
```


Apple Lisa Computer Technical Information

```
000128             END;
000129
000130     {$IFC issAndE}
000131     iSftSysVariant = RECORD
000132             DSysIdx: integer;
000133             DVersId: longint;
000134             END;
000135     {$ENDC}
000136
000137     iStringVariant = RECORD
000138             ObjectNumber: integer;
000139             NameAddr: FileAddr;
000140             END;
000141
000142     iJumpTVariant = RECORD
000143             JumpL: integer;
000144             AbsAddr: MemAddr;
000145             END;
000146
000147     iOldJumpTV    = RECORD
000148             RelOffset: longint;
000149             Noop: integer;
000150             Jump: integer;
000151             PCRel: integer;
000152             END;
000153
000154     iMACJumpTV    = RECORD
000155             RelOffset: integer;
000156             MoveW: integer;
000157             SegNum: integer;
000158             TrapInst: integer;
000159             END;
000160
000161     iJump4TV      = RECORD
000162             JumpPC: integer;
000163             RelOffSet: integer;
000164             END;
000165
000166     iOldJTSegV    = RECORD
000167             Addr1: MemAddr;
000168             FileLoc: FileAddr;
000169             CodeSize: longint;
000170             MemLoc: MemAddr;
000171             RetAddr: MemAddr;
000172             RefCount: longint;
000173             ActiveList: MemAddr;
000174             Reserved: longint;
000175             END;
000176
000177     iJTSegVariant = RECORD
000178             SegmentAddr: FileAddr;
000179             SizePacked: integer;
000180             SizeUnpacked: integer;
000181             MemLoc: MemAddr;
000182             END;
000183
000184     iModuleName   = RECORD
000185             ModuleName, SegmentName: NameString;
000186             CSize: longint;
000187             END;
000188
000189     iEndBlock     = RECORD
000190             CSize: longint;
000191             END;
000192
000193     iEntryPoint   = RECORD
```

Apple Lisa Computer Technical Information

```
000194             LinkName, UserName: NameString;
000195             Loc: SegAddr;
000196             END;
000197
000198 iExternal      = RECORD
000199             LinkName, UserName: NameString;
000200             END;
000201
000202 iStartAddress = RECORD
000203             Start: SegAddr;
000204             GSize: longint;
000205             END;
000206
000207 iCodeBlock     = RECORD
000208             Addr: SegAddr;
000209             END;
000210
000211 iRelocation    = RECORD
000212             Reserved: integer;
000213             END;
000214
000215 iCommonRelocation = RECORD
000216             CommonName: NameString;
000217             END;
000218
000219 iCommonDefinition = RECORD
000220             CommonName: NameString;
000221             DSize: longint;
000222             END;
000223
000224 iShortExternal = RECORD
000225             LinkName, UserName: NameString;
000226             END;
000227
000228 iDInitDef      = RECORD
000229             d_areaName: NameString;
000230             DSize: longint;
000231             END;
000232
000233 iDInitData     = RECORD
000234             d_areaName: NameString;
000235             dAddr: longint;
000236             DSize: longint;
000237             END;
000238
000239 iDInitRef      = RECORD
000240             d_areaName: NameString;
000241             END;
000242
000243 iDInitDRef     = RECORD
000244             d_areaName: NameString;
000245             d_Off: longint;
000246             rAreaName: NameString;
000247             END;
000248
000249 iQuickLoad     = RECORD
000250             StartLoc: SegAddr;
000251             DataSize: longint;
000252             END;
000253
000254 iLibModule     = RECORD
000255             ModuleName: NameString;
000256             ModSize: longint;
000257             CodeAddr, TextAddr: FileAddr;
000258             TextSize: longint;
000259             NrMods: integer;
```

Apple Lisa Computer Technical Information

```
000260             END;
000261
000262     iLibEntry      = RECORD
000263         LinkName: NameString;
000264         Module: integer;
000265         Address: SegAddr;
000266     END;
000267
000268     iUnitBlock     = RECORD
000269         UnitName: NameString;
000270         CodeAddr, TextAddr: FileAddr;
000271         TextSize, GlobalSize: longint;
000272         UnitType: integer;
000273     END;
000274
000275     iInterfLoc     = RECORD
000276         Reserved: integer;
000277     END;
000278
000279     iExecutable    = RECORD
000280         JTLaddr: MemAddr;
000281         JTSIZE, DataSize, MainSize, JTSegDelta, StkSegDelta,
000282         DynStack, MaxStack, MinHeap, MaxHeap: longint;
000283     END;
000284
000285     iOldExecutable = RECORD
000286         JTLaddr: MemAddr;
000287         JTSIZE, DataSize: longint;
000288     END;
000289
000290     iPhysicalExec  = RECORD
000291         JTLaddr: MemAddr;
000292         JTSIZE, DataSize, MainSize, JTSegDelta,
000293         StkSegDelta: longint;
000294     END;
000295
000296     iMACExecutable = RECORD
000297         JTLaddr: MemAddr;
000298         JTSIZE, DataSize, MainSize, JTSegDelta,
000299         StkSegDelta: longint;
000300     END;
000301
000302     iVersionCtrl   = RECORD
000303         CASE boolean OF
000304             true:
000305                 (sysNum, minSys, maxSys, Reserv1, Reserv2,
000306                 Reserv3: longint);
000307             false:
000308                 (FileType: ObjFileType;
000309                 ILibNum: VersNum;
000310                 Config: VersVal;
000311                 ModLevel: ModVal;
000312                 IntfTime, ImplTime: longint;
000313                 MinConfig, MaxConfig: VersVal);
000314     END;
000315
000316     iSegmentTable  = RECORD
000317         nSegments: integer;
000318     END;
000319
000320     iUnitTable     = RECORD
000321         nUnits, maxunit: integer;
000322     END;
000323
000324     iSegLocation   = RECORD
000325         nSegments: integer;
```

Apple Lisa Computer Technical Information

```
000326             END;
000327
000328     iOldUntLoc    = RECORD
000329                 nUnits: integer;
000330             END;
000331
000332     iUnitLocation = RECORD
000333                 nUnits: integer;
000334             END;
000335
000336     iOSData       = RECORD
000337                 Reserved: integer;
000338             END;
000339
000340     iStringBlock  = RECORD
000341                 nStrings: integer;
000342             END;
000343
000344     {$IFC isSAndE}
000345     iSoftSystem   = RECORD
000346                 SysNumber: integer;
000347                 PublicSeg: integer;
000348                 SysVersion: VersVal;
000349                 Reserved: longint;
000350             END;
000351     {$ENDC}
000352
000353     iPackedCode   = RECORD
000354                 Addr: MemAddr;
000355                 CSize: longint;
000356             END;
000357
000358     iPackTable    = RECORD
000359                 packversion: longint;
000360             END;
000361
000362     IDInitCRef    = RECORD
000363                 d_areaName: NameString;
000364                 d_Off: longint;
000365                 entryName: NameString;
000366             END;
000367
000368     iDebugSymbols = RECORD
000369                 UserName, SegName: NameString;
000370                 ProcBase, ProcSyms, ProcStmt, ProcNode,
000371                 UsesSize: longint;
000372                 HoleBase, HoleTop, MapBase, MapTop: longint;
000373                 MapName: NameString;
000374             END;
000375
000376     iDebugEntry   = RECORD
000377                 UserName: NameString;
000378                 EntrySeg: longint;
000379                 EntryLoc: SegAddr;
000380             END;
000381
000382     iDebugCommon  = RECORD
000383                 UnitName: NameString;
000384                 CommonBase: MemAddr;
000385             END;
000386
000387     iUnknown      = RECORD
000388                 { nothing here }
000389             END;
000390
000391     ObjBlock      = RECORD
```

Apple Lisa Computer Technical Information

```

000392      Variant      : VariantType;
000393      NrVariants: LongInt;
000394
000395      CASE BlockHeader: BlockType OF
000396          ModuleName  : (bModuleName  : iModuleName);
000397          EndBlock    : (bEndBlock    : iEndBlock);
000398          EntryPoint  : (bEntryPoint  : iEntryPoint);
000399          External    : (bExternal    : iExternal);
000400          StartAddress : (bStartAddress : iStartAddress);
000401          CodeBlock   : (bCodeBlock   : iCodeBlock);
000402          Relocation  : (bRelocation  : iRelocation);
000403          CommonReloc : (bCommonReloc : iCommonReloc);
000404          CommonDef   : (bCommonDef   : iCommonDef);
000405          ShortExternal: (bShortExternal: iShortExternal);
000406          DInitDef    : (bDInitDef    : iDInitDef);
000407          DInitData   : (bDInitData   : iDInitData);
000408          DInitRef    : (bDInitRef    : iDInitRef);
000409          DInitDRef   : (bDInitDRef   : iDInitDRef);
000410          QuickLoad   : (bQuickLoad   : iQuickLoad);
000411          OldExecutable: (bOldExecutable: iOldExecutable);
000412          LibModule   : (bLibModule   : iLibModule);
000413          LibEntry    : (bLibEntry    : iLibEntry);
000414          UnitBlock   : (bUnitBlock   : iUnitBlock);
000415          InterfLoc   : (bInterfLoc   : iInterfLoc);
000416          PhysicalExec : (bPhysicalExec : iPhysicalExec);
000417          MACExecutable: (bMACExecutable: iMACExecutable);
000418          Executable  : (bExecutable  : iExecutable);
000419          VersionCtrl : (bVersionCtrl : iVersionCtrl);
000420          SegmentTable : (bSegmentTable : iSegmentTable);
000421          UnitTable   : (bUnitTable   : iUnitTable);
000422          SegLocation : (bSegLocation : iSegLocation);
000423          UnitLocation : (bUnitLocation : iUnitLocation);
000424          OldUntLoc   : (bOldUntLoc   : iOldUntLoc);
000425          StringBlock : (bStringBlock : iStringBlock);
000426          PackedCode  : (bPackedCode  : iPackedCode);
000427          PackTable   : (bPackTable   : iPackTable);
000428          DInitCRef   : (bDInitCRef   : iDInitCRef);
000429          {$IFC isSAndE}
000430          SoftSystem  : (bSoftSystem  : iSoftSystem);
000431          {$ENDC}
000432          OSData      : (bOSData      : iOSData);
000433          DebugSymbols : (bDebugSymbols : iDebugSymbols);
000434          DebugEntry   : (bDebugEntry   : iDebugEntry);
000435          DebugCommon  : (bDebugCommon  : iDebugCommon);
000436          UnknownBlock : (bUnknown    : iUnknownBlock);
000437      END;
000438
000439      ObjVarBlock = RECORD
000440          CASE VarHeader : VariantType OF
000441              RefVariant  : (bRefVariant  : iRefVariant);
000442              ShortRef    : (bShortRef    : iShortRef);
000443              ModVariant  : (bModVariant  : iModVariant);
000444              SegVariant  : (bSegVariant  : iSegVariant);
000445              UnitVariant : (bUnitVariant : iUnitVariant);
000446              IntfLocVariant: (bIntfLocVariant: iIntfLocVariant);
000447              SegLocVariant : (bSegLocVariant : iSegLocVariant);
000448              UnitLocVariant: (bUnitLVariant : iUnitLVariant);
000449              {$IFC isSAndE}
000450              SftSysVariant : (bSftSysVariant : iSftSysVariant);
000451              {$ENDC}
000452              OldULVariant : (bOldULVariant : iOldULVariant);
000453              StringVariant : (bStringVariant : iStringVariant);
000454              OldJumpTV    : (bOldJumpTV    : iOldJumpTV);
000455              MACJumpTV    : (bMACJumpTV    : iMACJumpTV);
000456              Jump4TV      : (bJump4TV      : iJump4TV);
000457              OldJTSegV    : (bOldJTSegV    : iOldJTSegV);

```

Apple Lisa Computer Technical Information

```
000458             JumpTVariant : (bJumpTVariant : iJumpTVariant);
000459             JTSegVariant   : (bJTSegVariant   : iJTSegVariant)
000460             END;
000461
000462     ObjHandle     = ^ObjDesc;
000463
000464     ObjDesc       = RECORD
000465                 ObjFile: FileHandle;
000466                 NextBlock: FileAddr;
000467             END;
000468
000469     VAR
000470     VariantSize: ARRAY [VariantType] OF integer;
000471     OIAllowAbort: BOOLEAN;
000472
000473     PROCEDURE InitObjFile(VAR ObjPtr: ObjHandle; nBlocks: integer);
000474
000475     PROCEDURE OpenObjFile(VAR ObjPtr: ObjHandle; FileName: LString;
000476                           NewFile: boolean);
000477
000478     PROCEDURE ZeroObjEnd(ObjPtr: ObjHandle);
000479
000480     PROCEDURE CloseObjFile(ObjPtr: ObjHandle; Save: boolean);
000481
000482     PROCEDURE GetObjPtr(ObjPtr: ObjHandle; VAR BytePtr: FileAddr);
000483
000484     PROCEDURE GetObjBlockPtr(ObjPtr: ObjHandle; VAR BytePtr: FileAddr);
000485
000486     PROCEDURE SetObjPtr(ObjPtr: ObjHandle; BytePtr: FileAddr);
000487
000488     PROCEDURE SetObjBlockPtr(ObjPtr: ObjHandle; BytePtr: FileAddr);
000489
000490     PROCEDURE SkipObjBytes(ObjPtr: ObjHandle; NrBytes: longint);
000491
000492     PROCEDURE SetObjInvar(VAR B: ObjBlock; InvarType: BlockType;
000493                           VarSize: longint);
000494
000495     PROCEDURE CopyObjSeq(InObj, OutObj: ObjHandle; NrBytes: integer);
000496
000497     PROCEDURE GetObjInvar(ObjPtr: ObjHandle; VAR Stuff: ObjBlock);
000498
000499     PROCEDURE GetObjVar(ObjPtr: ObjHandle; VarType: VariantType;
000500                           VAR Stuff: ObjVarBlock);
000501
000502     PROCEDURE GetObjName(ObjPtr: ObjHandle; VAR N: NameString);
000503
000504     PROCEDURE GetObjSeq(ObjPtr: ObjHandle; Stuff: Ptr; NrBytes: integer);
000505
000506     PROCEDURE GetObjByte(ObjPtr: ObjHandle; VAR B: Byte);
000507
000508     PROCEDURE GetObjWord(ObjPtr: ObjHandle; VAR W: integer);
000509
000510     PROCEDURE GetObjLong(ObjPtr: ObjHandle; VAR L: longint);
000511
000512     PROCEDURE PutObjInvar(ObjPtr: ObjHandle; VAR Stuff: ObjBlock);
000513
000514     PROCEDURE PutObjVar(ObjPtr: ObjHandle; VarType: VariantType;
000515                           VAR Stuff: ObjVarBlock);
000516
000517     PROCEDURE PutObjName(ObjPtr: ObjHandle; N: NameString);
000518
000519     PROCEDURE PutObjSeq(ObjPtr: ObjHandle; Stuff: Ptr; NrBytes: integer);
000520
000521     PROCEDURE PutObjByte(ObjPtr: ObjHandle; B: Byte);
000522
000523     PROCEDURE PutObjWord(ObjPtr: ObjHandle; W: integer);
```

Apple Lisa Computer Technical Information

```
000524
000525     PROCEDURE PutObjLong(ObjPtr: ObjHandle; L: longint);
000526
000527     *****
000528     *
000529     *                               THAT'S ALL FOLKS ...
000530     *
000531     *****
000532
```

End of File -- Lines: 532 Characters: 19082

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PARTS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PARTS
000004 *
000005 *****
000006
000007 USES {$U+} PARTS;
000008
000009
000010 {$IFC isIntrin} INTRINSIC; {$ENDC}
000011
000012 INTERFACE
000013
000014     USES
000015         {$IFC SrcOnOS }
000016             (*$U OBJ/PASDEFS.OBJ *) PasDefs,
000017             (*$U OBJ/MEMMAN.OBJ *) MemMan,
000018             (*$U OBJ/LISTS.OBJ *) Lists;
000019         {$ELSEC }
000020             (*$U OBJ: PASDEFS.OBJ *) PasDefs,
000021             (*$U OBJ: MEMMAN.OBJ *) MemMan,
000022             (*$U OBJ: LISTS.OBJ *) Lists;
000023         {$ENDC }
000024
000025     TYPE
000026
000027
000028         Class           = integer;
000029         Member          = integer;
000030         MemberRec      = RECORD
000031             memberOf: Class;
000032             nextMember: Member;
000033             firstMember: Member;
000034         END;
000035         MemRowPtr      = ^MemRow;
000036         MemRow         = ARRAY [ - 128..127] OF MemberRec;
000037         MemArrPtr      = ^MemArr;
000038         MemArr         = ARRAY [ - 128..127] OF MemRowPtr;
000039         PRHandle       = ^Partition;
000040         Partition      = RECORD
000041             nClasses, nMembers: longint;
000042             classes, members: MemArrPtr;
000043         END;
000044         PartHandle     = ^PartObject;
000045         PartObject     = RECORD
000046             storP: PRHandle;
000047             partL: ListHandle;
000048         END;
000049
000050     PROCEDURE NewClass(partH: PartHandle; cl: Class);
000051
000052     PROCEDURE AddMember(partH: PartHandle; mem: Member; cl: Class);
000053
000054     PROCEDURE DeleteMember(partH: PartHandle; mem: Member; cl: Class);
000055
000056     FUNCTION IsEmptyClass(partH: PartHandle; cl: Class): boolean;
000057
000058     PROCEDURE EachMember(partH: PartHandle; cl: Class; PROCEDURE
000059         Visit(m: Member));
000060
000061     FUNCTION ClassOf(partH: PartHandle; mem: Member): Class;
```


Apple Lisa Computer Technical Information

```
000062
000063 PROCEDURE RevMembers(partH: PartHandle; cl: Class);
000064
000065 PROCEDURE InitPLists;
000066
000067 FUNCTION MoreMembers(partH: PartHandle; n, newRow: integer): boolean;
000068
000069 FUNCTION MoreClasses(partH: PartHandle; n, newRow: integer): boolean;
000070
000071 FUNCTION InitPartition(nCl, nMem: longint; abst: PartHandle): PartHandle;
000072
000073 *****
000074 *
000075 *                THAT'S ALL FOLKS ...
000076 *
000077 *****
000078
```

End of File -- Lines: 78 Characters: 2253

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PASDEFS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PASDEFS
000004 *
000005 *****
000006
000007 USES { $U+ } PASDEFS;
000008
000009
000010 { $IFC isIntrin } INTRINSIC; { $ENDC }
000011
000012 INTERFACE
000013
000014     USES
000015         { $IFC SrcOnOS }
000016         { $U OBJ/STDUNIT } STDUNIT;
000017         { $ELSEC }
000018         { $U OBJ:STDUNIT } STDUNIT;
000019         { $ENDC }
000020
000021     { $SETC RANGEF = 0 }
000022     { $SETC DEBUGF = 0 }
000023     { $SETC INFOF = 1 }
000024
000025     CONST
000026
000027
000028         NameStrLen    = 8;
000029         MaxLStringLen = 79;
000030         Blank         = ' ';
000031         Empty         = '';
000032
000033     TYPE
000034         NameString    = PACKED ARRAY [1..NameStrLen] OF char;
000035         LString       = String[MaxLStringLen];
000036         PLString      = ^LString;
000037         MemPtr        = ^integer;
000038         ProcPtr       = ^integer;
000039
000040     VAR
000041         errors:       integer;
000042         ListingFile: Text;
000043         ListFlag:     boolean;
000044
000045     PROCEDURE InitPasDefs;
000046
000047     PROCEDURE PasHalt;
000048
000049     PROCEDURE SumErrors;
000050
000051     PROCEDURE Warning(s: LString);
000052
000053     PROCEDURE Error(s: LString);
000054
000055     PROCEDURE FatalError(s: LString);
000056
000057 *****
000058 *
000059 *                 THAT'S ALL FOLKS ...
000060 *
000061 *****
```

Apple Lisa Computer Technical Information

000062

End of File -- Lines: 62 Characters: 1217

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PASHEAP.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PASHEAP
000004 *
000005 *****
000006
000007 USES {$U+} PASHEAP;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         (*$U libos/syscall.obj*) syscall;
000016
000017     TYPE
000018         PHTB          = - 128..127;
000019         PHTP          = ^PHTB;
000020
000021     PROCEDURE PLINITHEAP(VAR ernum, refnum: integer; size, delta: longint;
000022                         ldsn: integer; swapable: boolean);
000023
000024     PROCEDURE %_NEW(VAR ptr: PHTP; size: integer);
000025
000026     PROCEDURE %_NEWL(VAR ptr: PHTP; lsize: longint);
000027
000028     PROCEDURE %_MARK(VAR ptr: PHTP);
000029
000030     PROCEDURE %_RELSE(VAR ptr: PHTP);
000031
000032     FUNCTION %_MEMAV: LONGINT;
000033
000034     FUNCTION %_HHeapRes: integer;
000035
000036     PROCEDURE %_pphpreinit;
000037
000038     PROCEDURE %_pphreinit;
000039
000040     FUNCTION %_phwordsavail: longint;
000041
000042     PROCEDURE %_disp(VAR ptr: PHTP; size: longint);
000043
000044 *****
000045 *
000046 *             THAT'S ALL FOLKS ...
000047 *
000048 *****
000049
```

End of File -- Lines: 49 Characters: 1057

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PMDECL.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PMDECL
000004 *
000005 *****
000006
000007 USES { $U+ } PMDECL;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015     { $U libos/syscall } syscall;
000016
000017     CONST
000018     ThreshHold    = 8;
000019
000020     TYPE
000021     TNibble       = 0..15;
000022     TBootStuff   = RECORD
000023         ExtendMem: Boolean;
000024         BootVol: TNibble;
000025     END;
000026     TOutPutStuff = RECORD
000027         NormCont: TNibble;
000028         DimCont: TNibble;
000029         BeepVol: TNibble;
000030         FadeDelay: TNibble;
000031     END;
000032     TInputStuff  = RECORD
000033         BeginRepeat: TNibble;
000034         SubRepeat: TNibble;
000035         DoubleClick: TNibble;
000036     END;
000037     TInternalStuff = RECORD
000038         Mem_Loss: integer;
000039         Mouseon: Boolean;
000040         ScaleMouse: Boolean;
000041     END;
000042     TypeConnect  = string[32];
000043     Ch_info      = RECORD
000044         version: integer;
000045         driver_id: longint;
000046         bootable: boolean;
000047         preload: boolean;
000048         permanent: boolean;
000049         devicetype: devtype;
000050         size_exten: integer;
000051         def_cdinfo: ARRAY [1..3] OF integer;
000052         removable: boolean;
000053         ejectable: boolean;
000054         fs_start_block: longint;
000055         start_block: longint;
000056         numattach: integer;
000057         numConnectors: integer;
000058     END;
000059     cddEntry     = RECORD
000060         drv_name: e_name;
000061         active: integer;
```

Apple Lisa Computer Technical Information

```
000062             info: ch_info;
000063             END;
000064     cddHeader    = RECORD
000065                 version: integer;
000066                 cdcount: integer;
000067             END;
000068     systemCdd    = RECORD
000069                 descr: cddheader;
000070             END;
000071
000072 *****
000073 *
000074 *             THAT'S ALL FOLKS ...
000075 *
000076 *****
000077
```

End of File -- Lines: 77 Characters: 2435

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PMM.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PMM
000004 *
000005 *****
000006
000007 USES {$U+} PMM;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libhw/hwint} HWInt,
000016         {$U libos/Syscall} SysCall,
000017         {$U libsm/UnitStd} UnitStd,
000018         {$U libpm/PMDecl} PMDecl;
000019         {$SETC DOPXPM := FALSE}
000020
000021     VAR
000022         PMBeep_Flash: TNibble;
000023
000024     PROCEDURE PmInit(VAR error: integer);
000025
000026     PROCEDURE PMReadConfig(VAR error: integer; VAR NextEntry: longint;
000027                             VAR Config: ConfigDev);
000028
000029     PROCEDURE PMWriteConfig(VAR error: integer; VAR NextEntry: longint;
000030                             VAR Config: ConfigDev);
000031
000032     PROCEDURE PMReadInput(VAR InputInfo: TInputStuff);
000033
000034     PROCEDURE PMWriteInput(InputInfo: TInputStuff);
000035
000036     PROCEDURE PMReadOutPut(VAR OutputInfo: TOutputStuff);
000037
000038     PROCEDURE PMWriteOutPut(OutputInfo: TOutputStuff);
000039
000040     PROCEDURE PMReadBoot(VAR BootInfo: TBootStuff);
000041
000042     PROCEDURE PMWriteBoot(BootInfo: TBootStuff);
000043
000044     PROCEDURE PMWrite;
000045
000046     FUNCTION DimConvert(Contrast: Integer): Integer;
000047
000048     FUNCTION ConvertCont(Contrast: Integer): Integer;
000049
000050     PROCEDURE PMReadDevolp(VAR IntrnlInfo: TInternalStuff);
000051
000052     PROCEDURE PMWriteDevolp(IntrnlInfo: TInternalStuff);
000053
000054     PROCEDURE DefaultPM;
000055
000056     PROCEDURE DefaultConfig;
000057     {$IFC DOPXPM}
000058
000059     PROCEDURE PxPm;
000060     {$ENDC}
000061
```

Apple Lisa Computer Technical Information

```
000062 *****  
000063 *  
000064 *          THAT'S ALL FOLKS ...  
000065 *  
000066 *****  
000067
```

End of File -- Lines: 67 Characters: 1564

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 POOLER.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : POOLER
000004 *
000005 *****
000006
000007 USES { $U+ } POOLER;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $IFC OSBuilt }
000016         { $U libsm/unitstd.obj } unitstd,
000017         { $U libdb/dbenv.obj } dbenv,
000018         { $U libin/INTRLIBp.obj } international,
000019         { $U libdb/dbdecl1.obj } dbdecl1,
000020         { $U libos/syscall.obj } syscall,
000021         { $U libos/psyscall.obj } psyscall,
000022         { $U libdb/lowlevel.obj } lowlevel;
000023     { $ELSEC }
000024         { $U OBJ:dbenv.obj } dbenv,
000025         { $U INTRLIB.obj } international,
000026         { $U OBJ:dbdecl1.obj } dbdecl1,
000027         { $U OBJ:syscall.obj } syscall,
000028         { $U OBJ:lowlevel.obj } lowlevel;
000029     { $ENDC }
000030
000031     Const
000032
000033
000034         { $SETC NOSWAP := FALSE }
000035         { $SETC debug := FALSE }
000036         { $IFC PRERELEASE }
000037         { $SETC debugpool := TRUE }
000038         { $ELSEC }
000039         { $SETC debugpool := FALSE }
000040         { $ENDC }
000041     V0myprod_vers = 0;
000042     myprod_vers   = 1;
000043     nilptr        = - 1;
000044     nilval        = - 1;
000045     fisbtree     = 0;
000046     fhasnull     = 1;
000047     fmaster      = 2;
000048     fdetail      = 3;
000049     ffileok      = 4;
000050     fdata        = 5;
000051     writenon     = 5;
000052     noerr        = 0;
000053     notfound     = 1;
000054     duplkey      = 2;
000055     eos          = 3;
000056     fileempty    = 4;
000057     nobuffs      = 3404;
000058     ioerr        = 3405;
000059     noroom       = 3406;
000060     inconsistent = 3407;
000061     badscanid    = 3408;
```

Apple Lisa Computer Technical Information

```
000062      ticketoverflow = 3409;
000063      ticketchanged = 3410;
000064      badrecaddress = 3412;
000065      oddrecsize    = 3413;
000066      badnsearch    = 3414;
000067      badwhich      = 3415;
000068      illegalscan   = 3416;
000069      rectoobig     = 3417;
000070      notdbfile     = 3418;
000071      wrongfile     = 3419;
000072      badversion    = 3420;
000073      hasschema     = 3421;
000074      badpagesize   = 3425;
000075      DBVersionMismatch = 3426;
000076      DBTooOld      = 3427;
000077      DBTooNew      = 3428;
000078      empty         = 0;
000079      clean         = 1;
000080      dirty         = 2;
000081      readonly     = 0;
000082      update        = 1;
000083      labelreadonly = 2;
000084      labelupdate   = 3;
000085      first         = 0;
000086      last          = 1;
000087      approx        = 2;
000088      exact         = 3;
000089      current       = 4;
000090      prior         = 5;
000091      next          = 6;
000092      touch_by_vltree = 0;
000093      touch_delete  = 1;
000094      untouched     = 2;
000095      leaftype      = - 26215;
000096      nonleaftype   = - 26729;
000097      labeltype     = - 25701;
000098      freetype      = - 25187;
000099      labelkeys     = 3;
000100      labelflds     = 4;
000101      labelbnd     = 3;
000102
000103      TYPE
000104      uniqueid      = uid;
000105      filelabel     = RECORD
000106                  check_1_it: integer;
000107                  DBvers: byte;
000108                  USERvers: byte;
000109                  updateversion: integer;
000110                  flags: elemset;
000111                  firstpage, lastpage: fileptr;
000112                  nrecs: fileptr;
000113                  pagesize: integer;
000114                  nfields: integer;
000115                  nkeys: integer;
000116                  nsort: integer;
000117                  vflds: integer;
000118                  kvflds: integer;
000119                  depth: integer;
000120                  root: fileptr;
000121                  freelist: fileptr;
000122                  labellist: fileptr;
000123                  labeltickets: fileptr;
000124                  tickets: ticket;
000125                  ticketfld: integer;
000126                  funiqueid: uniqueid;
000127                  check_2_it: integer;
```

Apple Lisa Computer Technical Information

```
000128           END;
000129   pagedesc      = RECORD
000130               nkeys: integer;
000131               ptype: integer;
000132               CASE boolean OF
000133                 true:
000134                   (prior: fileptr;
000135                    next: fileptr);
000136                 false:
000137                   (free: fileptr; )
000138           END;
000139   ptrpagedesc    = ^pagedesc;
000140   vacttype       = (undone, del, split, split_replace, rot, rot_replace,
000141                    irot, irot_replace, merge, merge_replace, NoSibEmpty);
000142   stackrec       = RECORD
000143               numkeys, index: integer;
000144               page: fileptr;
000145               rec_lpage, rec_rpage: fileptr;
000146               CASE vact: vacttype OF
000147                 irot, irot_replace:
000148                   (unrotPt: integer;
000149                    insindex: integer);
000150                 del, rot, merge, NoSibEmpty:
000151                   (delsize: integer;
000152                    delrec: ptrdata);
000153           END;
000154   ptrstackrec    = ^stackrec;
000155   pagestack      = ARRAY [0..0] OF stackrec;
000156   ptrpagestack   = ^pagestack;
000157   labelentry     = RECORD
000158               ltype: integer;
000159               enum: fileptr;
000160               seq: integer;
000161               varfld: vfld
000162           END;
000163   ptrlabelentry  = ^labelentry;
000164   offsets        = ARRAY [0..0] OF integer;
000165   ptroffsets     = ^offsets;
000166   ptrscandesc    = ^scandesc;
000167   ptrfiledesc    = ^filedesc;
000168   filedesc       = RECORD
000169               check_1_it: integer;
000170               DBvers: byte;
000171               USERvers: byte;
000172               updateversion: integer;
000173               flags: elemset;
000174               firstpage, lastpage: fileptr;
000175               nrecs: fileptr;
000176               pagesize: integer;
000177               nfields: integer;
000178               nkeys: integer;
000179               nsort: integer;
000180               vflds: integer;
000181               kvflds: integer;
000182               depth: integer;
000183               root: fileptr;
000184               freelist: fileptr;
000185               labellist: fileptr;
000186               labeltickets: fileptr;
000187               tickets: ticket;
000188               ticketfld: integer;
000189               funiqueid: uniqueid;
000190               check_2_it: integer;
000191               openid: uid;
000192               pnext: integer;
000193               pkeyfiles: integer;
```

Apple Lisa Computer Technical Information

```
000194         readers, writers: integer;
000195         fstatus: integer;
000196         scans: integer;
000197         frefno: integer;
000198         pedesc: ptrrecptr;
000199         pkdesc: ptrrecptr;
000200         pfldset: ptrbits;
000201         firstvfld: integer;
000202         lastvfld: integer;
000203         lastvsfld: integer;
000204         END;
000205     scandesc = RECORD
000206         onfile, viafile: integer;
000207         fnext: integer;
000208         touched: integer;
000209         intent: integer;
000210         position: integer;
000211         ptrcos: ptrdata;
000212         cossize: integer;
000213         keypage: fileptr;
000214         qpage: fileptr;
000215         keyindex: integer;
000216         qindex: integer;
000217         ufldset: bits;
000218     END;
000219     lmhandle = fint8;
000220     diffdesc = RECORD
000221         diffid: integer;
000222         realid: integer;
000223         on, via: integer;
000224         vpage: fileptr;
000225         vindex: integer;
000226         curhandle: lmhandle;
000227         have_an_update, reversescan: boolean;
000228         whichway: integer;
000229         pdfilter: ptrdata;
000230         pkfilter: ptrdata;
000231         plow, phigh: ptrdata;
000232         nlflds, nhflds: integer;
000233         CheckPtSize: integer;
000234         nullset: bits;
000235     END;
000236     ptrdiffdesc = ^diffdesc;
000237     QChkPtDesc = RECORD
000238         CPtSize: integer;
000239         CPtLowSize: integer;
000240         CPtHighSize: integer;
000241         CPTDfilterSize: integer;
000242         CPTKfilterSize: integer;
000243         CPTnhflds, CPTnlflds: integer;
000244         CPTwhichway: integer;
000245         CPTvpage: fileptr;
000246         CPTvindex: integer;
000247         CPTcurhandle: lmhandle;
000248         CPTreversescan: boolean;
000249         hasVia: boolean;
000250         Clear: boolean;
000251         Good: boolean;
000252         viaUniqueid: uniqueid;
000253     END;
000254     ptrQChkPtDesc = ^QChkPtDesc;
000255     buffdesc = RECORD
000256         pbuff: ptrdata;
000257         state: integer;
000258         locks: integer;
000259         pfile: integer;
```

Apple Lisa Computer Technical Information

```
000260             page: fileptr;
000261             priority: integer;
000262             END;
000263     pool       = ARRAY [0..0] OF buffdesc;
000264     ptrpool    = ^pool;
000265     filetable  = ARRAY [0..0] OF ptrfiledesc;
000266     ptrfiletable = ^filetable;
000267     scantable  = ARRAY [0..0] OF ptrscandesc;
000268     ptrscantable = ^scantable;
000269     difftable  = ARRAY [0..0] OF ptrdiffdesc;
000270     ptrdifftable = ^difftable;
000271
000272     VAR
000273     { $IFC PRERELEASE }
000274     pdebug:      boolean;
000275     { $ENDC }
000276     psize:       integer;
000277     popenobj:    integer;
000278     pfiletable: ptrfiletable;
000279     nfiles:      integer;
000280     pscantable:  ptrscantable;
000281     nscans:      integer;
000282     pdifftable: ptrdifftable;
000283     ndiffs:      integer;
000284     lab_desc:    ARRAY [0..labelbnd] OF flddesc;
000285     pstack:      ptrpagestack;
000286     curdepth, maxdepth: integer;
000287     pbuff_segment: longint;
000288     buff_segment: integer;
000289     buffpool:    ptrpool;
000290     ticks:       integer;
000291     lastbuff:    integer;
000292     uiddesc:     flddesc;
000293     isleaf, isnonleaf, isroot: integer;
000294     dont_use_freelist: boolean;
000295     { $IFC DEBUGPOOL }
000296     labwrites, datawrites, datareads: integer;
000297     { $ENDC }
000298     { $IFC PRERELEASE }
000299
000300     PROCEDURE dumpbuff(first, last: integer);
000301     { $ENDC }
000302
000303     PROCEDURE lockbuff(VAR presult: integer; VAR buffer: integer);
000304
000305     PROCEDURE relbuff(VAR presult: integer; buffer: integer;
000306                       relpriority: integer);
000307
000308     PROCEDURE dowritenon(VAR presult: integer; fileid: integer);
000309
000310     PROCEDURE getpage(VAR presult: integer; VAR buffer: integer; fileid: integer;
000311                      rpage: fileptr; noinconsistent: boolean);
000312
000313     PROCEDURE putpage(VAR presult: integer; buffer: integer; fileid: integer;
000314                      wpage: fileptr);
000315
000316     PROCEDURE checkpt(VAR presult: integer; fileid: integer; newstate: integer);
000317
000318     PROCEDURE doclose(VAR presult: integer; fileid: integer);
000319
000320     FUNCTION no_room(presult: integer): boolean;
000321
000322     PROCEDURE force_out(VAR presult: integer; fileid: integer; wpage: fileptr);
000323
000324     PROCEDURE PageInvalidates(fileid: integer);
000325
```

Apple Lisa Computer Technical Information

```
000326 *****  
000327 *  
000328 *                THAT'S ALL FOLKS ...  
000329 *  
000330 *****  
000331
```

End of File -- Lines: 331 Characters: 11109

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PRBUF.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRBUF
000004 *
000005 *****
000006
000007 USES { $U+ } PRBUF;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U LibOS/SysCall      } SysCall,
000016         { $U LibPM/PMDecl      } PMDecl,
000017         { $U LibSM/UnitStd     } UnitStd,
000018         { $U LibSM/UnitHz      } UnitHz,
000019         { $U LibQD/Storage     } Storage,
000020         { $U LibQD/QuickDraw   } QuickDraw,
000021         { $U LibFM/FontMgr     } FontMgr,
000022         { $U LibPr/PrStdInfo   } PrStdInfo;
000023
000024     TYPE
000025         TPbCmd      = (pbCmdText, pbCmdHRule, pbCmdVRule);
000026         TPbCommand  = RECORD
000027             h: TC;
000028             v: TC;
000029             CASE pbCmd: TPbCmd OF
000030                 pbCmdText:
000031                     (cFont: TC;
000032                      seteface: Style;
000033                      cFirstChar: TC;
000034                      cLength: TC);
000035                 pbCmdHRule:
000036                     (cWidth: TC);
000037                 pbCmdVRule:
000038                     (cHeight: TC);
000039             END;
000040         TPbData      = PACKED ARRAY [0..32000] OF CHAR;
000041         TPPbData     = ^TPbData;
000042         THPbData     = ^TPPbData;
000043         TPbProcs     = ARRAY [TPbCmd] OF TProc;
000044         TPbInstall   = RECORD
000045             chkProcs: TPbProcs;
000046             dumpProcs: TPbProcs;
000047             scanLine: TProc;
000048         END;
000049
000050     VAR
000051         rPrBounds:   Rect;
000052         HOffset, VOffset: Integer;
000053
000054     PROCEDURE PbClose;
000055
000056     PROCEDURE PbDumpBand;
000057
000058     PROCEDURE PbHRule(cLength: TC);
000059
000060     PROCEDURE PbOffset(HOff, VOff: Integer);
000061
```

Apple Lisa Computer Technical Information

```
000062    PROCEDURE PbOpen(install: TPbInstall);
000063
000064    PROCEDURE PbText(cCount: TC; p: TP);
000065
000066    PROCEDURE PbVRule(cLength: TC);
000067
000068    *****
000069    *
000070    *                THAT'S ALL FOLKS ...
000071    *
000072    *****
000073
```

End of File -- Lines: 73 Characters: 2018

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PRDLGMGR.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRDLGMGR
000004 *
000005 *****
000006
000007 USES {$U+} PRDLGMGR;
000008
000009
000010 INTRINSIC;
000011     {$SETC fBtnRead = FALSE}
000012
000013 INTERFACE
000014
000015     USES
000016         {$U LibOS/SysCall      } SysCall,
000017         {$U LibSM/UnitStd     } UnitStd,
000018         {$U LibSM/UnitHz      } UnitHz,
000019         {$U LibHW/HWInt       } HWInt,
000020         {$U LibQD/QuickDraw   } QuickDraw,
000021         {$U LibFM/FontMgr     } FontMgr,
000022         {$U LibQD/Storage     } Storage,
000023         {$U LibWM/Events      } Events,
000024         {$U LibWM/Folders     } Folders,
000025         {$U LibSB/WmlStd      } WmlStd,
000026         {$U LibSB/WmlCrs      } WmlCrs,
000027         {$U LibPM/PMDecl      } PMDecl,
000028         {$U libPr/PrStdInfo   } PrStdInfo;
000029
000030     CONST
000031         cInfinite      = $7FFFFFFF;
000032         cZero          = 0;
000033         cCursorDelay   = 3500;
000034         cBtnMax        = 31;
000035         cMuMax         = 31;
000036         idMuMax        = 15;
000037         iBtnNil        = - 1;
000038         iMuNil         = - 1;
000039         iBtnStd        = - 2;
000040         dhCkfmMarg     = 6;
000041         dvCkfmMarg     = 4;
000042         dhSpToCk       = 4;
000043         dhSpToR        = 3;
000044         dvSpToR        = 1;
000045         cChNrMax       = 4;
000046         cChSpMax       = 15;
000047         dtDlgEnd       = 25;
000048         dtDlgMin       = 25;
000049         ckfmMax        = 5;
000050
000051     TYPE
000052         TDBox          = RECORD
000053             fKeyDlg: TF;
000054             timeStrt: TL;
000055             procEvent: TProc;
000056             procIdle: TProc;
000057             procEnd: TProc;
000058         END;
000059         TEbtn          = (ebtnSq, ebtnRR, ebtnOv);
000060         TEqd           = (eqdLI, eqdCI, eqdRI, eqdRO);
000061         TEact          = (eactIn, eactCk, eactNr, eactSp, eactIc);
```

Apple Lisa Computer Technical Information

```
000062      TEnib          = (enibSm, enibLg);
000063      TRgnib          = ARRAY [TEnib] OF Point;
000064      TLx              = PACKED RECORD
000065                  fam: 0..31;
000066                  fB: TF;
000067                  fI: TF;
000068                  eact: TEact;
000069                  eqd: TEqd;
000070                  ebtn: TEbtn;
000071                  enib: TEnib;
000072                  fBtnVis: TF;
000073                  mask: CHAR;
000074                  smax: 0..255;
000075      END;
000076      TBtn              = RECORD
000077                  r: Rect;
000078                  lx: TLx;
000079                  hsp: THsp;
000080      END;
000081      TRgbtn           = ARRAY [0..cBtnMax] OF TBtn;
000082      TIdMu            = 0..idMuMax;
000083      TSetIdMu         = SET OF TIdMu;
000084      THmu             = ^TPmu;
000085      TPmu             = ^TMu;
000086      TMu              = RECORD
000087                  rBB: Rect;
000088                  idMu: TIdMu;
000089                  iBtnOn: TB;
000090                  iBtnUsr: TB;
000091                  cBtn: TB;
000092                  fMuVis: TF;
000093                  fMUActv: TF;
000094                  rgbtn: TRgbtn;
000095      END;
000096      TRghmu           = ARRAY [0..cMuMax] OF THmu;
000097      THckfm           = ^TPckfm;
000098      TPckfm           = ^Tckfm;
000099      Tckfm            = RECORD
000100                  rBB: Rect;
000101                  iMuKey: TB;
000102                  iBtnKey: TB;
000103                  cMu: TB;
000104                  rghmu: TRghmu;
000105      END;
000106      Tckfmseg         = RECORD
000107                  ckfm: ARRAY [0..ckfmMax] OF THckfm;
000108      END;
000109
000110  VAR
000111      dbox:            TDBox;
000112      setIdMuAll:     TSetIdMu;
000113      hspNil:          THsp;
000114      finfoLx:         FontInfo;
000115      rgrib:           TRgnib;
000116      pPrRecUsr:       TPrRec;
000117      hckfmPrCk:       THckfm;
000118      procCkCmd:       TProc;
000119
000120  PROCEDURE DlgMgrInit;
000121
000122  PROCEDURE PauseEvt(dt: TC);
000123
000124  FUNCTION FMseInFldr(pwnd: WindowPtr; VAR pt: Point): TF;
000125
000126  PROCEDURE PrStrToNum(spNum: TSp; VAR c: TC);
000127
```

Apple Lisa Computer Technical Information

```
000128 PROCEDURE PrNumToStr(c: TC; VAR spNum: TSp);
000129
000130 PROCEDURE PrCkEvent(event: EventRecord);
000131
000132 PROCEDURE PrCkTrkMse;
000133
000134 PROCEDURE CaretMove(pwnd: WindowPtr; h, v: TC);
000135
000136 PROCEDURE CaretSynch;
000137
000138 PROCEDURE CaretKill;
000139
000140 PROCEDURE CaretFlip;
000141
000142 PROCEDURE CaretBlink;
000143
000144 PROCEDURE DlgInstall(fKey: TF; pEvent, pIdle, pEnd: TProc);
000145
000146 PROCEDURE DlgEvent(event: EventRecord);
000147
000148 PROCEDURE DlgIdle;
000149
000150 PROCEDURE DlgEnd(fDismiss: TF);
000151
000152 PROCEDURE DlgOpen(dv: TC);
000153
000154 FUNCTION FDlgEvent(VAR event: EventRecord): TF;
000155
000156 PROCEDURE DlgMain;
000157
000158 FUNCTION FDlgDone: TF;
000159
000160 FUNCTION FckfmSelect(hckfm: THckfm; ptMse: Point; VAR iMu, iBtn: TB): TF;
000161
000162 PROCEDURE CkfmSetUp(hckfm: THckfm; fInit: TF; setIdMuActv: TSetIdMu);
000163
000164 PROCEDURE CkfmShow(hckfm: THckfm; fInit: TF);
000165
000166 PROCEDURE CkfmTrkMse(hckfm: THckfm; pWnd: WindowPtr);
000167
000168 PROCEDURE CkfmKeys(hckfm: THckfm; iMu, iBtn: TB);
000169
000170 PROCEDURE CkfmAddCh(hckfm: THckfm; ch: CHAR);
000171
000172 FUNCTION FckfmNxtKey(hckfm: THckfm; VAR iMuNxt, iBtnNxt: TB): TF;
000173
000174 PROCEDURE CkfmSp(hckfm: THckfm; iMu, iBtn: TB; VAR sp: TSp; fNew: TF);
000175
000176 FUNCTION FckfmHit(hckfm: THckfm; iMu, iBtn: TB): TF;
000177 {$IFC fBtnRead}
000178
000179 FUNCTION HspAlloc(sp: TSp; cChMax: TC): THsp;
000180
000181 PROCEDURE HspFree(VAR hsp: THsp);
000182
000183 FUNCTION HckfmRead(VAR ckFile: Text): THckfm;
000184 {$ENDC}
000185
000186 PROCEDURE MuShow(hmu: THmu);
000187
000188 PROCEDURE MuErase(hmu: THmu);
000189
000190 PROCEDURE MuPush(hmu: THmu; iBtn: TB);
000191
000192 FUNCTION FMuHit(hmu: THmu; ptMse: Point; VAR iBtn: TB): TF;
000193
```

Apple Lisa Computer Technical Information

```
000194     PROCEDURE BtnDraw(btn: TBtn);
000195
000196     PROCEDURE BtnErase(btn: TBtn);
000197
000198     PROCEDURE BtnFill(btn: TBtn; fFlip: TF);
000199
000200     PROCEDURE BtnPtSp(btn: TBtn; VAR ptSp: Point; VAR dhSp: TC);
000201
000202     PROCEDURE BtnWrite(btn: TBtn; sp: TSp);
000203
000204     PROCEDURE BtnAddCh(btn: TBtn; ch: CHAR);
000205
000206     PROCEDURE BtnBkSp(btn: TBtn);
000207
000208     PROCEDURE BtnBind(VAR lx: TLx);
000209
000210     PROCEDURE BtnSetSp(btn: TBtn; sp: TSp; showIt: TF);
000211
000212     PROCEDURE BtnReappear(hmu: THmu; iBtn: TB);
000213
000214     PROCEDURE BtnVanish(hmu: THmu; iBtn: TB);
000215
000216 *****
000217 *
000218 *           THAT'S ALL FOLKS ...
000219 *
000220 *****
000221
```

End of File -- Lines: 221 Characters: 5648

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PREVENTS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PREVENTS
000004 *
000005 *****
000006
000007 USES { $U+ } PREVENTS;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U LibSM/UnitStd   } UnitStd,
000016         { $U LibOS/SysCall  } SysCall,
000017         { $U LibPM/PMDecl   } PMDecl,
000018         { $U LibSM/UnitHz   } UnitHz,
000019         { $U LibQD/Storage   } Storage,
000020         { $U LibQD/QuickDraw } QuickDraw,
000021         { $U LibQD/GrafUtil  } GrafUtil,
000022         { $U LibFM/FontMgr   } FontMgr,
000023         { $U LibPr/PrStdInfo } PrStdInfo,
000024         { $U LibPr/PrStdProc } PrStdProc;
000025
000026     CONST
000027         ECNameSize   = 8;
000028         maxstuff     = 38;
000029
000030     TYPE
000031         PrMessages   = (ENull, EWake, EDone, EPrintRequest, EPrintDialog,
000032             ESettingsDialog, EPrListDialog, EMonitorDialog,
000033             ESendData, ETalkToMe, EMetrics, EJobCount,
000034             EPrintTerminate, EReply, EAbort, ESTuffed, EConfigure);
000035         PrEventBlk   = RECORD
000036             event_header: t_eheader;
000037             CASE INTEGER OF
000038                 0:
000039                     (event_text: t_event_text);
000040                 1:
000041                     (eventBytes: ARRAY [0..39] OF - 128..127);
000042                 2:
000043                     (eventInts: ARRAY [0..19] OF INTEGER);
000044                 3:
000045                     (eventLongs: ARRAY [0..9] OF LONGINT);
000046             END;
000047         Channel       = RECORD
000048             ChName: Pathname;
000049             RefNum: INTEGER;
000050             Exname: t_ex_name;
000051             EvBlock: PrEventBlk;
000052             Interval: timestmp_interval;
000053             ClkTime: time_rec;
000054             Waits: t_waitlist;
000055             END;
000056
000057     FUNCTION PrAccept(VAR Chan: Channel): PrMessage;
000058
000059     PROCEDURE PrCall(VAR cError: TC; PrID: TL; Message: PrMessages; mess1, mess2,
000060         mess3, mess4: TL);
000061
```

Apple Lisa Computer Technical Information

```
000062 PROCEDURE PrCloseChan(VAR Chan: Channel);
000063
000064 PROCEDURE PrDeferToBgd(PrID: TL; Message: PrMessages; mess1, mess2, mess3,
000065 mess4: TL);
000066
000067 PROCEDURE PrIDtoPath(PrID: TL; VAR path: PathName);
000068
000069 PROCEDURE Proffer(VAR Chan: Channel; Message: PrMessages);
000070
000071 PROCEDURE PrOpenChan(VAR Chan: Channel; Rx: TF);
000072
000073 PROCEDURE PrOpenCall(VAR Chan: Channel; ExProc: LongAdr);
000074
000075 PROCEDURE PrJobCount(PrID: TL; VAR cjob: INTEGER);
000076
000077 PROCEDURE PrNotify(PrID: TL; Message: PrMessages; mess1, mess2, mess3,
000078 mess4: TL);
000079
000080 PROCEDURE PrPrintRequest(PrID: TL; lFileName, ncopies, PrntrID, PortID: TL);
000081
000082 PROCEDURE PrPrintStop(PrID: TL; lFileName, ncopies, PrntrID, PortID: TL);
000083
000084 PROCEDURE PrSendMessage(VAR cError: INTEGER; VAR Chan: Channel;
000085 Message: PrMessages);
000086
000087 FUNCTION PrWaitMessage(VAR Chan: Channel): PrMessage;
000088
000089 PROCEDURE PrStuffData(PrID: TL; dataPtr: TP; nbytes: TC);
000090
000091 PROCEDURE PrUpdate;
000092
000093 *****
000094 *
000095 * THAT'S ALL FOLKS ...
000096 *
000097 *****
000098
```

End of File -- Lines: 98 Characters: 3195

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PRFILEPR.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRFILEPR
000004 *
000005 *****
000006
000007 USES { $U+ } PRFILEPR;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U LibOS/SysCall   } SysCall,
000016         { $U LibPM/PMDecl   } PMDecl,
000017         { $U LibSM/UnitStd   } UnitStd,
000018         { $U LibSM/UnitHz    } UnitHz,
000019         { $U LibQD/Storage   } Storage,
000020         { $U LibQD/QuickDraw } QuickDraw,
000021         { $U LibFM/FontMgr   } FontMgr,
000022         { $U LibPr/PrStdInfo } PrStdInfo,
000023         { $U LibPr/PrStdProc } PrStdProc;
000024
000025     CONST
000026         pfPassword   = - 13264;
000027         pfPicHRule    = 20;
000028         pfPicVRule    = 21;
000029         pfPicTab      = 22;
000030         HeapSize      = 12000;
000031
000032     TYPE
000033         TStr80        = String[80];
000034
000035     PROCEDURE OpenBlock(VAR error: Integer; VAR FileRec: TPrFileRec;
000036                         access: MSet);
000037
000038     PROCEDURE WriteBlock(VAR error: Integer; p: TP; lCount: LongInt;
000039                          RefNum: Integer);
000040
000041     PROCEDURE ReadBlock(VAR error: Integer; p: TP; lCount: LongInt;
000042                         RefNum: Integer);
000043
000044     PROCEDURE PfSetPos(VAR error: Integer; lPos: TL; RefNum: Integer);
000045
000046     PROCEDURE PfSavePos(VAR error: Integer; VAR lPos: TL; RefNum: Integer);
000047
000048 *****
000049 *
000050 *             THAT'S ALL FOLKS ...
000051 *
000052 *****
000053
```

End of File -- Lines: 53 Characters: 1424

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PROGCOMM.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PROGCOMM
000004 *
000005 *****
000006
000007 USES { $U+ } PROGCOMM;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015     { $U StdUnit   } StdUnit,
000016     { $U ShellComm } ShellComm;
000017
000018     CONST
000019     PCNone      = - 1;
000020     PCAny       = 0;
000021     PCText      = 1;
000022     PCBufMax   = 1023;
000023     PC_SetReallyStop = 1;
000024     PC_GetReallyStop = 2;
000025     PC_SetUnSavedEdits = 6960;
000026     PC_GetUnSavedEdits = 8751;
000027
000028     TYPE
000029     PCBufrP     = ^PCBufr;
000030     PCBufr      = PACKED ARRAY [0..PCBufMax] OF CHAR;
000031
000032     VAR
000033     PCBufrPtr:   PCBufrP;
000034
000035     PROCEDURE PCInit;
000036
000037     PROCEDURE PCSetRunCmd(RC: SUStr);
000038
000039     PROCEDURE PCSetRetStr(RS: SUStr);
000040
000041     PROCEDURE PCReWrite(WriteType: INTEGER; Key: SUStr);
000042
000043     FUNCTION PCReset(ReadType: INTEGER; Key: SUStr): BOOLEAN;
000044
000045     FUNCTION PCClose(KillBuf: BOOLEAN; Key: SUStr): BOOLEAN;
000046
000047     FUNCTION PCPutCh(Ch: CHAR): BOOLEAN;
000048
000049     FUNCTION PCGetCh(VAR Ch: CHAR): BOOLEAN;
000050
000051     FUNCTION PCPutLine(L: SUStr): BOOLEAN;
000052
000053     FUNCTION PCGetLine(VAR L: SUStr): BOOLEAN;
000054
000055     FUNCTION PCShellCmd(Cmd: INTEGER; P: SUStrP): BOOLEAN;
000056
000057 *****
000058 *
000059 *                 THAT'S ALL FOLKS ...
000060 *
000061 *****
```


Apple Lisa Computer Technical Information

000062

End of File -- Lines: 62 Characters: 1355

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PRPUBLIC.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRPUBLIC
000004 *
000005 *****
000006
000007 USES {$U+} PRPUBLIC;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U LibOS/SysCall   } SysCall,
000016         {$U LibPM/PMDecl   } PMDecl,
000017         {$U LibSM/UnitStd   } UnitStd,
000018         {$U LibSM/UnitHz    } UnitHz,
000019         {$U LibQD/Storage   } Storage,
000020         {$U LibQD/QuickDraw } QuickDraw,
000021         {$U LibFM/FontMgr   } FontMgr,
000022         {$U LibPr/PrStdInfo } PrStdInfo;
000023
000024     PROCEDURE PsYieldCpu;
000025
000026     PROCEDURE PrGlobalInit;
000027
000028     FUNCTION FPrPrfValid(VAR prprf: TPrRec): TF;
000029
000030     PROCEDURE PrPrfDefault(VAR prprf: TPrRec);
000031
000032     FUNCTION fPrPrfEq(VAR prprf1, prprf2: TPrRec): TF;
000033
000034     FUNCTION PrDocStart(VAR PrRec: TPrRec; VAR PrPort: TPrPort; LDSN: INTEGER;
000035         DoSpoolAlert: Boolean): Boolean;
000036
000037     FUNCTION PrPageStart(VAR prinsIn: TPrRec; VAR PrPort: TPrPort): TF;
000038
000039     FUNCTION PrPageEnd(VAR prinsIn: TPrRec; VAR PrPort: TPrPort): TF;
000040
000041     PROCEDURE PrDocEnd(VAR prinsIn: TPrRec; VAR PrPort: TPrPort);
000042
000043     PROCEDURE PrMgrInit;
000044
000045     PROCEDURE PrPrfDlg(VAR prprf: TPrRec; VAR fNewStl: TF; fPgSzSuppress: TF);
000046
000047     FUNCTION FPrInsDlg(VAR prprf: TPrRec; VAR fNewStl: TF;
000048         prmode: PrMenuSuppress): TF;
000049
000050     PROCEDURE PrSpoolAbort;
000051
000052     PROCEDURE PrBgdDlg;
000053
000054     PROCEDURE PrDlgDefault(fPgSzSuppress: TF; Prmode: PrMenuSuppress; MaxHeight,
000055         MaxWidth, MinHeight, MinWidth: TC);
000056
000057     PROCEDURE PrLFntID(VAR prprf: TPrRec; famIn: TC; setIn: Style;
000058         VAR lFntid: TLFntID);
000059
000060 *****
000061 *
```

Apple Lisa Computer Technical Information

```
000062 *                THAT'S ALL FOLKS ...
000063 *
000064 *****
000065
```

End of File -- Lines: 65 Characters: 1780

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PRSTDINF.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRSTDINF
000004 *
000005 *****
000006
000007 USES { $U+ } PRSTDINF;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U LibOS/SysCall    } SysCall,
000016         { $U LibPM/PMDecl    } PMDecl,
000017         { $U LibSM/UnitStd    } UnitStd,
000018         { $U LibSM/UnitHz     } UnitHz,
000019         { $U LibQD/Storage    } Storage,
000020         { $U LibQD/QuickDraw  } QuickDraw,
000021         { $U LibFM/FontMgr    } FontMgr;
000022
000023         { $setc doTracePR := TRUE }
000024         { $setc fTracePR  := doTracePR and fTrace }
000025
000026         { $SETC PrDebug    = fdbgok }
000027         { $SETC PrSymbols  = fsymok }
000028
000029     CONST
000030         cPrPrfVersion = 11;
000031         prdVersion    = 1;
000032         cbPrPrfMax    = 64;
000033         PrPageMax     = $07FFF;
000034         LastBlock     = 9999;
000035         iPgOther      = 0;
000036         iPg8x11       = 1;
000037         iPg8x14       = 2;
000038         iPg14x11      = 3;
000039         iPgA4         = 4;
000040         iPg210x12     = 5;
000041         iPg310x8      = 6;
000042         iPgSpec       = 7;
000043         iPgFst        = 0;
000044         iPgMax        = 15;
000045         prPgFract     = 120;
000046         prNILProcess  = - 1;
000047         prNilRefNum   = - 1;
000048         prNilPrinter  = - 1;
000049         prNilPort     = - 1;
000050         prErrAbort    = 3050;
000051         prErrPrShop   = 3068;
000052         NoPRD         = 3055;
000053         NoPMDrivers   = 3056;
000054         PMbutNoPRD    = 3057;
000055         NoPrProcess   = 3058;
000056         NoBtnFile     = 3059;
000057         PrParentProcess = 'System.Print';
000058         spPrAlrt      = 'System.Pr.';
000059         sParentSeg     = 'System.PrData';
000060         ldsnshared    = 12;
000061         ldsndialog    = 9;
```

Apple Lisa Computer Technical Information

```
000062
000063     TYPE
000064     T1         = 0..1;
000065     T4         = 0..15;
000066     T8         = - 128..127;
000067     T16        = PACKED RECORD
000068                 CASE INTEGER OF
000069                 0:
000070                     (by1, by0: - 128..127);
000071                 1:
000072                     (c1, c0: CHAR);
000073                 2:
000074                     (usb1, usb0: 0..255);
000075                 3:
000076                     (sb1, sb0: - 128..127);
000077                 4:
000078                     (n3, n2, n1, n0: T4);
000079                 5:
000080                     (f15, f14, f13, f12, f11, f10, f9, f8, f7, f6,
000081                      f5, f4, f3, f2, f1, f0: BOOLEAN);
000082                 6:
000083                     (b15, b14, b13, b12, b11, b10, b9, b8, b7, b6,
000084                      b5, b4, b3, b2, b1, b0: T1);
000085                 7:
000086                     (i: INTEGER);
000087                 END;
000088     T32        = PACKED RECORD
000089                 CASE INTEGER OF
000090                 0:
000091                     (i1, i0: T16);
000092                 1:
000093                     (l: LONGINT);
000094                 2:
000095                     (p: TP);
000096                 3:
000097                     (h: TH);
000098                 6:
000099                     (aby: PACKED ARRAY [0..3] OF - 128..127);
000100                 END;
000101     TDirection = (Portrait, LandScape);
000102     TPgSize    = RECORD
000103                 PaperDirect: TDirection;
000104                 PrintDirect: TDirection;
000105                 Width: INTEGER;
000106                 Height: INTEGER;
000107                 Extend: LongInt;
000108                 END;
000109     TDlgPgSz   = RECORD
000110                 pgSzBtn: INTEGER;
000111                 widInch: INTEGER;
000112                 htInch: INTEGER;
000113                 wid16th: INTEGER;
000114                 ht16th: INTEGER;
000115                 widMM: INTEGER;
000116                 htMM: INTEGER;
000117                 END;
000118     TPrJobInfo = RECORD
000119                 DocPgFst: Integer;
000120                 DocPgLst: Integer;
000121                 SpooledPages: Integer;
000122                 Copies: Integer;
000123                 Immediate: Boolean;
000124                 Extend: LongInt;
000125                 END;
000126     TPrInfo    = RECORD
000127                 ADev: TADev;
```

Apple Lisa Computer Technical Information

```
000128           PrinterID: Integer;
000129           NumColor: Integer;
000130           HRes, VRes: Integer;
000131           rPaper: Rect;
000132           rPrintable: Rect;
000133           PgSize: TPgSize;
000134           Port: Integer;
000135           DlgPgSz: TDlgPgSz;
000136           Extend: LongInt;
000137           END;
000138 TPrDevInfo    = ARRAY [1..128] OF - 128..127;
000139 TPrPort       = RECORD
000140             gport: GrafPort;
000141             gProcs: QDProcs;
000142             Extend: LongInt;
000143           END;
000144 TpPrRec       = ^TPrRec;
000145 ThPrRec       = ^TPPrRec;
000146 TPrRec        = RECORD
000147             PrVersion: Integer;
000148             PrLDSN: Integer;
000149             PrJobInfo: TPrJobInfo;
000150             PrInfo: TPrInfo;
000151             PrDevInfo: TPrDevInfo;
000152             Extend: LongInt;
000153           END;
000154 TPfPage       = RECORD
000155             Start: LongInt;
000156             Length: LongInt;
000157           END;
000158 TPfPages      = ARRAY [1..999] OF TPfPage;
000159 TPPfPages     = ^TPfPages;
000160 THPfPages     = ^TPPfPages;
000161 TPrFileId     = RECORD
000162             sFileName: Str255;
000163             CASE Boolean OF
000164               True:
000165                 (LongId: LongInt);
000166               False:
000167                 (Id, BlockNum: Integer);
000168             END;
000169 TPrFileRec    = RECORD
000170             FileName: PathName;
000171             RefNum: Integer;
000172           END;
000173 TPrDSRec      = RECORD
000174             SegName: PathName;
000175             InitSize: LongInt;
000176             RefNum: Integer;
000177             LDSN: Integer;
000178             BegAddr: LongInt;
000179           END;
000180 PrMenuSuppress = (ePrNormal, ePgRangeSuppress, ePrDialogSuppress);
000181 DlgOption      = (eDlgOK, eDlgCancel, eDlgAnother);
000182 TPrnrtrID     = 0..16383;
000183 TPrExtWord    = PACKED RECORD
000184             CASE INTEGER OF
000185               0:
000186                 (printer_flag: TF;
000187                  default_flag: TF;
000188                  PrDeviceID: TPrnrtrID);
000189               1:
000190                 (ExtWord: INTEGER);
000191             END;
000192 DriverChoice  = RECORD
000193             Connector: TypeConnect;
```

Apple Lisa Computer Technical Information

```
000194             cd_Driver: e_name;
000195             END;
000196     DN_Info      = RECORD
000197             printer_id: LongInt;
000198             numDrivers: Integer;
000199             END;
000200     prdHeader    = RECORD
000201             version: Integer;
000202             prdcount: Integer;
000203             END;
000204     prdEntry     = RECORD
000205             entryLen: Integer;
000206             drv_r_name: e_name;
000207             info: dn_info;
000208             END;
000209     PRD_Info     = RECORD
000210             descr: prdHeader;
000211             info: ARRAY [0..0] OF prdEntry;
000212             END;
000213
000214     VAR
000215     cPrError:    Integer;
000216     fBackGround: TF;
000217     fInitialized: TF;
000218     lParentID:   TL;
000219     PrAlertUP:   Boolean;
000220     prsslot, prsconnector, prsdevice, prsAserial, prsBserial, prsparallel,
000221     prsnone:     TSp;
000222     PrExtras:    ARRAY [0..10] OF LONGINT;
000223     PrDebug:     Boolean;
000224     PrBugpName:  PathName;
000225     PrBugFile:   Text;
000226
000227     FUNCTION PrRec68K: TpPrRec;
000228
000229     *****
000230     *
000231     *             THAT'S ALL FOLKS ...
000232     *
000233     *****
000234
```

End of File -- Lines: 234 Characters: 7814

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 PRSTDPRO.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRSTDPRO
000004 *
000005 *****
000006
000007 USES { $U+ } PRSTDPRO;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U LibOS/SysCall   } SysCall,
000016         { $U LibPM/PMDecl   } PMDecl,
000017         { $U LibHW/HWint    } HWint,
000018         { $U LibSM/UnitStd   } UnitStd,
000019         { $U LibSM/UnitHz    } UnitHz,
000020         { $U LibQD/Storage   } Storage,
000021         { $U LibQD/QuickDraw } QuickDraw,
000022         { $U LibFM/FontMgr   } FontMgr,
000023         { $U LibPr/PrStdInfo } PrStdInfo;
000024
000025     CONST
000026         PortA      = $0A10;
000027         PortB      = $0A20;
000028
000029     TYPE
000030         Str80      = String[80];
000031
000032     PROCEDURE PsCopy(Src, Dst: TL; len: TC);
000033
000034     PROCEDURE PsOpenPort(VAR Error: Integer; Port: Integer);
000035
000036     PROCEDURE PsClosePort(VAR Error: Integer; Port: Integer);
000037
000038     PROCEDURE PsInPort(VAR Error: Integer; p: TP; c: TC);
000039
000040     PROCEDURE PsOutPort(VAR Error: Integer; p: TP; c: TC);
000041
000042     PROCEDURE PsPortName(Port: Integer; shandle: THsp);
000043
000044     PROCEDURE PsPortToDevName(Port: Integer; VAR DevName: e_name);
000045
000046     PROCEDURE PsPreemptive;
000047
000048     PROCEDURE PsNonPreemptive;
000049
000050     PROCEDURE PsNumToStr(c, w: TC; pstr: TpSp; z: CHAR);
000051
000052     PROCEDURE PsYieldCpu;
000053
000054     FUNCTION fClrBitmap(VAR bits: Bitmap; rCheck: Rect): TF;
000055
000056     PROCEDURE InitDeBug;
000057
000058     FUNCTION PrAnd(c1, c2: TC): TC;
000059
000060     PROCEDURE ClrBitmap(pBitmap: TP);
000061
```


Apple Lisa Computer Technical Information

```
000062 FUNCTION fClrBits(VAR bits: Bitmap; rCheck: Rect): TF;
000063
000064 FUNCTION fEQBlk(a, b: TP; l: TL): TF;
000065
000066 PROCEDURE MoveBlk(a, b: TP; l: TL);
000067
000068 FUNCTION PrShRight(c1, n2: TC): TC;
000069
000070 FUNCTION PrShLeft(c1, n2: TC): TC;
000071
000072 PROCEDURE GetFileName(VAR PrFileId: TPrFileId);
000073
000074 *****
000075 *
000076 *                THAT'S ALL FOLKS ...
000077 *
000078 *****
000079
```

End of File -- Lines: 79 Characters: 1812

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 QUEUES.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : QUEUES
000004 *
000005 *****
000006
000007 USES {$U+} QUEUES;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U LibSM/UnitStd } UnitStd,
000016         {$U LibSM/UnitHz  } UnitHz;
000017
000018     TYPE
000019         pQueue      = ^Queue;
000020         hQueue      = ^pQueue;
000021         Queue       = RECORD
000022             Head: hQueue;
000023             Tail: hQueue;
000024             END;
000025
000026     PROCEDURE QueueFree(Queue: hQueue);
000027
000028     PROCEDURE QueueInit(Queue: hQueue);
000029
000030     PROCEDURE QueueInUse(Queue: hQueue);
000031
000032     PROCEDURE Enqueue(Queue, Item: hQueue);
000033
000034     FUNCTION Dequeue(Queue: hQueue): hQueue;
000035
000036     FUNCTION InsertBefore(Queue, Successor, Item: hQueue): TF;
000037
000038     FUNCTION InsertAfter(Queue, Predecessor, Item: hQueue): TF;
000039
000040     FUNCTION Unqueue(Queue, Item: hQueue): TF;
000041
000042     FUNCTION QueueLength(Queue: hQueue): TC;
000043
000044     FUNCTION QueuePeek(Queue: hQueue): hQueue;
000045
000046     FUNCTION EnumerateQueue(Queue: hQueue; FUNCTION
000047         FuncHandle(Queue: hQueue; arg: hQueue): BOOLEAN;
000048         arg: hQueue): hQueue;
000049
000050 *****
000051 *
000052 *             THAT'S ALL FOLKS ...
000053 *
000054 *****
000055
```

End of File -- Lines: 55 Characters: 1339

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 QUICKDRAW.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : QUICKDRAW
000004 *
000005 *****
000006
000007 USES {$U+} QUICKDRAW;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     CONST
000015         srcCopy      = 0;
000016         srcOr        = 1;
000017         srcXor       = 2;
000018         srcBic       = 3;
000019         notSrcCopy   = 4;
000020         notSrcOr     = 5;
000021         notSrcXor    = 6;
000022         notSrcBic    = 7;
000023         patCopy      = 8;
000024         patOr        = 9;
000025         patXor       = 10;
000026         patBic       = 11;
000027         notPatCopy   = 12;
000028         notPatOr     = 13;
000029         notPatXor    = 14;
000030         notPatBic    = 15;
000031         normalBit    = 0;
000032         inverseBit   = 1;
000033         redBit       = 4;
000034         greenBit     = 3;
000035         blueBit      = 2;
000036         cyanBit      = 8;
000037         magentaBit   = 7;
000038         yellowBit    = 6;
000039         blackBit     = 5;
000040         blackColor   = 33;
000041         whiteColor   = 30;
000042         redColor     = 205;
000043         greenColor   = 341;
000044         blueColor    = 409;
000045         cyanColor    = 273;
000046         magentaColor = 137;
000047         yellowColor  = 69;
000048         picLParen    = 0;
000049         picRParen    = 1;
000050
000051     TYPE
000052         QDByte       = - 128..127;
000053         QDPtr        = ^QDByte;
000054         QDHandle     = ^QDPtr;
000055         Str255       = String[255];
000056         Pattern      = PACKED ARRAY [0..7] OF 0..255;
000057         Bits16       = ARRAY [0..15] OF INTEGER;
000058         VHSelect     = (v, h);
000059         GrafVerb     = (frame, paint, erase, invert, fill);
000060         StyleItem    = (bold, italic, underline, outline, shadow, condense,
000061             extend, onlymetricswidths);
```

Apple Lisa Computer Technical Information

```
000062      Style          = SET OF StyleItem;
000063      FontInfo         = RECORD
000064                      ascent: INTEGER;
000065                      descent: INTEGER;
000066                      widMax: INTEGER;
000067                      leading: INTEGER;
000068                      END;
000069      Point             = RECORD
000070                      CASE INTEGER OF
000071                        0:
000072                          (v: INTEGER;
000073                           h: INTEGER);
000074                        1:
000075                          (vh: ARRAY [VHSelect] OF INTEGER);
000076                      END;
000077      Rect              = RECORD
000078                      CASE INTEGER OF
000079                        0:
000080                          (top: INTEGER;
000081                           left: INTEGER;
000082                           bottom: INTEGER;
000083                           right: INTEGER);
000084                        1:
000085                          (topLeft: Point;
000086                           botRight: Point);
000087                      END;
000088      BitMap            = RECORD
000089                      baseAddr: QDPtr;
000090                      rowBytes: INTEGER;
000091                      bounds: Rect;
000092                      END;
000093      Cursor            = RECORD
000094                      data: Bits16;
000095                      mask: Bits16;
000096                      hotSpot: Point;
000097                      END;
000098      PenState          = RECORD
000099                      pnLoc: Point;
000100                      pnSize: Point;
000101                      pnMode: INTEGER;
000102                      pnPat: Pattern;
000103                      END;
000104      PolyHandle         = ^PolyPtr;
000105      PolyPtr           = ^Polygon;
000106      Polygon           = RECORD
000107                      polySize: INTEGER;
000108                      polyBBox: Rect;
000109                      polyPoints: ARRAY [0..0] OF Point;
000110                      END;
000111      RgnHandle         = ^RgnPtr;
000112      RgnPtr            = ^Region;
000113      Region            = RECORD
000114                      rgnSize: INTEGER;
000115                      rgnBBox: Rect;
000116                      END;
000117      PicHandle         = ^PicPtr;
000118      PicPtr            = ^Picture;
000119      Picture           = RECORD
000120                      picSize: INTEGER;
000121                      picFrame: Rect;
000122                      END;
000123      QDProcsPtr        = ^QDProcs;
000124      QDProcs           = RECORD
000125                      textProc: QDPtr;
000126                      lineProc: QDPtr;
000127                      rectProc: QDPtr;
```

Apple Lisa Computer Technical Information

```
000128         rRectProc: QDPtr;
000129         ovalProc: QDPtr;
000130         arcProc: QDPtr;
000131         polyProc: QDPtr;
000132         rgnProc: QDPtr;
000133         bitsProc: QDPtr;
000134         commentProc: QDPtr;
000135         txMeasProc: QDPtr;
000136         getPicProc: QDPtr;
000137         putPicProc: QDPtr;
000138         END;
000139 GrafPtr    = ^GrafPort;
000140 GrafPort   = RECORD
000141         device: INTEGER;
000142         portBits: BitMap;
000143         portRect: Rect;
000144         visRgn: RgnHandle;
000145         clipRgn: RgnHandle;
000146         bkPat: Pattern;
000147         fillPat: Pattern;
000148         pnLoc: Point;
000149         pnSize: Point;
000150         pnMode: INTEGER;
000151         pnPat: Pattern;
000152         pnVis: INTEGER;
000153         txFont: INTEGER;
000154         txFace: Style;
000155         txMode: INTEGER;
000156         txSize: INTEGER;
000157         spExtra: LongInt;
000158         fgColor: LongInt;
000159         bkColor: LongInt;
000160         colrBit: INTEGER;
000161         patStretch: INTEGER;
000162         picSave: QDHandle;
000163         rgnSave: QDHandle;
000164         polySave: QDHandle;
000165         grafProcs: QDProcsPtr;
000166         END;
000167
000168 VAR
000169     thePort: GrafPtr;
000170     white: Pattern;
000171     black: Pattern;
000172     gray: Pattern;
000173     ltGray: Pattern;
000174     dkGray: Pattern;
000175     arrow: Cursor;
000176     screenBits: BitMap;
000177     randSeed: LongInt;
000178
000179     PROCEDURE InitGraf(globalPtr: QDPtr);
000180
000181     PROCEDURE OpenPort(port: GrafPtr);
000182
000183     PROCEDURE InitPort(port: GrafPtr);
000184
000185     PROCEDURE ClosePort(port: GrafPtr);
000186
000187     PROCEDURE SetPort(port: GrafPtr);
000188
000189     PROCEDURE GetPort(VAR port: GrafPtr);
000190
000191     PROCEDURE GrafDevice(device: INTEGER);
000192
000193     PROCEDURE SetPortBits(bm: BitMap);
```

Apple Lisa Computer Technical Information

```
000194
000195     PROCEDURE PortSize(width, height: INTEGER);
000196
000197     PROCEDURE MovePortTo(leftGlobal, topGlobal: INTEGER);
000198
000199     PROCEDURE SetOrigin(h, v: INTEGER);
000200
000201     PROCEDURE SetClip(rgn: RgnHandle);
000202
000203     PROCEDURE GetClip(rgn: RgnHandle);
000204
000205     PROCEDURE ClipRect(r: Rect);
000206
000207     PROCEDURE BackPat(pat: Pattern);
000208
000209     PROCEDURE InitCursor;
000210
000211     PROCEDURE SetCursor(crsr: Cursor);
000212
000213     PROCEDURE HideCursor;
000214
000215     PROCEDURE ShowCursor;
000216
000217     PROCEDURE ObscureCursor;
000218
000219     PROCEDURE HidePen;
000220
000221     PROCEDURE ShowPen;
000222
000223     PROCEDURE GetPen(VAR pt: Point);
000224
000225     PROCEDURE GetPenState(VAR pnState: PenState);
000226
000227     PROCEDURE SetPenState(pnState: PenState);
000228
000229     PROCEDURE PenSize(width, height: INTEGER);
000230
000231     PROCEDURE PenMode(mode: INTEGER);
000232
000233     PROCEDURE PenPat(pat: Pattern);
000234
000235     PROCEDURE PenNormal;
000236
000237     PROCEDURE MoveTo(h, v: INTEGER);
000238
000239     PROCEDURE Move(dh, dv: INTEGER);
000240
000241     PROCEDURE LineTo(h, v: INTEGER);
000242
000243     PROCEDURE Line(dh, dv: INTEGER);
000244
000245     PROCEDURE TextFont(font: INTEGER);
000246
000247     PROCEDURE TextFace(face: Style);
000248
000249     PROCEDURE TextMode(mode: INTEGER);
000250
000251     PROCEDURE TextSize(size: INTEGER);
000252
000253     PROCEDURE SpaceExtra(extra: LongInt);
000254
000255     PROCEDURE DrawChar(ch: char);
000256
000257     PROCEDURE DrawString(s: Str255);
000258
000259     PROCEDURE DrawText(textBuf: QDPtr; firstByte, byteCount: INTEGER);
```

Apple Lisa Computer Technical Information

000260
000261 FUNCTION CharWidth(ch: CHAR): INTEGER;
000262
000263 FUNCTION StringWidth(s: Str255): INTEGER;
000264
000265 FUNCTION TextWidth(textBuf: QDPtr; firstByte, byteCount: INTEGER): INTEGER;
000266
000267 PROCEDURE GetFontInfo(VAR info: FontInfo);
000268
000269 PROCEDURE AddPt(src: Point; VAR dst: Point);
000270
000271 PROCEDURE SubPt(src: Point; VAR dst: Point);
000272
000273 PROCEDURE SetPt(VAR pt: Point; h, v: INTEGER);
000274
000275 FUNCTION EqualPt(pt1, pt2: Point): BOOLEAN;
000276
000277 PROCEDURE ScalePt(VAR pt: Point; fromRect, toRect: Rect);
000278
000279 PROCEDURE MapPt(VAR pt: Point; fromRect, toRect: Rect);
000280
000281 PROCEDURE LocalToGlobal(VAR pt: Point);
000282
000283 PROCEDURE GlobalToLocal(VAR pt: Point);
000284
000285 PROCEDURE SetRect(VAR r: Rect; left, top, right, bottom: INTEGER);
000286
000287 FUNCTION EqualRect(rect1, rect2: Rect): BOOLEAN;
000288
000289 FUNCTION EmptyRect(r: Rect): BOOLEAN;
000290
000291 PROCEDURE OffsetRect(VAR r: Rect; dh, dv: INTEGER);
000292
000293 PROCEDURE MapRect(VAR r: Rect; fromRect, toRect: Rect);
000294
000295 PROCEDURE InsetRect(VAR r: Rect; dh, dv: INTEGER);
000296
000297 FUNCTION SectRect(src1, src2: Rect; VAR dstRect: Rect): BOOLEAN;
000298
000299 PROCEDURE UnionRect(src1, src2: Rect; VAR dstRect: Rect);
000300
000301 FUNCTION PtInRect(pt: Point; r: Rect): BOOLEAN;
000302
000303 PROCEDURE Pt2Rect(pt1, pt2: Point; VAR dstRect: Rect);
000304
000305 PROCEDURE FrameRect(r: Rect);
000306
000307 PROCEDURE PaintRect(r: Rect);
000308
000309 PROCEDURE EraseRect(r: Rect);
000310
000311 PROCEDURE InvertRect(r: Rect);
000312
000313 PROCEDURE FillRect(r: Rect; pat: Pattern);
000314
000315 PROCEDURE FrameRoundRect(r: Rect; ovWd, ovHt: INTEGER);
000316
000317 PROCEDURE PaintRoundRect(r: Rect; ovWd, ovHt: INTEGER);
000318
000319 PROCEDURE EraseRoundRect(r: Rect; ovWd, ovHt: INTEGER);
000320
000321 PROCEDURE InvertRoundRect(r: Rect; ovWd, ovHt: INTEGER);
000322
000323 PROCEDURE FillRoundRect(r: Rect; ovWd, ovHt: INTEGER; pat: Pattern);
000324
000325 PROCEDURE FrameOval(r: Rect);

Apple Lisa Computer Technical Information

```
000326
000327   PROCEDURE PaintOval(r: Rect);
000328
000329   PROCEDURE EraseOval(r: Rect);
000330
000331   PROCEDURE InvertOval(r: Rect);
000332
000333   PROCEDURE FillOval(r: Rect; pat: Pattern);
000334
000335   PROCEDURE FrameArc(r: Rect; startAngle, arcAngle: INTEGER);
000336
000337   PROCEDURE PaintArc(r: Rect; startAngle, arcAngle: INTEGER);
000338
000339   PROCEDURE EraseArc(r: Rect; startAngle, arcAngle: INTEGER);
000340
000341   PROCEDURE InvertArc(r: Rect; startAngle, arcAngle: INTEGER);
000342
000343   PROCEDURE FillArc(r: Rect; startAngle, arcAngle: INTEGER; pat: Pattern);
000344
000345   PROCEDURE PtToAngle(r: Rect; pt: Point; VAR angle: INTEGER);
000346
000347   FUNCTION OpenPoly: PolyHandle;
000348
000349   PROCEDURE ClosePoly;
000350
000351   PROCEDURE KillPoly(poly: PolyHandle);
000352
000353   PROCEDURE OffsetPoly(poly: PolyHandle; dh, dv: INTEGER);
000354
000355   PROCEDURE MapPoly(poly: PolyHandle; fromRect, toRect: Rect);
000356
000357   PROCEDURE FramePoly(poly: PolyHandle);
000358
000359   PROCEDURE PaintPoly(poly: PolyHandle);
000360
000361   PROCEDURE ErasePoly(poly: PolyHandle);
000362
000363   PROCEDURE InvertPoly(poly: PolyHandle);
000364
000365   PROCEDURE FillPoly(poly: PolyHandle; pat: Pattern);
000366
000367   FUNCTION NewRgn: RgnHandle;
000368
000369   PROCEDURE DisposeRgn(rgn: RgnHandle);
000370
000371   PROCEDURE CopyRgn(srcRgn, dstRgn: RgnHandle);
000372
000373   PROCEDURE SetEmptyRgn(rgn: RgnHandle);
000374
000375   PROCEDURE SetRectRgn(rgn: RgnHandle; left, top, right, bottom: INTEGER);
000376
000377   PROCEDURE RectRgn(rgn: RgnHandle; r: Rect);
000378
000379   PROCEDURE OpenRgn;
000380
000381   PROCEDURE CloseRgn(dstRgn: RgnHandle);
000382
000383   PROCEDURE OffsetRgn(rgn: RgnHandle; dh, dv: INTEGER);
000384
000385   PROCEDURE MapRgn(rgn: RgnHandle; fromRect, toRect: Rect);
000386
000387   PROCEDURE InsetRgn(rgn: RgnHandle; dh, dv: INTEGER);
000388
000389   PROCEDURE SectRgn(srcRgnA, srcRgnB, dstRgn: RgnHandle);
000390
000391   PROCEDURE UnionRgn(srcRgnA, srcRgnB, dstRgn: RgnHandle);
```


Apple Lisa Computer Technical Information

```
000392
000393     PROCEDURE DiffRgn(srcRgnA, srcRgnB, dstRgn: RgnHandle);
000394
000395     PROCEDURE XorRgn(srcRgnA, srcRgnB, dstRgn: RgnHandle);
000396
000397     FUNCTION EqualRgn(rgnA, rgnB: RgnHandle): BOOLEAN;
000398
000399     FUNCTION EmptyRgn(rgn: RgnHandle): BOOLEAN;
000400
000401     FUNCTION PtInRgn(pt: Point; rgn: RgnHandle): BOOLEAN;
000402
000403     FUNCTION RectInRgn(r: Rect; rgn: RgnHandle): BOOLEAN;
000404
000405     PROCEDURE FrameRgn(rgn: RgnHandle);
000406
000407     PROCEDURE PaintRgn(rgn: RgnHandle);
000408
000409     PROCEDURE EraseRgn(rgn: RgnHandle);
000410
000411     PROCEDURE InvertRgn(rgn: RgnHandle);
000412
000413     PROCEDURE FillRgn(rgn: RgnHandle; pat: Pattern);
000414
000415     PROCEDURE ScrollRect(dstRect: Rect; dh, dv: INTEGER; updateRgn: rgnHandle);
000416
000417     PROCEDURE CopyBits(srcBits, dstBits: BitMap; srcRect, dstRect: Rect;
000418                       mode: INTEGER; maskRgn: RgnHandle);
000419
000420     FUNCTION OpenPicture(picFrame: Rect): PicHandle;
000421
000422     PROCEDURE ClosePicture;
000423
000424     PROCEDURE DrawPicture(myPicture: PicHandle; dstRect: Rect);
000425
000426     PROCEDURE PicComment(kind, dataSize: INTEGER; dataHandle: QDHandle);
000427
000428     PROCEDURE KillPicture(myPicture: PicHandle);
000429
000430     PROCEDURE SetStdProcs(VAR procs: QDProcs);
000431
000432     PROCEDURE StdText(count: INTEGER; textAddr: QDPtr; numer, denom: Point);
000433
000434     PROCEDURE StdLine(newPt: Point);
000435
000436     PROCEDURE StdRect(verb: GrafVerb; r: Rect);
000437
000438     PROCEDURE StdRRect(verb: GrafVerb; r: Rect; ovWd, ovHt: INTEGER);
000439
000440     PROCEDURE StdOval(verb: GrafVerb; r: Rect);
000441
000442     PROCEDURE StdArc(verb: GrafVerb; r: Rect; startAngle, arcAngle: INTEGER);
000443
000444     PROCEDURE StdPoly(verb: GrafVerb; poly: PolyHandle);
000445
000446     PROCEDURE StdRgn(verb: GrafVerb; rgn: RgnHandle);
000447
000448     PROCEDURE StdBits(VAR srcBits: BitMap; VAR srcRect, dstRect: Rect;
000449                      mode: INTEGER; maskRgn: RgnHandle);
000450
000451     PROCEDURE StdComment(kind, dataSize: INTEGER; dataHandle: QDHandle);
000452
000453     FUNCTION StdTxMeas(count: INTEGER; textAddr: QDPtr; VAR numer, denom: Point;
000454                       VAR info: FontInfo): INTEGER;
000455
000456     PROCEDURE StdGetPic(dataPtr: QDPtr; byteCount: INTEGER);
000457
```

Apple Lisa Computer Technical Information

```
000458    PROCEDURE StdPutPic(dataPtr: QDPtr; byteCount: INTEGER);
000459
000460    FUNCTION GetPixel(h, v: INTEGER): BOOLEAN;
000461
000462    FUNCTION Random: INTEGER;
000463
000464    PROCEDURE StuffHex(thingptr: QDPtr; s: Str255);
000465
000466    PROCEDURE ForeColor(color: LongInt);
000467
000468    PROCEDURE BackColor(color: LongInt);
000469
000470    PROCEDURE ColorBit(whichBit: INTEGER);
000471
000472    *****
000473    *
000474    *                THAT'S ALL FOLKS ...
000475    *
000476    *****
000477
```

End of File -- Lines: 477 Characters: 12804

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 RECOVERY.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : RECOVERY
000004 *
000005 *****
000006
000007 USES { $U+ } RECOVERY;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015     { $IFC OSBUILT }
000016     { $U libsm/unitstd.obj } unitstd,
000017     { $U libdb/dbenv.obj } dbenv,
000018     { $U libin/INTRLIBp.obj } international,
000019     { $U libdb/dbdecl1.obj } dbdecl1,
000020     { $U libos/syscall.obj } syscall,
000021     { $U libos/psyscall.obj } psyscall,
000022     { $U libdb/lowlevel.obj } lowlevel,
000023     { $U libdb/pooler.obj } pooler,
000024     { $U libdb/heap.obj } heap,
000025     { $U libdb/czcompact.obj } czCompact,
000026     { $U libdb/vltree.obj } vltree,
000027     { $U libdb/scan.obj } scan,
000028     { $U libdb/labscan.obj } labscan,
000029     { $U libdb/schema.obj } schema;
000030     { $ELSEC }
000031     { $U OBJ:dbenv.obj } dbenv,
000032     { $U INTRLIB.obj } international,
000033     { $U OBJ:dbdecl1.obj } dbdecl1,
000034     { $U OBJ:syscall.obj } syscall,
000035     { $U OBJ:lowlevel.obj } lowlevel,
000036     { $U OBJ:pooler.obj } pooler,
000037     { $U OBJ:heap.obj } heap,
000038     { $U OBJ:czcompact.obj } czCompact,
000039     { $U OBJ:vltree.obj } vltree,
000040     { $U OBJ:scan.obj } scan,
000041     { $U OBJ:labscan.obj } labscan,
000042     { $U OBJ:schema.obj } schema;
000043     { $ENDC }
000044
000045     const
000046
000047
000048     WarnIndex = 0;
000049     WarnNoSchema = 1;
000050     WarnOldSchema = 2;
000051     BadFileDesc = 3400;
000052     BadFileSchema = 3401;
000053
000054     PROCEDURE FileRecover(VAR ddresult: integer; VAR warnings: elemset;
000055                           oldprefix, newprefix, pfname, pnewname,
000056                           precname: ptrpathname;
000057
000058                           PROCEDURE
000059                           ECheck(EC: ecstr; ftype: integer; VAR Check: checkinfo;
000060                                   VAR result: integer);
000061
```

Apple Lisa Computer Technical Information

```
000062          PROCEDURE
000063          indexmake(VAR ddresult: integer; ifilename, filename,
000064                    tempdir1, tempdir2: ptrpathname;
000065                    temp_ldsn: integer; nisort, nikeys: integer;
000066                    piflds: ptridesc; duplok: boolean);
000067
000068          PROCEDURE
000069          MarksRevenge(VAR result: integer;
000070                      VAR warnings: elemset; lsid: integer);
000071
000072          ToolsMarketCode: integer);
000073
000074 *****
000075 *
000076 *          THAT'S ALL FOLKS ...
000077 *
000078 *****
000079
```

End of File -- Lines: 79 Characters: 2585

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 REFS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : REFS
000004 *
000005 *****
000006
000007 USES {$U+} REFS;
000008
000009
000010 {$IFC isIntrin} INTRINSIC; {$ENDC}
000011
000012 INTERFACE
000013
000014     USES
000015         {$IFC SrcOnOS }
000016             (*$U OBJ/PASDEFS.OBJ *) PasDefs,
000017             (*$U OBJ/MEMMAN.OBJ *) MemMan,
000018             (*$U OBJ/LISTS.OBJ *) Lists;
000019         {$ELSEC }
000020             (*$U OBJ: PASDEFS.OBJ *) PasDefs,
000021             (*$U OBJ: MEMMAN.OBJ *) MemMan,
000022             (*$U OBJ: LISTS.OBJ *) Lists;
000023         {$ENDC }
000024
000025     TYPE
000026
000027
000028         Ref           = integer;
000029         RefHRow       = ARRAY [ - 128..127] OF Ref;
000030         RefHRowPtr    = ^RefHRow;
000031         RefHArr       = ARRAY [ - 128..127] OF RefHRowPtr;
000032         RefHPtr       = ^RefHArr;
000033         RefRec        = RECORD
000034             refloc: integer;
000035             nextref: Ref;
000036         END;
000037         RefRow        = ARRAY [ - 128..127] OF RefRec;
000038         RefRowPtr     = ^RefRow;
000039         RefArr        = ARRAY [ - 128..127] OF RefRowPtr;
000040         RefArrPtr     = ^RefArr;
000041         RefHandle     = ^RefObject;
000042         RefObject     = RECORD
000043             memRefs: MMHandle;
000044             refHeads: RefHPtr;
000045             refStore: RefArrPtr;
000046             refsLH: ListHandle;
000047         END;
000048
000049     FUNCTION AddRef(rh: RefHandle; head: Ref; l: integer; FUNCTION
000050         GetMore(n, row: integer): boolean): Ref;
000051
000052     PROCEDURE EachRef(rh: RefHandle; head: Ref; PROCEDURE
000053         Visit(i: Ref));
000054
000055     FUNCTION MoreHeads(rh: RefHandle; n, row: integer): boolean;
000056
000057     PROCEDURE NewRefList(rh: RefHandle; head: Ref);
000058
000059     PROCEDURE InitRLists;
000060
000061     FUNCTION InitRefs(nHeads, nRefs: longint): RefHandle;
```

Apple Lisa Computer Technical Information

```
000062
000063     PROCEDURE CopyRefLists(rh: RefHandle; old, new: Ref);
000064
000065     *****
000066     *
000067     *                 THAT'S ALL FOLKS ...
000068     *
000069     *****
000070
```

End of File -- Lines: 70 Characters: 1976

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 SCAN.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SCAN
000004 *
000005 *****
000006
000007 USES {$U+} SCAN;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$IFC OSBUILT }
000016             {$U libsm/unitstd.obj} unitstd,
000017             {$U libdb/dbenv.obj } dbenv,
000018             {$U libin/INTRLIBp.obj } international,
000019             {$U libdb/dbdecl1.obj } dbdecl1,
000020             {$U libos/syscall.obj } syscall,
000021             {$U libos/psyscall.obj } psyscall,
000022             {$U libdb/lowlevel.obj } lowlevel,
000023             {$U libdb/pooler.obj } pooler,
000024             {$U libdb/heap.obj } heap,
000025             {$U libdb/czcompact.obj } czCompact,
000026             {$U libdb/vltree.obj } vltree;
000027         {$ELSEC}
000028             {$U OBJ:dbenv.obj } dbenv,
000029             {$U INTRLIB.obj } international,
000030             {$U OBJ:dbdecl1.obj } dbdecl1,
000031             {$U OBJ:syscall.obj } syscall,
000032             {$U OBJ:lowlevel.obj } lowlevel,
000033             {$U OBJ:pooler.obj } pooler,
000034             {$U OBJ:heap.obj } heap,
000035             {$U OBJ:czcompact.obj } czCompact,
000036             {$U OBJ:vltree.obj } vltree;
000037         {$ENDC}
000038
000039     FUNCTION SParmsBad(scanid: integer; which: integer; nsearch: integer;
000040                       pentry: ptrdata): integer;
000041
000042     FUNCTION scanidbad(scanid: integer): boolean;
000043
000044     PROCEDURE pstack_validate(VAR sresult: integer);
000045
000046     FUNCTION keysize(fileid: integer; pentry: ptrdata): integer;
000047
000048     PROCEDURE einser(VAR sresult: integer; VAR offender: integer;
000049                     scanid: integer; pentry: ptrdata; size: integer;
000050                     newticket: boolean);
000051
000052     PROCEDURE efetch(VAR sresult: integer; scanid: integer; which: integer;
000053                     nsearch: integer; pentry: ptrdata);
000054
000055     PROCEDURE eupdate(VAR sresult: integer; VAR offender: integer;
000056                      scanid: integer; which: integer; nsearch: integer;
000057                      pentry: ptrdata; pnewrec: ptrdata; size: integer);
000058
000059     PROCEDURE edelete(VAR sresult: integer; scanid: integer; which: integer;
000060                      nsearch: integer; pentry: ptrdata);
000061
```

Apple Lisa Computer Technical Information

```
000062 *****  
000063 *  
000064 *           THAT'S ALL FOLKS ...  
000065 *  
000066 *****  
000067
```

End of File -- Lines: 67 Characters: 2224

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 SCHEMA.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SCHEMA
000004 *
000005 *****
000006
000007 USES { $U+ } SCHEMA;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $IFC OSBUILT }
000016             { $U libsm/unitstd.obj } unitstd,
000017             { $U libdb/dbenv.obj } dbenv,
000018             { $U libin/INTRLIBp.obj } international,
000019             { $U libdb/dbdecl1.obj } dbdecl1,
000020             { $U libos/syscall.obj } syscall,
000021             { $U libos/psyscall.obj } psyscall,
000022             { $U libdb/lowlevel.obj } lowlevel,
000023             { $U libdb/pooler.obj } pooler,
000024             { $U libdb/heap.obj } heap,
000025             { $U libdb/czcompact.obj } czCompact,
000026             { $U libdb/vltree.obj } vltree,
000027             { $U libdb/scan.obj } scan,
000028             { $U libdb/labscan.obj } labscan;
000029         { $ELSEC }
000030             { $U OBJ:dbenv.obj } dbenv,
000031             { $U INTRLIB.obj } international,
000032             { $U OBJ:dbdecl1.obj } dbdecl1,
000033             { $U OBJ:syscall.obj } syscall,
000034             { $U OBJ:lowlevel.obj } lowlevel,
000035             { $U OBJ:pooler.obj } pooler,
000036             { $U OBJ:heap.obj } heap,
000037             { $U OBJ:czcompact.obj } czCompact,
000038             { $U OBJ:vltree.obj } vltree,
000039             { $U OBJ:scan.obj } scan,
000040             { $U OBJ:labscan.obj } labscan;
000041         { $ENDC }
000042
000043     const
000044
000045
000046         indexfile      = 0;
000047         fileschema     = 1;
000048         queryCheckPoint = 2;
000049         not_needed     = 0;
000050         via_only       = 1;
000051         on_only        = 2;
000052         on_and_via     = 3;
000053         new_index      = 4;
000054
000055     TYPE
000056         isegment       = PACKED RECORD
000057                         aord: 0..1;
000058                         field: 0..32767;
000059                         END;
000060         ptrisegment    = ^isegment;
000061         idesc          = ARRAY [0..0] OF isegment;
```

Apple Lisa Computer Technical Information

```
000062      ptridesc      = ^idesc;
000063      ientry         = RECORD
000064                      lab: labelentry;
000065                      iduplok: boolean;
000066                      isort: integer;
000067                      ikeys: integer;
000068                      iunique: uniqueid;
000069                      iflds: idesc;
000070                      END;
000071      ptrientry      = ^ientry;
000072
000073      PROCEDURE filldesc(VAR ddresult: integer; pfile: ptrfiledesc);
000074
000075      PROCEDURE makefile(VAR ddresult: integer; pname: ptrpathname;
000076                        pfdesc: ptrfiledesc);
000077
000078      PROCEDURE openindexes(VAR ddresult: integer; VAR viafile: integer;
000079                          scanid: integer; ifilename: ptrpathname;
000080                          intent: integer; labelusage: integer);
000081
000082      PROCEDURE cvt_scan(VAR ddresult: integer; scanid, sintent,
000083                       sviafile: integer);
000084
000085      PROCEDURE initpool(VAR ddresult: integer; first_time: boolean;
000086                       heap_ldsn: integer; nbuff: integer; dheapsize: integer;
000087                       pheapname: ptrpathname; dstype: Tdtype;
000088                       pagesize: integer);
000089
000090      PROCEDURE flushscan(VAR ddresult: integer; scanid: integer);
000091
000092      PROCEDURE closescan(VAR ddresult: integer; scanid: integer);
000093
000094      PROCEDURE openscan(VAR ddresult: integer; onname, vianame: ptrpathname;
000095                       VAR scanid: integer; sintent: integer);
000096
000097      PROCEDURE makedescfile(VAR ddresult: integer; pname: ptrpathname;
000098                            pfdesc: ptrfiledesc);
000099
000100      PROCEDURE clonefile(VAR ddresult: integer; sourcefile, filename: ptrpathname;
000101                        needtomakefile: boolean);
000102
000103      PROCEDURE makeformfile(VAR ddresult: integer; pform: ptrdata;
000104                            pname: ptrpathname; needtomakefile: boolean);
000105
000106      PROCEDURE quick_label_scan(VAR ddresult: integer; scanid: integer;
000107                                sintent: integer; VAR lscanid: integer);
000108
000109      PROCEDURE get_header(VAR ddresult: integer; scanid: integer;
000110                          VAR head: header);
000111
000112      PROCEDURE getform(VAR ddresult: integer; scanid: integer; pdesc: ptrdata);
000113
000114      PROCEDURE deletefile(VAR ddresult: integer; pname: ptrpathname);
000115
000116      PROCEDURE deleteindex(VAR ddresult: integer; piname, pname: ptrpathname);
000117
000118      *****
000119      *
000120      *                      THAT'S ALL FOLKS ...
000121      *
000122      *****
000123
```

End of File -- Lines: 123 Characters: 4043

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 SCRAP.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SCRAP
000004 *
000005 *****
000006
000007 USES {$U+} SCRAP;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/UnitStd      } UnitStd,
000016         {$U libsm/UnitHz      } UnitHz,
000017         {$U libos/SysCall     } SysCall,
000018         {$U libsu/UnitFile    } UnitFile,
000019         {$U libqd/QuickDraw   } QuickDraw,
000020         {$U libfm/FontMgr     } FontMgr,
000021         {$U libqd/Storage     } Storage,
000022         {$U libwm/Events      } Events,
000023         {$U libwm/Folders     } Folders,
000024         {$U libsb/WmlStd      } WmlStd,
000025         {$U libsb/Wmlsb       } Wmlsb,
000026         {$U libsu/UnitFmt     } UnitFmt,
000027         {$U libpm/PmDecl      } PmDecl,
000028         {$U libpr/PrStdInfo   } PrStdInfo,
000029         {$U libsu/UnitCS      } UnitCS,
000030         {$U libsu/UnitFigAtom } UnitFigAtom,
000031         {$U libsu/UnitFf      } UnitFf;
000032
000033     CONST
000034         ScrapNil      = 0;
000035         ScrapFE       = 2;
000036         ScrapMtx      = 3;
000037         ScrapBGraf    = 4;
000038         ScrapList     = 5;
000039         ScrapDwg      = 6;
000040         ScrapPert     = 7;
000041         ScrapCs       = 12;
000042         ScrapUG       = 13;
000043         SecondOpenScrapCall = 4050;
000044         HZINITfailed  = 4051;
000045         NoUndoScrap   = 4052;
000046         TooManyProcesses = 4053;
000047         ProcCalledByNonOwner = 4054;
000048         ProcessNotFound = 4055;
000049         ScrapNotPicture = 4059;
000050         DataSegNotOpen = - 4060;
000051
000052     TYPE
000053         ScrapType     = 0..15;
000054
000055     VAR
000056         icsFfScrap:   TB;
000057         ifilScrap:    TB;
000058         CurrScrapSet: SET OF ScrapType;
000059         Scrap1Figures_RefNum, Scrap2Figures_RefNum, Scrap1Lotus_RefNum,
000060         Scrap2Lotus_RefNum: TL;
000061
```

Apple Lisa Computer Technical Information

```
000062     PROCEDURE InitScrap(VAR InitErr: Integer);
000063
000064     PROCEDURE KillScrapOwner(Who: ProcessId; VAR KillErr: TC);
000065
000066     PROCEDURE ReviveScrapOwner(Who: ProcessId);
000067
000068     PROCEDURE DrawUScrap(VAR DrawErr: TC);
000069
000070     PROCEDURE BackOutOfScrap;
000071
000072     PROCEDURE OpenScrap(VAR OpenErr: Integer);
000073
000074     FUNCTION DSegOfScrap: Integer;
000075
000076     FUNCTION AddrOfScrapDseg: LongInt;
000077
000078     FUNCTION HzOfScrap: THz;
000079
000080     PROCEDURE ClaimScrap;
000081
000082     PROCEDURE InheritScrap(SaveOld: Boolean);
000083
000084     PROCEDURE UndoInheritScrap(VAR UndoErr: integer);
000085
000086     PROCEDURE AcceptInheritScrap;
000087
000088     PROCEDURE EraseScrapData(VAR EraseErr: integer);
000089
000090     PROCEDURE StartPutScrap(VAR PutErr: integer);
000091
000092     PROCEDURE PutScrap(which: ScrapType; what: TH; VAR PutErr: integer);
000093
000094     PROCEDURE PutGrScrap(What: PicHandle; VAR PutErr: integer);
000095
000096     PROCEDURE PutCsScrap(icsContents: TB; VAR PutCSErr: integer);
000097
000098     PROCEDURE EndPutScrap(VAR PutErr: integer);
000099
000100     PROCEDURE StartGetScrap(VAR GetErr: integer);
000101
000102     PROCEDURE GetScrap(VAR which: ScrapType; VAR what: TH);
000103
000104     PROCEDURE GetGrScrap(VAR What: PicHandle);
000105
000106     PROCEDURE GetCsScrap(VAR Contents: TB);
000107
000108     PROCEDURE EndGetScrap(VAR GetErr: integer);
000109
000110     PROCEDURE bindUTDseg(VAR BindErr: integer);
000111
000112     PROCEDURE unBindUTDseg(VAR UnBindErr: integer);
000113
000114     *****
000115     *
000116     *           THAT'S ALL FOLKS ...
000117     *
000118     *****
000119
```

End of File -- Lines: 119 Characters: 2953

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 SHELLCOMM.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SHELLCOM
000004 *
000005 *****
000006
000007 USES { $U+ } SHELLCOM;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015     { $U StdUnit } StdUnit;
000016
000017     CONST
000018         SCNone          = - 1;
000019         SCAny           = 0;
000020         SCText          = 1;
000021         SCBufMax       = 1023;
000022         SC_SetReallyStop = 1;
000023         SC_GetReallyStop = 2;
000024         SC_SetUnSavedEdits = 6960;
000025         SC_GetUnSavedEdits = 8751;
000026
000027     VAR
000028         SCRetStr:      SUStr;
000029         SCRunCmd:     SUStr;
000030         SCBuf:        PACKED ARRAY [0..1023] OF CHAR;
000031         SCExecAbort:  BOOLEAN;
000032         SCReallyStopExec: BOOLEAN;
000033         SCUnSavedEdits: BOOLEAN;
000034
000035     PROCEDURE SCInit;
000036
000037     PROCEDURE SCSetRunCmd(RC: SUStr);
000038
000039     PROCEDURE SCSetRetStr(RS: SUStr);
000040
000041     PROCEDURE SCSetExecAbort(B: BOOLEAN);
000042
000043     PROCEDURE SCReWrite(WriteType: INTEGER; Key: SUStr);
000044
000045     FUNCTION SCReset(ReadType: INTEGER; Key: SUStr): BOOLEAN;
000046
000047     FUNCTION SCClose(KillBuf: BOOLEAN; Key: SUStr): BOOLEAN;
000048
000049     FUNCTION SCPutCh(Ch: CHAR): BOOLEAN;
000050
000051     FUNCTION SCGetCh(VAR Ch: CHAR): BOOLEAN;
000052
000053     FUNCTION SCPutLine(L: SUStr): BOOLEAN;
000054
000055     FUNCTION SCGetLine(VAR L: SUStr): BOOLEAN;
000056
000057     FUNCTION SCShellCmd(Cmd: INTEGER; P: SUStr): BOOLEAN;
000058
000059 *****
000060 *
000061 *                                     THAT'S ALL FOLKS ...
```

Apple Lisa Computer Technical Information

000062 *
000063 *****
000064

End of File -- Lines: 64 Characters: 1441

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 STDUNIT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : STDUNIT
000004 *
000005 *****
000006
000007 USES {$U+} STDUNIT;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libOS/SysCall.obj } SysCall,
000016         {$U libPL/PasLibCall.obj } PasLibCall,
000017         {$U libPL/PPasLibC.obj } PPasLibC;
000018
000019     CONST
000020         SUMaxStrLeng = 255;
000021         SUNullstr    = '';
000022         SUSpace      = ' ';
000023         SUOrdCR      = 13;
000024         SUMaxPNLeng  = 66;
000025         SUMaxVNLeng  = 33;
000026         SUMaxFNLeng  = 32;
000027         SUVolSuffix  = '-';
000028
000029     TYPE
000030         SUSetOfChar  = SET OF CHAR;
000031         SUStrP       = ^SUSTR;
000032         SUSTR        = STRING[255];
000033         SUVolName    = STRING[SUMaxVNLeng];
000034         SUFile       = FILE;
000035         SUFileP      = ^SUFile;
000036         PromptState  = (SUDefault, SUEscape, SUNone, SUOptions, SUValid,
000037             SUInvalid);
000038         ErrTextRet   = (SUOK, SUBadEFOpen, SUBadEFRead, SUErrNNotFound);
000039         ConvNState   = (SUValidN, SUNoN, SUBadN, SUNOverFlow);
000040
000041     VAR
000042         SUOsBootV:   SUVolName;
000043         SUMyProcV:   SUVolName;
000044         SUBell, SUBackSpace, SUCr, SUTab, SUEsc, SUDle, SUNul: CHAR;
000045         SUNulls:     SUSTR;
000046         SUKeyBoard:  INTERACTIVE;
000047
000048     PROCEDURE SUInit;
000049
000050     PROCEDURE SUDone;
000051
000052     FUNCTION SUUpCh(Ch: CHAR): CHAR;
000053
000054     FUNCTION SULowCh(Ch: CHAR): CHAR;
000055
000056     PROCEDURE SUUpStr(S: SUSTRP);
000057
000058     PROCEDURE SULowStr(S: SUSTRP);
000059
000060     FUNCTION SUEqStr(S1: SUSTRP; S2: SUSTRP): BOOLEAN;
000061
```

Apple Lisa Computer Technical Information

```
000062 FUNCTION SUEq2Str(S1: SStrP; S2: SStr): BOOLEAN;
000063
000064 PROCEDURE SUTrimLeading(S: SStrP);
000065
000066 PROCEDURE SUTrimTrailing(S: SStrP);
000067
000068 PROCEDURE SUTrimBlanks(S: SStrP);
000069
000070 PROCEDURE SUAddCh(S: SStrP; Ch: CHAR; MaxStrLeng: INTEGER;
000071 VAR OverFlow: BOOLEAN);
000072
000073 PROCEDURE SUConcat(S1: SStrP; S2: SStrP);
000074
000075 PROCEDURE SUAddStr(S1: SStrP; S2: SStrP; MaxStrLeng: INTEGER;
000076 VAR OverFlow: BOOLEAN);
000077
000078 PROCEDURE SUSetStr(Dest: SStrP; Src: SStrP);
000079
000080 PROCEDURE SUCopyStr(Dest: SStrP; Src: SStrP; Start, Count: INTEGER);
000081
000082 FUNCTION SUIsVolName(FN: SStrP): BOOLEAN;
000083
000084 PROCEDURE SUVolPart(PathN: SStrP; VolN: SStrP);
000085
000086 PROCEDURE SUAddExtension(FN: SStrP; DefExt: SStr; MaxStrLeng: INTEGER;
000087 VAR OverFlow: BOOLEAN);
000088
000089 PROCEDURE SUSplitFN(PathN: SStrP; CatN: SStrP; FN: SStrP; Ext: SStrP);
000090
000091 PROCEDURE SUMakeFN(PathN: SStrP; CatN: SStrP; FN: SStrP; Ext: SStr;
000092 VAR OverFlow: BOOLEAN);
000093
000094 PROCEDURE SUCHkFN(FN: SStrP; VAR PState: PromptState; DefVol: SStr;
000095 DefFN: SStr; DefExt: SStr);
000096
000097 PROCEDURE SUGetCh(VAR Ch: CHAR);
000098
000099 PROCEDURE SUGetLine(S: SStrP; VAR PState: PromptState);
000100
000101 PROCEDURE SUGetStr(S: SStrP; VAR PState: PromptState; DefVal: SStr);
000102
000103 PROCEDURE SUGetFN(FN: SStrP; VAR PState: PromptState; DefVol: SStr;
000104 DefFN: SStr; DefExt: SStr);
000105
000106 PROCEDURE SUGetInt(VAR I: INTEGER; VAR PState: PromptState; DefVal: INTEGER);
000107
000108 PROCEDURE SUWaitEscOrSp(VAR PState: PromptState);
000109
000110 PROCEDURE SUWaitSp;
000111
000112 PROCEDURE SUGetChInSet(VAR Ch: CHAR; Chars: SSetOfChar);
000113
000114 FUNCTION SUGetYesNo: BOOLEAN;
000115
000116 FUNCTION SUGetBool(Default: BOOLEAN): BOOLEAN;
000117
000118 PROCEDURE SUGetErrText(ErrFN: SStr; ErrN: INTEGER; ErrMsg: SStrP;
000119 VAR ErrRet: ErrTextRet);
000120
000121 PROCEDURE SUErrText(ErrFN: SStr; ErrN: INTEGER; ErrMsg: SStrP);
000122
000123 PROCEDURE SUStopExec(VAR ErrNum: INTEGER);
000124
000125 PROCEDURE SUCloseExec(VAR ErrNum: INTEGER);
000126
000127 PROCEDURE SUInitSysVols;
```


Apple Lisa Computer Technical Information

```
000128
000129     PROCEDURE SUSysReset(F: SUFileP; FN: SStr; VAR IOSTatus: INTEGER);
000130
000131     PROCEDURE SUIntToStr(N: INTEGER; S: SStrP);
000132
000133     PROCEDURE SULIntToStr(N: LONGINT; S: SStrP);
000134
000135     PROCEDURE SStrToInt(NS: SStrP; VAR N: INTEGER; VAR CState: ConvNState);
000136
000137     PROCEDURE SStrToLInt(NS: SStrP; VAR N: LONGINT; VAR CState: ConvNState);
000138
000139 *****
000140 *
000141 *           THAT'S ALL FOLKS ...
000142 *
000143 *****
000144
```

End of File -- Lines: 144 Characters: 3939

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 STORAGE.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : STORAGE
000004 *
000005 *****
000006
000007 USES { $U+ } STORAGE;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES                               { $U libsm/UnitStd } UnitStd,
000015     { $U libsm/UnitHz   } UnitHz;
000016
000017     TYPE
000018     Ptr           = TP;
000019     Handle        = TH;
000020     RelHandle     = INTEGER;
000021     ProcPtr       = TProc;
000022
000023     VAR
000024     theHeap:      THz;
000025     ordHeap:      LongInt;
000026
000027     PROCEDURE InitHeap(startPtr, limitPtr: Ptr; errorProc: ProcPtr);
000028
000029     PROCEDURE SetHeap(hz: Thz);
000030
000031     PROCEDURE GetHeap(VAR hz: Thz);
000032
000033     FUNCTION NewPtr(byteCount: INTEGER): Ptr;
000034
000035     PROCEDURE DisposePtr(p: Ptr);
000036
000037     FUNCTION NewHandle(byteCount: INTEGER): Handle;
000038
000039     PROCEDURE DisposeHandle(h: Handle);
000040
000041     PROCEDURE SetSize(h: Handle; newSize: INTEGER);
000042
000043     FUNCTION GetSize(h: Handle): INTEGER;
000044
000045 *****
000046 *
000047 *                               THAT'S ALL FOLKS ...
000048 *
000049 *****
000050
```

End of File -- Lines: 50 Characters: 1075

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 SYS1LOCK.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SYS1LOCK
000004 *
000005 *****
000006
000007 USES {$U+} SYS1LOCK;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/UnitStd      } UnitStd,
000016         {$U libos/SysCall.Obj  } SysCall,
000017         {$U libos/PSysCall.obj } PSysCall;
000018
000019     PROCEDURE LkSYSINIT(VAR err: INTEGER);
000020
000021     PROCEDURE LkOPSEG1(VAR err: INTEGER);
000022
000023     PROCEDURE LkOPSEG2(VAR err: INTEGER);
000024
000025     PROCEDURE LkACTIVATE(VAR err: Integer);
000026
000027     PROCEDURE LkWORKSET(VAR err: INTEGER);
000028
000029     PROCEDURE LkGRAFSEG(VAR err: INTEGER);
000030
000031     PROCEDURE LkMOVERS(VAR err: INTEGER);
000032
000033     PROCEDURE LkOPEN1LW(VAR err: Integer);
000034
000035     PROCEDURE LkLWORK(VAR err: Integer);
000036
000037     PROCEDURE LkOTHRWORK(VAR err: Integer);
000038
000039     PROCEDURE LkNEVER(VAR err: INTEGER);
000040
000041     PROCEDURE LkFmgrutil(VAR err: Integer);
000042
000043     PROCEDURE LkPMMSeg(VAR err: INTEGER);
000044
000045     PROCEDURE LkWMwarm(VAR err: Integer);
000046
000047     PROCEDURE LkWMJrnl(VAR err: Integer);
000048
000049     PROCEDURE LkWMalert(VAR err: INTEGER);
000050
000051     PROCEDURE Lkalert(VAR err: INTEGER);
000052
000053     PROCEDURE LkSMcold(VAR err: INTEGER);
000054
000055     PROCEDURE LkSUCold(VAR err: INTEGER);
000056
000057     PROCEDURE LkFECold(VAR err: INTEGER);
000058
000059     PROCEDURE LkSBCold(VAR err: INTEGER);
000060
000061     PROCEDURE LkINCold(VAR err: INTEGER);
```

Apple Lisa Computer Technical Information

```
000062
000063     PROCEDURE LkWMcold(VAR err: Integer);
000064
000065     *****
000066     *
000067     *               THAT'S ALL FOLKS ...
000068     *
000069     *****
000070
```

End of File -- Lines: 70 Characters: 1513

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 TEENV.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : TEENV
000004 *
000005 *****
000006
000007 USES { $U+ } TEENV;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U libsm/UnitStd } UnitStd,
000016         { $U libsm/UnitHz } UnitHz,
000017         { $U libqd/Storage } Storage,
000018         { $U libqd/QuickDraw } QuickDraw,
000019         { $U libfm/FontMgr } FontMgr,
000020         { $U libdb/dbenv } dbenv,
000021         { $U libfe/fedec } FEDec,
000022         { $U libfe/fld } fieldedit;
000023         { $SETC teProduction := NOT FDbgOK }
000024         { $SETC teSym := FSymOK }
000025         { $SETC TEDEBUG := NOT teProduction }
000026         { $SETC doTraceTE := TRUE }
000027         { $SetC fTraceTE := doTraceTE AND fTRACE }
000028
000029     CONST
000030         OK                = 0;
000031         OutOfMem          = 1;
000032         InvalidCell      = 2;
000033         rmDel             = 3;
000034         rmOk              = 4;
000035         rmReName         = 5;
000036         parseErr         = 6;
000037         undoErr          = 7;
000038         badField         = 8;
000039         cvnotenuff       = 9;
000040         crnotenuff       = 10;
000041         nullSnip         = 0;
000042         pInfSnip         = - 2;
000043         mInfSnip         = - 1;
000044         nullTimeout      = 0;
000045         dfltTimeout      = - 1;
000046         cFolderpnl      = 0;
000047         cCellPnl         = 1;
000048         cTblPnl          = 2;
000049         cWidePnl         = 3;
000050         cDialogPnl       = 4;
000051         cScrapPnl        = 5;
000052         nullPnl          = - 1;
000053         cBitmapH         = 720;
000054         cBitmapV         = 20;
000055         { $IFC NOT TEDEBUG }
000056         TraceFmgr        = FALSE;
000057         { $ENDC }
000058
000059     TYPE
000060         bitptr            = ^Bitmap;
000061         PnlIndex         = INTEGER;
```

Apple Lisa Computer Technical Information

```

000062   Panel           = RECORD
000063                   PnlPort: GrafPtr;
000064                   Offset: Point;
000065                   PnlRect: Rect;
000066                   END;
000067   panelptr        = ^Panel;
000068   pnlHandle       = ^panelptr;
000069   STR40           = string[40];
000070   str9            = string[9];
000071   str100          = string[100];
000072   pixel          = integer;
000073   idType          = integer;
000074   rgKind          = (aNullRg, aCellRg, aRectRg, aRowRg, aColRg, aRowGrid,
000075                   aColGrid, aRowHedRg, aColHedRg, aWTBoxRg, aTblHedRg);
000076   rgJust          = (topJ, botJ, leftJ, rightJ, botLeftJ, topLeftJ, topRightJ,
000077                   nearJ);
000078   rgX             = ^range;
000079   range           = RECORD
000080                   rKind: rgKind;
000081                   loRow: idType;
000082                   loCol: idType;
000083                   hiRow: idType;
000084                   hiCol: idType;
000085                   END;
000086   fract           = INTEGER;
000087   snipKind        = (rowKind, colKind);
000088   SelKind         = (aNullSl, aCellTxtSl, aCellSl, aRectRgSl, aRowRgSl,
000089                   aColRgSl, aRowGrdSl, aColGrdSl, aRowHedSl, aColHedSl,
000090                   aTblHedSl, aWTBoxSl);
000091   TblPart         = (outOfTbl, cellArea, rowGrid, colGrid, rowHandl, colHandl,
000092                   rowHed, colHed, whTbBox, tblHed);
000093   TableRecord     = RECORD
000094                   SplitTable: BOOLEAN;
000095                   RectBorder: BOOLEAN;
000096                   Marquee: BOOLEAN;
000097                   ColHandles: BOOLEAN;
000098                   ColGrids: BOOLEAN;
000099                   ColGridHandles: BOOLEAN;
000100                   ColRangeOk: BOOLEAN;
000101                   EditColTitle: BOOLEAN;
000102                   InsertCols: BOOLEAN;
000103                   EdBlankCol: BOOLEAN;
000104                   RowHandles: BOOLEAN;
000105                   RowGrids: BOOLEAN;
000106                   RowRangeOk: BOOLEAN;
000107                   RowGridHandles: BOOLEAN;
000108                   EditRowTitle: BOOLEAN;
000109                   InsertRows: BOOLEAN;
000110                   EdBlankRow: BOOLEAN;
000111                   EditTable: BOOLEAN;
000112                   ShoFormulas: BOOLEAN;
000113                   Preview: BOOLEAN;
000114                   RectRgOk: BOOLEAN;
000115                   FieldPad: INTEGER;
000116                   scrollincr: ARRAY [rowkind..colkind] OF INTEGER;
000117                   END;
000118   tmBand          = RECORD
000119                   tmbKind: SnipKind;
000120                   tmbLoP: pixel;
000121                   tmbHiP: pixel;
000122                   tmbHideP: pixel;
000123                   tmbLoId: idtype;
000124                   END;
000125   cellParType     = (cRecord);
000126   idModeType      = (mmfirst, mmlast, mmmnext, mmprior, mmfraction);
000127   mcType          = (mmRight, mmLeft, mmUp, mmDown, mmNone, mmRange,

```

Apple Lisa Computer Technical Information

```
000128           mmmBegOfRow, mmmBegOfCol, mmmRightPeg, mmmLeftPeg);
000129 IOModeType   = (mmRead, mmWrite, mmIONull);
000130 errRecord     = RECORD
000131             errSpot: interval;
000132             mess1: str40;
000133             mess2: str40;
000134             status: integer;
000135             END;
000136 cellRecord    = RECORD
000137             align: integer;
000138             font: Tlfntid;
000139             protected: boolean;
000140             hiLite: boolean;
000141             NumRuns: integer;
000142             dim: boolean;
000143             END;
000144 BGrowResult   = (BSame, BGrew, BWent, nxBWent);
000145 ScrollType    = (incrScroll, pageScroll, jmpScroll);
000146 ScrollDir     = (ScrollBak, ScrollFwd);
000147 TMstate       = RECORD
000148             marqsaved: boolean;
000149             rg: range;
000150             isSel: boolean;
000151             isOn: boolean;
000152             isBorder: boolean;
000153             END;
000154
000155 VAR
000156     {$IFC TEDEBUG}
000157     traceFMGR:   BOOLEAN;
000158     {$ENDC }
000159     dfltNbrhood: Rect;
000160     nullNbrhood: Rect;
000161     TblPars:     TableRecord;
000162     FolderPnl:  PnlIndex;
000163     CellPnl:    PnlIndex;
000164     TblPnl:     PnlIndex;
000165     WidePnl:    PnlIndex;
000166     DialogPnl:  PnlIndex;
000167     ScrapPnl:   PnlIndex;
000168
000169     PROCEDURE AdjRect(Pnl: PnlIndex; VAR r: rect);
000170
000171     PROCEDURE AdjToPnl(Pnl: PnlIndex; P: Point; VAR relPoint: Point);
000172
000173     PROCEDURE allocPnl(thePnl: PnlIndex);
000174
000175     PROCEDURE ChgPnlSize(P: PnlIndex; Width, Height: Pixel);
000176
000177     PROCEDURE FreePnl(thePnl: PnlIndex);
000178
000179     PROCEDURE GetPnlBits(P: pnlIndex; VAR b: bitptr);
000180
000181     PROCEDURE GetPnloffset(Pnl: pnlIndex; VAR offset: point);
000182
000183     PROCEDURE GetPnlRect(PNL: PnlIndex; VAR aPnlRect: Rect);
000184
000185     PROCEDURE InitPnls(VAR status: integer);
000186
000187     PROCEDURE MovePnl(P: PnlIndex; Offset: point);
000188     {$IFC TEDEBUG}
000189
000190     PROCEDURE prntClip;
000191
000192     PROCEDURE PRNTPORT;
000193
```

Apple Lisa Computer Technical Information

```
000194     PROCEDURE prntRect(aRect: rect);
000195
000196     PROCEDURE PrntRg(s: str40; rg: range);
000197     {$ENDC}
000198
000199     FUNCTION ptInPnl(portptr: grafptr; loc: point; hipnl: pnlindex): PnlIndex;
000200
000201     FUNCTION rgEq(aRg, bRg: range): boolean;
000202
000203     FUNCTION SectPnl(Pnl: pnlIndex; portptr: grafptr; updrgn: rgnhandle;
000204         resultrgn: rgnhandle): BOOLEAN;
000205
000206     PROCEDURE SetPnlPort(P: PnlIndex);
000207
000208     PROCEDURE setRg(VAR aRg: range; aKind: rgKind; aLoRow, aLoCol, aHiRow,
000209         aHiCol: idType);
000210
000211     PROCEDURE SetUpPnl(P: PnlIndex; portptr: grafptr; width, Height: pixel;
000212         loc: point);
000213
000214     *****
000215     *
000216     *                THAT'S ALL FOLKS ...
000217     *
000218     *****
000219
```

End of File -- Lines: 219 Characters: 6969

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 TM.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : TM
000004 *
000005 *****
000006
000007 USES { $U+ } TM;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U libsm/UnitHz } UnitHz,
000016         { $U libqd/Storage } Storage,
000017         { $U libqd/QuickDraw } QuickDraw,
000018         { $U libfm/FontMgr } FontMgr,
000019         { $U libdb/dbenv } dbenv,
000020         { $U libfe/FEdec } FEdec,
000021         { $U libfe/flD } fieldedit,
000022         { $U libte/teenv } teenv,
000023         { $U libsb/WMLstd } WMLstd,
000024         { $U libsb/WMLsb } WMLsb;
000025         { $ifc teProduction }
000026         { $SetC tmdebug := False }
000027         { $elsec }
000028         { $SetC tmdebug := True }
000029         { $endc }
000030         { $SetC TMSDebug := tmDebug }
000031
000032     CONST
000033         { $IFC NOT tmdebug }
000034         traceSMgr      = False;
000035         tracetmgr      = false;
000036         traceXTmgr     = false;
000037         fwriteln       = false;
000038         { $ENDC }
000039         ascCR          = 13;
000040         maxString      = 130;
000041         splitDh        = dhSkewer;
000042         splitDv        = dvSkewer;
000043         maxrun         = 5;
000044
000045     TYPE
000046         cellSelType    = (selectCell, selectContents, PutCaretAtEnd);
000047         { $IFC tmdebug }
000048
000049     VAR
000050         traceSMgr, traceTMgr, traceXTMgr, fwriteln: boolean;
000051         { $ENDC }
000052
000053     PROCEDURE ChgRgBorders(newRg: range; shoMarquee: boolean);
000054
000055     PROCEDURE DumpSel;
000056
000057     PROCEDURE EndRgBorders;
000058
000059     PROCEDURE FindTblPt(hv: point; VAR PrtOfTbl: tblPart; VAR rg: range;
000060         VAR virtRect, screenRect: rect);
000061
```

Apple Lisa Computer Technical Information

```
000062 PROCEDURE getCR(VAR rg: range);
000063
000064 PROCEDURE getEditCell(VAR theRow: idType; VAR theCol: idType);
000065
000066 FUNCTION grayOfBand(aKind: snipKind; aP: pixel): fract;
000067
000068 PROCEDURE growBand(aKind: snipKind; aP: pixel; VAR newP: pixel;
000069 VAR result: BGrowResult);
000070
000071 PROCEDURE LDtmgr;
000072
000073 PROCEDURE LDtmLoFreq;
000074
000075 PROCEDURE NewRgBorders(newRg: range; useMarquee: boolean);
000076
000077 PROCEDURE NewRgImg(rg: range);
000078
000079 FUNCTION PtInLCrg(hv: point; targRg: range): boolean;
000080
000081 PROCEDURE ReBldTbl(initRg: range);
000082
000083 PROCEDURE ScrBand(aKind: snipKind; aP: pixel; scrScale: ScrollType;
000084 scrDir: ScrollDir; scrFract: fract; VAR didMove: boolean);
000085
000086 PROCEDURE setCR(rg: range);
000087
000088 PROCEDURE shoNewRg(oldRg: range; newRg: range; just: rgJust);
000089
000090 PROCEDURE ShoNewWid(aKind: snipKind; loSnip, hiSnip: idType);
000091
000092 PROCEDURE ShoTbl(shoRgnH: rgnHandle);
000093
000094 FUNCTION snipAllInBand(aKind: snipKind; aP: pixel; snipId: idType): boolean;
000095
000096 PROCEDURE SplitBand(aKind: snipKind; VAR atP: pixel; VAR didSplit: boolean);
000097
000098 PROCEDURE tmCurVwPt(VAR vwPt: point);
000099
000100 PROCEDURE tmCurVwSet(vwPt: point);
000101
000102 PROCEDURE tmFact(fActivate: boolean);
000103
000104 PROCEDURE tmInit;
000105
000106 PROCEDURE tmLoadBands;
000107
000108 PROCEDURE tmLoadNextBand(aTmBand: tmBand);
000109
000110 PROCEDURE tmLoadState(atmstate: tmstate);
000111
000112 PROCEDURE tmNewPars;
000113
000114 PROCEDURE tmNewSnips(aKind: snipKind; priorSnip: idType; scrollit: boolean);
000115
000116 PROCEDURE tmNextBandRg(VAR bandsnip: snipKind; VAR bandloId,
000117 bandhiId: idType);
000118
000119 PROCEDURE tmReDoViews;
000120
000121 PROCEDURE tmReSize;
000122
000123 PROCEDURE tmSaveBands;
000124
000125 PROCEDURE tmSaveNextBand(VAR aTmBand: tmBand);
000126
000127 PROCEDURE tmsavestate(VAR atmstate: tmstate);
```

Apple Lisa Computer Technical Information

```
000128
000129 PROCEDURE tmSetRgFB(cRg: range; useMarquee: boolean);
000130
000131 PROCEDURE tmShoBreaks;
000132
000133 PROCEDURE tmShoNewTbl;
000134
000135 PROCEDURE TurnOffMarquee;
000136
000137 PROCEDURE TurnOnMarquee;
000138
000139 FUNCTION underEdit(aRow, aCol: idType): boolean;
000140
000141 PROCEDURE visRgOfView(aPt: point; VAR aRg: range);
000142
000143 PROCEDURE InitProcInfo(CellInfo: ProcPtr; CellRun: ProcPtr;
000144                      CellValue: ProcPtr; FoldedFormula: ProcPtr;
000145                      GetSnipId: ProcPtr; mmBreakSnip: ProcPtr;
000146                      mmMapSnip: ProcPtr; SetSnipWidth: ProcPtr;
000147                      SnipLT: ProcPtr; SnipWidth: ProcPtr;
000148                      FatalError: ProcPtr);
000149
000150 PROCEDURE boldflds(VAR status: integer);
000151
000152 FUNCTION CellChanged: Boolean;
000153
000154 PROCEDURE ChangeCellFont;
000155
000156 PROCEDURE ClearSel(VAR status: integer);
000157
000158 PROCEDURE CopySel(VAR status: integer);
000159
000160 PROCEDURE CutSel(VAR status: integer);
000161
000162 PROCEDURE DoSelAct(Activate: boolean);
000163
000164 PROCEDURE DoSelCR(aRg: Range; VAR timeout: INTEGER; hilite: cellseltype);
000165
000166 PROCEDURE DoTblSel(P: point; VAR Nbh: Rect; VAR Timeout: INTEGER);
000167
000168 PROCEDURE DoWavSel(P: Point; VAR Nbh: Rect; VAR Timeout: Integer);
000169
000170 PROCEDURE EndCR;
000171
000172 PROCEDURE EnterSelection(VAR Status: Integer);
000173
000174 PROCEDURE EraseWav;
000175
000176 PROCEDURE ExitSel;
000177
000178 PROCEDURE ExtendSel(P: point; VAR nbh: Rect; VAR delta: INTEGER);
000179
000180 PROCEDURE GetSelKind(VAR theSelKind: selKind);
000181
000182 PROCEDURE InitSelMgr(wavActFlg: boolean; maxLenWav: integer;
000183                   growdata: integer; growrun: integer);
000184
000185 PROCEDURE InKey(ch: char; shiftflag: boolean; cmdflag: Boolean;
000186               resetrun: boolean; VAR Timeout: INTEGER;
000187               VAR status: INTEGER);
000188
000189 PROCEDURE LDsmgr;
000190
000191 PROCEDURE MovePointer(P: Point; VAR pRg: range; VAR nbh: rect);
000192
000193 PROCEDURE MseUp(P: Point; VAR delta: INTEGER);
```

Apple Lisa Computer Technical Information

```
000194
000195   PROCEDURE NewNbh(P: point; VAR nbh: Rect; VAR delta: INTEGER);
000196
000197   PROCEDURE NewNbhWav(P: Point; VAR nbh: Rect; VAR delta: Integer);
000198
000199   PROCEDURE NewTimOut(VAR delta: INTEGER);
000200
000201   PROCEDURE NewWsel(P: Point; Pnl: PnlIndex; VAR Nbh: Rect;
000202                   VAR timeout: integer);
000203
000204   PROCEDURE NullCR;
000205
000206   PROCEDURE NullSel;
000207
000208   PROCEDURE PackSMGR(hndsusdata: hnddata; offset: integer; curlen: integer;
000209                   VAR leninstalled: integer; VAR newlen: integer);
000210
000211   PROCEDURE PasteSel(pasteruns: boolean; PasteLimit: integer;
000212                   VAR status: integer);
000213
000214   PROCEDURE PointAtRg(P: Point; VAR pRg: range; VAR nbh: rect);
000215
000216   PROCEDURE reDrawSel(showselect: boolean);
000217
000218   PROCEDURE ReInitSelMgr;
000219
000220   PROCEDURE restoreRgFB(VAR timeout: integer);
000221
000222   PROCEDURE ReSelCR(VAR timeout: INTEGER; hilite: cellseltype);
000223
000224   PROCEDURE restoreflds(VAR curtimeout: integer);
000225
000226   PROCEDURE resizeflds;
000227
000228   PROCEDURE SelAllTxt(VAR timeout: INTEGER);
000229
000230   PROCEDURE SelNewCell(P: point; hilite: cellseltype; VAR nbh: Rect;
000231                   VAR timeout: integer);
000232
000233   PROCEDURE SetAnchorCell(RowId, colId: idtype);
000234
000235   PROCEDURE SetCellStr(str: str40);
000236
000237   PROCEDURE UNDOSEL(VAR status: integer);
000238
000239   FUNCTION ValidCoord(rg: range): Boolean;
000240
000241   PROCEDURE WhTblSel(rg: range; VAR nbh: Rect; VAR timeout: integer);
000242
000243   PROCEDURE unboldflds(VAR status: integer);
000244
000245   PROCEDURE UnpackSMGR(Hndsdata: Hnddata; offset: integer);
000246   { $IFC tmDebug }
000247
000248   PROCEDURE startTMtime;
000249
000250   PROCEDURE stopTMtime;
000251   { $ENDC }
000252
000253 *****
000254 *
000255 *           THAT'S ALL FOLKS ...
000256 *
000257 *****
000258
```

Apple Lisa Computer Technical Information

End of File -- Lines: 258 Characters: 6533

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 TREES.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : TREES
000004 *
000005 *****
000006
000007 USES { $U+ } TREES;
000008
000009
000010 { $IFC isIntrin } INTRINSIC; { $ENDC }
000011
000012 INTERFACE
000013
000014     USES
000015         { $IFC SrcOnOS }
000016             (* $U OBJ/PASDEFS *) PasDefs,
000017             (* $U OBJ/MEMMAN *) MemMan;
000018         { $ELSEC }
000019             (* $U OBJ/PASDEFS *) PasDefs,
000020             (* $U OBJ/MEMMAN *) MemMan;
000021         { $ENDC }
000022
000023     { $SETC DEBUGF = 0 }
000024
000025     CONST
000026
000027
000028         NullNode      = - 32768;
000029
000030     TYPE
000031         NRRowPtr      = ^NRRow;
000032         NRRow         = ARRAY [ - 128..127 ] OF NameString;
000033         NRHandle      = ^NameArr;
000034         NameArr       = ARRAY [ - 128..127 ] OF NRRowPtr;
000035         CompResult    = (ToLeft, Equal, ToRight);
000036         TreeType      = (NameT, Future1, Future2, Future3, Future4);
000037         Node          = integer;
000038         NodeRec       = RECORD
000039             lLink, rLink: Node;
000040         END;
000041         NodeRow       = ARRAY [ - 128..127 ] OF NodeRec;
000042         NodeRowPtr    = ^NodeRow;
000043         NodeArr       = ARRAY [ - 128..127 ] OF NodeRowPtr;
000044         NodeArrPtr    = ^NodeArr;
000045         TreeHandle    = ^TreeRec;
000046         TreeRec       = RECORD
000047             nodMem: MMHandle;
000048             nodes: NodeArrPtr;
000049             maxNodes, curNodes: longint;
000050             root: Node;
000051             compare: ProcPtr;
000052             store: ProcPtr;
000053             CASE TRType: TreeType OF
000054                 NameT:
000055                     (names: NRHandle);
000056                 Future1, Future2, Future3, Future4:
000057                     (FutPtr: longint);
000058             END;
000059
000060     FUNCTION EnterNode(t: TreeHandle; VAR newN: NameString;
000061         PROCEDURE duplicate
```

Apple Lisa Computer Technical Information

```
000062          (t: TreeHandle; VAR np: NameString; n: Node);
000063          PROCEDURE newNode
000064          (t: TreeHandle; VAR x: NameString; VAR newN: Node)): Node;
000065
000066          FUNCTION TLookUp(t: TreeHandle; VAR x: NameString): Node;
000067
000068          PROCEDURE InOTraverse(t: TreeHandle; PROCEDURE
000069          Visit(i: Node));
000070
000071          FUNCTION InitNameTree(n: longint): TreeHandle;
000072
000073          FUNCTION GrowTree(t: TreeHandle; n, row: integer): boolean;
000074
000075          *****
000076          *
000077          *          THAT'S ALL FOLKS ...
000078          *
000079          *****
000080
```

End of File -- Lines: 80 Characters: 2435

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNITCS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITCS
000004 *
000005 *****
000006
000007 USES {$U+} UNITCS;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/UnitStd    } UnitStd,
000016         {$U libsm/UnitHz    } UnitHz,
000017         {$U libos/SysCall   } SysCall,
000018         {$U libqd/Storage   } Storage,
000019         {$U libqd/QuickDraw } QuickDraw,
000020         {$U libfm/FontMgr   } FontMgr,
000021         {$U libpm/PmDecl    } PmDecl,
000022         {$U libpr/PrStdInfo } PrStdInfo,
000023         {$U libsu/UnitFmt   } UnitFmt;
000024
000025     CONST
000026         {$IFC NOT fDbgOk}
000027         ftstCs          = FALSE;           {$ENDC}
000028         clpdDft         = 4;
000029         icsLst          = 10;
000030         ointNil         = 0;
000031         orecNil         = MAXINT;
000032         lpMax           = 2000000000;
000033         iprocSeqLpd     = 0;
000034         iprocFindLpFixed = 1;
000035         iprocHilight   = 2;
000036         iprocCrd        = 3;
000037         iprocFreeIcs   = 4;
000038         iprocFSellLpBounds = 5;
000039         iprocPxHcs     = 6;
000040         iprocLstCs     = 6;
000041         iimgpScrn      = 0;
000042         iimgpQume      = 1;
000043         iimgpCIto      = 2;
000044         iimgpLst       = 2;
000045
000046     TYPE
000047         TLp           = TL;
000048         TArglp        = ARRAY [0..0] OF TLp;
000049         TRglp         = ^TArglp;
000050         TInt          = RECORD
000051             ointNxt: TC;
000052             ointPrv: TC;
000053             ointParent: TC;
000054             lpFst: TLp;
000055             lpLim: TLp;
000056             ics: TB;
000057             fValid: TF;
000058         END;
000059         TPint         = ^TInt;
000060         TArgint       = ARRAY [0..0] OF TInt;
000061         TRgint        = ^TArgint;
```


Apple Lisa Computer Technical Information

```

000062      THpic      = PicHandle;
000063      TPpic      = PicPtr;
000064      TFigd      = RECORD
000065                n: TN;
000066                ppic: TPpic;
000067      END;
000068      TTyset      = RECORD
000069                fRce: TF;
000070                fParBnds: TF;
000071                fRpe: TF
000072      END;
000073      TALpd      = RECORD
000074                ics: TB;
000075                ilpd: TB;
000076                fParSt: TF;
000077                lp: TLp;
000078                lpLim: TLp;
000079                lpSon: TLp;
000080                icsSon: TB;
000081                tyset: TTyset;
000082                rce: TRce;
000083                lpFstPar: TLp;
000084                lpLimPar: TLp;
000085                rpe: TRpe;
000086                arce: TARce;
000087                arpe: TARpe;
000088                figd: TFigd;
000089      END;
000090      TLpd      = ^TALpd;
000091      TTycs      = (tycsNil, tycsUs, tycsEd, tycsFf, tycsUf, tycsFm, tycsPg,
000092                tycsRuler, tycsLst, tycsFld, tycsMtrx, tycsFig);
000093      TCspd      = RECORD
000094                argproc: ARRAY [0..iproclstCs] OF TProc;
000095                argimgp: ARRAY [0..iimgplst] OF TProc;
000096      END;
000097      TTyxy      = (tyxyScrnl, tyxyIP, tyxyPgFract, tyxyMica, tyxyDotMx,
000098                tyxyHiDotMx, tyxyLDotMx, tyxyLHiDotMx, tyxyNew1,
000099                tyxyNew2, tyxyNew3, tyxyNew4, tyxyNew5, tyxyNew6,
000100                tyxyNew7, tyxyNew8, tyxyNil);
000101      TCs      = RECORD
000102                cspd: TCspd;
000103                hz: THz;
000104                tycs: TTycs;
000105                tyxy: TTyxy;
000106                ointFst: TC;
000107                orecConFst: TC;
000108                cRef: TC;
000109                prprf: TPrRec;
000110      END;
000111      TPcs      = ^TCs;
000112      THcs      = ^TPcs;
000113      TArchcs    = ARRAY [0..0] OF THcs;
000114      TRghcs    = ^TArchcs;
000115      TPglp      = RECORD
000116                icsHdr: TB;
000117                icsFtr: TB;
000118                yTopHdr: TY;
000119                dyHdr: TY;
000120                dyHtBody: TY;
000121                dyBotBody: TY;
000122                tyxy: TTyxy;
000123      END;
000124      TAmpicsprcs = ARRAY [0..0] OF TL;
000125      TMpicsprcs = ^TAmpicsprcs;
000126
000127      VAR

```

Apple Lisa Computer Technical Information

```
000128      {$IFC fDbgOk}
000129      fTstCs:      TF;                      {$ENDC}
000130      tysetRpe:     TTyset;
000131      tysetBnds:    TTyset;
000132      rghcs:        TRghcs;
000133      mpicsprcs:    TMPicsprcs;
000134      icsMac:        TC;
000135      argalpd:      ARRAY [0..clpdDft] OF TALpd;
000136      arglpd:       ARRAY [0..clpdDft] OF TLpd;
000137      lpdStd:       TLpd;
000138      tysetStd:     TTyset;
000139      lldInt:       TLld;
000140      tyxyLst:      TTyxy;
000141      tyxyCs:       TTyxy;
000142      pglpStd:      TPglp;
000143      fNoInvalidate: TF;
000144
000145      FUNCTION LpMinP(lp1: TLp; lp2: TLp): TLp;
000146
000147      FUNCTION LpMaxP(lp1: TLp; lp2: TLp): TLp;
000148
000149      PROCEDURE PxRgcs;
000150
000151      PROCEDURE PxAchad(VAR achad: TAchad);
000152
000153      PROCEDURE PxCs(ics: TB);
000154
000155      PROCEDURE FreeInt(ics: TB; oint: TC);
000156
000157      FUNCTION OintMark(oint: TC; ics: TB; lpFst: TLp; lpLim: TLp; fValid: TF;
000158                      ointParent: TC): TC;
000159
000160      FUNCTION IcsCreate(tyCs: TTyCs; cbCs: TC; hz: THz): TB;
000161
000162      PROCEDURE FreeIcs(ics: TB);
000163
000164      PROCEDURE PXLpd(lpD: TLpd; VAR achad: TAchad);
000165
000166      PROCEDURE SetLpd(lpD: TLpd; ics: TB; lp: TLp; tyset: TTyset;
000167                  VAR achad: TAchad);
000168
000169      PROCEDURE SeqLpd(lpD: TLpd; VAR achad: TAchad);
000170
000171      FUNCTION FParStart(ics: TB; lp: TLp): TF;
000172
000173      PROCEDURE FindLpFixed(ics: TB; lp: TLp; VAR lpFixed: TLp);
000174
000175      PROCEDURE FindLpFstPar(ics: TB; lp: TLp; VAR lpFixed: TLp);
000176
000177      PROCEDURE InitCs(hz: THz);
000178
000179      PROCEDURE PxInt(oInt: TC);
000180
000181      PROCEDURE AdjustLps(ics: TB; lpFstInv: TLp; lpLimInv: TLp; dlpAdjust: TLp);
000182
000183      FUNCTION FSelLpBounds(lp: TLp; ics: TB; VAR lpFstSel, lpLimSel, lpFstAtom,
000184                          lpLimAtom: TLp): TF;
000185
000186      FUNCTION TrueStdSelLpBounds(lp: TLp; ics: TB; VAR lpFstSel, lpLimSel,
000187                              lpFstAtom, lpLimAtom: TLp): TF;
000188
000189      FUNCTION FTextIcs(ics: TB): TF;
000190
000191      PROCEDURE PurgeIcsPrCs;
000192
000193      *****
```

Apple Lisa Computer Technical Information

```
000194 *  
000195 *          THAT'S ALL FOLKS ...  
000196 *  
000197 *****  
000198
```

End of File -- Lines: 198 Characters: 5836

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNITFF.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITFF
000004 *
000005 *****
000006
000007 USES {$U+} UNITFF;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/UnitStd    } UnitStd,
000016         {$U libsm/UnitHz    } UnitHz,
000017         {$U libos/SysCall   } SysCall,
000018         {$U libsu/UnitFile  } UnitFile,
000019         {$U libqd/QuickDraw } QuickDraw,
000020         {$U libfm/FontMgr   } FontMgr,
000021         {$U libqd/Storage   } Storage,
000022         {$U libsu/UnitFmt   } UnitFmt,
000023         {$U libpm/PmDecl    } PmDecl,
000024         {$U libpr/PrStdInfo } PrStdInfo,
000025         {$U libsu/UnitCS    } UnitCS;
000026
000027     CONST
000028         {$IFC NOT fDbgOk}
000029         ftstFf          = FALSE;           {$ENDC}
000030         pwFf           = - 13142;
000031         pwNotFf        = 0;
000032         cbPgff         = 512;
000033         dxMinTab       = 6;
000034         ffVersion      = 5;
000035         ffVerPrPrf     = 2;
000036         ffVerNewRgfbcS = 3;
000037         ffVerMsgFst    = 3;
000038         ffVerCommaDecimal = 4;
000039         ffVerPrRec     = 5;
000040
000041     TYPE
000042         TTyRlU          = (tyrluEnglish, tyrluMetric, tyrluPica, tyrluElite);
000043         TFfLeader      = RECORD
000044             password: TW;
000045             version: TC;
000046             lpHdrFst: TLp;
000047             lpFtrFst: TLp;
000048             lpTxtFst: TLp;
000049             dummyLpLimFf: TB;
000050             tyrlu: TTyRlU;
000051             pglp: TPglp;
000052             prprf: TPrRec;
000053         END;
000054
000055     VAR
000056         cspdFf:        TCspd;
000057         {$IFC fDbgOk}
000058         ftstFf:        TF;
000059         {$ENDC}
000060
000061     PROCEDURE PxHcsff(hcs: THcs);
```

Apple Lisa Computer Technical Information

```
000062
000063   PROCEDURE PxLeader(VAR leader: Tffleader);
000064
000065   PROCEDURE SeqLpdFf(lpd: TLpd; VAR achad: TACHad);
000066
000067   FUNCTION IcsFfCreate(ifil: TC; icsFig: TB; hz: THz;
000068                       VAR fffleader: Tffleader): TB;
000069
000070   PROCEDURE FreeFfIcs(icsFf: TB);
000071
000072   PROCEDURE WriteIcsFf(icsDoc: TB; pglp: TPglp; tyrluFf: TTyRlu; ifil: TB;
000073                       ifilFig: TB; hzDoc: THz);
000074
000075   PROCEDURE InitFF;
000076
000077   FUNCTION LpMacFf(icsFf: TB): TLp;
000078
000079   PROCEDURE SetLeaderOfIcsFfFromIFil(ics: TB; ifil: TC; hz: THz);
000080
000081   FUNCTION IfilOfFf(icsFf: TB): TB;
000082
000083   PROCEDURE GetLeaderOfIcsFf(icsFf: TB; VAR fffleader: TffLeader);
000084
000085 *****
000086 *
000087 *           THAT'S ALL FOLKS ...
000088 *
000089 *****
000090
```

End of File -- Lines: 90 Characters: 2363

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNITFIGA.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITFIGA
000004 *
000005 *****
000006
000007 USES {$U+} UNITFIGA;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/UnitStd    } UnitStd,
000016         {$U libsm/UnitHz    } UnitHz,
000017         {$U libos/SysCall   } SysCall,
000018         {$U libsu/UnitFile  } UnitFile,
000019         {$U libqd/QuickDraw } QuickDraw,
000020         {$U libfm/FontMgr   } FontMgr,
000021         {$U libqd/Storage   } Storage,
000022         {$U libsu/UnitFmt   } UnitFmt,
000023         {$U libpm/PmDecl    } PmDecl,
000024         {$U libpr/PrStdInfo } PrStdInfo,
000025         {$U libsu/UnitCS    } UnitCS;
000026
000027     CONST
000028         {$IFC NOT fDbgOk}
000029         ftstFig      = FALSE;           {$ENDC}
000030         tyfilFig     = - 13143;
000031         verFigCur   = 1;
000032         cErrFilNotValid = 5100;
000033
000034     TYPE
000035         TCsfig      = RECORD
000036             cs: TCs;
000037             ifil: TB;
000038             END;
000039         TPcsfig     = ^TCsfig;
000040         THcsfig     = ^TPcsfig;
000041
000042     VAR
000043         cspdFig:    TCspd;
000044         {$IFC fDbgOk}
000045         ftstFig:    TF;
000046         {$ENDC}
000047
000048     PROCEDURE PxHcsfig(hcsfig: THcsfig);
000049
000050     FUNCTION  IcsFigCreate(VAR cError: TC; path: Pathname; setaccess: TSetaccess;
000051                          hz: THz; password: E_name): TB;
000052
000053     PROCEDURE FreeFigIcs(ics: TB);
000054
000055     PROCEDURE FigFindLpFixed(ics: TB; lp: TLp; VAR lpFixed: TLp);
000056
000057     FUNCTION  LpFigAdd(ics: TB; hpic: THpic): TLp;
000058
000059     PROCEDURE SeqLpdFig(lpd: TLpd; VAR achad: TACHad);
000060
000061     FUNCTION  IfilOffFig(icsFig: TB): TB;
```

Apple Lisa Computer Technical Information

```
000062
000063     PROCEDURE InitFig;
000064
000065     *****
000066     *
000067     *               THAT'S ALL FOLKS ...
000068     *
000069     *****
000070
```

End of File -- Lines: 70 Characters: 1729

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNITFILE.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITFILE
000004 *
000005 *****
000006
000007 USES {$U+} UNITFILE;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/UnitStd} UnitStd,
000016         {$U libsm/UnitHz} UnitHz,
000017         {$U libos/SysCall} SysCall,
000018         {$U libos/PSystemCall} PSystemCall;
000019
000020     CONST
000021         ifilNil          = - 1;
000022         ifilRangeWarn   = - 4056;
000023         noMFileErr      = 4057;
000024         ifilNotOpenWarn = - 4058;
000025         ivodNil          = MAXINT;
000026         ivodArgvod       = 0;
000027         ivodFst          = 1;
000028         cbVofilOnFile   = 128;
000029         cvodMore         = 48;
000030         cbBlkOfVofil    = 512;
000031         {$IFC NOT fDbgOk}
000032         fTstFile        = FALSE;
000033         {$ENDC}
000034
000035     TYPE
000036         String255       = STRING[255];
000037         TVod             = RECORD
000038             ib: TL;
000039             cb: TL;
000040             END;
000041         TVofil           = RECORD
000042             tyfil: TW;
000043             ver: TC;
000044             ivodMac: TC;
000045             vodArgvod: TVod;
000046             END;
000047         TArgvod          = ARRAY [1..1] OF TVod;
000048         TRgvod           = ^TArgvod;
000049         TSetaccess       = MSet;
000050
000051     VAR
000052         Scrap_RefNum: TL;
000053         {$IFC fDbgOk}
000054         fTstFile:      TF;
000055         {$ENDC}
000056
000057     PROCEDURE InitFiles(hz: THz);
000058
000059     FUNCTION IfilOpen(VAR cError: TC; pathIn: String255; cbBlk: TC;
000060         setaccess: TSetaccess; hzWrite: THz;
000061         password: E_name): TB;
```


Apple Lisa Computer Technical Information

```
000062
000063 PROCEDURE CloseIfil(VAR cError: TC; ifil: TB);
000064
000065 FUNCTION FilCbOfN(n: TN): TC;
000066
000067 FUNCTION FilFSwapInN(hz: THz; n: TN): TF;
000068
000069 PROCEDURE FilSwapOutN(hz: THz; n: TN);
000070
000071 FUNCTION NFromIfilIb(ifil: TB; ib: TL): TN;
000072
000073 FUNCTION CBlkOfIfil(ifil: TC): TC;
000074
000075 PROCEDURE RenameFile(VAR cError: TC; pathIn: String255; enameIn: TSp;
000076 password: E_name);
000077
000078 PROCEDURE KillFile(VAR cError: TC; pathIn: String255; password: E_name);
000079
000080 FUNCTION FMonitorFile(path: String255): TF;
000081
000082 PROCEDURE PathOfIfil(ifil: TC; VAR path: String255; VAR password: e_name);
000083
000084 PROCEDURE SetIbLimOfIfil(ifil: TB; ibLim: TL);
000085
000086 FUNCTION RefnumOfIfil(ifil: TB): TC;
000087
000088 PROCEDURE CleanIfil(VAR cError: TC; hz: THz; ifil: TB; fIgnoreDirty: TF);
000089
000090 FUNCTION CErrOfIfil(ifil: TB): TC;
000091
000092 PROCEDURE GetVofil(ifil: TB; VAR vofil: TVofil);
000093
000094 FUNCTION IvodAdd(ifil: TB; hSrc: TH; cb: TL): TC;
000095
000096 PROCEDURE SetTyfilVer(ifil: TB; tyfil: TW; ver: TC);
000097
000098 *****
000099 *
000100 *           THAT'S ALL FOLKS ...
000101 *
000102 *****
000103
```

End of File -- Lines: 103 Characters: 2579

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNITFMT.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITFMT
000004 *
000005 *****
000006
000007 USES { $U+ } UNITFMT;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015     { $U libsm/UnitStd } UnitStd,
000016     { $U libsm/UnitHz } UnitHz,
000017     { $U libqd/QuickDraw } QuickDraw,
000018     { $U libfm/FontMgr } FontMgr;
000019     { $SETC doTraceSU := TRUE }
000020     { $SetC fTraceSU := doTraceSU AND fTRACE }
000021
000022     CONST
000023     { $IFC NOT fDbgOk }
000024     ftstFmt      = FALSE;           { $ENDC }
000025     ifldQuad     = 0;
000026     ifldXLfst    = 1;
000027     ifldXLBody   = 2;
000028     ifldXRt      = 3;
000029     ifldXtb      = 4;
000030     ifldTbQuad   = 5;
000031     ifldRgtbd    = 6;
000032     ifldDxtb     = 7;
000033     ifldBold     = 8;
000034     ifldItalic   = 9;
000035     ifldUnderline = 10;
000036     ifldFont     = 11;
000037     ifldVan      = 12;
000038     ifldSuperscript = 13;
000039     ifldYLine    = 14;
000040     ifldYLd      = 15;
000041     ifldMovTb    = 16;
000042     ifldClrTb    = 17;
000043     ifldFilTb    = 18;
000044     ifldKeep     = 19;
000045     ifldLpFig    = 20;
000046     ifldIcsFig   = 21;
000047     ifldTyfam    = 22;
000048     ifldFsp      = 23;
000049     ifldLst      = 23;
000050     fopcEnd      = - 1;
000051     ibRgtbd      = 14;
000052     cbRceVan     = 2;
000053     ifnt12Tile   = 0;
000054     ifntp10Tile  = 1;
000055     ifntp12Tile  = 2;
000056     ifnt18Tile   = 3;
000057     ifnt24Tile   = 4;
000058     ifnt12Cent   = 5;
000059     ifnt18Cent   = 6;
000060     ifnt24Cent   = 7;
000061     ifntFBold    = 8;
```

Apple Lisa Computer Technical Information

```
000062      ifntp10Cent   = 9;
000063      ifntp12Cent   = 10;
000064      ifntelite     = 11;
000065      ifntcourier   = 12;
000066      ifntp20Tile   = 13;
000067      ifntp15Tile   = 14;
000068      ifntp20Cent   = 15;
000069      ifntp15Cent   = 16;
000070      ifntLst       = 16;
000071      ibXTbd        = 0;
000072      ibChTbd       = 2;
000073      ibRgfbcb      = 1;
000074      cbHfccInit    = 10;
000075      itbdLst       = 58;
000076      icsNil        = 0;
000077      lpNil         = - 2;
000078      tyfamModern    = 0;
000079      tyfamClassic  = 1;
000080      tyfamLst      = 1;
000081      tyfam1Fill    = 2;
000082      tyfam2Fill    = 3;
000083      fsp8P20       = 0;
000084      fsp8P15       = 1;
000085      fsp12P12      = 2;
000086      fsp12P10     = 3;
000087      fsp12         = 4;
000088      fsp14         = 5;
000089      fsp18         = 6;
000090      fsp24         = 7;
000091      fspLst       = 7;
000092
000093      TYPE
000094      TFcc          = RECORD
000095                  cref: TB;
000096                  argfbc: TArgb;
000097                  END;
000098      TPfcc         = ^TFcc;
000099      THfcc         = ^TPfcc;
000100      TQuad        = (quadL, quadC, quadR, quadJ);
000101      TTyfill      = (tyfillNil, tyfillDots, tyfillHyph, tyfillUL);
000102      TTbdOld      = PACKED RECORD
000103                  x: TX;
000104                  fill14: 0..15;
000105                  quad: TQuad;
000106                  tyfill: TTyfill;
000107                  chLdr: TCh;
000108                  END;
000109      TTbd         = PACKED RECORD
000110                  x: TX;
000111                  fill13: 0..7;
000112                  fDecimalComma: TF;
000113                  quad: TQuad;
000114                  tyfill: TTyfill;
000115                  chLdr: TCh;
000116                  END;
000117      TArgtbd      = ARRAY [0..0] OF TTbd;
000118      TRgtbd       = ^TArgtbd;
000119      TArcpe       = PACKED RECORD
000120                  cb: TB;
000121                  b1: TB;
000122                  END;
000123      TRcpe        = ^TArcpe;
000124      TArce       = PACKED RECORD
000125                  cb: 0..255;
000126                  fVan: TF;
000127                  fBold: TF;
```

Apple Lisa Computer Technical Information

```
000128             fItalic: TF;
000129             fUnderline: TF;
000130             fill4: 0..15;
000131             cbSuperscript: TB;
000132             ifnt: TB;
000133             fKeep: TF;
000134             fOutline: TF;
000135             fShadow: TF;
000136             fFillB: TF;
000137             fFillC: TF;
000138             fFillD: TF;
000139             fFillE: TF;
000140             fFillF: TF;
000141             icsFig: TB;
000142             lpFig: TC;
000143             END;
000144     TRce       = ^TArce;
000145     TArpe     = PACKED RECORD
000146             cb: 0..255;
000147             sy: TB;
000148             xLftFst: TX;
000149             xLftBody: TX;
000150             xRt: TX;
000151             yLd: TY;
000152             fill1: TB;
000153             yLine: 0..63;
000154             quad: TQuad;
000155             itbLim: TW;
000156             argtbd: ARRAY [0..itbdLst] OF TTbd;
000157             END;
000158     TRpe       = ^TArpe;
000159     TTyfam     = 0..3;
000160     TFsp       = TB;
000161     TFstyle    = RECORD
000162             tyfam: TTyfam;
000163             fsp: TFsp;
000164             END;
000165     TTydst     = (tydstRce, tydstRpe, tydstTbd, tydstNil);
000166
000167     VAR
000168     { $IFC fdbgOk }
000169     ftstFmt:   TF;                               { $ENDC }
000170     argfam:    ARRAY [0..ifntLst] OF TFam;
000171     tbdNil:    TTbd;
000172     arceStd:   TArce;
000173     arpeStd:   TArpe;
000174     ampifntfstyle: ARRAY [0..ifntLst] OF TFstyle;
000175
000176     PROCEDURE PxArce(VAR arce: TArce);
000177
000178     PROCEDURE PxArpe(VAR arpe: TArpe);
000179
000180     FUNCTION CfbcApplyIfld(ifld: TC; wsva: TW; pval: TPC; rpe: TRpe;
000181             rce: TRce): TC;
000182
000183     FUNCTION IfbcLstApplyRgfbc(rgfbc: TRgb; rpe: TRpe; rce: TRce): TC;
000184
000185     PROCEDURE AddRgfbc(VAR hfcc: THfcc; hfccSrc: THfcc; hz: THz);
000186
000187     PROCEDURE AddFop(VAR hfcc: THfcc; ifld: TB; sval: TW; pval: TPC; hz: THz);
000188
000189     PROCEDURE GenFopsDiff(hfcc: THfcc; tydst: TTydst; rcpeStd, rcpe: TP;
000190             hz: THz);
000191
000192     PROCEDURE zzGenParFops(hfcc: THfcc; rpe: TRpe; hz: THz);
000193
```

Apple Lisa Computer Technical Information

```
000194     FUNCTION ItbFromX(rgtbd: TRgtbd; itbLst: TC; x: TX): TC;
000195
000196     PROCEDURE InitFmt;
000197
000198     PROCEDURE PxRgfbc(rgfbc: TRgb);
000199
000200 *****
000201 *
000202 *                THAT'S ALL FOLKS ...
000203 *
000204 *****
000205
```

End of File -- Lines: 205 Characters: 5721

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNITHZ.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITHZ
000004 *
000005 *****
000006
000007 USES {$U+} UNITHZ;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013   {$SETC fhz := FALSE}
000014   {$SETC fhzP := FALSE}
000015
000016   USES
000017     {$U libsm/UnitStd} UnitStd,
000018     {$U libos/SysCall} SysCall;
000019
000020   CONST
000021     tybkFree      = 0;
000022     tybkStd       = 1;
000023     tybkN         = 2;
000024     tybkNrel     = 3;
000025     {$IFC NOT fhz}
000026     ftstHz       = FALSE;           {$ENDC}
000027     {$IFC NOT fhz}
000028     ftstHzV     = FALSE;           {$ENDC}
000029     {$IFC NOT fhzP}
000030     ftstHzP     = FALSE;           {$ENDC}
000031     ipPoolMax    = 32000;
000032     cbMinAsk     = 512;
000033     MaxBlkSize   = 32767;
000034
000035   TYPE
000036     TH           = ^TP;
000037     TArgh       = ARRAY [0..0] OF TH;
000038     TRgh        = ^TArgh;
000039     TTybk       = 0..3;
000040     THz         = ^TAhz;
000041     TBp         = PACKED RECORD
000042                 fRelPBase: TF;
000043                 CASE TF OF
000044                   FALSE:
000045                     (ip: 0..32000);
000046                   TRUE:
000047                     (iwP: 0..32000);
000048                 END;
000049     TPpn        = ^TPn;
000050     TStn        = PACKED RECORD
000051                 tybk: TTybk;
000052                 fDirty: TF;
000053                 fLock: TF;
000054                 fill: 0..15;
000055                 ubt: TB;
000056                 END;
000057     TNob        = RECORD
000058                 n: TN;
000059                 stn: TStn;
000060                 bp: TBp;
000061                 data: TW;
```

Apple Lisa Computer Technical Information

```

000062             END;
000063     TPnob       = ^TNob;
000064     TPpnob      = ^TPnob;
000065     TArgpnob    = ARRAY [0..0] OF TPnob;
000066     TRgpnob     = ^TArgpnob;
000067     THrgpnob    = ^TRgpnob;
000068     TBk         = ^TAbk;
000069     TAbk        = RECORD
000070             CASE TF OF
000071                 FALSE:
000072                     (hdr: PACKED RECORD
000073                      tybk: TTybk;
000074                      cw: 0..16000;
000075                      END;
000076                     CASE TTybk OF
000077                         tybkNrel:
000078                             (dataNrel: TW);
000079                         tybkStd:
000080                             (bp: TBp;
000081                              dataStd: TW);
000082                         tybkN:
000083                             (SpaceForCompactRoutine: Integer;
000084                              nob: TNob); );
000085                 TRUE:
000086                     (cwFree: TL;
000087                      bkfNxt: TBk;
000088                      bkfPrv: TBk; );
000089             END;
000090     TArgwBase    = ARRAY [0..0] OF TW;
000091     TRgwBase     = ^TArgwBase;
000092     TAhz         = RECORD
000093             bkFst: TBk;
000094             bkLst: TBk;
000095             bkFFst: TBk;
000096             rgwBase: TRgwBase;
000097             ipPoolMac: TC;
000098             hFstFree: TH;
000099             cbFree: TL;
000100             hrgpnob: THrgpnob;
000101             mskIpnLst: TW;
000102             ipnCur: TC;
000103             cpnAvail: TC;
000104             ubtCur: TB;
000105             procCbMore: TProc;
000106             procCbOfN: TProc;
000107             procFSwapInN: TProc;
000108             procSwapOutN: TProc;
000109             fScramble: TF;
000110             hScramble: TH;
000111             fUpScramble: TF;
000112             fCheck: TF;
000113             cCompact: TC;
000114             HasConcrete: TF;
000115             argpPool: ARRAY [0..0] OF TP;
000116             END;
000117
000118     VAR
000119         hNil:      TH;
000120         pNil:      TP;
000121         hzNil:     THz;
000122         {$IFC fhz}
000123         fTstHzV:   TF;                {$ENDC}
000124         {$IFC fhz}
000125         fTstHz:    TF;                {$ENDC}
000126         {$IFC fhzP}
000127         fTstHzP:   TF;                {$ENDC}

```

Apple Lisa Computer Technical Information

```
000128
000129 FUNCTION HALlocate(hz: THz; cb: TC): TH;
000130
000131 FUNCTION HzInit(pfSt: TP; pLim: TP; pBase: TP; ipPoolMac: TC; logIpnLim: TC;
000132     procCbMore: TProc; procCbOfN: TProc; procFSwapInN: TProc;
000133     procSwapOutN: TProc): THz;
000134
000135 PROCEDURE PxHz(hz: THz);
000136
000137 PROCEDURE ChangeSizeH(hz: THz; h: TH; cbNew: TC);
000138
000139 PROCEDURE ChangeNSize(hz: THz; n: TN; cbNew: TC);
000140
000141 PROCEDURE FreeBk(hz: THz; h: TH; tybk: TTybk);
000142
000143 PROCEDURE AllocBk(hz: THz; hDst: TH; cb: TC; tybk: TTybk);
000144
000145 FUNCTION PMapN(hz: THz; nSrc: TN): TP;
000146
000147 FUNCTION PCreateNob(hz: THz; nSrc: TN; cbData: TC): TP;
000148
000149 PROCEDURE SetFDirty(hz: THz; n: TN; fDirty: TF);
000150
000151 PROCEDURE SetCbFree(hz: THz; cbFree: TL; fEnlargeHz: TF);
000152
000153 FUNCTION CbDataOfH(hz: THz; h: TH): TC;
000154
000155 FUNCTION PAllocate(hz: THz; cb: TC): TP;
000156
000157 PROCEDURE FreeH(hz: THz; h: TH);
000158
000159 PROCEDURE FreeP(hz: THz; p: TP);
000160
000161 PROCEDURE FreeN(hz: THz; n: TN);
000162
000163 PROCEDURE ReleaseBkNrel(hz: THz; pfStRelease: TP);
000164
000165 FUNCTION PLstFree(hz: THz): TP;
000166
000167 FUNCTION HzFromH(h: TH): THz;
000168
000169 FUNCTION FCheckHzOk(hz: THz; VAR cbKStd: TC): TF;
000170
000171 FUNCTION HLockN(hz: THz; n: TN; fNeedH: TF): TH;
000172
000173 PROCEDURE UnlockN(hz: THz; n: TN);
000174
000175 PROCEDURE LockHandleInConcrete(h: TH);
000176
000177 PROCEDURE UnlockHandleFromConcrete(h: TH);
000178
000179 PROCEDURE EnlargeHz(hz: THz; cbMore: TL);
000180
000181 FUNCTION CbShrinkHz(hz: THz; cbLess: TL): TL;
000182
000183 FUNCTION CbOfHz(hz: THz): TL;
000184 {$IFC fOS}
000185
000186 FUNCTION CbMoreGrowSeg(hz: THz; cbNeed: TC): TC;
000187 {$ENDC}
000188
000189 FUNCTION PnobFromN(hz: THz; n: TN): TPnob;
000190
000191 PROCEDURE Scramble(hz: THz);
000192
000193 *****
```


Apple Lisa Computer Technical Information

```
000194 *  
000195 *           THAT'S ALL FOLKS ...  
000196 *  
000197 *****  
000198
```

End of File -- Lines: 198 Characters: 5673

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNITS.TEXT"
=====
```

```
000001 *****
000002 *
000003 *           APPLE LISA DESKTOP 3.0 LIBRARIES UNIT TABLE
000004 *
000005 *****
000006
000007 UnitName Unit# File# Type   DataSize
000008 -----
000009 PASLIB      1    3  Intrin  000000
000010 UNITSTD    2    4  Intrin  000002
000011 GRAFUTIL   3    4  Intrin  000000
000012 UNITHZ     4    4  Shared  000024
000013 STORAGE   5    4  Intrin  000008
000014 QUICKDRA  6    4  Intrin  0000CE
000015 HWINT      7    4  Intrin  000000
000016 FEDEC      8    4  Intrin  000000
000017 FONTMGR   9    4  Shared  000054
000018 EVENTS    10   4  Shared  000210
000019 WINDOWS   11   4  Shared  0010F2
000020 FOLDERS   12   4  Shared  00000A
000021 MENUS     13   4  Shared  00009C
000022 FLDUT     14   4  Intrin  000000
000023 WMLSTD    15   4  Intrin  000018
000024 WMLCRS    16   4  Intrin  0003FE
000025 WMLGROW   17   4  Intrin  00000E
000026 WMLSB     18   4  Intrin  000038
000027 INTERNAT  19   4  Intrin  000000
000028 FPLIB     20   5  Intrin  000006
000029 ALERTMGR  21   4  Intrin  000300
000030 MATHLIB   22   5  Intrin  000016
000031 UNITFF    23   4  Shared  00004A
000032 UNITCS    24   4  Shared  000644
000033 UNITFILE  25   4  Shared  000156
000034 UNITFMT   26   4  Shared  0001F2
000035 BGENV     27   8  Intrin  000058
000036 MATMGR    28   8  Intrin  00188E
000037 PARSER    29   8  Intrin  0006FC
000038 COMPUTE   30   8  Intrin  000084
000039 STUBS     31   8  Intrin  000000
000040 UUNIVERS  32   4  Intrin  000222
000041 FEGLOBAL  33   8  Intrin  000398
000042 COMGR     34   8  Intrin  00033C
000043 USTDTERM  35   9  Intrin  0002CE
000044 UQPGRAPH  36   9  Intrin  000056
000045 TEEEXEC   37   8  Intrin  000004
000046 UQPPRINT  38   9  Intrin  000188
000047 LCFEXEC   39   8  Intrin  0001A6
000048 TEENV     40   6  Intrin  00080E
000049 TM        41   6  Intrin  000BCA
000050 FMGRUTIL  42   4  Intrin  000000
000051 PMM       43   4  Shared  000042
000052 PMDECL    44   4  Intrin  000000
000053 UNITFIGA  45   4  Shared  00002A
000054 SYS1LOCK  46   4  Intrin  000000
000055 UCLASCAL  47   3  Intrin  00004E
000056 UTKUNIVE  49  11  Intrin  0001CA
000057 UOBJECT   50  10  Intrin  000164
000058 UTEXT     51  11  Intrin  00027C
000059 UDRAW     52  10  Intrin  0001B2
000060 UABC      53  10  Intrin  0009F2
000061 UDIALOG   54  11  Intrin  000564
```

Apple Lisa Computer Technical Information

000062	LCUT	55	6	Intrin	00002C
000063	IOPRIMIT	56	12	Intrin	00057E
000064	SHELLCOM	57	12	Shared	00070C
000065	PROGCOMM	58	12	Intrin	000006
000066	RECOVERY	59	6	Intrin	000000
000067	LOWLEVEL	60	6	Intrin	000126
000068	DBDECL1	61	6	Intrin	000000
000069	POOLER	62	6	Intrin	000052
000070	DBENV	63	4	Intrin	000000
000071	HEAP	64	6	Intrin	000000
000072	VLTREE	65	6	Intrin	000000
000073	CZCOMPAC	66	6	Intrin	00001C
000074	LABSCAN	68	6	Intrin	000000
000075	SCHEMA	69	6	Intrin	000000
000076	SCAN	70	6	Intrin	000000
000077	FIELDEDI	71	4	Intrin	0000CA
000078	SCRAP	73	4	Shared	000250
000079	FILERCOM	75	4	Shared	0002E8
000080	PRPUBLIC	76	7	Intrin	000000
000081	PRSTDINF	77	4	Shared	0005D0
000082	PRSTDPRO	78	7	Intrin	000002
000083	PRFILEPR	79	7	Intrin	000000
000084	PRBUF	80	7	Intrin	000010
000085	PRSPool	81	7	Intrin	000346
000086	QUEUES	82	7	Intrin	000000
000087	PREVENTS	83	7	Intrin	000000
000088	PRDLGMR	84	7	Intrin	0000BA
000089	PRMGR	85	7	Intrin	000026
000090	UVT100	86	9	Intrin	00001C
000091	USOROC	87	9	Intrin	00001C
000092	STDUNIT	88	12	Intrin	0005B0
000093	IUMAN	90	1	Intrin	0000BC
000094	OBJIO	91	1	Intrin	00011C
000095	FILEIO	92	1	Intrin	000000
000096	GRAPHS	94	1	Intrin	000008
000097	TREES	95	1	Intrin	000000
000098	REFS	96	1	Intrin	000004
000099	PARTS	97	1	Intrin	000004
000100	LISTS	98	1	Intrin	000000
000101	MEMMAN	99	1	Intrin	000000
000102	PASDEFS	100	1	Intrin	00025E
000103	MPASLIB	101	2	Intrin	000000
000104	BLKIOINT	102	3	Intrin	0000DE
000105	BLOCKIO	103	3	Shared	0005C4
000106	PASHEAP	104	3	Intrin	000022
000107					
000108	*****				
000109					
000110					
000111					
000112	*****				
000113					

End of File -- Lines: 113 Characters: 3628

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UNITSTD.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITSTD
000004 *
000005 *****
000006
000007 USES {$U+} UNITSTD;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013     {$Setc fDbgOK      := FALSE}
000014     {$Setc fOS        := TRUE}
000015     {$Setc fSymOK     := FALSE}
000016     {$Setc fStd1      := FALSE}
000017     {$Setc fTrace     := FALSE}
000018     {$SETC doTraceSM := TRUE}
000019     {$SETC fTraceSM  := doTraceSM AND fTrace}
000020     {$IFC fTrace}
000021
000022     USES
000023         {$U libos/syscall.obj} Syscall;
000024         {$ENDC}
000025         (*$R-*)
000026
000027     CONST
000028         chLst      = 127;
000029         chSp       = 32;
000030         chCr       = 13;
000031         chTab      = 9;
000032         olleNil    = 0;
000033         procNil    = 1;
000034         {$IFC NOT fDbgOk}
000035         ftstStd    = FALSE;           {$ENDC}
000036         {$IFC NOT fStd1}
000037         ftstStd1   = FALSE;           {$ENDC}
000038
000039     TYPE
000040         TC          = INTEGER;
000041         TB          = - 128..127;
000042         TA          = 0..127;
000043         TP          = ^TB;
000044         TPc         = ^TC;
000045         TPa         = LONGINT;
000046         TF          = BOOLEAN;
000047         TX          = TC;
000048         TY          = TC;
000049         TL          = LONGINT;
000050         TW          = INTEGER;
000051         TCh         = TB;
000052         TCd         = TC;
000053         TN          = TL;
000054         TPn         = ^TN;
000055         TFd         = RECORD
000056             ifil: TB;
000057             bhi: TB;
000058             iblo: TW
000059             END;
000060         TPFd        = ^TFd;
000061         TNOOrFd     = PACKED RECORD
```

Apple Lisa Computer Technical Information

```
000062             CASE TC OF
000063                 1:
000064                     (fd: Tfd);
000065                 2:
000066                     (n: TN);
000067             END;
000068     TArgch         = ARRAY [0..0] OF TCh;
000069     TRgch          = ^TArgch;
000070     TAchad         = RECORD
000071         rgch: TRgch;
000072         ichFst: TC;
000073         ichLim: TC;
000074     END;
000075     TChad          = ^TAchad;
000076     TArgb         = ARRAY [0..0] OF TB;
000077     TRgb          = ^TArgb;
000078     THrgb         = ^TRgb;
000079     TAbad         = RECORD
000080         rgb: TRgb;
000081         ichFst: TC;
000082         ichLim: TC;
000083     END;
000084     TBad          = ^TAbad;
000085     TArgc         = ARRAY [0..0] OF TC;
000086     TRgc          = ^TArgc;
000087     TArgx         = ARRAY [0..0] OF TX;
000088     TRgx          = ^TArgx;
000089     TCmp          = (cmpLs, cmpEq, cmpGr);
000090     TAlld        = RECORD
000091         olleFreeFst: TC;
000092         olleLim: TC;
000093     END;
000094     TLld          = ^TAlld;
000095     TAlle        = RECORD
000096         olleNxt: TC;
000097         ollePrv: TC;
000098     END;
000099     TLle         = ^TAlle;
000100     TSp           = STRING[80];
000101     TPsp          = ^TSp;
000102     THsp         = ^TPsp;
000103     TARGCHAR      = PACKED ARRAY [0..0] OF CHAR;
000104     TRGCHAR       = ^TARGCHAR;
000105     TProc        = TP;
000106
000107     VAR
000108         Temp:      Integer;
000109         {$IFC fdbgOk}
000110         fTstStd:   TF;
000111         {$ENDC}
000112         {$IFC fstdl}
000113         fTstStd1:  TF;
000114         {$ENDC}
000115         {$IFC NOT fSymOk}                                {$D-} {$ENDC}
000116
000117     FUNCTION CMin(c1: TC; c2: TC): TC;
000118
000119     FUNCTION CMax(c1: TC; c2: TC): TC;
000120
000121     PROCEDURE MoveAchad(VAR achadDst: TAchad; VAR achadSrc: TAchad);
000122
000123     PROCEDURE MoveRgch(rgchDst: TRgch; rgchSrc: TRgch; cb: TC);
000124
000125     PROCEDURE MoveOvRgch(rgchDst: TRgch; rgchSrc: TRgch; cb: TC);
000126
000127     PROCEDURE DivMod(num: TC; denom: TC; VAR quotient: TC; VAR rem: TC);
```

Apple Lisa Computer Technical Information

```
000128
000129 FUNCTION IBinSearch(rgc: TRgc; icLst: TC; c: TC): TC;
000130
000131 PROCEDURE Break(bkcd: TC);
000132
000133 PROCEDURE MoveOlle(lld: TLld; olle: TC; VAR olleDstFst: TC;
000134             VAR olleSrcFst: TC);
000135
000136 PROCEDURE PxLld(lld: TLld);
000137
000138 PROCEDURE InitOlleFree(lld: TLld; olleFreeFst: TC; cbAlle: TC);
000139
000140 FUNCTION IchCr(VAR achad: TACHad; fFwd: TF): TC;
000141
000142 FUNCTION WAnd(w1, w2: TW): TW;
000143
000144 FUNCTION WXor(w1, w2: TW): TW;
000145
000146 FUNCTION WNot(w: TW): TW;
000147
000148 FUNCTION WShLft(w: TW; cbitSh: TW): TW;
000149
000150 FUNCTION WShRt(w: TW; cbitSh: TW): TW;
000151
000152 PROCEDURE AsgnField(p: TP; ib: TW; mskNotFld: TW; val: TW; cbitSh: TC);
000153
000154 FUNCTION ValOfField(p: TP; ib: TW; mskNotFld: TW; cbitSh: TC): TW;
000155
000156 FUNCTION UpperCh(ch: CHAR): CHAR;
000157
000158 PROCEDURE MakeSpUpper(VAR sp: TSp);
000159 {$IFC fTrace}
000160
000161 PROCEDURE InitTrCalls;
000162
000163 PROCEDURE LogCall;
000164
000165 PROCEDURE ResetTrace(LogNamesAndSegs: Boolean);
000166
000167 PROCEDURE StartCallLog;
000168
000169 PROCEDURE StopCallLog;
000170
000171 PROCEDURE ListCalls;
000172
000173 PROCEDURE SetCallTrace(Traceit: Boolean);
000174 {$ENDC}
000175
000176 *****
000177 *
000178 *             THAT'S ALL FOLKS ...
000179 *
000180 *****
000181
```

End of File -- Lines: 181 Characters: 4379

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 USOROC.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : USOROC
000004 *
000005 *****
000006
000007 USES {$U+} USOROC;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U tklib} UObject,
000016         {$U sysllib} QuickDraw,
000017         {$U tklib} UDraw,
000018         {$U tklib} UABC,
000019         {$U tklib} UTKUniversalText,
000020         {$U tklib} UText,
000021         {$U tklib} UDialog,
000022         {$U iospaslib} Blkiointr,
000023         {$U libqp/UStdTerm} UStdTerm;
000024
000025     CONST
000026         maxI           = 3;
000027
000028     TYPE
000029         states         = (q0, q1, q2, q3, q4, q6);
000030         Ilist          = ARRAY [1..maxI] OF CHAR;
000031         smallstr       = STRING[10];
000032         TSOROC        = SUBCLASS OF TStdTerm
000033             state: states;
000034             I: Ilist;
000035             newline: BOOLEAN;
000036             params, pos, top, bottom, rmarg: INTEGER;
000037             vReadBuff: TString;
000038             FUNCTION CREATE(object: TObject; heap: Theap): TSOROC;
000039             END;
000040
000041 *****
000042 *
000043 *                 THAT'S ALL FOLKS ...
000044 *
000045 *****
```

End of File -- Lines: 45 Characters: 1224

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 UVT100.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UVT100
000004 *
000005 *****
000006
000007 USES {$U+} UVT100;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U tklib} UObject,
000016         {$U sysllib} QuickDraw,
000017         {$U tklib} UDraw,
000018         {$U tklib} UABC,
000019         {$U tklib} UTKUniversalText,
000020         {$U tklib} UText,
000021         {$U tklib} UDialog,
000022         {$U iospaslib} Blkiointr,
000023         {$U libqp/UStdTerm} UStdTerm;
000024
000025     CONST
000026         maxp           = 5;
000027         maxI          = 5;
000028         bmarg         = 23;
000029         tmarg         = 0;
000030
000031     TYPE
000032         states         = (q0, q1, q2, q3, q4, q5, q6, q7);
000033         Plist          = ARRAY [1..maxp] OF Tstr255;
000034         Ilist          = ARRAY [1..maxI] OF CHAR;
000035         padtype        = (ansiAppl, vt52Appl, normal);
000036         curskeymodes  = (ansiSet, ansiReset, vt52set);
000037         smallstr       = string[10];
000038         typeStyle      = (GR, US, UK);
000039         fontStuff      = RECORD
000040             vcursorXY: TCoordinate;
000041             charInfo: TRAttributes;
000042             currType: typestyle;
000043         END;
000044     TVT100          = SUBCLASS OF TStdTerm
000045         state: states;
000046         P: Plist;
000047         I: Ilist;
000048         newline, ansi, vt52, LetHerScroll, ukPound,
000049         origabs: BOOLEAN;
000050         params, pos, top, bottom, rmarg: INTEGER;
000051         charAttr: fontStuff;
000052         vReadBuff: TString;
000053         SIFont, SOfont, currentFont: typeStyle;
000054         TILE12VTS, TILE20VTS: INTEGER;
000055         CursorKeyMode: curskeymodes;
000056         keyPad: padtype;
000057         FUNCTION CREATE(object: TObject; heap: Theap): TVT100;
000058     END;
000059
000060 *****
000061 *
```


Apple Lisa Computer Technical Information

```
000062 *                THAT'S ALL FOLKS ...
000063 *
000064 *****
000065
```

End of File -- Lines: 65 Characters: 2028

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 VLTREE.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : VLTREE
000004 *
000005 *****
000006
000007 USES { $U+ } VLTREE;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $IFC OSBUILT }
000016         { $U libsm/unitstd.obj } unitstd,
000017         { $U libdb/dbenv.obj } dbenv,
000018         { $U libin/INTRLIBp.obj } international,
000019         { $U libdb/dbdecl1.obj } dbdecl1,
000020         { $U libos/syscall.obj } syscall,
000021         { $U libos/psyscall.obj } psyscall,
000022         { $U libdb/lowlevel.obj } lowlevel,
000023         { $U libdb/pooler.obj } pooler,
000024         { $U libdb/heap.obj } heap;
000025     { $ELSEC }
000026         { $U OBJ:dbenv.obj } dbenv,
000027         { $U INTRLIB.obj } international,
000028         { $U OBJ:dbdecl1.obj } dbdecl1,
000029         { $U OBJ:syscall.obj } syscall,
000030         { $U OBJ:lowlevel.obj } lowlevel,
000031         { $U OBJ:pooler.obj } pooler,
000032         { $U OBJ:heap.obj } heap;
000033     { $ENDC }
000034
000035     Const
000036
000037
000038         noflow           = 0;
000039         treeempty       = 1;
000040         rotate          = 2;
000041         coalesce        = 3;
000042         pageempty       = 4;
000043         isplit          = 5;
000044
000045     FUNCTION entrysize(firstfld, lastfld, doitfld, isize: integer;
000046                       pedesc: ptrrecptr; prec: ptrdata): integer;
000047
000048     FUNCTION quickinsert(fileid: integer; pagetype: integer; pnewrec: ptrdata;
000049                          size: integer; pfptr: ptrfileptr; canmove: boolean;
000050                          index, buffl: integer; VAR ipage: fileptr;
000051                          VAR iindex: integer): boolean;
000052
000053     FUNCTION quickDelete(VAR kresult: integer; onfile: integer; qpage: fileptr;
000054                          qindex: integer): boolean;
000055
000056     FUNCTION quickUpdate(VAR kresult: integer; onfile: integer; pnewrec: ptrdata;
000057                          size: integer; qpage: fileptr;
000058                          qindex: integer): boolean;
000059
000060     FUNCTION splitORrotate(VAR kresult: integer; fileid: integer;
000061                            pagetype: integer; pnewrec: ptrdata; size: integer;
```

Apple Lisa Computer Technical Information

```
000062         pfptr: ptrfileptr; canmove: boolean; index,
000063         buff1: integer; leftsib, rightsib: fileptr;
000064         VAR buff2: integer; VAR ipage: fileptr;
000065         VAR iindex: integer; ps: ptrstackrec;
000066         VAR sibpage: fileptr): integer;
000067
000068     PROCEDURE firstleaf(VAR kresult: integer; fileid: integer; pagetype: integer;
000069         pnewrec: ptrdata; size: integer; VAR ipage: fileptr;
000070         VAR iindex: integer);
000071
000072     FUNCTION pagedelete(VAR kresult: integer; fileid: integer; pagetype: integer;
000073         buffer, index: integer; closestsib: fileptr;
000074         VAR rbuff: integer; VAR dpage: fileptr;
000075         VAR dindex: integer): integer;
000076
000077     PROCEDURE findduplicate(VAR kresult: integer; fileid: integer; pedesc,
000078         psdesc: ptrrecptr; nsearch: integer; is_insert,
000079         duplimpossible: boolean; pkey: ptrdata; VAR buffer,
000080         index: integer; VAR stackinvalid: integer);
000081
000082     PROCEDURE nextprior(VAR kresult: integer; fileid: integer; VAR buffer,
000083         index: integer; which: integer);
000084
000085     PROCEDURE find(VAR kresult: integer; which: integer; nsearch: integer;
000086         fileid: integer; psdesc: ptrrecptr; pkey: ptrdata;
000087         VAR buffer: integer);
000088
000089     PROCEDURE findkeyed(VAR kresult: integer; which: integer; nsearch: integer;
000090         fileid: integer; psdesc: ptrrecptr; pkey: ptrdata;
000091         VAR buffer, buffindex: integer);
000092
000093     PROCEDURE insertkeyed(VAR kresult: integer; fileid: integer; pkey: ptrdata;
000094         size: integer; VAR ipage: fileptr;
000095         VAR iindex: integer);
000096
000097     PROCEDURE deletekeyed(VAR kresult: integer; fileid: integer; pkey: ptrdata;
000098         VAR dpage: fileptr; VAR dindex: integer);
000099
000100 *****
000101 *
000102 *             THAT'S ALL FOLKS ...
000103 *
000104 *****
000105
```

End of File -- Lines: 105 Characters: 4054

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 WINDOWS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WINDOWS
000004 *
000005 *****
000006
000007 USES { $U+ } WINDOWS;
000008
000009
000010 INTRINSIC SHARED;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U libsm/UnitStd      } UnitStd,
000016         { $U libsm/UnitHz      } UnitHz,
000017         { $U libqd/Storage     } Storage,
000018         { $U libqd/QuickDraw   } QuickDraw,
000019         { $U libfm/FontMgr     } FontMgr,
000020         { $U libos/SysCall     } SysCall,
000021         { $U libwm/Events      } Events;
000022
000023     CONST
000024         lastWindow      = 24;
000025
000026     TYPE
000027         StringPtr      = ^Str255;
000028         StringHandle   = ^StringPtr;
000029         WindowPeek     = ^WindowRecord;
000030         WindowRecord   = RECORD
000031             port: GrafPort;
000032             strucRgn: RgnHandle;
000033             contRgn: RgnHandle;
000034             updateRgn: RgnHandle;
000035             windowPic: PicHandle;
000036             windowProc: ProcPtr;
000037             dataHandle: Handle;
000038             refCon: LongInt;
000039             process: LongInt;           { window process owner }
000040             titleHandle: StringHandle;
000041             titleWidth: INTEGER;
000042             nextWindow: WindowPeek;
000043             closeIcon: INTEGER;
000044             iconOverlay: INTEGER;
000045             available: BOOLEAN;
000046             visible: BOOLEAN;
000047             hilited: BOOLEAN;
000048             hasTab: BOOLEAN;
000049             okMove: BOOLEAN;
000050             okGrow: BOOLEAN;
000051             okFront: BOOLEAN;
000052             okClose: BOOLEAN;
000053             okActivate: BOOLEAN;
000054             okDispose: BOOLEAN;
000055             END;
000056         WindowMessage = (draw, hit, calcRgns);
000057
000058     VAR
000059         buttonWindow: WindowPeek;
000060         grayRgn:      RgnHandle;
000061         wmPort:       GrafPtr;
```

Apple Lisa Computer Technical Information

```
000062     saveUpdate:   BOOLEAN;
000063     paintWhite:   BOOLEAN;
000064     windowArray:  ARRAY [0..lastWindow] OF WindowRecord;
000065     dPort, wPort: GrafPort;
000066
000067     PROCEDURE BringToFront(window: WindowPeek; reDraw: BOOLEAN);
000068
000069     PROCEDURE CalcVis(window: WindowPeek);
000070
000071     PROCEDURE CalcVisBehind(startWindow: WindowPeek; clobbered: RgnHandle);
000072
000073     PROCEDURE ChangeProcess(window: WindowPtr; newProcess: ProcessID);
000074
000075     PROCEDURE CheckWindow(VAR event: EventRecord);
000076
000077     PROCEDURE ClipAbove(window: WindowPeek);
000078
000079     PROCEDURE DeleteWindow(window: WindowPeek);
000080
000081     PROCEDURE DisposeString(sh: StringHandle);
000082
000083     PROCEDURE DrawNew(window: WindowPeek; fUpdate: BOOLEAN);
000084
000085     PROCEDURE GetWindTitle(window: WindowPeek; VAR title: Str255);
000086
000087     PROCEDURE HideWindow(window: WindowPeek);
000088
000089     PROCEDURE HiLiteWindow(window: WindowPeek; fHiLite: BOOLEAN);
000090
000091     PROCEDURE HitContent(window: WindowPeek; event: EventRecord);
000092
000093     PROCEDURE InsertWindow(window, behind: WindowPeek);
000094
000095     PROCEDURE MakeActive(window: WindowPeek; event: EventRecord);
000096
000097     PROCEDURE MoveWindow(windPeek: WindowPeek; dh, dv: INTEGER;
000098                       bringFront: BOOLEAN);
000099
000100     FUNCTION NewString(str: Str255): StringHandle;
000101
000102     PROCEDURE PaintBehind(startWindow: WindowPeek; clobbered: RgnHandle);
000103
000104     PROCEDURE PaintOne(window: WindowPeek; clobbered: RgnHandle);
000105
000106     FUNCTION PrevWindow(window: WindowPeek): WindowPeek;
000107
000108     PROCEDURE SaveOld(window: WindowPeek);
000109
000110     PROCEDURE SendUpdate;
000111
000112     PROCEDURE SetActWindow(window: WindowPeek);
000113
000114     PROCEDURE SetString(sh: StringHandle; str: Str255);
000115
000116     PROCEDURE SetWindTitle(window: WindowPeek; title: Str255);
000117
000118     PROCEDURE ShowWindow(window: WindowPeek);
000119
000120     PROCEDURE TopActive;
000121
000122     FUNCTION UpShift(ch: CHAR): CHAR;
000123
000124     PROCEDURE WindowSize(window: WindowPeek; width, height: INTEGER;
000125                       fUpdate: BOOLEAN);
000126
000127     *****
```

Apple Lisa Computer Technical Information

```
000128 *  
000129 *          THAT'S ALL FOLKS ...  
000130 *  
000131 *****  
000132
```

End of File -- Lines: 132 Characters: 3916

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 WMLCRS.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WMLCRS
000004 *
000005 *****
000006
000007 USES { $U+ } WMLCRS;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         { $U libsm/Unitstd } UnitStd,
000016         { $U libsm/UnitHz } UnitHz,
000017         { $U libqd/Storage } Storage,
000018         { $U libqd/QuickDraw } QuickDraw,
000019         { $U libfm/FontMgr } FontMgr,
000020         { $U libfm/FmgrUtil } FmgrUtil,
000021         { $U libsb/WmlStd } WmlStd;
000022
000023     CONST
000024         icrsFirst      = - 1;
000025         icrsHidden     = - 1;
000026         icrsEscape     = 0;
000027         icrsInactive   = 1;
000028         icrsUpDwn      = 2;
000029         icrsLftRt      = 3;
000030         icrsGrow       = 4;
000031         icrsIbeam      = 5;
000032         icrsMenu       = 6;
000033         icrsVSkewer    = 7;
000034         icrsHSkewer    = 8;
000035         icrsLCcross    = 9;
000036         icrsXIbeam     = 10;
000037         icrsHrGlass    = 11;
000038         icrsCheck      = 12;
000039         icrsGECross    = 13;
000040         icrsLFIbeam    = 14;
000041         icrsGEIbeam    = 15;
000042         icrsLast       = 15;
000043
000044     PROCEDURE InitWmlCrs(VAR cError: TC);
000045
000046     PROCEDURE SetStdCursor(icrs: TC);
000047
000048     FUNCTION IcrsCurrent: TC;
000049
000050 *****
000051 *
000052 *             THAT'S ALL FOLKS ...
000053 *
000054 *****
000055
```

End of File -- Lines: 55 Characters: 1283

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 WMLGROW.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WMLGROW
000004 *
000005 *****
000006
000007 USES { $U+ } WMLGROW;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES                                     { $U libsm/UnitStd } UnitStd,
000015     { $U libsm/UnitHz                       } UnitHz,
000016     { $U libqd/Storage                       } Storage,
000017     { $U libqd/QuickDraw                     } QuickDraw,
000018     { $U libfm/FontMgr                       } FontMgr,
000019     { $U libsb/WmlStd                         } WmlStd,
000020     { $IFC FWMLoS }
000021     { $U libos/SysCall } SysCall,
000022     { $ELSEC }
000023     { $U obj:OSStub } SysCall,
000024     { $ENDC }
000025     { $U libwm/Events                         } Events,
000026     { $U libwm/Folders                       } Folders;
000027
000028     CONST
000029         dhGrow          = 24;
000030         dvGrow          = 16;
000031
000032     PROCEDURE GetGrowRect(VAR rGrow: TR);
000033
000034     PROCEDURE PaintGrow;
000035
000036     FUNCTION FGrowHit(pt: TPt): TF;
000037
000038     PROCEDURE DragFrame(ptMouse: TPt; fDrawScrolls: TF; VAR ptNewBR: TPt);
000039
000040     PROCEDURE InitXorFrame;
000041
000042     PROCEDURE XorFrame(ptFrameBR: TPt; fDrawScrolls: TF);
000043
000044     PROCEDURE EndXorFrame;
000045
000046 *****
000047 *
000048 *                                     THAT'S ALL FOLKS ...
000049 *
000050 *****
000051
```

End of File -- Lines: 51 Characters: 1218

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 WMLSB.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WMLSB
000004 *
000005 *****
000006
000007 USES {$U+} WMLSB;
000008
000009
000010 INTRINSIC;
000011
000012 INTERFACE
000013
000014     USES
000015         {$U libsm/UnitStd } UnitStd,
000016         {$U libsm/UnitHz } UnitHz,
000017         {$U libqd/Storage } Storage,
000018         {$U libqd/QuickDraw } QuickDraw,
000019         {$U libfm/FontMgr } FontMgr,
000020         {$U libsb/WmlStd } WmlStd,
000021         {$IFC FWMLOS}
000022         {$U libos/Syscall } Syscall,
000023         {$ELSEC}
000024         {$U obj:OsStub } Syscall,
000025         {$ENDC}
000026         {$U libwm/Events } Events;
000027
000028     CONST
000029         dhSbox      = 24;
000030         dvSbox      = 16;
000031         dhThumb     = 24;
000032         dvThumb     = 16;
000033         dhSkewer    = 6;
000034         dvSkewer    = 4;
000035
000036     TYPE
000037         TIcon       = (iconSkewer, iconArwA, iconPagA, iconGryA, iconThumb,
000038                       iconGryB, iconPagB, iconArwB);
000039         TSicon      = SET OF TIcon;
000040         THSb        = ^TPSb;
000041         TPSb        = ^TSb;
000042         TSb         = RECORD
000043             r: TR;
000044             tyvh: TTyvh;
000045             sicon: TSicon;
000046             siconVis: TSicon;
000047             cThumb: TC;
000048             refCon: TL;
000049             fHead: TF;
000050             hsbPrv: THSb;
000051             hsbNxt: THSb;
000052         END;
000053         TSbList     = RECORD
000054             hz: THz;
000055             hsbFst: THSb;
000056         END;
000057         TPosts      = ARRAY [iconSkewer..iconArwB] OF TCd;
000058
000059     VAR
000060         hsbNil:     THSb;
000061         dptSbox:    TPT;
```

Apple Lisa Computer Technical Information

```
000062      dptThumb:      TPt;
000063      dptSkewer:     TPt;
000064      TyvhNOT:      ARRAY [TTyvh] OF TTyvh;
000065      fTstSb:       TF;
000066
000067      PROCEDURE InitWmlSb;
000068
000069      PROCEDURE InitsbList(VAR sbList: TSbList; hz: THz);
000070
000071      FUNCTION SbCreate(VAR sbList: TSbList; hsbBefore: THSb; tyvhSb: TTyvh;
000072                      ptTopLeft: TPt; cdBotRt: TCd): THSb;
000073
000074      PROCEDURE KillSb(VAR sbList: TSbList; hsbKill: THSb);
000075
000076      PROCEDURE GetSbRect(hsb: THSb; VAR r: TR);
000077
000078      PROCEDURE SetSbRect(hsb: THSb; r: TR);
000079
000080      FUNCTION RefconSb(hsb: THSb): TL;
000081
000082      PROCEDURE SetSbRefcon(hsb: THSb; refcon: TL);
000083
000084      FUNCTION HsbFirst(VAR sbList: TSbList): THSb;
000085
000086      FUNCTION HsbNext(hsb: THSb): THSb;
000087
000088      FUNCTION HsbPrev(hsb: THSb): THSb;
000089
000090      FUNCTION TyvhOfSb(hsb: THSb): TTyvh;
000091
000092      PROCEDURE SetSbIcons(hsb: THSb; siconNew: TSicon);
000093
000094      PROCEDURE GetSbIcons(hsb: THSb; VAR sicon: TSicon);
000095
000096      PROCEDURE GetVisIcons(hsb: THSb; VAR siconVis: TSicon);
000097
000098      FUNCTION CThumbPos(hsb: THSb): TC;
000099
000100      PROCEDURE SetThumb(hsb: THSb; cNewThumb: TC);
000101
000102      PROCEDURE MoveSb(hsb: THSb; ptNewTL: TPt);
000103
000104      PROCEDURE AdjsbCorner(hsb: THSb; dcdAdjust: TCd; fTopLeft: TF);
000105
000106      PROCEDURE PaintSb(hsb: THSb);
000107
000108      PROCEDURE EraseSb(hsb: THSb);
000109
000110      PROCEDURE PaintArw(hsb: THSb; iconWhichArw: TIcon; fBlack: TF);
000111
000112      FUNCTION HsbFromPt(VAR sbList: TSbList; pt: TPt): THSb;
000113
000114      FUNCTION FSbHit(VAR sbList: TSbList; pt: TPt; VAR hsbHit: THSb;
000115                   VAR iconHit: TIcon): TF;
000116
000117      PROCEDURE FlushRects(VAR rH, rV: TR);
000118
000119      PROCEDURE MkPosts(hsb: THSb; VAR posts: TPosts);
000120
000121      PROCEDURE SplitSb(VAR sbList: TSbList; hsbOld: THSb; VAR hsbNew: THSb;
000122                    cdCut: TCd);
000123
000124      PROCEDURE AdjSplitBetween(hsbBefore, hsbAfter: THSb; dcdAdjust: TCd);
000125
000126      FUNCTION HsbNextOnSbar(hsb: THSb): THSb;
000127
```

Apple Lisa Computer Technical Information

```
000128 FUNCTION HsbPrevOnSbar(hsb: THSb): THSb;
000129
000130 FUNCTION HsbSbarHead(hsb: THSb): THSb;
000131
000132 FUNCTION HsbSbarTail(hsb: THSb): THSb;
000133
000134 FUNCTION FSbarHead(hsb: THSb): TF;
000135
000136 PROCEDURE SlideSbar(hsb: THSb; dcdAdjust: TCd);
000137
000138 PROCEDURE PaintSbar(hsb: THSb);
000139
000140 PROCEDURE EraseSbar(hsb: THSb);
000141
000142 PROCEDURE SetupMvThumb(hsb: THSb);
000143
000144 PROCEDURE MoveThumb(cNewThumb: TC);
000145
000146 PROCEDURE DragThumb(hsb: THSb; ptMouse: TPt; VAR cThumbUp: TC);
000147
000148 PROCEDURE FixRLimits(hsb: THSb; VAR rLimits: TR);
000149
000150 PROCEDURE DragSkewer(hsbDwn: THSb; ptMouse: TPt; rLimits: TR;
000151                     VAR ptSkewerUp: TPt);
000152
000153 PROCEDURE HideSBorder(r: TR);
000154 {$IFC WmlDebug }
000155
000156 PROCEDURE PxSbList(VAR sbList: TSbList);
000157
000158 PROCEDURE PxSicon(sicon: TSicon);
000159
000160 PROCEDURE PxSb(hsb: THSb);
000161
000162 PROCEDURE PxSbar(hsb: THSb);
000163 {$ENDC }
000164
000165 *****
000166 *
000167 *                 THAT'S ALL FOLKS ...
000168 *
000169 *****
000170
```

End of File -- Lines: 170 Characters: 4238

Apple Lisa Computer Technical Information

```
=====
FILE: "LISA LIB 3 WMLSTD.TEXT"
=====
```

```
000001 *****
000002 *
000003 * APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WMLSTD
000004 *
000005 *****
000006
000007 USES { $U+ } WMLSTD;
000008
000009
000010 INTRINSIC;
000011     { $SETC FWML0S := TRUE }
000012
000013 INTERFACE
000014
000015     USES                               { $U libsm/UnitStd } UnitStd,
000016     { $U libsm/UnitHz } UnitHz,
000017     { $U libqd/Storage } Storage,
000018     { $U libqd/QuickDraw } QuickDraw,
000019     { $U libfm/FontMgr } FontMgr;
000020     { $SETC WmlDebug := FDbgOK }
000021     { $SETC WmlSymbols := FSymOk }
000022     { $SETC doTraceSB := TRUE }
000023     { $SetC fTraceSB := doTraceSB AND fTRACE }
000024
000025     CONST
000026         tyvhV          = V;
000027         tyvhH          = H;
000028
000029     TYPE
000030         TAport         = GrafPort;
000031         TPort          = GrafPtr;
000032         TBmp           = BitMap;
000033         TPt            = Point;
000034         TTyvH          = VHSelect;
000035         TR              = Rect;
000036
000037     PROCEDURE PenSave;
000038
000039     PROCEDURE PenRestore;
000040
000041     PROCEDURE WmlTxtSave;
000042
000043     PROCEDURE WmlTxtRestore;
000044
000045 *****
000046 *
000047 *                               THAT'S ALL FOLKS ...
000048 *
000049 *****
000050
```

End of File -- Lines: 50 Characters: 1110

SUMMARY:

```
Total number of files : 76
Total file lines       : 9648
Total file characters  : 278266
```