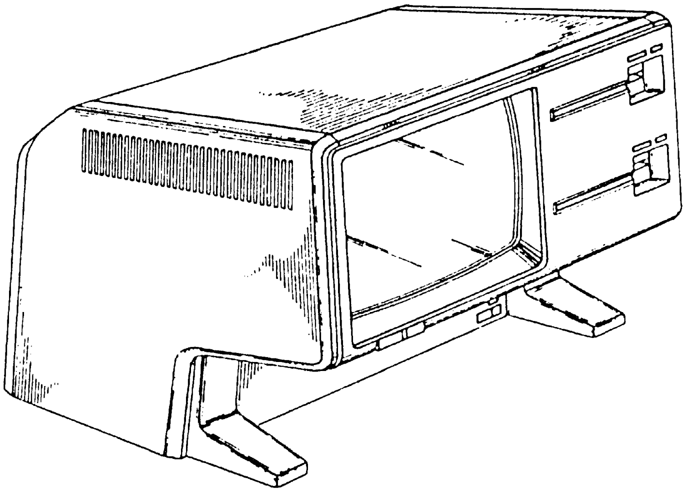
Apple Lisa

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).

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

```
**************************************************************************
*
*              APPLE LISA DESKTOP 3.0 LIBRARIES UNIT TABLE
*
**************************************************************************

UnitName Unit# File# Type   DataSize
-------- ----- ----- ------ --------
PASLIB      1    3  Intrin   000000
UNITSTD     2    4  Intrin   000002
GRAFUTIL    3    4  Intrin   000000
UNITHZ      4    4  Shared   000024
STORAGE     5    4  Intrin   000008
QUICKDRA    6    4  Intrin   0000CE
HWINT       7    4  Intrin   000000
FEDEC       8    4  Intrin   000000
FONTMGR     9    4  Shared   000054
EVENTS     10    4  Shared   000210
WINDOWS    11    4  Shared   0010F2
FOLDERS    12    4  Shared   00000A
MENUS      13    4  Shared   00009C
FLDUT      14    4  Intrin   000000
WMLSTD     15    4  Intrin   000018
WMLCRS     16    4  Intrin   0003FE
WMLGROW    17    4  Intrin   00000E
WMLSB      18    4  Intrin   000038
INTERNAT   19    4  Intrin   000000
FPLIB      20    5  Intrin   000006
ALERTMGR   21    4  Intrin   000300
MATHLIB    22    5  Intrin   000016
UNITFF     23    4  Shared   00004A
UNITCS     24    4  Shared   000644
UNITFILE   25    4  Shared   000156
UNITFMT    26    4  Shared   0001F2
BGENV      27    8  Intrin   000058
MATMGR     28    8  Intrin   00188E
PARSER     29    8  Intrin   0006FC
COMPUTE    30    8  Intrin   000084
STUBS      31    8  Intrin   000000
UUNIVERS   32    4  Intrin   000222
FEGLOBAL   33    8  Intrin   000398
COMGR      34    8  Intrin   00033C
USTDTERM   35    9  Intrin   0002CE
UQPGRAPH   36    9  Intrin   000056
TEEXEC     37    8  Intrin   000004
UQPPRINT   38    9  Intrin   000188
LCFEXEC    39    8  Intrin   0001A6
TEENV      40    6  Intrin   00080E
TM         41    6  Intrin   000BCA
FMGRUTIL   42    4  Intrin   000000
PMM        43    4  Shared   000042
PMDECL     44    4  Intrin   000000
UNITFIGA   45    4  Shared   00002A
SYS1LOCK   46    4  Intrin   000000
UCLASCAL   47    3  Intrin   00004E
UTKUNIVE   49   11  Intrin   0001CA
UOBJECT    50   10  Intrin   000164
UTEXT      51   11  Intrin   00027C
UDRAW      52   10  Intrin   0001B2
UABC       53   10  Intrin   0009F2
UDIALOG    54   11  Intrin   000564
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  PRDLGMGR   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
```

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : ALERTMGR
*
********************************************************************************

USES {$U+} ALERTMGR;


INTRINSIC;

INTERFACE

   USES
      {$U libhw/HWINT} HwInt,
      {$U libsm/UNITSTD} UnitStd,
      {$U libsm/UNITHZ} UnitHz,
      {$U libqd/STORAGE} Storage,
      {$U libqd/QUICKDRAW} QuickDraw,
      {$U libfm/FONTMGR} FontMgr,
      {$U libfm/FMGRUTIL} FMgrUtil,
      {$U libos/SYSCALL} SysCall,
      {$U libos/PSYSCALL} PSysCall,
      {$U libpm/PMDECL} PMDecl,
      {$U libpm/PMM} PMM,
      {$U libwm/EVENTS} Events,
      {$U libwm/FOLDERS} Folders,
      {$U libwm/MENUS} Menus,
      {$U libwm/WINDOWS} Windows,
      {$U libsb/WMLSTD} WmlStd,
      {$U libsb/WMLCRS} WmlCrs,
      {$U libdb/DBENV} dbenv;
      {$setc alSymbols := FSymOK}
      {$setc alDebug := FDbgOK}
      {$setc dbgAlWould := alDebug}
      {$setc dbgAlrt := false}
      {$setc dbgAlEvt := false}
      {$setc dbgAlRdErr := false}
      {$setc dbgAlOpErr := false}

   CONST
      maxButn       = 10;
      noButn        = 11;

   TYPE
      TButn         = 0..maxButn;
      TAlertFile    = ^PAlertRec;
      PAlertRec     = ^TAlertRec;
      TAlertRec     = RECORD
                         falerts: longint;
                         fnumAlerts: integer;
                         frefNum: integer;
                         fPos: integer;
                      END;
      TParamAlert   = String[40];
      TArgAlert     = 0..5;
      TCountAlert   = 7..9;
      TPstr255      = ^Str255;
      TAlertKind    = (getProc, drawProc, askProc, waitProc, stopProc, noteProc,
                       cautionProc, cautionOKproc, cautionCancelProc,
                       cautionStopProc, cautionNoteProc, cautionInsistProc);
```

```

VAR
   alertError:   integer;
   refuseDeactivate: boolean;
   traceWouldAlert: boolean;
   preventReentry: boolean;
   BackGrInProgress: boolean;
   {$ifc dbgAlRdErr}
   fakeAlError:  integer;
   {$endc}

FUNCTION AskAlert(alertFile: TAlertFile; alertNumber: integer): integer;

PROCEDURE ArgAlert(n: TArgAlert; s: Str255);

FUNCTION BackgroundAlert(alertFile: TAlertFile; alertNumber: integer;
                         alertType: TAlertKind): integer;

PROCEDURE BeepAlert(volume: integer);

FUNCTION ButnPushed(first, last: TButn; VAR pushed: TButn;
                    pt: Point): boolean;

FUNCTION CautionAlert(alertFile: TAlertFile; alertNumber: integer): boolean;

FUNCTION CalcButWidth(topButn, botButn: TButn; VAR height: integer): integer;

PROCEDURE CountAlert(whichCounter: TCountAlert; countValue: integer);

PROCEDURE DrawAlert(alertFile: TAlertFile; alertNumber: integer;
                    marginRect: Rect);

PROCEDURE DTAlert(alertFile: TAlertFile; osDT: longint;
                  VAR userString: TParamAlert);

PROCEDURE EndWaitAlert;

PROCEDURE GetAlert(alertFile: TAlertFile; alertNumber: integer;
                   pStr: TPstr255);

PROCEDURE GetButn(d: TButn; pStr: TPstr255);

PROCEDURE HideButn(d: TButn);

PROCEDURE InitAlerts(cacheSize, cacheBytes: integer; hz: THz; idleProcPtr,
                     errProcPtr: procPtr);

PROCEDURE initMP(myHeap: Thz; toolPreFix: TParamAlert;
                 ptrToAlertProc: procPtr; phraseversion: integer;
                 myMenus: rMenuPtr; VAR alertFile: TAlertFile;
                 VAR error: integer);

PROCEDURE LdSgAl;

PROCEDURE LocateAlert(top: integer);

PROCEDURE LockAlert(alertFile: TAlertFile; alertNumber: integer);

PROCEDURE NoteAlert(alertFile: TAlertFile; alertNumber: integer);

FUNCTION NthAlert(ordinal: INTEGER; alertFile: TAlertFile; VAR alertNumber,
                  numStages: integer; VAR alertKind: TAlertKind): BOOLEAN;

PROCEDURE OpenPhraseFile(VAR refNum: integer; path: pathName);

PROCEDURE ParamAlert(cite1, cite2, cite3: Str255);
```

```

   PROCEDURE PushButn(d: TButn; f: boolean);

   FUNCTION ReadAlerts(refNum, version: integer): TAlertFile;

   PROCEDURE ReshowButn(d: TButn);

   PROCEDURE SetButWidths(width: integer);

   PROCEDURE ShowButn(d: TButn; h, v: integer; fDefault: boolean);

   PROCEDURE StopAlert(alertFile: TAlertFile; alertNumber: integer);

   PROCEDURE UnlockAlerts;

   PROCEDURE WaitAlert(alertFile: TAlertFile; alertNumber: integer);

   PROCEDURE WouldAlert(VAR menu: menuInfo; itemIndex: integer);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 152 Characters: 4546

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : BLKIOINT
*
*******************************************************************************

USES {$U+} BLKIOINT;


INTRINSIC; {$%+}

INTERFACE

   USES
      {$U libos/syscall.obj } syscall,
      {$U libpl/UClascal.obj} UClascal;

      {$SETC MONITOR  := false}
      {$SETC DEBUG    := false}
      {$SETC fRANGECK := TRUE }
      {$SETC fSYMBOL  := TRUE }

   CONST
      MMaxInt       = $7FFF;
      MaxUnit       = 20;
      MaxDir        = 77;
      MaxDir1       = 78;
      VIDLeng       = 7;
      TIDLeng       = 15;
      MaxSeg        = 15;
      FBlkSize      = 512;
      DirBlk        = 2;
      NameLen       = 23;
      FillrLn       = 11;
      SysUnit       = 5;
      DirSize       = 2048;
      DirNBlocks    = 4;
      unitScConsole = 1;
      unitScKeyboard = 2;
      unitWiConsole = 4;
      unitWiKeyboard = 5;
      unitScPrinter = 6;
      unitWiPrinter = 7;
      unitOSFile    = 10;

   TYPE
      {$IFC MONITOR}
      IORsltWd      = (INoError, IBadBlock, IBadUnit, IBadMode, ITimeOut,
                         ILostUnit, ILostFile, IBadTitle, INoRoom, INoUnit,
                         INoFile, IDupFile, INotClosed, INotOpen, IBadFormat,
                         IStrgOvfl, IWrProt, IDevError, INoDirSpace);
      {$ELSEC}
      IORsltWd      = (INoError);
      {$ENDC}
      DateRec       = PACKED RECORD
                           Year: 0..100;
                           Day: 0..31;
                           Month: 0..12;
                        END;
      UnitNum       = 0..MaxUnit;
      VID           = STRING[VIDLENG];
```

```
DirRange      = 0..MaxDir;
dirrang1      = 0..MaxDir1;
TitleID       = STRING[TIDLENG];
FullID        = STRING[NAMELEN];
FileKind      = (UNTYPEDFILE, XDSKFILE, CODEFILE, TEXTFILE, INFOFILE,
                 DATAFILE, GRAFFILE, FOTOFILE, SECUREDIR);
DirEntry      = PACKED RECORD
                   DFIRSTBLK: INTEGER;
                   DLASTBLK: INTEGER;
                   STATUS: BOOLEAN;
                   FILLER1: 0..1024;
                   CASE DFKIND: FILEKIND OF
                      SECUREDIR, UNTYPEDFILE:
                         (DVID: VID;
                          DEOVBLK: INTEGER;
                          DNUMFILES: INTEGER;
                          DLOADTIME: INTEGER;
                          DLASTBOOT: DATEREC);
                      XDSKFILE, CODEFILE, TEXTFILE, INFOFILE, DATAFILE,
                      GRAFFILE, FOTOFILE:
                         (DTID: TitleID;
                          DLASTBYTE: integer;
                          CASE BOOLEAN OF
                             TRUE:
                                (DORefnum: integer; );
                             FALSE:
                                (DACCESS: DATEREC; ); )
                END;
DirP          = ^Directory;
Directory     = ARRAY [DirRange] OF DirEntry;
CloseType     = (CNormal, CLock, CPurge, CCrunch);
CharArray     = PACKED ARRAY [0..FBlkSize] OF CHAR;
WindowP       = ^Window;
Window        = CharArray;
FileState     = (FJandW, FNeedChar, FGotChar);
FIBP          = ^FIB;
FVIDrec       = RECORD
                   CASE BOOLEAN OF
                      TRUE:
                         (FTrefnum: integer);
                      FALSE:
                         (FVID: VID; );
                END;
FIB           = RECORD
                   FWindow: WindowP;
                   FEOLN, FEOF: BOOLEAN;
                   FIsOS: BOOLEAN;
                   FState: FileState;
                   FRecSize: INTEGER;
                   CASE FIsOpen: BOOLEAN OF
                      TRUE:
                         (FIsBlkd: BOOLEAN;
                          FNewFile: BOOLEAN;
                          FUnit: UnitNum;
                          FVIDstuff: FVIDrec;
                          FMaxBlk, FNxtBlk, FReptCnt: INTEGER;
                          HiByt3: 0..100;
                          FModified: BOOLEAN;
                          FHeader: DIRENTRY;
                          HiByt4: 0..100;
                          CASE FSoftBuf: BOOLEAN OF
                             TRUE:
                                (FMaxByte, FNxtByte: INTEGER;
                                 HiByt5: BOOLEAN;
                                 FBufChngd: BOOLEAN;
                                 FBuffer: CharArray))
                   END;
   {$IFC MONITOR}
   UTBLENTRY     = RECORD
                      UVID: VID;
                      LOGGED: BOOLEAN;
                      lockdir: boolean;
                      CASE UISBLKD: BOOLEAN OF
                         TRUE:
                            (UEOVBLK: INTEGER)
                   END;
   {$ENDC}
   envirRec      = RECORD
                      A6_init: longint;
                      PC_2setup: longint;
                      PC_retinit: longint;
                      SP_init: longint;
                   END;
   longaddr      = ^longint;
   Tstr255       = string[255];
   TTerminal     = SUBCLASS OF NIL
                      FUNCTION CREATE: TTerminal; ABSTRACT;
                      PROCEDURE VWrite(VAR str: Tstr255); ABSTRACT;
                      FUNCTION VRead: CHAR; ABSTRACT;
                      PROCEDURE VGotoxy(x, y: INTEGER); ABSTRACT;
                      FUNCTION VKeyPress: BOOLEAN; ABSTRACT;
                      FUNCTION VAbortKey: BOOLEAN; ABSTRACT;
                      PROCEDURE VScreenCtr(contrfun: INTEGER); ABSTRACT;
                      FUNCTION OpenPrinter: INTEGER; ABSTRACT;
                      FUNCTION PrWrite(VAR str: Tstr255): INTEGER; ABSTRACT;
                      FUNCTION ClosePrinter: INTEGER; ABSTRACT;
                      PROCEDURE CleanUp; ABSTRACT;
                      PROCEDURE SetupIOFile;
                      PROCEDURE InWindow(yesno: BOOLEAN);
                   END;

VAR
   IORslt:       INTEGER;
   PMyid:        longint;
   PAbortKeyF:   boolean;
   PResProgram:  boolean;
   PBeginEnv:    envirRec;
   PTrLisaChar:  Boolean;
   PConsWindow:  BOOLEAN;
   PInWindow:    INTEGER;
   PInputfwindow: CHAR;
   PInputfile:   FILE;
   POutfwindow:  CHAR;
   POutputfile:  FILE;
   PasTerm:      TTerminal;
   TermMptr:     ^longint;
   Termdata:     RECORD
                    classptr: longint;
                 END;
   haltaddress:  LONGINT;
   fpcb:         INTEGER;
   {$IFC MONITOR}
   gdirectory:   PACKED ARRAY [1..dirsize] OF CHAR;
   {$ENDC}

FUNCTION %_FIORESULT: INTEGER;

PROCEDURE %_Setiorslt(ior: integer);

PROCEDURE %_Backupinput(VAR f: text);

FUNCTION %%%zenviron: longint;
```

```

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 200 Characters: 7086

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : BLOCKIO
*
********************************************************************************

USES {$U+} BLOCKIO;


INTRINSIC SHARED; {$%+}

INTERFACE

   USES
      {$U libos/syscall.obj} syscall,
      {$U libos/psyscall.obj} psyscall,
      {$U libpl/blkiointr.obj} blkiointr;

   CONST
      CclearScreen  = 1;
      CclearEScreen = 2;
      CclearELine   = 3;
      CgoHome       = 11;
      CleftArrow    = 12;
      CrightArrow   = 13;
      CupArrow      = 14;
      CdownArrow    = 15;

   TYPE
      consoledest   = (alscreen, mainscreen, xsorocA, xsorocB, folder, spare1,
                       spare2, spare3);
      dsProcCode    = (dsResProg, dsSoftPwbtn, dsPrintDev, dsSetGPrefix,
                       dsEnbDisk, dsGetDiskEnbF);
      dsProcParam   = RECORD
                         CASE ProcCode: dsProcCode OF
                            dsResProg:
                               (RProcessId: longint);
                            dsSoftPwbtn:
                               (SPButton: boolean);
                            dsPrintDev:
                               (PrDevice: e_name);
                            dsSetGPrefix:
                               (errnum: INTEGER;
                                prefix: pathname);
                            dsEnbDisk:
                               (toEnbDisk: boolean);
                            dsGetDiskEnbF:
                               (diskEnbF: BOOLEAN);
                      END;
      pdsProcCode   = (pdsSomething);
      pdsProcParam  = RECORD
                         CASE ProcCode: pdsProcCode OF
                            pdsSomething:
                               (something: longint);
                      END;

   PROCEDURE %_FINIT(VAR F: FIB; WINDOW: WINDOWP; recbytes: INTEGER);

   PROCEDURE %_FOPEN(VAR F: FIB; VAR FTITLE: pathname; FOPENOLD: BOOLEAN;
                     JUNK: longint);

```

```
    PROCEDURE %_FCLOSE(VAR F: FIB; FTYPE: integer);

    FUNCTION %_FBLOCKIO(VAR F: FIB; A: longint; NBLOCKS, RBLOCK: INTEGER;
                        DOREAD: BOOLEAN): INTEGER;

    PROCEDURE %_FGET(VAR F: FIB);

    PROCEDURE %_FPUT(VAR F: FIB);

    PROCEDURE %_WRITELN(VAR F: FIB);

    PROCEDURE %_WRITECHAR(VAR F: FIB; charaddr: longaddr; count: integer);

    PROCEDURE %_READLN(VAR F: FIB);

    FUNCTION %_READCHAR(VAR F: FIB): CHAR;

    PROCEDURE %_FSEEK(VAR F: FIB; RECORDNUM: integer);

    PROCEDURE %_FGOTOXY(X, Y: INTEGER);

    FUNCTION PAbortFlag: boolean;

    PROCEDURE GetGPrefix(VAR prefix: pathname);

    PROCEDURE ScreenCtr(contrfun: integer);

    PROCEDURE GetPrDevice(VAR PrDevice: e_name);

    PROCEDURE PTranLisaChar(toTranslate: boolean);

    FUNCTION PaslibVersion: INTEGER;

    PROCEDURE BlockIOInit;

    PROCEDURE BlockIOdisinit;

    PROCEDURE lockPaslib(VAR errnum: integer);

    PROCEDURE lockPasiolib(VAR errnum: integer);

    PROCEDURE moveconsole(VAR errnum: integer; applconsole: consoledest);

    PROCEDURE ExecReset(VAR errnum: INTEGER; VAR execfile: pathname;
                        stopexec: BOOLEAN);

    FUNCTION ExecFlag: BOOLEAN;

    PROCEDURE OutputRedirect(VAR errnum: INTEGER; VAR outfile: pathname;
                             stopoutput: BOOLEAN);

    FUNCTION OutputRFlag: BOOLEAN;

    PROCEDURE DSPaslibCall(VAR ProcParam: dsProcParam);

    PROCEDURE %_PDSPaslibCall(VAR ProcParam: pdsProcParam);

    PROCEDURE %_blockiosetup(firsttime: boolean; VAR regis: envirRec);

    FUNCTION %_CkResident(VAR regis: envirRec): boolean;

    FUNCTION %_pkeypress(VAR f: fib): boolean;

    PROCEDURE %_funitiocall(unitno: integer; bufaddr: longint; nbytes: integer;
                            blocknum: integer; mode: integer; unitcode: integer);
```

```
   FUNCTION %_funitbusy(unitno: integer): boolean;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 135 Characters: 3909

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : CZCOMPAC
*
********************************************************************************

USES {$U+} CZCOMPAC;


INTRINSIC;

INTERFACE

   USES
      {$IFC OSBuilt }
         {$U libsm/unitstd.obj} unitstd,
         {$U libdb/dbenv.obj } dbenv,
         {$U libdb/dbdecl1.obj } dbdecl1,
         {$U libos/syscall.obj } syscall,
         {$U libos/psyscall.obj } psyscall,
         {$U libdb/lowlevel.obj } lowlevel,
         {$U libdb/pooler.obj } pooler,
         {$U libdb/heap.obj } heap;
      {$ELSEC }
         {$U obj:dbenv.obj } dbenv,
         {$U obj:dbdecl1.obj } dbdecl1,
         {$U obj:syscall.obj } syscall,
         {$U obj:lowlevel.obj } lowlevel,
         {$U obj:pooler.obj } pooler,
         {$U obj:heap.obj } heap;
      {$ENDC }

   {$SETC debug := false }

   Type


      states_de_compact = (no_try, first_try);

   VAR
      pheap_segment: longint;
      heap_segment: integer;
      heapsize:     longint;
      czone:        zonedesc;
      compact_state: states_de_compact;
      {$IFC PRERELEASE }
      debug_compact: boolean;
      {$ENDC }

   PROCEDURE heap_compact;

   FUNCTION init_compact_state: states_de_compact;

   FUNCTION retry_compact(VAR error_code: integer;
                          VAR local_state: states_de_compact): boolean;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************
```

```
000062

End of File -- Lines: 62 Characters: 1552
```

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : DBDECL1
*
*******************************************************************************

USES {$U+} DBDECL1;


INTRINSIC;

INTERFACE
   {$SETC OSBUILT := TRUE }

   USES
      {$IFC OSBUILT }
         {$U libsm/unitstd.obj} unitstd,
         {$U libdb/dbenv.obj } dbenv,
         {$U libos/syscall.obj } syscall,
         {$U libos/psyscall.obj } psyscall,
         {$U libin/intrlibp.obj } international;
      {$ELSEC }
         {$U OBJ:dbenv.obj } dbenv,
         {$U INTRLIB.obj } international;
      {$ENDC }

   {$SETC PRERELEASE := FALSE }
   {$SETC DOLLARD    := NOT fSymOk }
   {$SETC doTraceDB  := TRUE}
   {$SetC fTraceDB   := doTraceDB AND fTRACE}

   Const


      lt            = 0;
      gt            = 1;
      eq            = 2;
      wordsize      = 2;
      elemsize      = 16;
      halfelemsize  = 8;
      maxrsize      = 1000;
      ascending     = 0;
      descending    = 1;
      ftypnull      = 0;
      ftypint1      = 1;
      ftypint2      = 2;
      ftypint4      = 3;
      ftypint8      = 4;
      ftypfptr      = 5;
      ftyptick      = 6;
      ftyppack2     = 7;
      ftypstr       = 8;
      ftypvstr      = 9;
      ftypvnull     = 10;
      ftypsvstr     = 11;
      ftypsvnull    = 12;
      ftypes        = 12;
      nullint1      = - 128;
      nullint2      = - 32768;
      nullint4      = - 2147483648;
      fnilval       = - 1;
   dbleft        = 0;
   dbright       = 1;
   dbcentered    = 2;
   dbwordwrap    = 3;
   dbnullok      = 4;
   dbdescend     = 5;
   dbisvisible   = 6;
   none          = 0;
   ss            = 1;
   chk           = 2;
   zip           = 3;
   ph            = 4;
   dol           = 5;
   num           = 6;
   tim           = 7;
   dt            = 8;
   lets          = 9;
   tick          = 10;
   OEMtick       = 11;
   OEMFlets      = 12;
   OEMSVlets     = 13;
   OEMFGen       = 14;
   OEMVGen       = 15;
   OEMSVGen      = 16;
   ecstrBnd      = 40;

TYPE
   short         = - 32767..32767;
   ptrinteger    = ^integer;
   ptrlongint    = ^longint;
   fileptr       = longint;
   ptrfileptr    = ^fileptr;
   ticket        = RECORD
                      high: integer;
                      low: longint;
                   END;
   ptrticket     = ^ticket;
   fstring       = PACKED ARRAY [0..0] OF char;
   ptrfstring    = ^fstring;
   fint8         = RECORD
                      high: longint;
                      low: longint;
                   END;
   ptrfint8      = ^fint8;
   vfld          = RECORD
                      size: integer;
                      offset: integer;
                   END;
   ptrvfld       = ^vfld;
   svfld         = PACKED RECORD
                      offset: 0..1023;
                      size: 0..63;
                   END;
   ptrsvfld      = ^svfld;
   bitrange      = 0..15;
   elemset       = SET OF bitrange;
   bits          = ARRAY [0..0] OF elemset;
   ptrbits       = ^bits;
   fieldrange    = ftypnull..ftypes;
   ptr_field     = RECORD
                      CASE fieldrange OF
                         ftypnull, ftypint1, ftyppack2:
                            (ptr: ptrdata);
                         ftypint2:
                            (pint2: ptrinteger);
                         ftypint4:
                                 (pint4: ptrlongint);
                              ftypint8:
                                 (pint8: ptrfint8);
                              ftypfptr:
                                 (pfptr: ptrfileptr);
                              ftyptick:
                                 (ptick: ptrticket);
                              ftypstr:
                                 (pstr: ptrfstring);
                              ftypvstr, ftypvnull:
                                 (pvfld: ptrvfld);
                              ftypsvstr, ftypsvnull:
                                 (psvfld: ptrsvfld)
                        END;
        bitoffset     = RECORD
                           nbits: bitrange;
                           sbit: bitrange;
                        END;
        flddesc       = RECORD
                           offset: integer;
                           direction: byte;
                           ftype: byte;
                           CASE boolean OF
                              false:
                                 (size: integer);
                              true:
                                 (bo: bitoffset)
                        END;
        recptr        = ARRAY [0..0] OF flddesc;
        ptrrecptr     = ^recptr;
        V0header      = RECORD
                           space: integer;
                           version: integer;
                           nfields: short;
                           nkeys: short;
                           nsort: short;
                           keybytes: short;
                           recsize: integer;
                           flags: elemset;
                        END;
        header        = RECORD
                           space: integer;
                           version: integer;
                           nfields: short;
                           nkeys: short;
                           nsort: short;
                           keybytes: short;
                           recsize: integer;
                           flags: elemset;
                           MarketCode: integer;
                        END;
        checkinfo     = ARRAY [1..5] OF integer;
        ecstr         = string[ecstrBnd];
        ptrdbfield    = ^dbfield;
        dbfield       = RECORD
                           fldsize: integer;
                           name: short;
                           display: short;
                           highval: short;
                           lowval: short;
                           defval: short;
                           editcheck: checkinfo;
                           fldtype: integer;
                           options: elemset;
                           reptype: short;
                           offset: integer;
```

```
                          CASE boolean OF
                             false:
                                (size: integer);
                             true:
                                (bo: bitoffset);
                       END;
       monthstr      = string[9];
       longstr       = string[255];

    PROCEDURE movelg(p1, p2: ptrdata; nbytes: integer);

    PROCEDURE moverg(p1, p2: ptrdata; nbytes: integer);

    PROCEDURE movelf(p1, p2: ptrdata; nbytes: integer);

    PROCEDURE moverf(p1, p2: ptrdata; nbytes: integer);

    FUNCTION extract(pbyte: ptrdata; nbit, sbit: integer): integer;

    PROCEDURE deposit(value: integer; pbyte: ptrdata; nbit, sbit: integer);

    FUNCTION mult424(m4: fileptr; m2: integer): fileptr;

    FUNCTION comp44(s1, s2: fileptr): integer;

    FUNCTION comphand(VAR s1, s2: fint8): integer;

    FUNCTION div424(m4: fileptr; m2: integer; VAR r2: integer): fileptr;

    PROCEDURE setset(pset: ptrdata; lastelement, nelements: integer);

    PROCEDURE clearset(pset: ptrdata; elements: integer);

    PROCEDURE clearbit(pset: ptrdata; bit: integer);

    FUNCTION inpset(pset: ptrdata; bit: integer): boolean;

    PROCEDURE setbit(pset: ptrdata; bit: integer);

    FUNCTION pintersect(pset1, pset2: ptrbits; setelements: integer): boolean;

    FUNCTION compare(desc1, desc2: ptrrecptr; nfields: integer; prec1,
                     prec2: ptrdata): integer;

    PROCEDURE extractkey(pdesc1, pdesc2: ptrrecptr; nflds: integer;
                         pentry: ptrdata; pwhere: ptrdata; vflds: integer);

    FUNCTION findchanges(pdesc1: ptrrecptr; nfields: integer; pbefore,
                         pafter: ptrdata; pset: ptrbits): boolean;

    PROCEDURE GetRepType(VAR result: integer; DataType: INTEGER;
                         CheckArray: CheckInfo; VAR RepType: Short;
                         VAR Size: INTEGER);

    PROCEDURE ValidRepType(VAR result: integer; DataType: INTEGER;
                           CheckArray: CheckInfo; RepType: Short;
                           Size: INTEGER);

 ********************************************************************************
 *
 *                             THAT'S ALL FOLKS ...
 *
 ********************************************************************************

```

End of File -- Lines: 257 Characters: 8002

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : DBENV
*
*******************************************************************************

USES {$U+} DBENV;


INTRINSIC;

INTERFACE

   TYPE
      byte          = - 128..127;
      data          = ARRAY [0..0] OF byte;
      ptrdata       = ^data;
      hnddata       = ^ptrdata;
      intdata       = ARRAY [0..0] OF integer;
      ptrintdata    = ^intdata;
      hndintdata    = ^ptrintdata;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 28 Characters: 684

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

```
******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : EVENTS
*
******************************************************************************

USES {$U+} EVENTS;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libos/SysCall   } SysCall,
      {$U libos/PSysCall  } PSysCall,
      {$U libhw/HWInt     } HWInt,
      {$U libsm/UnitStd   } UnitStd,
      {$U libsm/UnitHz    } UnitHz,
      {$U libqd/Storage   } Storage,
      {$U libpm/PmDecl    } PmDecl,
      {$U libpm/PmM       } PmM,
      {$U libqd/QuickDraw } QuickDraw,
      {$U libfm/FontMgr   } FontMgr;

      {$SetC wmOS      := TRUE   }
      {$SetC wmDebug   := FALSE }
      {$SetC wmSymbols := fSymOk }
      {$SetC wmJournal := TRUE  }
      {$SetC wmMonkey  := FALSE  }
      {$SetC wmScrn2File := TRUE }
      {$SETC doTraceWM := TRUE}
      {$SetC fTraceWM := doTraceWM AND fTRACE}

   CONST
      nilEvent      = 0;
      buttonDown    = 1;
      buttonUp      = 2;
      keyDown       = 3;
      folderActivate = 4;
      folderDeactivate = 5;
      folderUpdate  = 6;
      folderMoved   = 7;
      filerEvent    = 8;
      abortEvent    = 9;
      diedEvent     = 10;
      private1      = 11;
      private2      = 12;
      private3      = 13;
      diskEvent     = 14;
      catalogEvent  = 15;
      disk1InCode   = 1;
      disk1OutCode  = 2;
      disk2InCode   = 3;
      disk2OutCode  = 4;
      paraCode      = 5;
      buttonCode    = 6;
      mouseCode     = 7;
      powerCode     = 8;
      microInCode   = 11;
      micro1InCode  = 12;
   micro2InCode  = 13;
   micro3InCode  = 14;
   microOutCode  = 15;
   micro1Outcode = 16;
   micro2OutCode = 17;
   micro3OutCode = 18;
   optionCode    = 104;
   shiftCode     = 126;
   commandCode   = 127;
   nilUserData   = 0;
   nilProcess    = 0;
   whyNot        = 0;
   whyClick      = 300;
   whyClose      = 301;
   whyDisk       = 302;
   whyDied       = 303;
   whyFind       = 304;

TYPE
   KeyCode       = 0..127;
   KeySet        = SET OF KeyCode;
   ProcessId     = LongInt;
   WindowPtr     = GrafPtr;
   EventHandle   = ^EventPtr;
   EventKind     = INTEGER;
   EventPtr      = ^EventRecord;
   EventRecord   = RECORD
                      who: WindowPtr;
                      what: EventKind;
                      where: Point;
                      when: LongInt;
                      why: INTEGER;
                      shiftKey: BOOLEAN;
                      alphaKey: BOOLEAN;
                      codeKey: BOOLEAN;
                      appleKey: BOOLEAN;
                      mouseKey: BOOLEAN;
                      repeatKey: BOOLEAN;
                      keyCap: KeyCode;
                      ascii: CHAR;
                      toProcess: ProcessId;
                      fromProcess: ProcessId;
                      fromFolder: WindowPtr;
                      userData: LongInt;
                   END;

VAR
   wmHeap:       Thz;
   deskPort:     GrafPtr;
   menuFolder:   WindowPtr;
   alertFolder:  WindowPtr;
   dialogFolder: WindowPtr;
   filerFolder:  WindowPtr;
   scrapFolder:  WindowPtr;
   activeFolder: WindowPtr;
   keyWindow:    WindowPtr;
   filerProcess: ProcessId;
   scrapProcess: ProcessId;
   activeProcess: ProcessId;
   keyProcess:   ProcessId;
   clickDelay:   LongInt;
   caretOnTime:  LongInt;
   caretOffTime: LongInt;
   eventDebug:   BOOLEAN;
   trapKeys:     KeySet;
   keyTrapProc:  ProcPtr;
```

```
     trapPrcs:     ProcessId;
     _CA_StartFlag: BOOLEAN;
     queueLocked:  BOOLEAN;
     recordFlag:   BOOLEAN;
     playFlag:     BOOLEAN;
     monkeyFlag:   BOOLEAN;
     monkeyWorld:  BOOLEAN;
     wmspare1:     LONGINT;
     wmspare2:     LONGINT;
     wmspare3:     LONGINT;
     wmspare4:     LONGINT;
     wmspare5:     LONGINT;

  FUNCTION Abort: BOOLEAN;

  FUNCTION Button: BOOLEAN;

  PROCEDURE CheckEvents(updateOK: BOOLEAN);

  PROCEDURE DeleteEvent(evHandle: EventHandle);

  FUNCTION ElapsTime: LongInt;

  FUNCTION EventAvail: BOOLEAN;

  FUNCTION FirstEvent: EventHandle;

  PROCEDURE GetEvent(VAR event: EventRecord);

  PROCEDURE GetEvFrom(fromProcess: ProcessID; VAR event: EventRecord);

  PROCEDURE GetKeys(VAR keys: KeySet);

  PROCEDURE GetMouse(VAR pt: Point);

  FUNCTION ImActive: BOOLEAN;

  PROCEDURE InitEvents;

  PROCEDURE LetOthersRun;

  FUNCTION NextEvent(evHandle: EventHandle): EventHandle;

  FUNCTION PeekEvent(VAR event: EventRecord): BOOLEAN;

  PROCEDURE PrintScreen;

  PROCEDURE PushEvent(VAR event: EventRecord);

  PROCEDURE SendEvent(VAR event: EventRecord; toPrcs: ProcessId);

  FUNCTION StillDown: BOOLEAN;

  FUNCTION Time: LongInt;

  PROCEDURE WakeUpHead;
  {$IFC wmJournal }

  PROCEDURE StartPlayback(fileName: PathName; journalMode: INTEGER);

  PROCEDURE StartRecording(fileName: PathName; journalMode: INTEGER);

  PROCEDURE StopPlayback;

  PROCEDURE StopRecording;
  {$ENDC }
```

```

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

End of File -- Lines: 200 Characters: 5104
```

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FEDEC
*
********************************************************************************

USES {$U+} FEDEC;


INTRINSIC;

INTERFACE

   USES
      {$U libsm/unitstd.obj} UnitStd,
      {$U libsm/unithz.obj} UnitHz,
      {$U libqd/QuickDraw} QuickDraw,
      {$U libfm/FontMgr} FontMgr,
      {$U libqd/Storage} Storage,
      {$U libdb/dbenv} dbenv;
      {$SETC doTraceFE := TRUE}
      {$SetC fTraceFE := doTraceFE AND fTRACE}

   CONST
      fldFull       = 1;
      fldProtected  = 2;
      nullSelect    = 3;
      invScrapType  = 4;
      invOpType     = 5;
      noRoomForRuns = 6;
      invalidEdit   = 7;
      FEallocationFailed = 8;
      FENotUndoable = 9;
      left          = 1;
      right         = 2;
      center        = 3;
      setNormal     = 1;
      setUnderline  = 2;
      setBold       = 3;
      setItalics    = 4;
      setOutline    = 5;
      setShadow     = 6;
      setFont       = 7;
      setToModern   = 8;
      setToClassic  = 9;
      noOp          = 0;
      cutOp         = 1;
      copyOp        = 2;
      pasteOp       = 3;
      chInputOp     = 4;
      clearOp       = 5;
      formatOp      = 6;
      valueToFormulaOp = 7;
      formulaToValueOp = 8;
      MaxFieldLngth = 1000;

   TYPE
      Tstr          = string[50];
      Ttriple       = ^Tstr;
      interval      = RECORD
                           lpFst: integer;
                           lpLim: integer;
                        END;
        TtySel        = (tySelPt, tySelWd, tySelPar);
        TtyHilight    = (tyHiNil, tyHiInvert, tyHiCaret, tyHiDimInvert,
                         tyHiDimCaret);
        txSel         = RECORD
                           int: interval;
                           tySel: TtySel;
                           tyHilight: TtyHilight;
                        END;
        hndRuns       = ^ptrRuns;
        ptrRuns       = ^runs;
        run           = RECORD
                           lpFst: integer;
                           font: integer;
                           face: Style;
                        END;
        runs          = ARRAY [0..0] OF run;
        field         = RECORD
                           coords: Rect;
                           maxLen: integer;
                           growLen: integer;
                           curLen: integer;
                           align: QDByte;
                           drawPad: QDByte;
                           curValue: hndData;
                           maxFmts: integer;
                           growFmts: integer;
                           curFmts: integer;
                           fmtInfo: hndRuns;
                           protect: boolean;
                        END;
        ptrField      = ^field;
        hndField      = ^ptrField;
        fieldState    = RECORD
                           select: txSel;
                           anchor: txSel;
                           selectOn: boolean;
                           visCaret: boolean;
                           selectRect: Rect;
                           bsPtr: integer;
                           space: interval;
                           valid: boolean;
                           validLp: integer;
                           newSel: boolean;
                           changed: boolean;
                        END;
        ptrFState     = ^fieldState;
        hndFState     = ^ptrFState;
        undoInfo      = RECORD
                           curField: hndField;
                           curFS: hndFState;
                           oldSelFld: hndField;
                           oldSelFS: hndFState;
                           select: txSel;
                           endselect: txSel;
                           anchor: txSel;
                           endanchor: txSel;
                           oldValFld: hndField;
                           oldValFS: hndFState;
                           curLen: integer;
                           curValue: hndData;
                           curFmts: integer;
                           fmtInfo: hndRuns;
                           lastOp: integer;
                        END;
```

```
      ptrUndoInfo   = ^undoInfo;
      hndUndoInfo   = ^ptrUndoInfo;
      HndScrap      = ^ptrScrap;
      PtrScrap      = ^ScrapRec;
      ScrapRec      = RECORD
                         Scrapfld: Field;
                         isWord: boolean;
                      END;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 142 Characters: 4377

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FIELDEDI
*
*******************************************************************************

USES {$U+} FIELDEDI;


INTRINSIC;

INTERFACE

   USES
      {$U libsm/unitStd } UnitStd,
      {$U libsm/unitHz} UnitHz,
      {$U libqd/Storage} Storage,
      {$U libqd/QuickDraw} QuickDraw,
      {$U libfm/FontMgr} FontMgr,
      {$U libos/SysCall} SysCall,
      {$U libwm/Events} Events,
      {$U libwm/Folders} Folders,
      {$U libsb/WMLStd} WMLStd,
      {$U libsb/WMLSb} WMLSb,
      {$U libdb/dbenv} dbenv,
      {$U libfe/FEDec} FEDec,
      {$U libsu/Scrap} Scrap,
      {$U libsu/UnitFmt} UnitFmt,
      {$U libpm/PmDecl} PmDecl,
      {$U libpr/PrStdInfo} PrStdInfo,
      {$U libsu/UnitCs} UnitCs,
      {$U libfe/Fldut} FldUT;
      {$SETC FLDDEBUG := FDBGOK }
      {$SETC FLDSYMBOLS := FSYMOK }

   CONST
      MaxFontNumber = 24;
      {$IFC NOT FLDDEBUG }
      fldTest       = false;
      {$ENDC }

   VAR
      {$IFC FLDDEBUG }
      fldTest:      boolean;
      {$ENDC }
      ToModern, ToClassic: ARRAY [0..MaxFontNumber] OF - 128..127;
      ScrapHnd:     HndScrap;
      fldHeap:      THz;
      fmtSize:      integer;
      uInfo:        undoInfo;
      SubFont:      Boolean;
      SubDev:       Integer;

   PROCEDURE FEInit(hz: THz; passtrpl: Ttriple; VAR errnum: integer);

   PROCEDURE FETerminate;
   {$IFC FLDDEBUG}

   PROCEDURE SetFldTest(testOn: boolean);

   PROCEDURE DumpFldInfo(hf: hndField; hfs: hndFState);
```

```
{$ENDC}

PROCEDURE LdFESeg;

PROCEDURE UseSubstituteFont(Substitute: Boolean; DevtoSub: Integer);

PROCEDURE SetFCoords(hf: hndField; hfs: hndFState; c: Rect);

PROCEDURE SetFAlign(hf: hndField; hfs: hndFState; a: byte);

PROCEDURE SetFPad(hf: hndField; hfs: hndFState; pad: byte);

PROCEDURE SetFProtect(hf: hndField; hfs: hndFState; p: boolean);

PROCEDURE InitFState(hf: hndField; hfs: hndFState);

FUNCTION CreateField(c: Rect; iLen, gLen: integer; a: byte; pad: byte; iFmts,
                     gFmts: integer; fontNum: integer; p: boolean): hndField;

PROCEDURE RemoveField(hf: hndField);

FUNCTION FmtFromLp(hf: hndField; hfs: hndFState; lp: integer): integer;

PROCEDURE FmtFromInt(hf: hndField; hfs: HndFState; Int: interval;
                     VAR Plain: boolean; VAR result: run);

PROCEDURE SetRunInterval(hf: hndField; hfs: hndFState; fontNum: integer;
                         format: integer; int: interval;
                         VAR errNum: integer);

PROCEDURE SetRunFormat(hf: hndField; hfs: hndFState; fontNum: integer;
                       format: integer; VAR errNum: integer);

PROCEDURE UpdateRun(hf: hndField; hfs: hndFState; fontNum: integer;
                    format: integer; VAR errNum: integer);

PROCEDURE VertCoords(hf: hndField; VAR ascent, descent: integer);

PROCEDURE MoveField(hf: hndField; hfs: hndFState; dx: Point);

PROCEDURE DrawField(hf: hndField; hfs: hndFState; showSelect,
                    erase: boolean);

PROCEDURE DrawFldAt(hf: hndField; hfs: hndFState; dx: Point; showSelect,
                    erase: boolean);

PROCEDURE SetSel(hf: hndField; hfs: hndFState; lpFst: integer;
                 lpLim: integer; selType: TtySel);

PROCEDURE SelectAll(hf: hndField; hfs: hndFState; VAR t: integer);

PROCEDURE ExitField(hf: hndField; hfs: hndFState);

PROCEDURE CompactField(hf: hndField);

PROCEDURE ErrorSelect(hf: hndField; hfs: hndFState; lpFst: integer;
                      lpLim: integer; VAR t: integer);

PROCEDURE HilightSel(hf: hndField; hfs: hndFState; fShow: boolean);

PROCEDURE ChangeSelHilight(hf: hndField; hfs: hndFState; toDim: boolean);

FUNCTION LpFromX(x: integer; hf: hndField; hfs: hndFState;
                 seltype: ttysel): integer;

PROCEDURE LwFromLp(orig: integer; hf: hndField; hfs: hndFState;
                      VAR lWord: interval; VAR FoundWord: boolean);

   FUNCTION CompressedLp(lp: integer; hf: hndField; hfs: hndFState;
                         VAR valid: boolean): integer;

   FUNCTION ExpandedLp(lp: integer; hf: hndField; hfs: hndFState): integer;

   PROCEDURE RepSelect(lp: integer; hf: hndField; hfs: hndFState);

   PROCEDURE Select(dxy: Point; hf: hndField; hfs: hndFState; VAR n: Rect;
                    VAR t: integer);

   PROCEDURE RepWordSelect(int: interval; hf: hndField; hfs: hndFState);

   PROCEDURE SelectWord(dxy: Point; hf: hndField; hfs: hndFState; VAR n: rect;
                        VAR t: integer);

   PROCEDURE RepGrowSel(chlp: integer; hf: hndField; hfs: hndFState);

   PROCEDURE GrowSel(dxy: Point; hf: hndField; hfs: hndFState; VAR n: Rect);

   PROCEDURE CaretErased(hf: hndField; hfs: hndFState);

   PROCEDURE StartBlink(VAR t: integer);

   PROCEDURE BlinkCaret(hf: hndField; hfs: hndFState; VAR t: integer);

   PROCEDURE CaretOn(hf: hndField; hfs: hndFState);

   PROCEDURE InsCh(ch: char; hf: hndField; hfs: hndFState; VAR errNum: integer);

   PROCEDURE Backspace(hf: hndField; hfs: hndFState; VAR errNum: integer);

   PROCEDURE Forwardspace(hf: hndField; hfs: hndFState; VAR errNum: integer);

   PROCEDURE BackWord(hf: hndField; hfs: hndFState; VAR errNum: integer);

   PROCEDURE ForwardWord(hf: hndField; hfs: hndFState; VAR errNum: integer);

   PROCEDURE DrawScrap(VAR errnum: integer);

   PROCEDURE CutCopyField(hf: hndField; hfs: hndFState; fCut: boolean;
                          fIntoScrap: boolean; VAR errNum: integer);

   PROCEDURE PasteField(hf: hndField; hfs: hndFState; VAR errNum: integer;
                        pasteFmt: boolean; PasteLimit: integer);

   PROCEDURE UndoEdit(hf: hndField; hfs: hndFState; VAR errNum: integer);

   PROCEDURE ClearField(hf: hndField; hfs: hndFState; VAR errNum: integer);

   FUNCTION qualifies(lp: integer; hf: hndfield; hfs: hndfstate): boolean;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 186 Characters: 5951

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FILEIO
*
*******************************************************************************

USES {$U+} FILEIO;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/PASDEFS.OBJ *) PasDefs,
         (*$U OBJ/MEMMAN.OBJ  *) Memman;
      {$ELSEC }
         (*$U OBJ:PASDEFS.OBJ *) PasDefs,
         (*$U OBJ:MEMMAN.OBJ  *) Memman;
      {$ENDC }

   CONST


      MaxBuffTop    = 32767;
      BuffBlock     = 512;
      Null          = '';
      CtrlC         = 3;
      Left          = 8;
      Down          = 10;
      Up            = 11;
      Right         = 12;
      CR            = 13;
      Esc           = 27;

   TYPE
      Byte          = 0..255;
      SetOfChar     = SET OF Char;
      Ptr           = ^Integer;
      FileByte      = - 128..127;
      Buffer        = ARRAY [0..MaxBuffTop] OF FileByte;
      BuffPtr       = ^Buffer;
      FileHandle    = ^FileDesc;
      FileDesc      = RECORD
                           Info: FILE;
                           OldFile: boolean;
                           NewLastBlk: boolean;
                           Buff: BuffPtr;
                           MaxBlks, CurBlks: Integer;
                           BlockNr0: Integer;
                           BuffTop, BuffIdx: Integer;
                           BuffChd, ReadEOF: boolean;
                           LastBlk: Integer;
                        END;

   PROCEDURE InitFile(VAR FilePtr: FileHandle; nBlocks: Integer);

   FUNCTION IAnd(I, J: Integer): Integer;

   FUNCTION IOr(I, J: Integer): Integer;

   FUNCTION IXOr(I, J: Integer): Integer;

   FUNCTION INot(I: Integer): Integer;

   FUNCTION PeekB(P: Ptr; SkipB: Integer): Byte;

   PROCEDURE PokeB(P: Ptr; SkipB: Integer; B: Byte);

   PROCEDURE ToHex(P: Ptr; SkipN, NrN: Integer; VAR S: LString);

   PROCEDURE WriteHex(VAR F: Text; P: Ptr; SkipN, NrN: Integer);

   PROCEDURE OpenFile(VAR FilePtr: FileHandle; FileName: LString;
                      NewFile: boolean);

   PROCEDURE ZeroFileEnd(FilePtr: FileHandle);

   PROCEDURE CloseFile(FilePtr: FileHandle; Save: boolean);

   PROCEDURE GetFilePtr(FilePtr: FileHandle; VAR BytePtr: LongInt);

   PROCEDURE SetFilePtr(FilePtr: FileHandle; BytePtr: LongInt);

   PROCEDURE SkipBytes(FilePtr: FileHandle; NrBytes: LongInt);

   FUNCTION FileEOF(FilePtr: FileHandle): boolean;

   PROCEDURE CopySeq(InFile, OutFile: FileHandle; NrBytes: LongInt);

   PROCEDURE GetSeq(FilePtr: FileHandle; Stuff: Ptr; NrBytes: LongInt);

   PROCEDURE GetByte(FilePtr: FileHandle; VAR B: Byte);

   PROCEDURE GetWord(FilePtr: FileHandle; VAR W: Integer);

   PROCEDURE GetLong(FilePtr: FileHandle; VAR L: LongInt);

   PROCEDURE PutSeq(FilePtr: FileHandle; Stuff: Ptr; NrBytes: LongInt);

   PROCEDURE PutByte(FilePtr: FileHandle; B: Byte);

   PROCEDURE PutWord(FilePtr: FileHandle; W: Integer);

   PROCEDURE PutLong(FilePtr: FileHandle; L: LongInt);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 113 Characters: 3042

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

```
******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FILERCOM
*
******************************************************************************

USES {$U+} FILERCOM;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libos/SysCall    } SysCall,
      {$U libos/PSysCall   } PSysCall,
      {$U libsm/UnitStd    } UnitStd,
      {$U libsm/UnitHz     } UnitHz,
      {$U libqd/Storage    } Storage,
      {$U libqd/QuickDraw  } QuickDraw,
      {$U libqd/GrafUtil   } GrafUtil,
      {$U libsb/WmlStd     } WmlStd,
      {$U libsb/WmlSb      } WmlSb,
      {$U libfm/FontMgr    } FontMgr,
      {$U libwm/Events     } Events,
      {$U libwm/windows    } Windows,
      {$U libwm/folders    } Folders,
      {$U libsu/Scrap      } Scrap;
      {$SETC fcDebug = fDbgOk }
      {$SETC fcSymbols = fSymOk }

   CONST
      scrapRef      = 8;
      cmAppInfo     = 1268;
      cmIconInfo    = 1269;
      nameSeperator = '<';
      fceNoErrors   = 0;
      fceAborted    = 4033;
      fceBadEventType = 4025;
      fceBadReason  = 4026;
      fceCantRead   = 4027;
      fceCantWrite  = 4028;
      fceInUse      = 4029;
      fceNoMemory   = 4030;
      fceOutOfDiskSpace = 4031;
      fceBadLDSN    = 4032;
      fceBadPassword = 4033;
      fceDuplicateName = 890;
      fceInvalidParam = 971;
      fceNameNotFound = 972;
      cmdClose      = 1001;
      cmdClosAll    = 1002;
      deskTool      = 0;
      writeTool     = 1;
      graphTool     = 2;
      matrixTool    = 3;
      drawTool      = 4;
      listTool      = 5;
      ballsTool     = 6;
      manualTool    = 7;
      projectTool   = 8;
   termEmTool    = 10;
   configTool    = 11;
   calcTool      = 12;
   clockTool     = 13;
   MwriteTool    = 14;
   MpaintTool    = 15;
   MdrawTool     = 16;
   maxKnownTool  = 16;
   iconWidth     = 48;
   iconHt        = 32;
   nilKind       = 0;
   fileKind      = 1;
   deskKind      = 2;
   folderkind    = 3;
   docKind       = 4;
   docPad        = 5;
   trashKind     = 6;
   printKind     = 7;
   calcKind      = 8;
   trayKind      = 9;
   computerKind  = 10;
   comp2Kind     = 11;
   inBox1Kind    = 12;
   inBox2Kind    = 13;
   outBox1Kind   = 14;
   outBox2Kind   = 15;
   folderPad     = 16;
   clipBdKind    = 17;
   clockKind     = 19;
   letterKind    = 20;
   letterPad     = 21;
   toolKind      = 24;
   diskKind      = 25;
   disk1Kind     = 26;
   disk2Kind     = 27;
   drawerKind    = 28;
   profileKind   = 29;
   priamKind     = 30;
   slotKind      = 31;
   slot1Kind     = 32;
   slot2Kind     = 33;
   slot3Kind     = 34;
   MdocKind      = 35;
   MdiskKind     = 36;
   lastKind      = 36;

TYPE
   FilingCmd     = LONGINT;
   FilerOp       = (fcClose, fcCopy, fcDfClose, fcNone, fcPrint, fcPut,
                    fcResume, fcShred, fcSuspend, fcTerminate,
                    fcNameToPrefix, fcPrefixToName, fcDupIcon, fcMakeTool);
   FReply        = (dfClosed, dfNotClosed, docClosd, docNotClosed, docXfered,
                    docNotXfered, InitFailed);
   FReason       = (allOK, badData, cantConvert, cantRead, cantWrite,
                    dirtyDoc, internalError, needToConvert, newerDoc,
                    noDiskSpace, noMemory, noMoreDocs, okButNoMore,
                    docPutBack, aUserAbort, wrongPassword);
   FilerExt      = RECORD
                      theFlrOp: FilerOp;
                      theErr: INTEGER;
                      theOffset: INTEGER;
                      theDf: INTEGER;
                      thePassword: E_Name;
                      thePrefix: Pathname;
                      theResult: Pathname;
                   END;
   FCopyOp       = (fcDocCopy, fcDocMove, fcDocBackup);
   ObjectKind    = nilKind..lastKind;
   TIconRef      = RECORD
                      kind: ObjectKind;
                      toolID: LONGINT;
                      userName: STRING[80];
                      password: E_Name;
                      diskName: Pathname;
                   END;
   PIconRef      = ^TIconRef;
   HIconRef      = ^PIconRef;
   TAppInfo      = RECORD
                      window: WindowPtr;
                      proc: INTEGER;
                   END;
   PAppInfo      = ^TAppInfo;
   HAppInfo      = ^PAppInfo;
   hFilerExt     = ^pFilerExt;
   pFilerExt     = ^FilerExt;
   ReplyPtr      = ^Reply;
   Reply         = RECORD
                      theReply: FReply;
                      theReason: FReason;
                   END;

VAR
   iconData:     ARRAY [ObjectKind] OF CHAR;
   iconMask:     ARRAY [ObjectKind] OF CHAR;
   iconOverlay:  ARRAY [deskTool..maxKnownTool] OF CHAR;
   iconBoxes:    ARRAY [ObjectKind] OF Rect;
   tinyData:     ARRAY [ObjectKind] OF CHAR;
   tinyMask:     ARRAY [ObjectKind] OF CHAR;
   tinyOverlay:  ARRAY [deskTool..maxKnownTool] OF CHAR;

PROCEDURE CopyDoc(VAR error: INTEGER; fromPrefix, toPrefix: Pathname;
                  useLdsn: INTEGER; theOp: FCopyOp; VAR docSize: LONGINT);

PROCEDURE DoFilingCmd(whichCmd: FilingCmd);

PROCEDURE GetAddParms(VAR error: INTEGER; theEvent: EventRecord;
                      VAR theFilerExt: FilerExt);

PROCEDURE TellFiler(VAR error: INTEGER; what: FReply; why: FReason;
                    myFolder: WindowPtr);

PROCEDURE NameToPrefix(VAR err, offset: INTEGER; myFolder: WindowPtr;
                       name: Pathname; VAR prefix: Pathname);

PROCEDURE PrefixToName(VAR err, offset: INTEGER; myFolder: WindowPtr;
                       prefix: Pathname; VAR name: Pathname);

PROCEDURE StartIconRef(VAR err: INTEGER; myWindow: WindowPtr);

PROCEDURE AddIconRef(iconInfo: TIconRef);

PROCEDURE EndIconRef;

PROCEDURE DrawRefScrap(VAR err: INTEGER);

PROCEDURE ReadIconRefs(VAR err: INTEGER; VAR proc: ProcessID;
                       VAR window: WindowPtr; VAR identity: TiconRef;
                       IDProc: TProc);

PROCEDURE CopyDiskfile(VAR err: INTEGER; source, destination: Pathname;
                       bufrAdrs, bufrSize: LONGINT; theOp: FCopyOp;
                       VAR osErr: INTEGER);

   PROCEDURE InitFC;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 202 Characters: 6379

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FLDUT
*
********************************************************************************

USES {$U+} FLDUT;


INTRINSIC;

INTERFACE

   USES
      {$U libsm/unitstd.obj} UnitStd,
      {$U libsm/unithz.obj} UnitHz,
      {$U libqd/QuickDraw} QuickDraw,
      {$U libfm/FontMgr} FontMgr,
      {$U libqd/Storage} Storage,
      {$U libos/SysCall} SysCall,
      {$U libwm/Events} Events,
      {$U libdb/dbenv} dbenv,
      {$U libfe/FEDec} FEDec,
      {$U libpm/pmdecl} pmdecl,
      {$U libpr/PrStdInfo} PrStdInfo,
      {$U libsu/unitfmt.obj} Unitfmt,
      {$U libsu/unitcs.obj} UnitCs,
      {$U libsu/scrap.obj} Scrap;
      {$SETC FLDUTDBG := FDBGOK}

   TYPE
      PFldUT        = ^TFldUT;
      TFldUT        = RECORD
                         cs: Tcs;
                         UTfield: hndfield;
                      END;

   FUNCTION CreateFldUt(StreamHeap: THz): TB;

   PROCEDURE SeqLpdFld(Lpd: TLpd; VAR achad: Tachad);
   {$IFC FLDUTDBG}

   PROCEDURE PxFldUT(hcs: thcs);
   {$ENDC}

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 51 Characters: 1275

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FMGRUTIL
*
*******************************************************************************

USES {$U+} FMGRUTIL;


{$SETC fmOS := TRUE}

INTRINSIC;

INTERFACE

   USES
      {$U libsm/UnitStd} UnitStd,
      {$U libsm/UnitHz} UnitHz,
      {$U libos/SysCall} SysCall,
      {$U libqd/QuickDraw} QuickDraw,
      {$U libfm/FontMgr} FontMgr,
      {$U libqd/Storage} Storage;
      {$SETC fmdebug := fdbgok}
      {$SETC fmSymbols := fsymok}

   FUNCTION GetIcon(fam: Tfam; cc: TCc; VAR rSrc: Rect; VAR bmSrc: BitMap;
                    VAR cError: TC): TF;

   PROCEDURE DrawIcon(fam: Tfam; rDst: Rect; cc: TCc; cmode: TC;
                      VAR cError: TC);

   PROCEDURE GetPattern(fam: Tfam; ccPat: TCc; VAR pat: Pattern;
                        VAR cError: TC);

   PROCEDURE GetCursor(fam: Tfam; ccData, ccMask: TCc; VAR curse: Cursor;
                       VAR cError: TC);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************


End of File -- Lines: 43 Characters: 1176
```

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FOLDERS
*
********************************************************************************

USES {$U+} FOLDERS;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libhw/HWInt       } HWInt,
      {$U libsm/UnitStd     } UnitStd,
      {$U libsm/UnitHz      } UnitHz,
      {$U libqd/Storage     } Storage,
      {$U libqd/QuickDraw   } QuickDraw,
      {$U libfm/FontMgr     } FontMgr,
      {$U libos/SysCall     } SysCall,
      {$U libwm/Events      } Events,
      {$IFC wmOS }
      {$U libpm/PmDecl      } PmDecl,
      {$U libpm/Pmm         } Pmm,
      {$ENDC }
      {$U libwm/Windows     } Windows;

   CONST
      maxWindWidth  = 720;
      maxWindHeight = 364;
      minWindWidth  = 50;
      minWindHeight = 14;
      noIcon        = 255;

   TYPE
      WindowInfo    = RECORD
                         visible: BOOLEAN;
                         hasTab: BOOLEAN;
                         okMove: BOOLEAN;
                         okGrow: BOOLEAN;
                         okFront: BOOLEAN;
                         okClose: BOOLEAN;
                         okActivate: BOOLEAN;
                      END;

   PROCEDURE BeginUpdate(folder: WindowPtr);

   PROCEDURE DialogHeight(height: INTEGER; fUpdate: BOOLEAN);

   PROCEDURE DisposeFolder(folder: WindowPtr);

   PROCEDURE EndUpdate(folder: WindowPtr);

   PROCEDURE FlushInput(window: WindowPtr);

   PROCEDURE FlushWindow(window: WindowPtr);

   PROCEDURE FlushProcess(process: ProcessID);

   PROCEDURE FolderSize(folder: WindowPtr; width, height: INTEGER;
                         fUpdate: BOOLEAN);

    FUNCTION GetFldrRefCon(folder: WindowPtr): LongInt;

    PROCEDURE GetFldrTitle(folder: WindowPtr; VAR title: Str255);

    PROCEDURE GetWindInfo(window: WindowPtr; VAR info: WindowInfo);

    PROCEDURE GiveControl(event: EventRecord);

    PROCEDURE GiveWindow(window: WindowPtr; toProcess: ProcessId);

    PROCEDURE HeDied(deadProcess: ProcessID);

    PROCEDURE HideFolder(folder: WindowPtr);

    PROCEDURE HiLiteTab(folder: WindowPtr; fHiLite: BOOLEAN);

    PROCEDURE ImDying;

    PROCEDURE InitWM;

    PROCEDURE InvalidateRgn(folder: WindowPtr; rgn: RgnHandle);

    PROCEDURE InvalRect(badRect: Rect);

    PROCEDURE InvalRgn(badRgn: RgnHandle);

    PROCEDURE LocateDialog(top, height: integer);

    PROCEDURE MakeFldrActive(folder: WindowPtr; reasonWhy: LongInt);

    PROCEDURE MakeTopActive;

    PROCEDURE MoveFolder(folder: WindowPtr; hGlobal, vGlobal: INTEGER);

    FUNCTION NewFolder(folderRect: Rect; title: Str255; visible: BOOLEAN;
                       behind: WindowPtr; refCon: LongInt; process: ProcessId;
                       whichIcon: INTEGER; overlay: INTEGER): WindowPtr;

    PROCEDURE OpenWM;

    PROCEDURE SetFldrTitle(folder: WindowPtr; title: Str255);

    PROCEDURE SetFldrRefCon(folder: WindowPtr; data: LongInt);

    PROCEDURE SetWindInfo(window: WindowPtr; VAR info: WindowInfo);

    PROCEDURE ShowFolder(folder: WindowPtr);

    PROCEDURE TakeControl(event: EventRecord; keepMenus, keepDialog: BOOLEAN);

    PROCEDURE TakeWindow(window: WindowPtr);

    PROCEDURE ValidRect(goodRect: Rect);

    PROCEDURE ValidRgn(goodRgn: RgnHandle);

    PROCEDURE WMClosePicture;

    PROCEDURE WMKillPicture(window: WindowPtr);

    PROCEDURE WMKillProcess(process: ProcessID);

    PROCEDURE WMOpenPicture(window: WindowPtr);

```

```
   PROCEDURE WMStartDoc(window: WindowPtr);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 135 Characters: 3537

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FONTMGR
*
*******************************************************************************

USES {$U+} FONTMGR;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libqd/QuickDraw} QuickDraw,
      {$U libsm/UnitStd} UnitStd,
      {$U libsm/UnitHz } UnitHz,
      {$U libos/SysCall} SysCall,
      {$U libos/PSysCall} PSysCall;
      {$SETC fmdebug := FALSE}
      {$SETC fmSymbols := fsymok}
      {$setC fTraceFM := fTrace}

   CONST
      ccMax         = 255;
      ccBS          = 8;
      ccHT          = 9;
      ccLF          = 10;
      ccVT          = 11;
      ccFF          = 12;
      ccCR          = 13;
      ccESC         = 27;
      ccPara        = 166;
      ccSec         = 164;
      ccCopy        = 169;
      ccReg         = 168;
      ccTM          = 170;
      ccDeg         = 161;
      ccDag         = 160;
      ccBul         = 165;
      ccPound       = 163;
      ccElip        = 201;
      ccSysApple    = 202;
      ccLisaLogo    = 0;
      ccCheck       = 142;
      ccApple       = 143;
      ccLreg        = 0;
      ccLalt        = 1;
      ccLdash       = 2;
      ccRreg        = 3;
      ccRalt        = 4;
      ccRdash       = 5;
      ccHreg        = 6;
      ccHalt        = 7;
      ccHdash       = 8;
      ccVreg        = 9;
      ccValt        = 10;
      ccVdash       = 11;
      ccG90f        = 12;
      ccG90c        = 13;
      ccG45f        = 14;
```

```
ccG45c        = 15;
ccWhite       = 16;
ccXLtGray     = 17;
ccLtGray      = 18;
ccMedGray     = 19;
ccGray        = 20;
ccDkGray      = 21;
ccBlack       = 22;
ccNil         = 255;
ccWhiteMask   = 255;
sysText       = 0;
wmfont        = 1;
sysPat        = 2;
sysCursor     = 3;
tile12        = 4;
tile18        = 5;
tile24        = 6;
p15Tile       = 7;
p12Tile       = 8;
p10Tile       = 9;
cent12        = 10;
cent18        = 11;
cent24        = 12;
p12cent       = 13;
p10cent       = 14;
elite         = 15;
courier       = 16;
boldface      = 17;
calcFont      = 18;
p20Tile       = 19;
marker        = 20;
tile7         = 21;
fIconFont     = 22;
sysLogo       = 23;
devScreen     = 0;
famLst        = 23;
substitution  = - 3024;
BigCache      = 98000;
HighCache     = 45000;
LowCache      = 30000;
{$IFC NOT fmdebug}
fdebug        = FALSE;
fdebug1       = FALSE;
fdebug2       = FALSE;
fdebug3       = FALSE;
fdebug4       = FALSE;
fdebug5       = FALSE;
fdebug6       = FALSE;
fdebug7       = FALSE;
fdebug8       = FALSE;
fdebug9       = FALSE;
fdebug10      = FALSE;
fdebug11      = FALSE;
fdebug12      = FALSE;
fdebug13      = FALSE;
fdebug14      = FALSE;
fdebug15      = FALSE;
fdebug16      = FALSE;
fdebug17      = FALSE;
fdebug18      = FALSE;
fdebug19      = FALSE;
fdebug20      = FALSE;
fdebug21      = FALSE;
fdebug22      = FALSE;
fdebug23      = FALSE;
fdebug24      = FALSE;
```

```
   fdebug25      = FALSE;
   fdebug26      = FALSE;
   fdebug27      = FALSE;
   fdebug28      = FALSE;
   fdebug29      = FALSE;
   fdebug30      = FALSE;
   fdebug31      = FALSE;
   fdebug32      = FALSE;
   fdebug33      = FALSE;
   fdebug34      = FALSE;
   fdebug35      = FALSE;
   fdebug36      = FALSE;
   fdebug37      = FALSE;
   fdebug38      = FALSE;
   fdebug39      = FALSE;
   fdebug40      = FALSE;
   {$ENDC}

TYPE
   TCc           = 0..255;
   TStr30        = STRING[30];
   TFam          = TC;
   TDev          = - 128..127;
   TEdev         = (edevScreen, edevDotMx, edevHDotMx, edevTyper, edevLaser,
                    edev1, edev2);
   TEwhcl        = (ewhcl20, ewhcl15, ewhcl12, ewhcl10, ewhclPS, ewhclMF,
                    ewhclMP, ewhclLCS);
   TEskcl        = (eskclMPPS, eskclMPFP, eskclMFR, eskclMFI, eskclApple,
                    eskclFQume, eskclPQume);
   TAdev         = PACKED RECORD
                      CASE BOOLEAN OF
                         TRUE:
                            (dev: TDev);
                         FALSE:
                            (fOffline: TF;
                             CASE edev: TEdev OF
                                edevScreen, edevLaser:
                                   ();
                                edevDotMx, edevHDotMx:
                                   (fHiRes: TF;
                                    fPortrait: TF);
                                edevTyper:
                                   (fSpoke: TF;
                                    ewhcl: TEwhcl);
                                edev1, edev2:
                                   ());
                    END;
   TFntid        = PACKED RECORD
                      fam: TFam;
                      seteface: Style;
                    END;
   TLfntid       = PACKED RECORD
                      fam: TFam;
                      seteface: Style;
                      CASE BOOLEAN OF
                         TRUE:
                            (dev: TDev);
                         FALSE:
                            (fOffline: TF;
                             CASE edev: TEdev OF
                                edevScreen, edevLaser:
                                   ();
                                edevDotMx, edevHDotMx:
                                   (fHiRes: TF;
                                    fPortrait: TF);
                                edevTyper:
                                      (fSpoke: TF;
                                       ewhcl: TEwhcl);
                                   edev1, edev2:
                                      ());
                      END;
      TWid          = 0..255;
      TMpccwid      = PACKED ARRAY [TCc] OF TWid;
      TPmpccwid     = ^TMpccwid;
      THmpccwid     = ^TPmpccwid;
      TPcharStyle   = ^TCharStyle;
      TCharstyle    = PACKED RECORD
                         hks: TH;
                         hmpccwid: THmpccwid;
                         bBold: TB;
                         bItalic: TB;
                         bUnderline: TB;
                         bShadow: TB;
                         cExtra: TC;
                      END;
      TMpefaceb     = ARRAY [StyleItem] OF TB;
      TModifier     = RECORD
                         seteface: Style;
                         mpefaceb: TMpefaceb;
                      END;

   VAR
      {$IFC fmdebug}
      fdebug:       TF;
      fdebug1:      TF;
      fdebug2:      TF;
      fdebug3:      TF;
      fdebug4:      TF;
      fdebug5:      TF;
      fdebug6:      TF;
      fdebug7:      TF;
      fdebug8:      TF;
      fdebug9:      TF;
      fdebug10:     TF;
      fdebug11:     TF;
      fdebug12:     TF;
      fdebug13:     TF;
      fdebug14:     TF;
      fdebug15:     TF;
      fdebug16:     TF;
      fdebug17:     TF;
      fdebug18:     TF;
      fdebug19:     TF;
      fdebug20:     TF;
      fdebug21:     TF;
      fdebug22:     TF;
      fdebug23:     TF;
      fdebug24:     TF;
      fdebug25:     TF;
      fdebug26:     TF;
      fdebug27:     TF;
      fdebug28:     TF;
      fdebug29:     TF;
      fdebug30:     TF;
      fdebug31:     TF;
      fdebug32:     TF;
      fdebug33:     TF;
      fdebug34:     TF;
      fdebug35:     TF;
      fdebug36:     TF;
      fdebug37:     TF;
      fdebug38:     TF;
      fdebug39:     TF;
      fdebug40:     TF;
      {$ENDC}
      InitialCacheSize: TL;
      MinCache:     TL;
      cRefnumLib:   TC;

   PROCEDURE FMinit(VAR cError: TC);

   PROCEDURE FMOpen(VAR cError: TC);

   PROCEDURE FMCloseLib(VAR cError: TC);

   PROCEDURE FMCacheSize(lCacheSize: TL; VAR cError: TC);

   FUNCTION FMAddFont(str30: TStr30; modifier: TModifier; dev: TDev;
                      VAR cError: TC): TFam;

   PROCEDURE FMDelfont(lfntid: TLfntid);

   FUNCTION FMSwapFont(VAR lfntid: TLfntid; VAR cError: TC): TPcharstyle;

   PROCEDURE FMLockFont(VAR lfntid: TLfntid; flock: TF; VAR cError: TC);

   FUNCTION FMFontMetrics(VAR lfntid: TLfntid; VAR finfo: FontInfo;
                          VAR cError: TC): TF;

   FUNCTION FMFontWidths(VAR lfntid: TLfntid; VAR hmpccwid: THmpccwid;
                         VAR dExtra: TB; VAR cError: TC): TF;

   PROCEDURE FMMapFont(VAR lfntid: TLfntid; VAR cError: TC);
   {$IFC fmdebug}

   PROCEDURE FMDebug(fdbg: TF);

   PROCEDURE FMDumpHeurTable;
   {$ENDC}

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 303 Characters: 8457

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : FPLIB
*
********************************************************************************

USES {$U+} FPLIB;


INTRINSIC;

INTERFACE

   {$C Copyright 1983, 1984, Apple Computer Inc. }

   CONST
      SIGDIGLEN     = 20;
      DECSTRLEN     = 255;

   TYPE
      Single        = real;
      Double        = ARRAY [0..3] OF integer;
      Comp          = ARRAY [0..3] OF integer;
      Extended      = ARRAY [0..4] OF integer;
      SigDig        = string[SIGDIGLEN];
      DecStr        = string[DECSTRLEN];
      Decimal       = RECORD
                         sgn: 0..1;
                         exp: integer;
                         sig: SigDig
                      END;
      Environ       = integer;
      RoundDir      = (TONEAREST, UPWARD, DOWNWARD, TOWARDZERO);
      RelOp         = (GT, LT, GL, EQ, GE, LE, GEL, UNORD);
      Exception     = (INVALID, UNDERFLOW, OVERFLOW, DIVBYZERO, INEXACT);
      NumClass      = (SNAN, QNAN, INFINITE, ZERO, NORMAL, DENORMAL);
      DecForm       = RECORD
                         style: (FLOATDECIMAL, FIXEDDECIMAL);
                         digits: integer
                      END;

   PROCEDURE AddS(x: Single; VAR y: Extended);

   PROCEDURE AddD(x: Double; VAR y: Extended);

   PROCEDURE AddC(x: Comp; VAR y: Extended);

   PROCEDURE AddX(x: Extended; VAR y: Extended);

   PROCEDURE SubS(x: Single; VAR y: Extended);

   PROCEDURE SubD(x: Double; VAR y: Extended);

   PROCEDURE SubC(x: Comp; VAR y: Extended);

   PROCEDURE SubX(x: Extended; VAR y: Extended);

   PROCEDURE MulS(x: Single; VAR y: Extended);

   PROCEDURE MulD(x: Double; VAR y: Extended);

```

```
  PROCEDURE MulC(x: Comp; VAR y: Extended);

  PROCEDURE MulX(x: Extended; VAR y: Extended);

  PROCEDURE DivS(x: Single; VAR y: Extended);

  PROCEDURE DivD(x: Double; VAR y: Extended);

  PROCEDURE DivC(x: Comp; VAR y: Extended);

  PROCEDURE DivX(x: Extended; VAR y: Extended);

  FUNCTION CmpX(x: Extended; r: RelOp; y: Extended): boolean;

  FUNCTION RelX(x, y: Extended): RelOp;

  PROCEDURE I2X(x: integer; VAR y: Extended);

  PROCEDURE S2X(x: Single; VAR y: Extended);

  PROCEDURE D2X(x: Double; VAR y: Extended);

  PROCEDURE C2X(x: Comp; VAR y: Extended);

  PROCEDURE X2X(x: Extended; VAR y: Extended);

  PROCEDURE X2I(x: Extended; VAR y: integer);

  PROCEDURE X2S(x: Extended; VAR y: Single);

  PROCEDURE X2D(x: Extended; VAR y: Double);

  PROCEDURE X2C(x: Extended; VAR y: Comp);

  PROCEDURE L2X(x: longint; VAR y: Extended);

  PROCEDURE X2L(x: Extended; VAR y: longint);

  PROCEDURE S2Dec(f: DecForm; x: Single; VAR y: Decimal);

  PROCEDURE D2Dec(f: DecForm; x: Double; VAR y: Decimal);

  PROCEDURE C2Dec(f: DecForm; x: Comp; VAR y: Decimal);

  PROCEDURE X2Dec(f: DecForm; x: Extended; VAR y: Decimal);

  PROCEDURE Dec2S(x: Decimal; VAR y: Single);

  PROCEDURE Dec2D(x: Decimal; VAR y: Double);

  PROCEDURE Dec2C(x: Decimal; VAR y: Comp);

  PROCEDURE Dec2X(x: Decimal; VAR y: Extended);

  PROCEDURE Str2Dec(s: DecStr; VAR index: integer; VAR d: Decimal;
                    VAR ValidPrefix: boolean);

  PROCEDURE Dec2Str(f: DecForm; d: Decimal; VAR s: DecStr);

  PROCEDURE S2Str(f: DecForm; x: Single; VAR y: DecStr);

  PROCEDURE D2Str(f: DecForm; x: Double; VAR y: DecStr);

  PROCEDURE C2Str(f: DecForm; x: Comp; VAR y: DecStr);

  PROCEDURE X2Str(f: DecForm; x: Extended; VAR y: DecStr);
```

```

    PROCEDURE Str2S(x: DecStr; VAR y: Single);

    PROCEDURE Str2D(x: DecStr; VAR y: Double);

    PROCEDURE Str2C(x: DecStr; VAR y: Comp);

    PROCEDURE Str2X(x: DecStr; VAR y: Extended);

    PROCEDURE RemX(x: Extended; VAR y: Extended; VAR quo: integer);

    PROCEDURE SqrtX(VAR x: Extended);

    PROCEDURE RintX(VAR x: Extended);

    PROCEDURE NegX(VAR x: Extended);

    PROCEDURE AbsX(VAR x: Extended);

    PROCEDURE CpySgnX(VAR x: Extended; y: Extended);

    PROCEDURE NextS(VAR x: Single; y: Single);

    PROCEDURE NextD(VAR x: Double; y: Double);

    PROCEDURE NextX(VAR x: Extended; y: Extended);

    FUNCTION ClassS(x: Single; VAR sgn: integer): NumClass;

    FUNCTION ClassD(x: Double; VAR sgn: integer): NumClass;

    FUNCTION ClassC(x: Comp; VAR sgn: integer): NumClass;

    FUNCTION ClassX(x: Extended; VAR sgn: integer): NumClass;

    PROCEDURE ScalbX(n: integer; VAR y: Extended);

    PROCEDURE LogbX(VAR x: Extended);

    PROCEDURE SetRnd(r: RoundDir);

    PROCEDURE SetEnv(e: Environ);

    FUNCTION GetRnd: RoundDir;

    PROCEDURE GetEnv(VAR e: Environ);

    FUNCTION TestXcp(x: Exception): boolean;

    PROCEDURE SetXcp(x: Exception; OnOff: boolean);

    FUNCTION TestHlt(x: Exception): boolean;

    PROCEDURE SetHlt(x: Exception; OnOff: boolean);

    PROCEDURE ProcEntry(VAR e: Environ);

    PROCEDURE ProcExit(e: Environ);

    PROCEDURE Log2X(VAR x: Extended);

    PROCEDURE LnX(VAR x: Extended);

    PROCEDURE Ln1X(VAR x: Extended);

    PROCEDURE Exp2X(VAR x: Extended);
```

```

   PROCEDURE ExpX(VAR x: Extended);

   PROCEDURE Exp1X(VAR x: Extended);

   PROCEDURE XpwrI(i: integer; VAR x: Extended);

   PROCEDURE XpwrY(y: Extended; VAR x: Extended);

   PROCEDURE Compound(r, n: Extended; VAR x: Extended);

   PROCEDURE Annuity(r, n: Extended; VAR x: Extended);

   PROCEDURE AtanX(VAR x: Extended);

   PROCEDURE SinX(VAR x: Extended);

   PROCEDURE CosX(VAR x: Extended);

   PROCEDURE TanX(VAR x: Extended);

   PROCEDURE RandomX(VAR x: Extended);

   FUNCTION GetHltAddress: longint;

   PROCEDURE SetHltAddress(HltAddress: longint);

   PROCEDURE InitFPLib;

   FUNCTION SANE_Environ: longint;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 230 Characters: 5565

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

```
******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : GRAFUTIL
*
******************************************************************************

USES {$U+} GRAFUTIL;


INTRINSIC;

INTERFACE

   USES                                {$U libqd/QuickDraw } QuickDraw;

   TYPE
      Fixed         = LongInt;
      Int64Bit      = RECORD
                         hiLong: LongInt;
                         loLong: LongInt;
                      END;

   FUNCTION BitAnd(long1, long2: LongInt): LongInt;

   FUNCTION BitOr(long1, long2: LongInt): LongInt;

   FUNCTION BitXor(long1, long2: LongInt): LongInt;

   FUNCTION BitNot(long: LongInt): LongInt;

   FUNCTION BitShift(long: LongInt; count: INTEGER): LongInt;

   FUNCTION BitTst(bytePtr: QDPtr; bitNum: LongInt): BOOLEAN;

   PROCEDURE BitSet(bytePtr: QDPtr; bitNum: LongInt);

   PROCEDURE BitClr(bytePtr: QDPtr; bitNum: LongInt);

   PROCEDURE LongMul(a, b: LongInt; VAR dst: Int64Bit);

   FUNCTION FixMul(a, b: Fixed): Fixed;

   FUNCTION FixRatio(numer, denom: INTEGER): Fixed;

   FUNCTION HiWord(x: Fixed): INTEGER;

   FUNCTION LoWord(x: Fixed): INTEGER;

   FUNCTION FixRound(x: Fixed): INTEGER;

******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
******************************************************************************

```

End of File -- Lines: 56 Characters: 1355

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : GRAPHS
*
*******************************************************************************

USES {$U+} GRAPHS;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/PASDEFS.OBJ *) PasDefs,
         (*$U OBJ/MEMMAN.OBJ  *) MemMan,
         (*$U OBJ/LISTS.OBJ   *) Lists;
      {$ELSEC }
         (*$U OBJ:PASDEFS.OBJ *) PasDefs,
         (*$U OBJ:MEMMAN.OBJ  *) MemMan,
         (*$U OBJ:LISTS.OBJ   *) Lists;
      {$ENDC }

   TYPE


      Vertex        = integer;
      Arc           = integer;
      VertexRec     = RECORD
                         firstIn, firstOut: Arc;
                      END;
      ArcRec        = RECORD
                         initial, terminal: Vertex;
                         nextIn, nextOut: Arc;
                      END;
      VtxRowPtr     = ^VtxRow;
      ArcRowPtr     = ^ArcRow;
      VtxRow        = ARRAY [ - 128..127] OF VertexRec;
      ArcRow        = ARRAY [ - 128..127] OF ArcRec;
      VtxArrPtr     = ^VtxArr;
      ArcArrPtr     = ^ArcArr;
      VtxArr        = ARRAY [ - 128..127] OF VtxRowPtr;
      ArcArr        = ARRAY [ - 128..127] OF ArcRowPtr;
      GRHandle      = ^GraphStore;
      GraphStore    = RECORD
                         memVertices, memArcs: MMHandle;
                         vertices: VtxArrPtr;
                         arcs: ArcArrPtr;
                      END;
      GraphHandle   = ^GraphObject;
      GraphObject   = RECORD
                         gstore: GRHandle;
                         inL, outL: ListHandle;
                      END;

   PROCEDURE AddArc(graph: GraphHandle; init, term: Vertex; VAR newA: Arc;
                    FUNCTION GetMore(n, row: integer): boolean);

   PROCEDURE AddVertex(graph: GraphHandle; VAR newV: Vertex; FUNCTION
                       GetMore(n, row: integer): boolean);

   PROCEDURE DeleteArc(graph: GraphHandle; oldArc: Arc);

   PROCEDURE DeleteVertex(graph: GraphHandle; oldVertex: Vertex);

   FUNCTION FindArc(graph: GraphHandle; init, term: Vertex): Arc;

   FUNCTION OppositeVertex(graph: GraphHandle; a: Arc;
                           oneVertex: Vertex): Vertex;

   FUNCTION InitVertex(graph: GraphHandle; a: Arc): Vertex;

   FUNCTION TermVertex(graph: GraphHandle; a: Arc): Vertex;

   FUNCTION IsSource(graph: GraphHandle; v: Vertex): boolean;

   FUNCTION IsSink(graph: GraphHandle; v: Vertex): boolean;

   FUNCTION IsIsolated(graph: GraphHandle; v: Vertex): boolean;

   PROCEDURE EachVertex(graph: GraphHandle; PROCEDURE
                        Visit(v: Vertex));

   PROCEDURE EachArc(graph: GraphHandle; PROCEDURE
                     Visit(a: Arc));

   PROCEDURE EachInArc(graph: GraphHandle; v: Vertex; PROCEDURE
                       Visit(inArc: Arc));

   PROCEDURE EachOutArc(graph: GraphHandle; v: Vertex; PROCEDURE
                        Visit(outArc: Arc));

   PROCEDURE EachIncidentArc(graph: GraphHandle; v: Vertex; PROCEDURE
                             Visit(incidentArc: Arc));

   PROCEDURE EachPredecessorVertex(graph: GraphHandle; v: Vertex; PROCEDURE
                                   Visit(predecessor: Vertex));

   PROCEDURE EachSuccessorVertex(graph: GraphHandle; v: Vertex; PROCEDURE
                                 Visit(successor: Vertex));

   PROCEDURE EachAdjacentVertex(graph: GraphHandle; v: Vertex; PROCEDURE
                                Visit(adjacent: Vertex));

   PROCEDURE InitGLists;

   PROCEDURE InitGraph(VAR graph: GraphHandle; nVertices, nArcs: longint;
                       FUNCTION MoreVerts(n, row: integer): boolean; FUNCTION
                       MoreArcs(n, row: integer): boolean);

   PROCEDURE DFSDir(g: GraphHandle; start: Vertex; PROCEDURE
                    First(vi, a, vt: Vertex); PROCEDURE
                    Again(vi, a, vt: Vertex); PROCEDURE
                    Done(v: Vertex));

   PROCEDURE BFSUnDir(g: GraphHandle; start: Vertex; FUNCTION
                      Test(v: Vertex; VAR cont: boolean): boolean);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 125 Characters: 4154

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : HEAP
*
*******************************************************************************

USES {$U+} HEAP;


INTRINSIC;

INTERFACE

   CONST
      noerror       = 0;
      nomem         = 3411;
      lhead         = 2;
      uhead         = 10;
      free_overhead = 12;
      allocover     = 8;

   TYPE
      _ptr          = ^integer;
      ptrptr        = ^_ptr;
      heapptr       = ^heapblk;
      heapblk       = RECORD
                         CASE boolean OF
                            true:
                               (size: integer;
                                prep: heapptr;
                                next: heapptr);
                            false:
                               (blk: ARRAY [0..0] OF - 127..127)
                      END;
      zonedesc      = RECORD
                         avail: heapptr;
                         heapmin, heapmax: _ptr;
                         lastdiff: integer;
                         availsize: integer;
                      END;
      ptrzonedesc   = ^zonedesc;

   PROCEDURE init_heap(pzone: ptrzonedesc; initsize: integer; where: _ptr);

   FUNCTION grow_heap(pzone: ptrzonedesc; growincr: integer): integer;

   FUNCTION nnew(pzone: ptrzonedesc; reqsize: integer;
                 pdataptr: ptrptr): integer;

   PROCEDURE ndispose(pzone: ptrzonedesc; pdata: _ptr);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 57 Characters: 1595

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : HWINT
*
********************************************************************************

USES {$U+} HWINT;


INTRINSIC;

INTERFACE

   TYPE
      Ascii         = Char;
      Pixels        = Integer;
      ManyPixels    = LongInt;
      CursorHeight  = Integer;
      CursorPtr     = ^Integer;
      LogicalAddress = LongInt;
      DateArray     = RECORD
                         year: Integer;
                         day: Integer;
                         hour: Integer;
                         minute: Integer;
                         second: Integer;
                      END;
      Frames        = LongInt;
      Seconds       = LongInt;
      MilliSeconds  = LongInt;
      MicroSeconds  = LongInt;
      AlarmNumber   = Integer;
      SpeakerVolume = Integer;
      ScreenContrast = Integer;
      VisibleScrn   = (PriScrn, AltScrn);
      KeybdQIndex   = 1..1000;
      KeybdId       = Integer;
      KeyCap        = 0..127;
      KeyCapSet     = SET OF KeyCap;
      KeyEvent      = PACKED RECORD
                         key: KeyCap;
                         ascii: Char;
                         state: Integer;
                         mouseX: Pixels;
                         mouseY: Pixels;
                         time: MilliSeconds;
                      END;

   PROCEDURE DriverInit;

   PROCEDURE DiskDriver(routine: LogicalAddress);

   PROCEDURE TwiggyDriver(routine: LogicalAddress);

   PROCEDURE DiskSync(busy: Boolean);

   PROCEDURE NMISync;

   PROCEDURE COPSSync;

   PROCEDURE Poll;
```

```

    PROCEDURE MouseLocation(VAR x: Pixels; VAR y: Pixels);

    PROCEDURE MouseUpdates(delay: MilliSeconds);

    PROCEDURE MouseScaling(scale: Boolean);

    PROCEDURE MouseThresh(threshold: Pixels);

    FUNCTION MouseOdometer: ManyPixels;

    PROCEDURE CursorLocation(x: Pixels; y: Pixels);

    PROCEDURE CursorTracking(track: Boolean);

    PROCEDURE CursorImage(hotX: Pixels; hotY: Pixels; height: CursorHeight;
                          data: CursorPtr; mask: CursorPtr);

    PROCEDURE CursorHide;

    PROCEDURE CursorShield(left: Pixels; top: Pixels; right: Pixels;
                           bottom: Pixels);

    PROCEDURE CursorDisplay;

    PROCEDURE CursorObscure;

    PROCEDURE CursorInit;

    PROCEDURE CursorReInit;

    PROCEDURE BusyImage(hotX: Pixels; hotY: Pixels; height: CursorHeight;
                        data: CursorPtr; mask: CursorPtr);

    PROCEDURE BusyDelay(delay: MilliSeconds);

    FUNCTION FrameCounter: Frames;

    PROCEDURE ScreenSize(VAR x: Pixels; VAR y: Pixels);

    FUNCTION ScreenAddr: LogicalAddress;

    FUNCTION AltScreenAddr: LogicalAddress;

    FUNCTION ScreenKeybd: VisibleScreen;

    PROCEDURE SetScreenKeybd(screen: VisibleScreen);

    FUNCTION Contrast: ScreenContrast;

    PROCEDURE SetContrast(contrast: ScreenContrast);

    PROCEDURE RampContrast(contrast: ScreenContrast);

    FUNCTION DimContrast: ScreenContrast;

    PROCEDURE SetDimContrast(contrast: ScreenContrast);

    FUNCTION FadeDelay: MilliSeconds;

    PROCEDURE SetFadeDelay(delay: MilliSeconds);

    PROCEDURE PowerDown;

    PROCEDURE PowerCycle(delay: Seconds);

    FUNCTION Volume: SpeakerVolume;

    PROCEDURE SetVolume(volume: SpeakerVolume);

    PROCEDURE Noise(waveLength: MicroSeconds);

    PROCEDURE Silence;

    PROCEDURE Beep(waveLength: MicroSeconds; duration: MilliSeconds);

    FUNCTION Keyboard: KeybdId;

    FUNCTION Legends: KeybdId;

    PROCEDURE SetLegends(id: KeybdId);

    FUNCTION KeyIsDown(key: KeyCap): Boolean;

    PROCEDURE KeyMap(VAR keys: KeyCapSet);

    FUNCTION KeybdPeek(repeats: Boolean; index: KeybdQIndex;
                       VAR event: KeyEvent): Boolean;

    FUNCTION AltKeyPeek(repeats: Boolean; index: KeybdQIndex;
                        VAR event: KeyEvent): Boolean;

    FUNCTION KeybdEvent(repeats: Boolean; wait: Boolean;
                        VAR event: KeyEvent): Boolean;

    FUNCTION AltKeyEvent(repeats: Boolean; wait: Boolean;
                         VAR event: KeyEvent): Boolean;

    PROCEDURE RepeatRate(VAR initial: MilliSeconds;
                         VAR subsequent: MilliSeconds);

    PROCEDURE SetRepeatRate(initial: MilliSeconds; subsequent: MilliSeconds);

    PROCEDURE KeyPushed(key: KeyCap);

    FUNCTION NMIKey: KeyCap;

    PROCEDURE SetNMIKey(key: KeyCap);

    FUNCTION ToggleKey: KeyCap;

    PROCEDURE SetToggleKey(key: KeyCap);

    FUNCTION KeyToAscii(key: KeyCap; State: Integer): Ascii;

    FUNCTION MicroTimer: MicroSeconds;

    FUNCTION Timer: MilliSeconds;

    PROCEDURE AlarmAssign(VAR alarm: AlarmNumber; routine: LogicalAddress);

    PROCEDURE AlarmReturn(alarm: AlarmNumber);

    PROCEDURE AlarmAbsolute(alarm: AlarmNumber; time: MilliSeconds);

    PROCEDURE AlarmRelative(alarm: AlarmNumber; delay: MilliSeconds);

    PROCEDURE AlarmOff(alarm: AlarmNumber);

    PROCEDURE DateTime(VAR date: DateArray);

    PROCEDURE SetDateTime(date: DateArray);
```

```

   PROCEDURE DateToTime(date: DateArray; VAR time: Seconds);

   FUNCTION TimeStamp: Seconds;

   PROCEDURE SetTimeStamp(time: Seconds);

   PROCEDURE TimeToDate(time: Seconds; VAR date: DateArray);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 208 Characters: 5154

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

```
*****************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : INTERNAT
*
*****************************************************************************

USES {$U+} INTERNAT;


INTRINSIC;

INTERFACE

   USES
     {$U libsm/UnitStd } UnitStd;

   CONST
      MaximumCharsMatching = 10;
      MCodeNil      = 0;
      MCodeUS       = 1;
      MCodeUK       = 2;
      MCodeGermany  = 3;
      MCodeFrance   = 4;
      MCodeItaly    = 5;
      MCodeSweden   = 6;
      MCodeSpain    = 7;

   TYPE
      KindOfCompare = (CompAbsolute, CompStrong, CompWeak, CompVeryWeak);
      CompChrP      = ^CompChr;
      CompChr       = PACKED ARRAY [0..0] OF CHAR;
      CompStrP      = ^CompStr;
      CompStr       = STRING[255];

   PROCEDURE InitInternational(VAR errnum: INTEGER);

   PROCEDURE UpStrShift(sP: CompStrP);

   PROCEDURE UpChrShift(pc: CompChrP; length: INTEGER);

   PROCEDURE DownStrShift(sP: CompStrP);

   PROCEDURE DownChrShift(pc: CompChrP; length: INTEGER);

   FUNCTION CompStrMagnitude(string1, string2: CompStrP;
                             veryWeak: BOOLEAN): INTEGER;

   FUNCTION CompChrMagnitude(char1, char2: CompChrP; length1, length2: INTEGER;
                             veryWeak: BOOLEAN): INTEGER;

   FUNCTION CompareChar(ch1, ch2: CHAR; kind: KindOfCompare): BOOLEAN;

   PROCEDURE CharsMatching(ch: CHAR; kind: KindOfCompare;
                           matchingChars: CompChrP;
                           VAR nOfMatchigChars: INTEGER);

   FUNCTION CompStrIdentity(string1, string2: CompStrP;
                            kind: KindOfCompare): BOOLEAN;

   FUNCTION CompChrIdentity(char1, char2: CompChrP; length1, length2: INTEGER;
                            kind: KindOfCompare): BOOLEAN;
```

```

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 68 Characters: 1915

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : IOPRIMIT
*
********************************************************************************

USES {$U+} IOPRIMIT;


INTRINSIC;

INTERFACE

   USES
      {$u SULib} StdUnit;

   CONST
      IOMaxStr      = 132;
      IOMaxStP1     = 133;
      IOBlkSize     = 6;
      IOByteSize    = 3072;
      IOBytSizeM1   = 3071;
      IOIncMax      = 5;
      IOPvtErr      = 32000;
      IOOpnErr      = 32001;
      IOIncErr      = 32002;
      IOBfrErr      = 32003;
      IOMem1Err     = 32004;
      IOMem2Err     = 32005;

   TYPE
      IOModes       = (IORead, IOWrite);
      IOCloseKind   = (IONormal, IOLock, IOPurge, IOCrunch);
      IODevType     = (IOBlkDev, IOKbDev, IOConsDev, IOPrDev);
      IOStrP        = ^IOStr;
      IOStr         = String[IOMaxStr];
      IOBufrP       = ^IOBufr;
      IOBufr        = PACKED ARRAY [0..IOBytSizeM1] OF Char;
      IOPBufP       = ^IOPBufr;
      IOPBufr       = RECORD
                         Next: IOPBufP;
                         Bufr: IOBufrP;
                      END;
      IOFCBP        = ^IOFCB;
      IOFCB         = RECORD
                         Filename: SUStr;
                         DevType: IODevType;
                         F: FILE;
                         LineNbr: Integer;
                         Bufr: IOBufrP;
                         BufrCp: Integer;
                         PvtBufr: IOPBufP;
                         PvtFCB: Boolean;
                         BlkNbr: Integer;
                         CurrLine: String[IOMaxStP1];
                         CurrLen: Integer;
                         Next, Prev, Avail: IOFCBP;
                         PtrToFCBPtr: ^IOFCBP;
                         Pushed: Boolean;
                         BufrSaved: Boolean;
                         CASE Mode: IOModes OF
                         IORead:
                            (LastBlk: Integer;
                             LastByte: Integer;
                             Eol: Boolean);
                         IOWrite:
                            (EndPage: Integer);
                   END;
   IOTextP       = ^Text;

VAR
   IOCurrFCB:    IOFCBP;
   IOIncDepth:   Integer;
   IOIncBufr:    IOBufrP;
   IOIncFCBs:    ARRAY [1..IOIncMax] OF IOFCBP;
   IOInTotal:    LongInt;
   IOOutTotal:   LongInt;
   IOEndfile:    Char;
   IONewline:    Char;
   IOKeyBoard:   Text;
   IOObjExt:     Boolean;
   IONoExt:      Boolean;
   IONoDLEs:     Boolean;

PROCEDURE InitIO;

PROCEDURE EndIO;

FUNCTION OpenF(Fname: SUStr; VAR FCB: IOFCBP; Mode: IOModes;
               Bufr: IOBufrP): Integer;

FUNCTION CreateF(Fname: SUStr; VAR FCB: IOFCBP; Bufr: IOBufrP): Integer;

FUNCTION SysOpenF(Fname: SUStr; VAR FCB: IOFCBP; Mode: IOModes;
                  Bufr: IOBufrP): Integer;

PROCEDURE CloseF(FCB: IOFCBP; CloseKind: IOCloseKind);

FUNCTION PushInput(Fname: SUStr): Integer;

FUNCTION PopInput: Boolean;

PROCEDURE NextPage(FCB: IOFCBP);

PROCEDURE FillBufr(FCB: IOFCBP);

PROCEDURE SeekBlock(FCB: IOFCBP; Block: Integer);

PROCEDURE FilePosition(FCB: IOFCBP; VAR Block, Byte: Integer);

FUNCTION GetcF(FCB: IOFCBP; VAR c: Char): Char;

FUNCTION GetLine(FCB: IOFCBP; Line: IOStrP): Boolean;

PROCEDURE PutcF(FCB: IOFCBP; c: Char);

PROCEDURE PutLineP(FCB: IOFCBP; Line: IOStrP);

PROCEDURE PutLineS(FCB: IOFCBP; Line: IOStr);

PROCEDURE PutStrP(FCB: IOFCBP; S: IOStrP; Width: Integer);

PROCEDURE PutStrS(FCB: IOFCBP; S: IOStr; Width: Integer);

FUNCTION PutIntP(N: LongInt; Width: Integer): IOStrP;

FUNCTION IOError(IOStatus: Integer; Msg: SUStr): Boolean;
```

```

   FUNCTION SysReset(F: IOTextP; FN: SUStr): Integer;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 136 Characters: 3872

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : IUMAN
*
*******************************************************************************

USES {$U+} IUMAN;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/STDUNIT.OBJ *) StdUnit,
         (*$U OBJ/PASDEFS.OBJ *) PasDefs,
         (*$U OBJ/MEMMAN.OBJ *) MemMan,
         (*$U OBJ/FILEIO.OBJ *) FileIO,
         (*$U OBJ/OBJIO.OBJ *) ObjIO;
      {$ELSEC }
         (*$U OBJ:STDUNIT.OBJ *) StdUnit,
         (*$U OBJ:PASDEFS.OBJ *) PasDefs,
         (*$U OBJ:MEMMAN.OBJ *) MemMan,
         (*$U OBJ:FILEIO.OBJ *) FileIO,
         (*$U OBJ:OBJIO.OBJ *) ObjIO;
      {$ENDC }

   CONST


      ConfigMax     = '127.127.127.127';

   TYPE
      pSegLocVariant = ^iSegLocVariant;
      pOldULVariant = ^iOldULVariant;
      pUnitLVariant = ^iUnitLVariant;
      {$IFC isSAndE}
      pSoftSystem   = ^iSoftSystem;
      pSftSysRec    = ^SftSysRec;
      SftSysRec     = RECORD
                         SDep: iSftSysVariant;
                         nxt: pSftSysRec;
                      END;
      {$ENDC}
      IUSegArr      = ARRAY [1..TMAXSEGS] OF pSegLocVariant;
      IUUnitArr     = ARRAY [1..TMAXUNITS] OF pUnitLVariant;
      IUStrArr      = ARRAY [1..TMAXFILES] OF PLString;
      PIUStrArr     = ^IUStrArr;
      {$IFC isSAndE}
      IUSysArr      = ARRAY [1..TMAXSYSS] OF pSoftSystem;
      IUSysDep      = ARRAY [1..TMAXSYSS] OF pSftSysRec;
      {$ENDC}

   VAR
      IUMaxSegs, IUMaxUnits, IUMaxFNames, IUMaxSyss: integer;
      iuLibSegs:    ^IUSegArr;
      iuLibUnts:    ^IUUnitArr;
      iuLibFNam:    PIUStrArr;
      {$IFC isSAndE}
      iuLibSyss:    ^IUSysArr;
      iuLibSNam:    PIUStrArr;
      iuLibSDep:    ^IUSysDep;
      {$ENDC}
      iuLibFile:    ObjHandle;
      iuLibName:    LString;
      OldFormat:    boolean;
      ThisIULib:    integer;
      sysConfig, sysMinConfig, sysMaxConfig: Longint;

   PROCEDURE InitIUMan;

   PROCEDURE EachIUSeg(PROCEDURE visit(s: integer));

   PROCEDURE EachIUUnit(PROCEDURE visit(u: integer));

   PROCEDURE EachIUFile(PROCEDURE visit(f: integer));
   {$IFC isSAndE}

   PROCEDURE EachIUSys(PROCEDURE visit(s: integer));
   {$ENDC}

   PROCEDURE VersToStr(VAR s: LString; VAR vers: Longint);

   PROCEDURE WriteVers(VAR f: text; vers: Longint; len: integer);

   PROCEDURE StrToVers(s: LString; VAR vers: Longint);

   PROCEDURE ReadVers(VAR vers: Longint; VAR stat: PromptState; DefL: Longint);

   PROCEDURE VerifyVers(minfile: Longint; maxfile: Longint; msg: LString);

   FUNCTION ReadIULib(allocDelta, unitsOnly, filter: boolean;
                      SSNam: PLString): boolean;

   PROCEDURE WriteIULib(VCtrl: ObjFileType; SysC: Longint; ModL: integer;
                        TimeS: Longint; VAR iuOutFile: ObjHandle;
                        RootIdx: integer);

   PROCEDURE WriteIUVers(VCtrl: ObjFileType; SysC: Longint; ModL: integer;
                         TimeS: Longint; VAR iuOutFile: ObjHandle;
                         RootIdx: integer; EmitOSBlk: boolean);

   FUNCTION IUInstall(VAR FName: LString; FNum: integer; VAR vFile: ObjHandle;
                      VAR PName: LString; PITSeg: integer): boolean;

   PROCEDURE CopyInter(inF, outF: ObjHandle; VAR outBlock: ObjBlock; NBytes,
                       bufP: Longint);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 115 Characters: 3471

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

```
******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : LABSCAN
*
******************************************************************************

USES {$U+} LABSCAN;


INTRINSIC;

INTERFACE

   USES
      {$IFC OSBUILT }
         {$U libsm/unitstd.obj} unitstd,
         {$U libdb/dbenv.obj } dbenv,
         {$U libin/INTRLIBp.obj } international,
         {$U libdb/dbdecl1.obj } dbdecl1,
         {$U libos/syscall.obj } syscall,
         {$U libos/psyscall.obj } psyscall,
         {$U libdb/lowlevel.obj } lowlevel,
         {$U libdb/pooler.obj } pooler,
         {$U libdb/heap.obj } heap,
         {$U libdb/czcompact.obj } czCompact,
         {$U libdb/vltree.obj } vltree,
         {$U libdb/scan.obj } scan;
      {$ELSEC}
         {$U OBJ:dbenv.obj } dbenv,
         {$U INTRLIB.obj } international,
         {$U OBJ:dbdecl1.obj } dbdecl1,
         {$U OBJ:syscall.obj } syscall,
         {$U OBJ:lowlevel.obj } lowlevel,
         {$U OBJ:pooler.obj } pooler,
         {$U OBJ:heap.obj } heap,
         {$U OBJ:czcompact.obj } czCompact,
         {$U OBJ:vltree.obj } vltree,
         {$U OBJ:scan.obj } scan;
      {$ENDC}

   {$SETC debug := false }

   {$IFC PRERELEASE }

   var


      labdebug:     boolean;
      {$ENDC }

   PROCEDURE lfetch(VAR ddresult: integer; scanid: integer; which: integer;
                    nsearch: integer; plabrec: ptrlabelentry);

   PROCEDURE linsert(VAR ddresult: integer; scanid: integer;
                     plabrec: ptrlabelentry; size: integer;
                     newticket: boolean);

   PROCEDURE ldelete(VAR ddresult: integer; scanid: integer; which: integer;
                     nsearch: integer; plabrec: ptrlabelentry);

******************************************************************************
```

```
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 66 Characters: 1901

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : LCUT
*
*******************************************************************************

USES {$U+} LCUT;


INTRINSIC;

INTERFACE

   USES
      (*$U libsm/UnitStd *) UnitStd,
      (*$U libsm/UnitHz *) UnitHz,
      (*$U libqd/Storage *) Storage,
      {$U libqd/QuickDraw } QuickDraw,
      {$U libfm/FontMgr   } FontMgr,
      (*$U libdb/dbenv *) dbenv,
      (*$U libfe/FEdec *) FEdec,
      (*$U libfe/fld *) fieldedit,
      (*$U libos/SysCall *) Syscall,
      (*$U libte/teenv *) teenv,
      (*$U libpr/PMDecl *) PMDecl,
      (*$U libpr/PrStdInfo *) PrStdInfo,
      (*$U libwm/events *) events,
      (*$U libsu/scrap *) scrap,
      (*$U libsu/unitFile  *) unitFile,
      (*$U libsu/unitFmt   *) unitFmt,
      (*$U libsu/unitCs    *) UnitCs,
      (*$U libsu/unitFF *) unitFF;
      (*$SETC LCUTdebug := not teProduction *)

   CONST
      LCUTAllOk     = 0;
      LCUTNoData    = 1;
      LCUTTabProb   = 2;
      LCUTWidProb   = 3;
      LCUTNoHeap    = 4;
      (*$IFC LCUTdebug *)
      TraceLCUT     = false
      (*$ENDC *)

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************


End of File -- Lines: 50 Characters: 1328
```

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : LISTS
*
*******************************************************************************

USES {$U+} LISTS;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/PASDEFS.OBJ *) PasDefs,
         (*$U OBJ/MEMMAN.OBJ  *) MemMan;
      {$ELSEC }
         (*$U OBJ:PASDEFS.OBJ *) PasDefs,
         (*$U OBJ:MEMMAN.OBJ  *) MemMan;
      {$ENDC }

   TYPE


      Head          = integer;
      Item          = integer;
      ListOpsPtr    = ^ListOpsRec;
      EnvrPtr       = ^integer;
      ListOpsRec    = RECORD
                         setFirst: ProcPtr;
                         getFirst: ProcPtr;
                         setNext: ProcPtr;
                         getNext: ProcPtr;
                         before: ProcPtr;
                      END;
      ListHandle    = ^ListStore;
      ListStore     = RECORD
                         ops: ListOpsPtr;
                         lstore: longint;
                      END;

   PROCEDURE AddItem(list: ListHandle; LHead: Head; NewItem: Item);

   PROCEDURE DeleteItem(list: ListHandle; LHead: Head; OldItem: Item);

   PROCEDURE EachItem(list: ListHandle; LHead: Head; PROCEDURE
                      Visit(LItem: Item));

   FUNCTION FirstItem(list: ListHandle; LHead: Head): Item;

   PROCEDURE InitList(list: ListHandle; NewList: Head);

   PROCEDURE InitLStore(VAR list: ListHandle; LO: ListOpsPtr; LS: longint);

   PROCEDURE InsertItem(list: ListHandle; LHead: Head; NewItem: Item);

   FUNCTION ListEmpty(list: ListHandle; LHead: Head): boolean;

   FUNCTION ListSize(list: ListHandle; LHead: Head): integer;

   PROCEDURE ReverseList(list: ListHandle; LHead: Head);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 69 Characters: 1923

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : LOWLEVEL
*
********************************************************************************

USES {$U+} LOWLEVEL;


INTRINSIC;

INTERFACE

   USES
      {$IFC OSBuilt }
         {$U libsm/unitstd.obj} unitstd,
         {$U libdb/dbenv.obj } dbenv,
         {$U libdb/dbdecl1.obj } dbdecl1,
         {$U libos/syscall.obj } syscall,
         {$U libos/psyscall.obj } psyscall;
      {$ELSEC }
         {$U obj:dbenv.obj } dbenv,
         {$U obj:dbdecl1.obj } dbdecl1,
         {$U obj:syscall.obj } syscall;
      {$ENDC }

   Type


      ptrpathname   = ^pathname;

   VAR
      prefix_length: integer;
      prefix:        pathname;
      passwd_length: integer;
      password:      e_name;

   PROCEDURE setprefix(newprefix: ptrpathname);

   PROCEDURE setpasswd(VAR newpasswd: e_name);

   PROCEDURE prefix_name(fname: ptrpathname);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 49 Characters: 1120

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

********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : MATHLIB
*
********************************************************************************

USES {$U+} MATHLIB;


INTRINSIC;

INTERFACE

   {$C Copyright 1983, 1984, Apple Computer Inc. }

   USES
     {$U- } {$U LibFP/NewFPLib } FPLib;

   CONST
      RandModulus   = 2147483647;
      LSigDigLen    = 30;

   TYPE
      RoundPrecision = (ExtPrecision, DblPrecision, RealPrecision);
      FP_Type       = (TFP_byte, TFP_integer, TFP_longint, TFP_Comp, TFP_real,
                       TFP_Double, TFP_Extended);
      Free_Format   = RECORD
                         MaxSig: integer;
                         Sig_FForm, Trail_Point, Int_EForm, Plus_EForm: boolean;
                      END;
      P_QR_Record   = longint;
      LongSigDig    = string[LSigDigLen];
      LongDecimal   = RECORD
                         sgn: 0..1;
                         exp: integer;
                         sig: LongSigDig;
                      END;

   PROCEDURE ASinX(VAR x: Extended);

   PROCEDURE ACosX(VAR x: Extended);

   PROCEDURE SinhX(VAR x: Extended);

   PROCEDURE CoshX(VAR x: Extended);

   PROCEDURE TanhX(VAR x: Extended);

   PROCEDURE Abs2X(x, y: Extended; VAR z: Extended);

   PROCEDURE ATan2X(x, y: Extended; VAR z: Extended);

   FUNCTION NextRandom(lastrandom: longint): longint;

   PROCEDURE ClearXcps;

   PROCEDURE ClearHlts;

   PROCEDURE SetPrecision(p: RoundPrecision);

   FUNCTION GetPrecision: RoundPrecision;
```

```

    PROCEDURE Math_Sort(first, last: integer; FUNCTION
                        sorted(i, j: integer): boolean; PROCEDURE
                        swap(i, j: integer)VAR error: boolean);

    FUNCTION SignOfX(x: Extended): boolean;

    FUNCTION FP_New(n: longint): longint;

    PROCEDURE FP_Size(x: Extended; VAR sgn: integer; VAR class: NumClass;
                      VAR size: FP_Type);

    PROCEDURE FP_Free_Ascii(x: Extended; width: integer; form: Free_Format;
                            VAR s: Decstr);

    PROCEDURE Fin_Npv(first, last, net: integer; rate: Extended;
                      VAR Npv: Extended; PROCEDURE
                      payment(i: integer; VAR pmt: Extended));

    PROCEDURE Fin_Return(first, last: integer; negperiod, posperiod: integer;
                         negrate, posrate: Extended; VAR ncs: integer;
                         VAR ret: Extended; PROCEDURE
                         payment(i: integer; VAR pmt: Extended));

    PROCEDURE Mat_Mult(n, p, m: integer; overlap: boolean; VAR error: boolean;
                       PROCEDURE afetch(i, j: integer;
                       VAR aij: Extended)PROCEDURE
                       xfetch(i, j: integer; VAR xij: Extended)PROCEDURE
                       bstore(i, j: integer; bij: Extended));

    PROCEDURE QR_Factor(n, p: integer; pivot: boolean; VAR QR: P_QR_Record;
                        PROCEDURE afetch(i, j: integer; VAR aij: Extended));

    PROCEDURE QR_Condition(QR: P_QR_Record; VAR cond: Extended);

    PROCEDURE QR_Determinant(QR: P_QR_Record; VAR det: Extended);

    PROCEDURE QR_Solve(m: integer; QR: P_QR_Record; VAR error: boolean; PROCEDURE
                       bfetch(i, j: integer; VAR bij: Extended); PROCEDURE
                       xstore(i, j: integer; xij: Extended));

    PROCEDURE QR_Residual(n, p: integer; m: integer; PROCEDURE
                          afetch(i, j: integer; VAR aij: Extended); PROCEDURE
                          bfetch(i, j: integer; VAR bij: Extended); PROCEDURE
                          xfetch(i, j: integer; VAR xij: Extended); PROCEDURE
                          rstore(i, j: integer; rij: Extended));

    PROCEDURE QR_Improve(m: integer; QR: P_QR_Record; VAR error: boolean;
                         PROCEDURE afetch(i, j: integer; VAR aij: Extended);
                         PROCEDURE bfetch(i, j: integer; VAR bij: Extended);
                         PROCEDURE xfetch(i, j: integer; VAR xij: Extended);
                         PROCEDURE xstore(i, j: integer; xij: Extended));

    PROCEDURE QR_TranSolve(m: integer; QR: P_QR_Record; VAR error: boolean;
                           PROCEDURE bfetch(i, j: integer; VAR bij: Extended);
                           PROCEDURE xstore(i, j: integer; xij: Extended));

    PROCEDURE QR_TranDeterminant(QR: P_QR_Record; VAR det: Extended);

    PROCEDURE X2LDec(f: DecForm; x: Extended; VAR y: LongDecimal);

    PROCEDURE LDec2X(prec: RoundPrecision; x: LongDecimal; VAR y: Extended);

    PROCEDURE Math_Solve(est1, est2: Extended; VAR result: Extended; PROCEDURE
                         f(x: Extended; VAR fx: Extended));

```

```
*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 133 Characters: 4707

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : MEMMAN
*
*******************************************************************************

USES {$U+} MEMMAN;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/PASDEFS.OBJ *) PasDefs;
      {$ELSEC }
         (*$U OBJ:PASDEFS.OBJ *) PasDefs;
      {$ENDC }

   CONST


      NullItem      = - 32768;

   TYPE
      MMStatus      = (MMAlloc, MMFree);
      MMStore       = ^MMStatArr;
      PMMStatRow    = ^MMStatRow;
      MMStatArr     = ARRAY [ - 128..127] OF PMMStatRow;
      MMStatRow     = PACKED ARRAY [ - 128..127] OF MMStatus;
      MMHandle      = ^MMRec;
      MMRec         = RECORD
                         maxElements, curElements: longint;
                         FirstFree: longint;
                         mstore: MMStore;
                      END;
      MMbyte        = - 128..127;
      MMIntBytes    = RECORD
                         CASE boolean OF
                            true:
                               (int: integer);
                            false:
                               (hi, lo: MMbyte)
                      END;

   PROCEDURE InitMem(VAR mem: MMHandle; numElements: longint; FUNCTION
                     GetMore(n, row: integer): boolean);

   FUNCTION MMAllocate(mem: MMHandle; FUNCTION
                       GetMore(n, row: integer): boolean): integer;

   PROCEDURE Reserve(mem: MMHandle; i: integer; FUNCTION
                     GetMore(n, row: integer): boolean);

   PROCEDURE Free(mem: MMHandle; i: integer);

   FUNCTION IsFree(mem: MMHandle; i: integer): boolean;

   PROCEDURE EachAllocated(mem: MMHandle; PROCEDURE
                           visit(i: integer));

   PROCEDURE HeapAlloc(VAR p: MemPtr; n: longint; m: longint);

   FUNCTION MMRows(n: longint): integer;

   FUNCTION NullMore(n, row: integer): boolean;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 74 Characters: 2079

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : MENUS
*
*******************************************************************************

USES {$U+} MENUS;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libhw/HWInt       } HWInt,
      {$U libsm/UnitStd     } UnitStd,
      {$U libsm/UnitHz      } UnitHz,
      {$U libqd/Storage     } Storage,
      {$U libqd/QuickDraw   } QuickDraw,
      {$U libfm/FontMgr     } FontMgr,
      {$U libos/SysCall     } SysCall,
      {$U libwm/Events      } Events,
      {$U libwm/Windows     } Windows,
      {$U libwm/Folders     } Folders;

   CONST
      maxItem       = 31;
      noMark        = 0;
      vertSpace     = 15;
      {$IFC wmOs }
      checkMark     = ccCheck;
      appleMark     = ccApple;
      {$ELSEC}
      checkMark     = 142;
      appleMark     = 143;
      {$ENDC}

   TYPE
      MenuPtr       = ^MenuInfo;
      MenuInfo      = RECORD
                           menuId: INTEGER;
                           menuWidth: INTEGER;
                           menuHeight: INTEGER;
                           DrawProc: ProcPtr;
                           ChooseProc: ProcPtr;
                           enableFlags: PACKED ARRAY [0..maxItem] OF BOOLEAN;
                           menuData: Handle;
                        END;
      rMenuInfo     = ARRAY [0..0] OF MenuInfo;
      rMenuPtr      = ^rMenuInfo;
      ItemPtr       = ^Str255;

   VAR
      menuDelay:    INTEGER;
      desktopMenu:  ARRAY [0..1] OF MenuInfo;

   PROCEDURE CalcMenuSize(VAR menu: MenuInfo);

   PROCEDURE CheckItem(VAR menu: MenuInfo; item: INTEGER; checked: BOOLEAN);

   PROCEDURE ChooseTxtItem(VAR menu: MenuInfo; menuRect: Rect; hitPt: Point;
                           VAR whichItem: INTEGER);

   PROCEDURE ClearMenuBar;

   PROCEDURE DeleteMenu(menuId: INTEGER);

   PROCEDURE DisableItem(VAR menu: MenuInfo; item: INTEGER);

   PROCEDURE DisableMenu(VAR menu: MenuInfo);

   PROCEDURE DrawMenuBar;

   PROCEDURE DrawTxtMenu(VAR menu: MenuInfo; menuRect: Rect);

   PROCEDURE EnableItem(VAR menu: MenuInfo; item: INTEGER);

   PROCEDURE EnableMenu(VAR menu: MenuInfo);

   PROCEDURE GetItem(VAR menu: MenuInfo; item: INTEGER; itemString: ItemPtr);

   PROCEDURE HiLiteMenu(menuId: INTEGER);

   PROCEDURE InitMenus;

   PROCEDURE InsertMenu(VAR menu: MenuInfo; beforeId: INTEGER);

   PROCEDURE MarkItem(VAR menu: MenuInfo; item: INTEGER; whichMark: INTEGER);

   PROCEDURE MenuSelect(startPt: Point; VAR whichMenu, whichItem: INTEGER);

   PROCEDURE MenuKey(ch: CHAR; VAR whichMenu, whichItem: INTEGER);

   PROCEDURE ReadMenu(VAR menuFile: TEXT; VAR menu: MenuInfo);

   PROCEDURE SetItem(VAR menu: MenuInfo; item: INTEGER; itemString: ItemPtr);

*******************************************************************************
*
*                              THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 103 Characters: 2851

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : OBJIO
*
*******************************************************************************

USES {$U+} OBJIO;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/PASDEFS.OBJ *) PasDefs,
         (*$U OBJ/FILEIO.OBJ  *) FileIO;
      {$ELSEC }
         (*$U OBJ:PASDEFS.OBJ *) PasDefs,
         (*$U OBJ:FILEIO.OBJ  *) FileIO;
      {$ENDC }
   CONST


      TMAXSEGS      = 4096;
      TMAXUNITS     = 4096;
      TMAXFILES     = 1024;
      TMAXSYSS      = 127;
      DLTASEGS      = 128;
      DLTAUNITS     = 128;
      DLTAFILES     = 32;
      DLTASYSS      = 16;

   TYPE
      BlockType     = (ModuleName, EndBlock, EntryPoint, External, StartAddress,
                       CodeBlock, Relocation, CommonReloc, CommonDef,
                       ShortExternal, DInitDef, DInitData, DInitRef, DInitDRef,
                       QuickLoad, OldExecutable, LibModule, LibEntry, UnitBlock,
                       InterfLoc, PhysicalExec, Executable, VersionCtrl,
                       SegmentTable, UnitTable, SegLocation, OldUntLoc,
                       StringBlock, MACExecutable, PackedCode, PackTable,
                       DInitCRef,
      {$IFC isSAndE}
                       SoftSystem,
      {$ENDC}
                       UnitLocation, OSData, DebugSymbols, DebugEntry,
                       DebugCommon, EOFMark, UnknownBlock);

      VariantType   = (NoVariant, RefVariant, ShortRef, ModVariant, Comments,
                       SegVariant, UnitVariant, IntfLocVariant, SegLocVariant,
                       UnitLocVariant, OldULVariant, StringVariant,
                       SftSysVariant, JumpTVariant, JTSegVariant, ObjectCode,
                       ProcHeap, OldJumpTV, OldJTSegV, MACJumpTV, Jump4TV,
                       UnknownVariant);

      FileAddr      = longint;
      MemAddr       = longint;
      SegAddr       = longint;

      ObjFileType   = (OldObjFile, IUDirectory, IULibrary, MainProg, IntrinUnit,
                       RegUnit, RegModule, DbgFile, PITFile);
```

```

VersNum       = - 1..127;
VersVal       = RECORD
                   CASE boolean OF
                      true:
                         (VersLong: longint);
                      false:
                         (System, Release, Version, InterFLevel: VersNum)
                END;

ModVal        = RECORD
                   CASE boolean OF
                      true:
                         (ModInt: integer);
                      false:
                         (ImplLevel, FixLevel: VersNum)
                END;

iRefVariant   = SegAddr;
iShortRef     = integer;
iModVariant   = integer;

iSegVariant   = RECORD
                   SegName: NameString;
                   SegNumber: integer;
                   Version1: longint;
                   Version2: longint;
                END;

iUnitVariant  = RECORD
                   UnitName: NameString;
                   UnitNumber: integer;
                   UnitType: integer;
                END;

iIntfLocVariant = RECORD
                     UnitName: NameString;
                     IfLoc: FileAddr;
                  END;

iSegLocVariant = RECORD
                    SegName: NameString;
                    SegNumber: integer;
                    UseDescPtr: longint;
                    UseCount: integer;
                    SysNumber: integer;
                    FileNumber: integer;
                    FileLocation: FileAddr;
                    SizePacked: integer;
                    SizeUnpacked: integer;
                 END;

iOldULVariant = RECORD
                   UnitName: NameString;
                   UnitNumber: integer;
                   FileNumber, UnitType: FileByte;
                   DataSize: longint;
                END;

iUnitLVariant = RECORD
                   UnitName: NameString;
                   UnitNumber: integer;
                   ShrdDataPtr: longint;
                   UseCount, SysNumber, FileNumber, DataPtrIdx,
                   UnitType: integer;
                   DataSize: longint;
                END;

{$IFC isSAndE}
iSftSysVariant = RECORD
                    DSysIdx: integer;
                    DVersId: longint;
                 END;
{$ENDC}

iStringVariant = RECORD
                    ObjectNumber: integer;
                    NameAddr: FileAddr;
                 END;

iJumpTVariant = RECORD
                   JumpL: integer;
                   AbsAddr: MemAddr;
                END;

iOldJumpTV    = RECORD
                   RelOffset: longint;
                   Noop: integer;
                   Jump: integer;
                   PCRel: integer;
                END;

iMACJumpTV    = RECORD
                   RelOffset: integer;
                   MoveW: integer;
                   SegNum: integer;
                   TrapInst: integer;
                END;

iJump4TV      = RECORD
                   JumpPC: integer;
                   RelOffSet: integer;
                END;

iOldJTSegV    = RECORD
                   Addr1: MemAddr;
                   FileLoc: FileAddr;
                   CodeSize: longint;
                   MemLoc: MemAddr;
                   RetAddr: MemAddr;
                   RefCount: longint;
                   ActiveList: MemAddr;
                   Reserved: longint;
                END;

iJTSegVariant = RECORD
                   SegmentAddr: FileAddr;
                   SizePacked: integer;
                   SizeUnpacked: integer;
                   MemLoc: MemAddr;
                END;

iModuleName   = RECORD
                   ModuleName, SegmentName: NameString;
                   CSize: longint;
                END;

iEndBlock     = RECORD
                   CSize: longint;
                END;

iEntryPoint   = RECORD
```

```
                           LinkName, UserName: NameString;
                           Loc: SegAddr;
                        END;

        iExternal     = RECORD
                           LinkName, UserName: NameString;
                        END;

        iStartAddress = RECORD
                           Start: SegAddr;
                           GSize: longint;
                        END;

        iCodeBlock    = RECORD
                           Addr: SegAddr;
                        END;

        iRelocation   = RECORD
                           Reserved: integer;
                        END;

        iCommonRelocation = RECORD
                               CommonName: NameString;
                            END;

        iCommonDefinition = RECORD
                               CommonName: NameString;
                               DSize: longint;
                            END;

        iShortExternal = RECORD
                            LinkName, UserName: NameString;
                         END;

        iDInitDef     = RECORD
                           d_areaName: NameString;
                           DSize: longint;
                        END;

        iDInitData    = RECORD
                           d_areaName: NameString;
                           dAddr: longint;
                           DSize: longint;
                        END;

        iDInitRef     = RECORD
                           d_areaName: NameString;
                        END;

        iDInitDRef    = RECORD
                           d_areaName: NameString;
                           d_Off: longint;
                           rAreaName: NameString;
                        END;

        iQuickLoad    = RECORD
                           StartLoc: SegAddr;
                           DataSize: longint;
                        END;

        iLibModule    = RECORD
                           ModuleName: NameString;
                           ModSize: longint;
                           CodeAddr, TextAddr: FileAddr;
                           TextSize: longint;
                           NrMods: integer;
                        END;

        iLibEntry     = RECORD
                           LinkName: NameString;
                           Module: integer;
                           Address: SegAddr;
                        END;

        iUnitBlock    = RECORD
                           UnitName: NameString;
                           CodeAddr, TextAddr: FileAddr;
                           TextSize, GlobalSize: longint;
                           UnitType: integer;
                        END;

        iInterfLoc    = RECORD
                           Reserved: integer;
                        END;

        iExecutable   = RECORD
                           JTLaddr: MemAddr;
                           JTSize, DataSize, MainSize, JTSegDelta, StkSegDelta,
                           DynStack, MaxStack, MinHeap, MaxHeap: longint;
                        END;

        iOldExecutable = RECORD
                            JTLaddr: MemAddr;
                            JTSize, DataSize: longint;
                         END;

        iPhysicalExec = RECORD
                           JTLaddr: MemAddr;
                           JTSize, DataSize, MainSize, JTSegDelta,
                           StkSegDelta: longint;
                        END;

        iMACExecutable = RECORD
                            JTLaddr: MemAddr;
                            JTSize, DataSize, MainSize, JTSegDelta,
                            StkSegDelta: longint;
                         END;

        iVersionCtrl  = RECORD
                           CASE boolean OF
                              true:
                                 (sysNum, minSys, maxSys, Reserv1, Reserv2,
                                  Reserv3: longint);
                              false:
                                 (FileType: ObjFileType;
                                  ILibNum: VersNum;
                                  Config: VersVal;
                                  ModLevel: ModVal;
                                  IntfTime, ImplTime: longint;
                                  MinConfig, MaxConfig: VersVal);
                        END;

        iSegmentTable = RECORD
                           nSegments: integer;
                        END;

        iUnitTable    = RECORD
                           nUnits, maxunit: integer;
                        END;

        iSegLocation  = RECORD
                           nSegments: integer;
```

```
                    END;

    iOldUntLoc    = RECORD
                       nUnits: integer;
                    END;

    iUnitLocation = RECORD
                       nUnits: integer;
                    END;

    iOSData       = RECORD
                       Reserved: integer;
                    END;

    iStringBlock  = RECORD
                       nStrings: integer;
                    END;

    {$IFC isSAndE}
    iSoftSystem   = RECORD
                       SysNumber: integer;
                       PublicSeg: integer;
                       SysVersion: VersVal;
                       Reserved: longint;
                    END;
    {$ENDC}

    iPackedCode   = RECORD
                       Addr: MemAddr;
                       CSize: longint;
                    END;

    iPackTable    = RECORD
                       packversion: longint;
                    END;

    iDInitCRef    = RECORD
                       d_areaName: NameString;
                       d_Off: longint;
                       entryName: NameString;
                    END;

    iDebugSymbols = RECORD
                       UserName, SegName: NameString;
                       ProcBase, ProcSyms, ProcStmt, ProcNode,
                       UsesSize: longint;
                       HoleBase, HoleTop, MapBase, MapTop: longint;
                       MapName: NameString;
                    END;

    iDebugEntry   = RECORD
                       UserName: NameString;
                       EntrySeg: longint;
                       EntryLoc: SegAddr;
                    END;

    iDebugCommon  = RECORD
                       UnitName: NameString;
                       CommonBase: MemAddr;
                    END;

    iUnknown      = RECORD
                      { nothing here }
                    END;

    ObjBlock      = RECORD
```

```
                           Variant   : VariantType;
                           NrVariants: LongInt;

                           CASE BlockHeader: BlockType OF
                              ModuleName   : (bModuleName   : iModuleName);
                              EndBlock     : (bEndBlock     : iEndBlock);
                              EntryPoint   : (bEntryPoint   : iEntryPoint);
                              External     : (bExternal     : iExternal);
                              StartAddress : (bStartAddress : iStartAddress);
                              CodeBlock    : (bCodeBlock    : iCodeBlock);
                              Relocation   : (bRelocation   : iRelocation);
                              CommonReloc  : (bCommonReloc  : iCommonReloc);
                              CommonDef    : (bCommonDef    : iCommonDef);
                              ShortExternal: (bShortExternal: iShortExternal);
                              DInitDef     : (bDInitDef     : iDInitDef);
                              DInitData    : (bDInitData    : iDInitData);
                              DInitRef     : (bDInitRef     : iDInitRef);
                              DInitDRef    : (bDInitDRef    : iDInitDRef);
                              QuickLoad    : (bQuickLoad    : iQuickLoad);
                              OldExecutable: (bOldExecutable: iOldExecutable);
                              LibModule    : (bLibModule    : iLibModule);
                              LibEntry     : (bLibEntry     : iLibEntry);
                              UnitBlock    : (bUnitBlock    : iUnitBlock);
                              InterfLoc    : (bInterfLoc    : iInterfLoc);
                              PhysicalExec : (bPhysicalExec : iPhysicalExec);
                              MACExecutable: (bMACExecutable: iMACExecutable);
                              Executable   : (bExecutable   : iExecutable);
                              VersionCtrl  : (bVersionCtrl  : iVersionCtrl);
                              SegmentTable : (bSegmentTable : iSegmentTable);
                              UnitTable    : (bUnitTable    : iUnitTable);
                              SegLocation  : (bSegLocation  : iSegLocation);
                              UnitLocation : (bUnitLocation : iUnitLocation);
                              OldUntLoc    : (bOldUntLoc    : iOldUntLoc);
                              StringBlock  : (bStringBlock  : iStringBlock);
                              PackedCode   : (bPackedCode   : iPackedCode);
                              PackTable    : (bPackTable    : iPackTable);
                              DInitCRef    : (bDInitCRef    : iDInitCRef);
                              {$IFC isSAndE}
                              SoftSystem   : (bSoftSystem   : iSoftSystem);
                              {$ENDC}
                              OSData       : (bOSData       : iOSData);
                              DebugSymbols : (bDebugSymbols : iDebugSymbols);
                              DebugEntry   : (bDebugEntry   : iDebugEntry);
                              DebugCommon  : (bDebugCommon  : iDebugCommon);
                              UnknownBlock : (bUnknown      : iUnknownBlock);
                        END;

        ObjVarBlock   = RECORD
                           CASE VarHeader   : VariantType OF
                              RefVariant    : (bRefVariant    : iRefVariant);
                              ShortRef      : (bShortRef      : iShortRef);
                              ModVariant    : (bModVariant    : iModVariant);
                              SegVariant    : (bSegVariant    : iSegVariant);
                              UnitVariant   : (bUnitVariant   : iUnitVariant);
                              IntfLocVariant: (bIntfLocVariant: iIntfLocVariant);
                              SegLocVariant : (bSegLocVariant : iSegLocVariant);
                              UnitLocVariant: (bUnitLVariant  : iUnitLVariant);
                              {$IFC isSAndE}
                              SftSysVariant : (bSftSysVariant : iSftSysVariant);
                              {$ENDC}
                              OldULVariant  : (bOldULVariant  : iOldULVariant);
                              StringVariant : (bStringVariant : iStringVariant);
                              OldJumpTV     : (bOldJumpTV     : iOldJumpTV);
                              MACJumpTV     : (bMACJumpTV     : iMACJumpTV);
                              Jump4TV       : (bJump4TV       : iJump4TV);
                              OldJTSegV     : (bOldJTSegV     : iOldJTSegV);
```

```
                         JumpTVariant  : (bJumpTVariant  : iJumpTVariant);
                         JTSegVariant  : (bJTSegVariant  : iJTSegVariant)
                   END;

   ObjHandle     = ^ObjDesc;

   ObjDesc       = RECORD
                      ObjFile: FileHandle;
                      NextBlock: FileAddr;
                   END;

VAR
   VariantSize:  ARRAY [VariantType] OF integer;
   OIAllowAbort: BOOLEAN;

PROCEDURE InitObjFile(VAR ObjPtr: ObjHandle; nBlocks: integer);

PROCEDURE OpenObjFile(VAR ObjPtr: ObjHandle; FileName: LString;
                      NewFile: boolean);

PROCEDURE ZeroObjEnd(ObjPtr: ObjHandle);

PROCEDURE CloseObjFile(ObjPtr: ObjHandle; Save: boolean);

PROCEDURE GetObjPtr(ObjPtr: ObjHandle; VAR BytePtr: FileAddr);

PROCEDURE GetObjBlockPtr(ObjPtr: ObjHandle; VAR BytePtr: FileAddr);

PROCEDURE SetObjPtr(ObjPtr: ObjHandle; BytePtr: FileAddr);

PROCEDURE SetObjBlockPtr(ObjPtr: ObjHandle; BytePtr: FileAddr);

PROCEDURE SkipObjBytes(ObjPtr: ObjHandle; NrBytes: longint);

PROCEDURE SetObjInvar(VAR B: ObjBlock; InvarType: BlockType;
                      VarSize: longint);

PROCEDURE CopyObjSeq(InObj, OutObj: ObjHandle; NrBytes: integer);

PROCEDURE GetObjInvar(ObjPtr: ObjHandle; VAR Stuff: ObjBlock);

PROCEDURE GetObjVar(ObjPtr: ObjHandle; VarType: VariantType;
                    VAR Stuff: ObjVarBlock);

PROCEDURE GetObjName(ObjPtr: ObjHandle; VAR N: NameString);

PROCEDURE GetObjSeq(ObjPtr: ObjHandle; Stuff: Ptr; NrBytes: integer);

PROCEDURE GetObjByte(ObjPtr: ObjHandle; VAR B: Byte);

PROCEDURE GetObjWord(ObjPtr: ObjHandle; VAR W: integer);

PROCEDURE GetObjLong(ObjPtr: ObjHandle; VAR L: longint);

PROCEDURE PutObjInvar(ObjPtr: ObjHandle; VAR Stuff: ObjBlock);

PROCEDURE PutObjVar(ObjPtr: ObjHandle; VarType: VariantType;
                    VAR Stuff: ObjVarBlock);

PROCEDURE PutObjName(ObjPtr: ObjHandle; N: NameString);

PROCEDURE PutObjSeq(ObjPtr: ObjHandle; Stuff: Ptr; NrBytes: integer);

PROCEDURE PutObjByte(ObjPtr: ObjHandle; B: Byte);

PROCEDURE PutObjWord(ObjPtr: ObjHandle; W: integer);
```

```

   PROCEDURE PutObjLong(ObjPtr: ObjHandle; L: longint);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

```
End of File -- Lines: 532 Characters: 19082
```

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PARTS
*
*******************************************************************************

USES {$U+} PARTS;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/PASDEFS.OBJ *) PasDefs,
         (*$U OBJ/MEMMAN.OBJ  *) MemMan,
         (*$U OBJ/LISTS.OBJ   *) Lists;
      {$ELSEC }
         (*$U OBJ:PASDEFS.OBJ *) PasDefs,
         (*$U OBJ:MEMMAN.OBJ  *) MemMan,
         (*$U OBJ:LISTS.OBJ   *) Lists;
      {$ENDC }

   TYPE


      Class         = integer;
      Member        = integer;
      MemberRec     = RECORD
                         memberOf: Class;
                         nextMember: Member;
                         firstMember: Member;
                      END;
      MemRowPtr     = ^MemRow;
      MemRow        = ARRAY [ - 128..127] OF MemberRec;
      MemArrPtr     = ^MemArr;
      MemArr        = ARRAY [ - 128..127] OF MemRowPtr;
      PRHandle      = ^Partition;
      Partition     = RECORD
                         nClasses, nMembers: longint;
                         classes, members: MemArrPtr;
                      END;
      PartHandle    = ^PartObject;
      PartObject    = RECORD
                         storP: PRHandle;
                         partL: ListHandle;
                      END;

   PROCEDURE NewClass(partH: PartHandle; cl: Class);

   PROCEDURE AddMember(partH: PartHandle; mem: Member; cl: Class);

   PROCEDURE DeleteMember(partH: PartHandle; mem: Member; cl: Class);

   FUNCTION IsEmptyClass(partH: PartHandle; cl: Class): boolean;

   PROCEDURE EachMember(partH: PartHandle; cl: Class; PROCEDURE
                        Visit(m: Member));

   FUNCTION ClassOf(partH: PartHandle; mem: Member): Class;

   PROCEDURE RevMembers(partH: PartHandle; cl: Class);

   PROCEDURE InitPLists;

   FUNCTION MoreMembers(partH: PartHandle; n, newRow: integer): boolean;

   FUNCTION MoreClasses(partH: PartHandle; n, newRow: integer): boolean;

   FUNCTION InitPartition(nCl, nMem: longint; abst: PartHandle): PartHandle;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 78 Characters: 2253

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PASDEFS
*
********************************************************************************

USES {$U+} PASDEFS;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         {$U OBJ/STDUNIT } STDUNIT;
      {$ELSEC }
         {$U OBJ:STDUNIT } STDUNIT;
      {$ENDC }

   {$SETC RANGEF = 0 }
   {$SETC DEBUGF = 0 }
   {$SETC INFOF  = 1 }

   CONST


      NameStrLen    = 8;
      MaxLStringLen = 79;
      Blank         = ' ';
      Empty         = '';

   TYPE
      NameString    = PACKED ARRAY [1..NameStrLen] OF char;
      LString       = String[MaxLStringLen];
      PLString      = ^LString;
      MemPtr        = ^integer;
      ProcPtr       = ^integer;

   VAR
      errors:       integer;
      ListingFile:  Text;
      ListFlag:     boolean;

   PROCEDURE InitPasDefs;

   PROCEDURE PasHalt;

   PROCEDURE SumErrors;

   PROCEDURE Warning(s: LString);

   PROCEDURE Error(s: LString);

   PROCEDURE FatalError(s: LString);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************
```

```
000062
End of File -- Lines: 62 Characters: 1217
```

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PASHEAP
*
*******************************************************************************

USES {$U+} PASHEAP;


INTRINSIC;

INTERFACE

   USES
      (*$U libos/syscall.obj*) syscall;

   TYPE
      PHTB          = - 128..127;
      PHTP          = ^PHTB;

   PROCEDURE PLINITHEAP(VAR ernum, refnum: integer; size, delta: longint;
                        ldsn: integer; swapable: boolean);

   PROCEDURE %_NEW(VAR ptr: PHTP; size: integer);

   PROCEDURE %_NEWL(VAR ptr: PHTP; lsize: longint);

   PROCEDURE %_MARK(VAR ptr: PHTP);

   PROCEDURE %_RELSE(VAR ptr: PHTP);

   FUNCTION %_MEMAV: LONGINT;

   FUNCTION %_HHeapRes: integer;

   PROCEDURE %_pphhpreinit;

   PROCEDURE %_pphhreinit;

   FUNCTION %_phwordsavail: longint;

   PROCEDURE %_disp(VAR ptr: PHTP; size: longint);

*******************************************************************************
*
*                              THAT'S ALL FOLKS ...
*
*******************************************************************************


End of File -- Lines: 49 Characters: 1057
```

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PMDECL
*
********************************************************************************

USES {$U+} PMDECL;


INTRINSIC;

INTERFACE

   USES
      {$U libos/syscall } syscall;

   CONST
      ThreshHold    = 8;

   TYPE
      TNibble       = 0..15;
      TBootStuff    = RECORD
                         ExtendMem: Boolean;
                         BootVol: TNibble;
                      END;
      TOutPutStuff  = RECORD
                         NormCont: TNibble;
                         DimCont: TNibble;
                         BeepVol: TNibble;
                         FadeDelay: TNibble;
                      END;
      TInputStuff   = RECORD
                         BeginRepeat: TNibble;
                         SubRepeat: TNibble;
                         DoubleClick: TNibble;
                      END;
      TInternalStuff = RECORD
                          Mem_Loss: integer;
                          Mouseon: Boolean;
                          ScaleMouse: Boolean;
                       END;
      TypeConnect   = string[32];
      Ch_info       = RECORD
                         version: integer;
                         driver_id: longint;
                         bootable: boolean;
                         preload: boolean;
                         permanent: boolean;
                         devicetype: devtype;
                         size_exten: integer;
                         def_cdinfo: ARRAY [1..3] OF integer;
                         removable: boolean;
                         ejectable: boolean;
                         fs_start_block: longint;
                         start_block: longint;
                         numattach: integer;
                         numConnectors: integer;
                      END;
      cddEntry      = RECORD
                         drvr_name: e_name;
                         active: integer;
                         info: ch_info;
                      END;
      cddHeader     = RECORD
                         version: integer;
                         cdcount: integer;
                      END;
      systemCdd     = RECORD
                         descr: cddheader;
                      END;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 77 Characters: 2435

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PMM
*
*******************************************************************************

USES {$U+} PMM;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libhw/hwint} HWInt,
      {$U libos/Syscall} SysCall,
      {$U libsm/UnitStd} UnitStd,
      {$U libpm/PMDecl} PMDecl;
      {$SETC DOPXPM := FALSE}

   VAR
      PMBeep_Flash: TNibble;

   PROCEDURE PmInit(VAR error: integer);

   PROCEDURE PMReadConfig(VAR error: integer; VAR NextEntry: longint;
                          VAR Config: ConfigDev);

   PROCEDURE PMWriteConfig(VAR error: integer; VAR NextEntry: longint;
                           VAR Config: ConfigDev);

   PROCEDURE PMReadInput(VAR InputInfo: TInputStuff);

   PROCEDURE PMWriteInput(InputInfo: TInputStuff);

   PROCEDURE PMReadOutPut(VAR OutputInfo: TOutputStuff);

   PROCEDURE PMWriteOutPut(OutputInfo: TOutputStuff);

   PROCEDURE PMReadBoot(VAR BootInfo: TBootStuff);

   PROCEDURE PMWriteBoot(BootInfo: TBootStuff);

   PROCEDURE PMWrite;

   FUNCTION DimConvert(Contrast: Integer): Integer;

   FUNCTION ConvertCont(Contrast: Integer): Integer;

   PROCEDURE PMReadDevolp(VAR IntrnlInfo: TInternalStuff);

   PROCEDURE PMWriteDevolp(IntrnlInfo: TInternalStuff);

   PROCEDURE DefaultPM;

   PROCEDURE DefaultConfig;
   {$IFC DOPXPM}

   PROCEDURE PxPm;
   {$ENDC}

```

```
*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 67 Characters: 1564

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

********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : POOLER
*
********************************************************************************

USES {$U+} POOLER;


INTRINSIC;

INTERFACE

   USES
      {$IFC OSBuilt }
         {$U libsm/unitstd.obj} unitstd,
         {$U libdb/dbenv.obj } dbenv,
         {$U libin/INTRLIBp.obj } international,
         {$U libdb/dbdecl1.obj } dbdecl1,
         {$U libos/syscall.obj } syscall,
         {$U libos/psyscall.obj } psyscall,
         {$U libdb/lowlevel.obj } lowlevel;
      {$ELSEC }
         {$U OBJ:dbenv.obj } dbenv,
         {$U INTRLIB.obj } international,
         {$U OBJ:dbdecl1.obj } dbdecl1,
         {$U OBJ:syscall.obj } syscall,
         {$U OBJ:lowlevel.obj } lowlevel;
      {$ENDC }

   Const


      {$SETC NOSWAP := FALSE }
      {$SETC debug := FALSE }
      {$IFC PRERELEASE }
      {$SETC debugpool := TRUE }
      {$ELSEC }
      {$SETC debugpool := FALSE }
      {$ENDC}
      V0myprod_vers = 0;
      myprod_vers   = 1;
      nilptr        = - 1;
      nilval        = - 1;
      fisbtree      = 0;
      fhasnull      = 1;
      fmaster       = 2;
      fdetail       = 3;
      ffileok       = 4;
      fdata         = 5;
      writenon      = 5;
      noerr         = 0;
      notfound      = 1;
      duplkey       = 2;
      eos           = 3;
      fileempty     = 4;
      nobuffs       = 3404;
      ioerr         = 3405;
      noroom        = 3406;
      inconsistent  = 3407;
      badscanid     = 3408;
      ticketoverflow = 3409;
      ticketchanged = 3410;
      badrecaddress = 3412;
      oddrecsize    = 3413;
      badnsearch    = 3414;
      badwhich      = 3415;
      illegalscan   = 3416;
      rectoobig     = 3417;
      notdbfile     = 3418;
      wrongfile     = 3419;
      badversion    = 3420;
      hasschema     = 3421;
      badpagesize   = 3425;
      DBVersionMismatch = 3426;
      DBTooOld      = 3427;
      DBTooNew      = 3428;
      empty         = 0;
      clean         = 1;
      dirty         = 2;
      readonly      = 0;
      update        = 1;
      labelreadonly = 2;
      labelupdate   = 3;
      first         = 0;
      last          = 1;
      approx        = 2;
      exact         = 3;
      current       = 4;
      prior         = 5;
      next          = 6;
      touch_by_vltree = 0;
      touch_delete  = 1;
      untouched     = 2;
      leaftype      = - 26215;
      nonleaftype   = - 26729;
      labeltype     = - 25701;
      freetype      = - 25187;
      labelkeys     = 3;
      labelflds     = 4;
      labelbnd      = 3;

   TYPE
      uniqueid      = uid;
      filelabel     = RECORD
                         check_1_it: integer;
                         DBvers: byte;
                         USERvers: byte;
                         updateversion: integer;
                         flags: elemset;
                         firstpage, lastpage: fileptr;
                         nrecs: fileptr;
                         pagesize: integer;
                         nfields: integer;
                         nkeys: integer;
                         nsort: integer;
                         vflds: integer;
                         kvflds: integer;
                         depth: integer;
                         root: fileptr;
                         freelist: fileptr;
                         labellist: fileptr;
                         labeltickets: fileptr;
                         tickets: ticket;
                         ticketfld: integer;
                         funiqueid: uniqueid;
                         check_2_it: integer;
```

```
                END;
pagedesc      = RECORD
                   nkeys: integer;
                   ptype: integer;
                   CASE boolean OF
                      true:
                         (prior: fileptr;
                          next: fileptr);
                      false:
                         (free: fileptr; )
                END;
ptrpagedesc   = ^pagedesc;
vacttype      = (undone, del, split, split_replace, rot, rot_replace,
                 irot, irot_replace, merge, merge_replace, NoSibEmpty);
stackrec      = RECORD
                   numkeys, index: integer;
                   page: fileptr;
                   rec_lpage, rec_rpage: fileptr;
                   CASE vact: vacttype OF
                      irot, irot_replace:
                         (unrotPt: integer;
                          insindex: integer);
                      del, rot, merge, NoSibEmpty:
                         (delsize: integer;
                          delrec: ptrdata);
                END;
ptrstackrec   = ^stackrec;
pagestack     = ARRAY [0..0] OF stackrec;
ptrpagestack  = ^pagestack;
labelentry    = RECORD
                   ltype: integer;
                   enum: fileptr;
                   seq: integer;
                   varfld: vfld
                END;
ptrlabelentry = ^labelentry;
offsets       = ARRAY [0..0] OF integer;
ptroffsets    = ^offsets;
ptrscandesc   = ^scandesc;
ptrfiledesc   = ^filedesc;
filedesc      = RECORD
                   check_1_it: integer;
                   DBvers: byte;
                   USERvers: byte;
                   updateversion: integer;
                   flags: elemset;
                   firstpage, lastpage: fileptr;
                   nrecs: fileptr;
                   pagesize: integer;
                   nfields: integer;
                   nkeys: integer;
                   nsort: integer;
                   vflds: integer;
                   kvflds: integer;
                   depth: integer;
                   root: fileptr;
                   freelist: fileptr;
                   labellist: fileptr;
                   labeltickets: fileptr;
                   tickets: ticket;
                   ticketfld: integer;
                   funiqueid: uniqueid;
                   check_2_it: integer;
                   openid: uid;
                   pnext: integer;
                   pkeyfiles: integer;
```

```
                           readers, writers: integer;
                           fstatus: integer;
                           scans: integer;
                           frefno: integer;
                           pedesc: ptrrecptr;
                           pkdesc: ptrrecptr;
                           pfldset: ptrbits;
                           firstvfld: integer;
                           lastvfld: integer;
                           lastvsfld: integer;
                        END;
        scandesc      = RECORD
                           onfile, viafile: integer;
                           fnext: integer;
                           touched: integer;
                           intent: integer;
                           position: integer;
                           ptrcos: ptrdata;
                           cossize: integer;
                           keypage: fileptr;
                           qpage: fileptr;
                           keyindex: integer;
                           qindex: integer;
                           ufldset: bits;
                        END;
        lmhandle      = fint8;
        diffdesc      = RECORD
                           diffid: integer;
                           realid: integer;
                           on, via: integer;
                           vpage: fileptr;
                           vindex: integer;
                           curhandle: lmhandle;
                           have_an_update, reversescan: boolean;
                           whichway: integer;
                           pdfilter: ptrdata;
                           pkfilter: ptrdata;
                           plow, phigh: ptrdata;
                           nlflds, nhflds: integer;
                           CheckPtSize: integer;
                           nullset: bits;
                        END;
        ptrdiffdesc   = ^diffdesc;
        QChkPtDesc    = RECORD
                           CPtSize: integer;
                           CPtLowSize: integer;
                           CPtHighSize: integer;
                           CPtDfilterSize: integer;
                           CPtKfilterSize: integer;
                           CPtnhflds, CPtnlflds: integer;
                           CPtwhichway: integer;
                           CPtvpage: fileptr;
                           CPtvindex: integer;
                           CPtcurhandle: lmhandle;
                           CPtreversescan: boolean;
                           hasVia: boolean;
                           Clear: boolean;
                           Good: boolean;
                           viaUniqueid: uniqueid;
                        END;
        ptrQChkPtDesc = ^QChkPtDesc;
        buffdesc      = RECORD
                           pbuff: ptrdata;
                           state: integer;
                           locks: integer;
                           pfile: integer;
```

```
                      page: fileptr;
                      priority: integer;
                   END;
   pool          = ARRAY [0..0] OF buffdesc;
   ptrpool       = ^pool;
   filetable     = ARRAY [0..0] OF ptrfiledesc;
   ptrfiletable  = ^filetable;
   scantable     = ARRAY [0..0] OF ptrscandesc;
   ptrscantable  = ^scantable;
   difftable     = ARRAY [0..0] OF ptrdiffdesc;
   ptrdifftable  = ^difftable;

VAR
   {$IFC PRERELEASE }
   pdebug:       boolean;
   {$ENDC}
   psize:        integer;
   popenobj:     integer;
   pfiletable:   ptrfiletable;
   nfiles:       integer;
   pscantable:   ptrscantable;
   nscans:       integer;
   pdifftable:   ptrdifftable;
   ndiffs:       integer;
   lab_desc:     ARRAY [0..labelbnd] OF flddesc;
   pstack:       ptrpagestack;
   curdepth, maxdepth: integer;
   pbuff_segment: longint;
   buff_segment: integer;
   buffpool:     ptrpool;
   ticks:        integer;
   lastbuff:     integer;
   uiddesc:      flddesc;
   isleaf, isnonleaf, isroot: integer;
   dont_use_freelist: boolean;
   {$IFC DEBUGPOOL }
   labwrites, datawrites, datareads: integer;
   {$ENDC }
   {$IFC PRERELEASE }

PROCEDURE dumpbuff(first, last: integer);
{$ENDC }

PROCEDURE lockbuff(VAR presult: integer; VAR buffer: integer);

PROCEDURE relbuff(VAR presult: integer; buffer: integer;
                  relpriority: integer);

PROCEDURE dowritenon(VAR presult: integer; fileid: integer);

PROCEDURE getpage(VAR presult: integer; VAR buffer: integer; fileid: integer;
                  rpage: fileptr; noinconsistent: boolean);

PROCEDURE putpage(VAR presult: integer; buffer: integer; fileid: integer;
                  wpage: fileptr);

PROCEDURE checkpt(VAR presult: integer; fileid: integer; newstate: integer);

PROCEDURE doclose(VAR presult: integer; fileid: integer);

FUNCTION no_room(presult: integer): boolean;

PROCEDURE force_out(VAR presult: integer; fileid: integer; wpage: fileptr);

PROCEDURE PageInvalidates(fileid: integer);

```

```
*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 331 Characters: 11109

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRBUF
*
*******************************************************************************

USES {$U+} PRBUF;


INTRINSIC;

INTERFACE

   USES
      {$U LibOS/SysCall      } SysCall,
      {$U LibPM/PMDecl       } PMDecl,
      {$U LibSM/UnitStd      } UnitStd,
      {$U LibSM/UnitHz       } UnitHz,
      {$U LibQD/Storage      } Storage,
      {$U LibQD/QuickDraw    } QuickDraw,
      {$U LibFM/FontMgr      } FontMgr,
      {$U LibPr/PrStdInfo    } PrStdInfo;

   TYPE
      TPbCmd        = (pbCmdText, pbCmdHRule, pbCmdVRule);
      TPbCommand    = RECORD
                         h: TC;
                         v: TC;
                         CASE pbCmd: TPbCmd OF
                            pbCmdText:
                               (cFont: TC;
                                seteface: Style;
                                cFirstChar: TC;
                                cLength: TC);
                            pbCmdHRule:
                               (cWidth: TC);
                            pbCmdVRule:
                               (cHeight: TC);
                      END;
      TPbData       = PACKED ARRAY [0..32000] OF CHAR;
      TPPbData      = ^TPbData;
      THPbData      = ^TPPbData;
      TPbProcs      = ARRAY [TPbCmd] OF TProc;
      TPbInstall    = RECORD
                         chkProcs: TPbProcs;
                         dumpProcs: TPbProcs;
                         scanLine: TProc;
                      END;

   VAR
      rPrBounds:    Rect;
      HOffSet, VOffSet: Integer;

   PROCEDURE PbClose;

   PROCEDURE PbDumpBand;

   PROCEDURE PbHRule(cLength: TC);

   PROCEDURE PbOffSet(HOff, VOff: Integer);

   PROCEDURE PbOpen(install: TPbInstall);

   PROCEDURE PbText(cCount: TC; p: TP);

   PROCEDURE PbVRule(cLength: TC);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 73 Characters: 2018

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRDLGMGR
*
*******************************************************************************

USES {$U+} PRDLGMGR;


INTRINSIC;
   {$SETC fBtnRead  = FALSE}

INTERFACE

   USES
      {$U LibOS/SysCall    } SysCall,
      {$U LibSM/UnitStd    } UnitStd,
      {$U LibSM/UnitHz     } UnitHz,
      {$U LibHW/HWInt      } HWInt,
      {$U LibQD/QuickDraw  } QuickDraw,
      {$U LibFM/FontMgr    } FontMgr,
      {$U LibQD/Storage    } Storage,
      {$U LibWM/Events     } Events,
      {$U LibWM/Folders    } Folders,
      {$U LibSB/WmlStd     } WmlStd,
      {$U LibSB/WmlCrs     } WmlCrs,
      {$U LibPM/PMDecl     } PMDecl,
      {$U libPr/PrStdInfo  } PrStdInfo;

   CONST
      cInfinite     = $7FFFFFFF;
      cZero         = 0;
      cCursorDelay  = 3500;
      cBtnMax       = 31;
      cMuMax        = 31;
      idMuMax       = 15;
      iBtnNil       = - 1;
      iMuNil        = - 1;
      iBtnStd       = - 2;
      dhCkfmMarg    = 6;
      dvCkfmMarg    = 4;
      dhSpToCk      = 4;
      dhSpToR       = 3;
      dvSpToR       = 1;
      cChNrMax      = 4;
      cChSpMax      = 15;
      dtDlgEnd      = 25;
      dtDlgMin      = 25;
      ckfmMax       = 5;

   TYPE
      TDBox         = RECORD
                           fKeyDlg: TF;
                           timeStrt: TL;
                           procEvent: TProc;
                           procIdle: TProc;
                           procEnd: TProc;
                        END;
      TEbtn         = (ebtnSq, ebtnRR, ebtnOv);
      TEqd          = (eqdLI, eqdCI, eqdRI, eqdRO);
      TEact         = (eactIn, eactCk, eactNr, eactSp, eactIc);
```

```
       TEnib         = (enibSm, enibLg);
       TRgnib        = ARRAY [TEnib] OF Point;
       TLx           = PACKED RECORD
                          fam: 0..31;
                          fB: TF;
                          fI: TF;
                          eact: TEact;
                          eqd: TEqd;
                          ebtn: TEbtn;
                          enib: TEnib;
                          fBtnVis: TF;
                          mask: CHAR;
                          smax: 0..255;
                       END;
       TBtn          = RECORD
                          r: Rect;
                          lx: TLx;
                          hsp: THsp;
                       END;
       TRgbtn        = ARRAY [0..cBtnMax] OF TBtn;
       TIdMu         = 0..idMuMax;
       TSetIdMu      = SET OF TIdMu;
       THmu          = ^TPmu;
       TPmu          = ^TMu;
       TMu           = RECORD
                          rBB: Rect;
                          idMu: TIdMu;
                          iBtnOn: TB;
                          iBtnUsr: TB;
                          cBtn: TB;
                          fMuVis: TF;
                          fMUActv: TF;
                          rgbtn: TRgbtn;
                       END;
       TRghmu        = ARRAY [0..cMuMax] OF THmu;
       THckfm        = ^TPckfm;
       TPckfm        = ^TCkfm;
       TCkfm         = RECORD
                          rBB: Rect;
                          iMuKey: TB;
                          iBtnKey: TB;
                          cMu: TB;
                          rghmu: TRghmu;
                       END;
       TCkfmseg      = RECORD
                          ckfm: ARRAY [0..ckfmMax] OF THckfm;
                       END;

    VAR
       dbox:         TDBox;
       setIdMuAll:   TSetIdMu;
       hspNil:       THsp;
       finfoLx:      FontInfo;
       rgnib:        TRgnib;
       pPrRecUsr:    TPprRec;
       hckfmPrCk:    THckfm;
       procCkCmd:    TProc;

    PROCEDURE DlgMgrInit;

    PROCEDURE PauseEvt(dt: TC);

    FUNCTION FMseInFldr(pwnd: WindowPtr; VAR pt: Point): TF;

    PROCEDURE PrStrToNum(spNum: TSp; VAR c: TC);

```

```
PROCEDURE PrNumToStr(c: TC; VAR spNum: TSp);

PROCEDURE PrCkEvent(event: EventRecord);

PROCEDURE PrCkTrkMse;

PROCEDURE CaretMove(pwnd: WindowPtr; h, v: TC);

PROCEDURE CaretSynch;

PROCEDURE CaretKill;

PROCEDURE CaretFlip;

PROCEDURE CaretBlink;

PROCEDURE DlgInstall(fKey: TF; pEvent, pIdle, pEnd: TProc);

PROCEDURE DlgEvent(event: EventRecord);

PROCEDURE DlgIdle;

PROCEDURE DlgEnd(fDismiss: TF);

PROCEDURE DlgOpen(dv: TC);

FUNCTION FDlgEvent(VAR event: EventRecord): TF;

PROCEDURE DlgMain;

FUNCTION FDlgDone: TF;

FUNCTION FCkfmSelect(hckfm: THckfm; ptMse: Point; VAR iMu, iBtn: TB): TF;

PROCEDURE CkFmSetUp(hckfm: THckfm; fInit: TF; setIdMuActv: TSetIdMu);

PROCEDURE CkfmShow(hckfm: THckfm; fInit: TF);

PROCEDURE CkfmTrkMse(hckfm: THckfm; pWnd: WindowPtr);

PROCEDURE CkfmKeys(hckfm: THckfm; iMu, iBtn: TB);

PROCEDURE CkfmAddCh(hckfm: THckfm; ch: CHAR);

FUNCTION FCkfmNxtKey(hckfm: THckfm; VAR iMuNxt, iBtnNxt: TB): TF;

PROCEDURE CkfmSp(hckfm: THckfm; iMu, iBtn: TB; VAR sp: TSp; fNew: TF);

FUNCTION FCkfmHit(hckfm: THckfm; iMu, iBtn: TB): TF;
{$IFC fBtnRead}

FUNCTION HspAlloc(sp: TSp; cChMax: TC): THsp;

PROCEDURE HspFree(VAR hsp: THsp);

FUNCTION HCkfmRead(VAR ckFile: Text): THckfm;
{$ENDC}

PROCEDURE MuShow(hmu: THmu);

PROCEDURE MuErase(hmu: THmu);

PROCEDURE MuPush(hmu: THmu; iBtn: TB);

FUNCTION FMuHit(hmu: THmu; ptMse: Point; VAR iBtn: TB): TF;

```

```
   PROCEDURE BtnDraw(btn: TBtn);

   PROCEDURE BtnErase(btn: TBtn);

   PROCEDURE BtnFill(btn: TBtn; fFlip: TF);

   PROCEDURE BtnPtSp(btn: TBtn; VAR ptSp: Point; VAR dhSp: TC);

   PROCEDURE BtnWrite(btn: TBtn; sp: TSp);

   PROCEDURE BtnAddCh(btn: TBtn; ch: CHAR);

   PROCEDURE BtnBkSp(btn: TBtn);

   PROCEDURE BtnBind(VAR lx: TLx);

   PROCEDURE BtnSetSp(btn: TBtn; sp: TSp; showIt: TF);

   PROCEDURE BtnReappear(hmu: THmu; iBtn: TB);

   PROCEDURE BtnVanish(hmu: THmu; iBtn: TB);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 221 Characters: 5648

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

```
******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PREVENTS
*
******************************************************************************

USES {$U+} PREVENTS;


INTRINSIC;

INTERFACE

   USES
      {$U LibSM/UnitStd    } UnitStd,
      {$U LIbOS/SysCall    } SysCall,
      {$U LibPM/PMDecl     } PMDecl,
      {$U LibSM/UnitHz     } UnitHz,
      {$U LibQD/Storage    } Storage,
      {$U LibQD/QuickDraw  } QuickDraw,
      {$U LibQD/GrafUtil   } GrafUtil,
      {$U LibFM/FontMgr    } FontMgr,
      {$U LibPr/PrStdInfo  } PrStdInfo,
      {$U LibPr/PrStdProcs } PrStdProcs;

   CONST
      ECNameSize    = 8;
      maxstuff      = 38;

   TYPE
      PrMessages    = (ENull, EWake, EDone, EPrintRequest, EPrintDialog,
                       ESettingsDialog, EPrListDialog, EMonitorDialog,
                       ESendData, ETalkToMe, EMetrics, EJobCount,
                       EPrintTerminate, EReply, EAbort, EStuffed, EConfigure);
      PrEventBlk    = RECORD
                         event_header: t_eheader;
                         CASE INTEGER OF
                            0:
                               (event_text: t_event_text);
                            1:
                               (eventBytes: ARRAY [0..39] OF - 128..127);
                            2:
                               (eventInts: ARRAY [0..19] OF INTEGER);
                            3:
                               (eventLongs: ARRAY [0..9] OF LONGINT);
                      END;
      Channel       = RECORD
                         ChName: Pathname;
                         RefNum: INTEGER;
                         Exname: t_ex_name;
                         EvBlock: PrEventBlk;
                         Interval: timestmp_interval;
                         ClkTime: time_rec;
                         Waits: t_waitlist;
                      END;

   FUNCTION PrAccept(VAR Chan: Channel): PrMessage;

   PROCEDURE PrCall(VAR cError: TC; PrID: TL; Message: PrMessages; mess1, mess2,
                    mess3, mess4: TL);

   PROCEDURE PrCloseChan(VAR Chan: Channel);

   PROCEDURE PrDeferToBgd(PrID: TL; Message: PrMessages; mess1, mess2, mess3,
                          mess4: TL);

   PROCEDURE PrIDtoPath(PrID: TL; VAR path: PathName);

   PROCEDURE PrOffer(VAR Chan: Channel; Message: PrMessages);

   PROCEDURE PrOpenChan(VAR Chan: Channel; Rx: TF);

   PROCEDURE PrOpenCall(VAR Chan: Channel; ExProc: LongAdr);

   PROCEDURE PrJobCount(PrID: TL; VAR cjob: INTEGER);

   PROCEDURE PrNotify(PrID: TL; Message: PrMessages; mess1, mess2, mess3,
                      mess4: TL);

   PROCEDURE PrPrintRequest(PrID: TL; lFileName, ncopies, PrntrID, PortID: TL);

   PROCEDURE PrPrintStop(PrID: TL; lFileName, ncopies, PrntrID, PortID: TL);

   PROCEDURE PrSendMessage(VAR cError: INTEGER; VAR Chan: Channel;
                           Message: PrMessages);

   FUNCTION PrWaitMessage(VAR Chan: Channel): PrMessage;

   PROCEDURE PrStuffData(PrID: TL; dataPtr: TP; nbytes: TC);

   PROCEDURE PrUpdate;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 98 Characters: 3195

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRFILEPR
*
*******************************************************************************

USES {$U+} PRFILEPR;


INTRINSIC;

INTERFACE

   USES
      {$U LibOS/SysCall   } SysCall,
      {$U LibPM/PMDecl    } PMDecl,
      {$U LibSM/UnitStd   } UnitStd,
      {$U LibSM/UnitHz    } UnitHz,
      {$U LibQD/Storage   } Storage,
      {$U LibQD/QuickDraw } QuickDraw,
      {$U LibFM/FontMgr   } FontMgr,
      {$U LibPr/PrStdInfo } PrStdInfo,
      {$U LibPr/PrStdProcs} PrStdProcs;

   CONST
      pfPassword    = - 13264;
      pfPicHRule    = 20;
      pfPicVRule    = 21;
      pfPicTab      = 22;
      HeapSize      = 12000;

   TYPE
      TStr80        = String[80];

   PROCEDURE OpenBlock(VAR error: Integer; VAR FileRec: TPrFileRec;
                       access: MSet);

   PROCEDURE WriteBlock(VAR error: Integer; p: TP; lCount: LongInt;
                        RefNum: Integer);

   PROCEDURE ReadBlock(VAR error: Integer; p: TP; lCount: LongInt;
                       RefNum: Integer);

   PROCEDURE PfSetPos(VAR error: Integer; lPos: TL; RefNum: Integer);

   PROCEDURE PfSavePos(VAR error: Integer; VAR lPos: TL; RefNum: Integer);

*******************************************************************************
*
*                              THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 53 Characters: 1424

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PROGCOMM
*
*******************************************************************************

USES {$U+} PROGCOMM;


INTRINSIC;

INTERFACE

   USES
      {$U StdUnit   } StdUnit,
      {$U ShellComm } ShellComm;

   CONST
      PCNone        = - 1;
      PCAny         = 0;
      PCText        = 1;
      PCBufrMax     = 1023;
      PC_SetReallyStop = 1;
      PC_GetReallyStop = 2;
      PC_SetUnSavedEdits = 6960;
      PC_GetUnSavedEdits = 8751;

   TYPE
      PCBufrP       = ^PCBufr;
      PCBufr        = PACKED ARRAY [0..PCBufrMax] OF CHAR;

   VAR
      PCBufrPtr:    PCBufrP;

   PROCEDURE PCInit;

   PROCEDURE PCSetRunCmd(RC: SUStr);

   PROCEDURE PCSetRetStr(RS: SUStr);

   PROCEDURE PCReWrite(WriteType: INTEGER; Key: SUStr);

   FUNCTION PCReset(ReadType: INTEGER; Key: SUStr): BOOLEAN;

   FUNCTION PCClose(KillBufr: BOOLEAN; Key: SUStr): BOOLEAN;

   FUNCTION PCPutCh(Ch: CHAR): BOOLEAN;

   FUNCTION PCGetCh(VAR Ch: CHAR): BOOLEAN;

   FUNCTION PCPutLine(L: SUStr): BOOLEAN;

   FUNCTION PCGetLine(VAR L: SUStr): BOOLEAN;

   FUNCTION PCShellCmd(Cmd: INTEGER; P: SUStrP): BOOLEAN;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************
```

```
000062

End of File -- Lines: 62 Characters: 1355
```

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRPUBLIC
*
*******************************************************************************

USES {$U+} PRPUBLIC;


INTRINSIC;

INTERFACE

   USES
      {$U LibOS/SysCall   } SysCall,
      {$U LibPM/PMDecl    } PMDecl,
      {$U LibSM/UnitStd   } UnitStd,
      {$U LibSM/UnitHz    } UnitHz,
      {$U LibQD/Storage   } Storage,
      {$U LibQD/QuickDraw } QuickDraw,
      {$U LibFM/FontMgr   } FontMgr,
      {$U LibPr/PrStdInfo } PrStdInfo;

   PROCEDURE PsYieldCpu;

   PROCEDURE PrGlobalInit;

   FUNCTION FPrPrfValid(VAR prprf: TPrRec): TF;

   PROCEDURE PrPrfDefault(VAR prprf: TPrRec);

   FUNCTION fPrPrfEq(VAR prprf1, prprf2: TPrRec): TF;

   FUNCTION PrDocStart(VAR PrRec: TPrRec; VAR PrPort: TPrPort; LDSN: INTEGER;
                       DoSpoolAlert: Boolean): Boolean;

   FUNCTION PrPageStart(VAR prinsIn: TPrRec; VAR PrPort: TPrPort): TF;

   FUNCTION PrPageEnd(VAR prinsIn: TPrRec; VAR PrPort: TPrPort): TF;

   PROCEDURE PrDocEnd(VAR prinsIn: TPrRec; VAR PrPort: TPrPort);

   PROCEDURE PrMgrInit;

   PROCEDURE PrPrfDlg(VAR prprf: TPrRec; VAR fNewStl: TF; fPgSzSuppress: TF);

   FUNCTION FPrInsDlg(VAR prprf: TPrRec; VAR fNewStl: TF;
                      prmode: PrMenuSuppress): TF;

   PROCEDURE PrSpoolAbort;

   PROCEDURE PrBgdDlg;

   PROCEDURE PrDlgDefault(fPgSzSuppress: TF; Prmode: PrMenuSuppress; MaxHeight,
                          MaxWidth, MinHeight, MinWidth: TC);

   PROCEDURE PrLFntID(VAR prprf: TPrRec; famIn: TC; setIn: Style;
                      VAR lFntid: TLFntID);

*******************************************************************************
*
```

```
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

End of File -- Lines: 65 Characters: 1780
```

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRSTDINF
*
********************************************************************************

USES {$U+} PRSTDINF;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U LibOS/SysCall   } SysCall,
      {$U LibPM/PMDecl    } PMDecl,
      {$U LibSM/UnitStd   } UnitStd,
      {$U LibSM/UnitHz    } UnitHz,
      {$U LibQD/Storage   } Storage,
      {$U LibQD/QuickDraw } QuickDraw,
      {$U LibFM/FontMgr   } FontMgr;

      {$setc doTracePR := TRUE}
      {$setc fTracePR  := doTracePR and fTrace}

      {$SETC PrDebug    = fdbgok}
      {$SETC PrSymbols  = fsymok}

   CONST
      cPrPrfVersion = 11;
      prdVersion    = 1;
      cbPrPrfMax    = 64;
      PrPageMax     = $07FFF;
      LastBlock     = 9999;
      iPgOther      = 0;
      iPg8x11       = 1;
      iPg8x14       = 2;
      iPg14x11      = 3;
      iPgA4         = 4;
      iPg210x12     = 5;
      iPg310x8      = 6;
      iPgSpec       = 7;
      iPgFst        = 0;
      iPgMax        = 15;
      prPgFract     = 120;
      prNILProcess  = - 1;
      prNilRefNum   = - 1;
      prNilPrinter  = - 1;
      prNilPort     = - 1;
      prErrAbort    = 3050;
      prErrPrShop   = 3068;
      NoPRD         = 3055;
      NoPMDrivers   = 3056;
      PMbutNoPRD    = 3057;
      NoPrProcess   = 3058;
      NoBtnFile     = 3059;
      PrParentProcess = 'System.Print';
      spPrAlrt      = 'System.Pr.';
      sParentSeg    = 'System.PrData';
      ldsnshared    = 12;
      ldsndialog    = 9;
```

```

  TYPE
     T1            = 0..1;
     T4            = 0..15;
     T8            = - 128..127;
     T16           = PACKED RECORD
                        CASE INTEGER OF
                           0:
                              (by1, by0: - 128..127);
                           1:
                              (c1, c0: CHAR);
                           2:
                              (usb1, usb0: 0..255);
                           3:
                              (sb1, sb0: - 128..127);
                           4:
                              (n3, n2, n1, n0: T4);
                           5:
                              (f15, f14, f13, f12, f11, f10, f9, f8, f7, f6,
                               f5, f4, f3, f2, f1, f0: BOOLEAN);
                           6:
                              (b15, b14, b13, b12, b11, b10, b9, b8, b7, b6,
                               b5, b4, b3, b2, b1, b0: T1);
                           7:
                              (i: INTEGER);
                     END;
     T32           = PACKED RECORD
                        CASE INTEGER OF
                           0:
                              (i1, i0: T16);
                           1:
                              (l: LONGINT);
                           2:
                              (p: TP);
                           3:
                              (h: TH);
                           6:
                              (aby: PACKED ARRAY [0..3] OF - 128..127);
                     END;
     TDirection    = (Portrait, LandScape);
     TPgSize       = RECORD
                        PaperDirect: TDirection;
                        PrintDirect: TDirection;
                        Width: INTEGER;
                        Height: INTEGER;
                        Extend: LongInt;
                     END;
     TDlgPgSz      = RECORD
                        pgSzBtn: INTEGER;
                        widInch: INTEGER;
                        htInch: INTEGER;
                        wid16th: INTEGER;
                        ht16th: INTEGER;
                        widMM: INTEGER;
                        htMM: INTEGER;
                     END;
     TPrJobInfo    = RECORD
                        DocPgFst: Integer;
                        DocPgLst: Integer;
                        SpooledPages: Integer;
                        Copies: Integer;
                        Immediate: Boolean;
                        Extend: LongInt;
                     END;
     TPrInfo       = RECORD
                        ADev: TADev;
```

```
                   PrinterID: Integer;
                   NumColor: Integer;
                   HRes, VRes: Integer;
                   rPaper: Rect;
                   rPrintable: Rect;
                   PgSize: TPgSize;
                   Port: Integer;
                   DlgPgSz: TDlgPgSz;
                   Extend: LongInt;
                END;
TPrDevInfo    = ARRAY [1..128] OF - 128..127;
TPrPort       = RECORD
                   gport: GrafPort;
                   gProcs: QDProcs;
                   Extend: LongInt;
                END;
TpPrRec       = ^TPrRec;
ThPrRec       = ^TPPrRec;
TPrRec        = RECORD
                   PrVersion: Integer;
                   PrLDSN: Integer;
                   PrJobInfo: TPrJobInfo;
                   PrInfo: TPrInfo;
                   PrDevInfo: TPrDevInfo;
                   Extend: LongInt;
                END;
TPfPage       = RECORD
                   Start: LongInt;
                   Length: LongInt;
                END;
TPfPages      = ARRAY [1..999] OF TPfPage;
TPPfPages     = ^TPfPages;
THPfPages     = ^TPPfPages;
TPrFileId     = RECORD
                   sFileName: Str255;
                   CASE Boolean OF
                      True:
                         (LongId: LongInt);
                      False:
                         (Id, BlockNum: Integer);
                END;
TPrFileRec    = RECORD
                   FileName: PathName;
                   RefNum: Integer;
                END;
TPrDSRec      = RECORD
                   SegName: PathName;
                   InitSize: LongInt;
                   RefNum: Integer;
                   LDSN: Integer;
                   BegAddr: LongInt;
                END;
PrMenuSuppress = (ePrNormal, ePgRangeSuppress, ePrDialogSuppress);
DlgOption     = (eDlgOK, eDlgCancel, eDlgAnother);
TPrntrID      = 0..16383;
TPrExtWord    = PACKED RECORD
                   CASE INTEGER OF
                      0:
                         (printer_flag: TF;
                          default_flag: TF;
                          PrDeviceID: TPrntrID);
                      1:
                         (ExtWord: INTEGER);
                END;
DriverChoice  = RECORD
                   Connector: TypeConnect;
```

```
                         cd_Driver: e_name;
                      END;
      DN_Info       = RECORD
                         printer_id: LongInt;
                         numDrivers: Integer;
                      END;
      prdHeader     = RECORD
                         version: Integer;
                         prdcount: Integer;
                      END;
      prdEntry      = RECORD
                         entryLen: Integer;
                         drvr_name: e_name;
                         info: dn_info;
                      END;
      PRD_Info      = RECORD
                         descr: prdHeader;
                         info: ARRAY [0..0] OF prdEntry;
                      END;

   VAR
      cPrError:     Integer;
      fBackGround:  TF;
      fInitialized: TF;
      lParentID:    TL;
      PrAlertUP:    Boolean;
      prsslot, prsconnector, prsdevice, prsAserial, prsBserial, prsparallel,
      prsnone:      TSp;
      PrExtras:     ARRAY [0..10] OF LONGINT;
      PrDebug:      Boolean;
      PrBugpName:   PathName;
      PrBugFile:    Text;

   FUNCTION PrRec68K: TpPrRec;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 234 Characters: 7814

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : PRSTDPRO
*
*******************************************************************************

USES {$U+} PRSTDPRO;


INTRINSIC;

INTERFACE

   USES
      {$U LibOS/SysCall   } SysCall,
      {$U LibPM/PMDecl    } PMDecl,
      {$U LibHW/HWint     } HWint,
      {$U LibSM/UnitStd   } UnitStd,
      {$U LibSM/UnitHz    } UnitHz,
      {$U LibQD/Storage   } Storage,
      {$U LibQD/QuickDraw } QuickDraw,
      {$U LibFM/FontMgr   } FontMgr,
      {$U LibPr/PrStdInfo } PrStdInfo;

   CONST
      PortA         = $0A10;
      PortB         = $0A20;

   TYPE
      Str80         = String[80];

   PROCEDURE PsCopy(Src, Dst: TL; len: TC);

   PROCEDURE PsOpenPort(VAR Error: Integer; Port: Integer);

   PROCEDURE PsClosePort(VAR Error: Integer; Port: Integer);

   PROCEDURE PsInPort(VAR Error: Integer; p: TP; c: TC);

   PROCEDURE PsOutPort(VAR Error: Integer; p: TP; c: TC);

   PROCEDURE PsPortName(Port: Integer; shandle: THsp);

   PROCEDURE PsPortToDevName(Port: Integer; VAR DevName: e_name);

   PROCEDURE PsPreemptive;

   PROCEDURE PsNonPreemptive;

   PROCEDURE PsNumToStr(c, w: TC; pstr: TpSp; z: CHAR);

   PROCEDURE PsYieldCpu;

   FUNCTION fClrBitmap(VAR bits: Bitmap; rCheck: Rect): TF;

   PROCEDURE InitDeBug;

   FUNCTION PrAnd(c1, c2: TC): TC;

   PROCEDURE ClrBitmap(pBitmap: TP);

   FUNCTION fClrBits(VAR bits: Bitmap; rCheck: Rect): TF;

   FUNCTION fEQBlk(a, b: TP; l: TL): TF;

   PROCEDURE MoveBlk(a, b: TP; l: TL);

   FUNCTION PrShRight(c1, n2: TC): TC;

   FUNCTION PrShLeft(c1, n2: TC): TC;

   PROCEDURE GetFileName(VAR PrFileId: TPrFileId);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 79 Characters: 1812

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : QUEUES
*
********************************************************************************

USES {$U+} QUEUES;


INTRINSIC;

INTERFACE

   USES
      {$U LibSM/UnitStd  } UnitStd,
      {$U LibSM/UnitHz   } UnitHz;

   TYPE
      pQueue        = ^Queue;
      hQueue        = ^pQueue;
      Queue         = RECORD
                         Head: hQueue;
                         Tail: hQueue;
                      END;

   PROCEDURE QueueFree(Queue: hQueue);

   PROCEDURE QueueInit(Queue: hQueue);

   PROCEDURE QueueInUse(Queue: hQueue);

   PROCEDURE Enqueue(Queue, Item: hQueue);

   FUNCTION Dequeue(Queue: hQueue): hQueue;

   FUNCTION InsertBefore(Queue, Successor, Item: hQueue): TF;

   FUNCTION InsertAfter(Queue, Predecessor, Item: hQueue): TF;

   FUNCTION Unqueue(Queue, Item: hQueue): TF;

   FUNCTION QueueLength(Queue: hQueue): TC;

   FUNCTION QueuePeek(Queue: hQueue): hQueue;

   FUNCTION EnumerateQueue(Queue: hQueue; FUNCTION
                           FuncHandle(Queue: hQueue; arg: hQueue): BOOLEAN;
                           arg: hQueue): hQueue;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 55 Characters: 1339

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : QUICKDRAW
*
********************************************************************************

USES {$U+} QUICKDRAW;


INTRINSIC;

INTERFACE

   CONST
      srcCopy       = 0;
      srcOr         = 1;
      srcXor        = 2;
      srcBic        = 3;
      notSrcCopy    = 4;
      notSrcOr      = 5;
      notSrcXor     = 6;
      notSrcBic     = 7;
      patCopy       = 8;
      patOr         = 9;
      patXor        = 10;
      patBic        = 11;
      notPatCopy    = 12;
      notPatOr      = 13;
      notPatXor     = 14;
      notPatBic     = 15;
      normalBit     = 0;
      inverseBit    = 1;
      redBit        = 4;
      greenBit      = 3;
      blueBit       = 2;
      cyanBit       = 8;
      magentaBit    = 7;
      yellowBit     = 6;
      blackBit      = 5;
      blackColor    = 33;
      whiteColor    = 30;
      redColor      = 205;
      greenColor    = 341;
      blueColor     = 409;
      cyanColor     = 273;
      magentaColor  = 137;
      yellowColor   = 69;
      picLParen     = 0;
      picRParen     = 1;

   TYPE
      QDByte        = - 128..127;
      QDPtr         = ^QDByte;
      QDHandle      = ^QDPtr;
      Str255        = String[255];
      Pattern       = PACKED ARRAY [0..7] OF 0..255;
      Bits16        = ARRAY [0..15] OF INTEGER;
      VHSelect      = (v, h);
      GrafVerb      = (frame, paint, erase, invert, fill);
      StyleItem     = (bold, italic, underline, outline, shadow, condense,
                       extend, onlymetricswidths);
```

```
Style         = SET OF StyleItem;
FontInfo      = RECORD
                   ascent: INTEGER;
                   descent: INTEGER;
                   widMax: INTEGER;
                   leading: INTEGER;
                END;
Point         = RECORD
                   CASE INTEGER OF
                      0:
                         (v: INTEGER;
                          h: INTEGER);
                      1:
                         (vh: ARRAY [VHSelect] OF INTEGER);
                END;
Rect          = RECORD
                   CASE INTEGER OF
                      0:
                         (top: INTEGER;
                          left: INTEGER;
                          bottom: INTEGER;
                          right: INTEGER);
                      1:
                         (topLeft: Point;
                          botRight: Point);
                END;
BitMap        = RECORD
                   baseAddr: QDPtr;
                   rowBytes: INTEGER;
                   bounds: Rect;
                END;
Cursor        = RECORD
                   data: Bits16;
                   mask: Bits16;
                   hotSpot: Point;
                END;
PenState      = RECORD
                   pnLoc: Point;
                   pnSize: Point;
                   pnMode: INTEGER;
                   pnPat: Pattern;
                END;
PolyHandle    = ^PolyPtr;
PolyPtr       = ^Polygon;
Polygon       = RECORD
                   polySize: INTEGER;
                   polyBBox: Rect;
                   polyPoints: ARRAY [0..0] OF Point;
                END;
RgnHandle     = ^RgnPtr;
RgnPtr        = ^Region;
Region        = RECORD
                   rgnSize: INTEGER;
                   rgnBBox: Rect;
                END;
PicHandle     = ^PicPtr;
PicPtr        = ^Picture;
Picture       = RECORD
                   picSize: INTEGER;
                   picFrame: Rect;
                END;
QDProcsPtr    = ^QDProcs;
QDProcs       = RECORD
                   textProc: QDPtr;
                   lineProc: QDPtr;
                   rectProc: QDPtr;
```

```
                      rRectProc: QDPtr;
                      ovalProc: QDPtr;
                      arcProc: QDPtr;
                      polyProc: QDPtr;
                      rgnProc: QDPtr;
                      bitsProc: QDPtr;
                      commentProc: QDPtr;
                      txMeasProc: QDPtr;
                      getPicProc: QDPtr;
                      putPicProc: QDPtr;
                   END;
   GrafPtr       = ^GrafPort;
   GrafPort      = RECORD
                      device: INTEGER;
                      portBits: BitMap;
                      portRect: Rect;
                      visRgn: RgnHandle;
                      clipRgn: RgnHandle;
                      bkPat: Pattern;
                      fillPat: Pattern;
                      pnLoc: Point;
                      pnSize: Point;
                      pnMode: INTEGER;
                      pnPat: Pattern;
                      pnVis: INTEGER;
                      txFont: INTEGER;
                      txFace: Style;
                      txMode: INTEGER;
                      txSize: INTEGER;
                      spExtra: LongInt;
                      fgColor: LongInt;
                      bkColor: LongInt;
                      colrBit: INTEGER;
                      patStretch: INTEGER;
                      picSave: QDHandle;
                      rgnSave: QDHandle;
                      polySave: QDHandle;
                      grafProcs: QDProcsPtr;
                   END;

VAR
   thePort:      GrafPtr;
   white:        Pattern;
   black:        Pattern;
   gray:         Pattern;
   ltGray:       Pattern;
   dkGray:       Pattern;
   arrow:        Cursor;
   screenBits:   BitMap;
   randSeed:     LongInt;

PROCEDURE InitGraf(globalPtr: QDPtr);

PROCEDURE OpenPort(port: GrafPtr);

PROCEDURE InitPort(port: GrafPtr);

PROCEDURE ClosePort(port: GrafPtr);

PROCEDURE SetPort(port: GrafPtr);

PROCEDURE GetPort(VAR port: GrafPtr);

PROCEDURE GrafDevice(device: INTEGER);

PROCEDURE SetPortBits(bm: BitMap);
```

```

    PROCEDURE PortSize(width, height: INTEGER);

    PROCEDURE MovePortTo(leftGlobal, topGlobal: INTEGER);

    PROCEDURE SetOrigin(h, v: INTEGER);

    PROCEDURE SetClip(rgn: RgnHandle);

    PROCEDURE GetClip(rgn: RgnHandle);

    PROCEDURE ClipRect(r: Rect);

    PROCEDURE BackPat(pat: Pattern);

    PROCEDURE InitCursor;

    PROCEDURE SetCursor(crsr: Cursor);

    PROCEDURE HideCursor;

    PROCEDURE ShowCursor;

    PROCEDURE ObscureCursor;

    PROCEDURE HidePen;

    PROCEDURE ShowPen;

    PROCEDURE GetPen(VAR pt: Point);

    PROCEDURE GetPenState(VAR pnState: PenState);

    PROCEDURE SetPenState(pnState: PenState);

    PROCEDURE PenSize(width, height: INTEGER);

    PROCEDURE PenMode(mode: INTEGER);

    PROCEDURE PenPat(pat: Pattern);

    PROCEDURE PenNormal;

    PROCEDURE MoveTo(h, v: INTEGER);

    PROCEDURE Move(dh, dv: INTEGER);

    PROCEDURE LineTo(h, v: INTEGER);

    PROCEDURE Line(dh, dv: INTEGER);

    PROCEDURE TextFont(font: INTEGER);

    PROCEDURE TextFace(face: Style);

    PROCEDURE TextMode(mode: INTEGER);

    PROCEDURE TextSize(size: INTEGER);

    PROCEDURE SpaceExtra(extra: LongInt);

    PROCEDURE DrawChar(ch: char);

    PROCEDURE DrawString(s: Str255);

    PROCEDURE DrawText(textBuf: QDPtr; firstByte, byteCount: INTEGER);
```

```

  FUNCTION CharWidth(ch: CHAR): INTEGER;

  FUNCTION StringWidth(s: Str255): INTEGER;

  FUNCTION TextWidth(textBuf: QDPtr; firstByte, byteCount: INTEGER): INTEGER;

  PROCEDURE GetFontInfo(VAR info: FontInfo);

  PROCEDURE AddPt(src: Point; VAR dst: Point);

  PROCEDURE SubPt(src: Point; VAR dst: Point);

  PROCEDURE SetPt(VAR pt: Point; h, v: INTEGER);

  FUNCTION EqualPt(pt1, pt2: Point): BOOLEAN;

  PROCEDURE ScalePt(VAR pt: Point; fromRect, toRect: Rect);

  PROCEDURE MapPt(VAR pt: Point; fromRect, toRect: Rect);

  PROCEDURE LocalToGlobal(VAR pt: Point);

  PROCEDURE GlobalToLocal(VAR pt: Point);

  PROCEDURE SetRect(VAR r: Rect; left, top, right, bottom: INTEGER);

  FUNCTION EqualRect(rect1, rect2: Rect): BOOLEAN;

  FUNCTION EmptyRect(r: Rect): BOOLEAN;

  PROCEDURE OffsetRect(VAR r: Rect; dh, dv: INTEGER);

  PROCEDURE MapRect(VAR r: Rect; fromRect, toRect: Rect);

  PROCEDURE InsetRect(VAR r: Rect; dh, dv: INTEGER);

  FUNCTION SectRect(src1, src2: Rect; VAR dstRect: Rect): BOOLEAN;

  PROCEDURE UnionRect(src1, src2: Rect; VAR dstRect: Rect);

  FUNCTION PtInRect(pt: Point; r: Rect): BOOLEAN;

  PROCEDURE Pt2Rect(pt1, pt2: Point; VAR dstRect: Rect);

  PROCEDURE FrameRect(r: Rect);

  PROCEDURE PaintRect(r: Rect);

  PROCEDURE EraseRect(r: Rect);

  PROCEDURE InvertRect(r: Rect);

  PROCEDURE FillRect(r: Rect; pat: Pattern);

  PROCEDURE FrameRoundRect(r: Rect; ovWd, ovHt: INTEGER);

  PROCEDURE PaintRoundRect(r: Rect; ovWd, ovHt: INTEGER);

  PROCEDURE EraseRoundRect(r: Rect; ovWd, ovHt: INTEGER);

  PROCEDURE InvertRoundRect(r: Rect; ovWd, ovHt: INTEGER);

  PROCEDURE FillRoundRect(r: Rect; ovWd, ovHt: INTEGER; pat: Pattern);

  PROCEDURE FrameOval(r: Rect);
```

```

   PROCEDURE PaintOval(r: Rect);

   PROCEDURE EraseOval(r: Rect);

   PROCEDURE InvertOval(r: Rect);

   PROCEDURE FillOval(r: Rect; pat: Pattern);

   PROCEDURE FrameArc(r: Rect; startAngle, arcAngle: INTEGER);

   PROCEDURE PaintArc(r: Rect; startAngle, arcAngle: INTEGER);

   PROCEDURE EraseArc(r: Rect; startAngle, arcAngle: INTEGER);

   PROCEDURE InvertArc(r: Rect; startAngle, arcAngle: INTEGER);

   PROCEDURE FillArc(r: Rect; startAngle, arcAngle: INTEGER; pat: Pattern);

   PROCEDURE PtToAngle(r: Rect; pt: Point; VAR angle: INTEGER);

   FUNCTION OpenPoly: PolyHandle;

   PROCEDURE ClosePoly;

   PROCEDURE KillPoly(poly: PolyHandle);

   PROCEDURE OffsetPoly(poly: PolyHandle; dh, dv: INTEGER);

   PROCEDURE MapPoly(poly: PolyHandle; fromRect, toRect: Rect);

   PROCEDURE FramePoly(poly: PolyHandle);

   PROCEDURE PaintPoly(poly: PolyHandle);

   PROCEDURE ErasePoly(poly: PolyHandle);

   PROCEDURE InvertPoly(poly: PolyHandle);

   PROCEDURE FillPoly(poly: PolyHandle; pat: Pattern);

   FUNCTION NewRgn: RgnHandle;

   PROCEDURE DisposeRgn(rgn: RgnHandle);

   PROCEDURE CopyRgn(srcRgn, dstRgn: RgnHandle);

   PROCEDURE SetEmptyRgn(rgn: RgnHandle);

   PROCEDURE SetRectRgn(rgn: RgnHandle; left, top, right, bottom: INTEGER);

   PROCEDURE RectRgn(rgn: RgnHandle; r: Rect);

   PROCEDURE OpenRgn;

   PROCEDURE CloseRgn(dstRgn: RgnHandle);

   PROCEDURE OffsetRgn(rgn: RgnHandle; dh, dv: INTEGER);

   PROCEDURE MapRgn(rgn: RgnHandle; fromRect, toRect: Rect);

   PROCEDURE InsetRgn(rgn: RgnHandle; dh, dv: INTEGER);

   PROCEDURE SectRgn(srcRgnA, srcRgnB, dstRgn: RgnHandle);

   PROCEDURE UnionRgn(srcRgnA, srcRgnB, dstRgn: RgnHandle);
```

```

    PROCEDURE DiffRgn(srcRgnA, srcRgnB, dstRgn: RgnHandle);

    PROCEDURE XorRgn(srcRgnA, srcRgnB, dstRgn: RgnHandle);

    FUNCTION EqualRgn(rgnA, rgnB: RgnHandle): BOOLEAN;

    FUNCTION EmptyRgn(rgn: RgnHandle): BOOLEAN;

    FUNCTION PtInRgn(pt: Point; rgn: RgnHandle): BOOLEAN;

    FUNCTION RectInRgn(r: Rect; rgn: RgnHandle): BOOLEAN;

    PROCEDURE FrameRgn(rgn: RgnHandle);

    PROCEDURE PaintRgn(rgn: RgnHandle);

    PROCEDURE EraseRgn(rgn: RgnHandle);

    PROCEDURE InvertRgn(rgn: RgnHandle);

    PROCEDURE FillRgn(rgn: RgnHandle; pat: Pattern);

    PROCEDURE ScrollRect(dstRect: Rect; dh, dv: INTEGER; updateRgn: rgnHandle);

    PROCEDURE CopyBits(srcBits, dstBits: BitMap; srcRect, dstRect: Rect;
                       mode: INTEGER; maskRgn: RgnHandle);

    FUNCTION OpenPicture(picFrame: Rect): PicHandle;

    PROCEDURE ClosePicture;

    PROCEDURE DrawPicture(myPicture: PicHandle; dstRect: Rect);

    PROCEDURE PicComment(kind, dataSize: INTEGER; dataHandle: QDHandle);

    PROCEDURE KillPicture(myPicture: PicHandle);

    PROCEDURE SetStdProcs(VAR procs: QDProcs);

    PROCEDURE StdText(count: INTEGER; textAddr: QDPtr; numer, denom: Point);

    PROCEDURE StdLine(newPt: Point);

    PROCEDURE StdRect(verb: GrafVerb; r: Rect);

    PROCEDURE StdRRect(verb: GrafVerb; r: Rect; ovWd, ovHt: INTEGER);

    PROCEDURE StdOval(verb: GrafVerb; r: Rect);

    PROCEDURE StdArc(verb: GrafVerb; r: Rect; startAngle, arcAngle: INTEGER);

    PROCEDURE StdPoly(verb: GrafVerb; poly: PolyHandle);

    PROCEDURE StdRgn(verb: GrafVerb; rgn: RgnHandle);

    PROCEDURE StdBits(VAR srcBits: BitMap; VAR srcRect, dstRect: Rect;
                      mode: INTEGER; maskRgn: RgnHandle);

    PROCEDURE StdComment(kind, dataSize: INTEGER; dataHandle: QDHandle);

    FUNCTION StdTxMeas(count: INTEGER; textAddr: QDPtr; VAR numer, denom: Point;
                       VAR info: FontInfo): INTEGER;

    PROCEDURE StdGetPic(dataPtr: QDPtr; byteCount: INTEGER);

```

```
   PROCEDURE StdPutPic(dataPtr: QDPtr; byteCount: INTEGER);

   FUNCTION GetPixel(h, v: INTEGER): BOOLEAN;

   FUNCTION Random: INTEGER;

   PROCEDURE StuffHex(thingptr: QDPtr; s: Str255);

   PROCEDURE ForeColor(color: LongInt);

   PROCEDURE BackColor(color: LongInt);

   PROCEDURE ColorBit(whichBit: INTEGER);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 477 Characters: 12804

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : RECOVERY
*
*******************************************************************************

USES {$U+} RECOVERY;


INTRINSIC;

INTERFACE

   USES
      {$IFC OSBUILT }
         {$U libsm/unitstd.obj   } unitstd,
         {$U libdb/dbenv.obj     } dbenv,
         {$U libin/INTRLIBp.obj  } international,
         {$U libdb/dbdecl1.obj   } dbdecl1,
         {$U libos/syscall.obj   } syscall,
         {$U libos/psyscall.obj  } psyscall,
         {$U libdb/lowlevel.obj  } lowlevel,
         {$U libdb/pooler.obj    } pooler,
         {$U libdb/heap.obj      } heap,
         {$U libdb/czcompact.obj } czCompact,
         {$U libdb/vltree.obj    } vltree,
         {$U libdb/scan.obj      } scan,
         {$U libdb/labscan.obj   } labscan,
         {$U libdb/schema.obj    } schema;
      {$ELSEC }
         {$U OBJ:dbenv.obj       } dbenv,
         {$U INTRLIB.obj         } international,
         {$U OBJ:dbdecl1.obj     } dbdecl1,
         {$U OBJ:syscall.obj     } syscall,
         {$U OBJ:lowlevel.obj    } lowlevel,
         {$U OBJ:pooler.obj      } pooler,
         {$U OBJ:heap.obj        } heap,
         {$U OBJ:czcompact.obj   } czCompact,
         {$U OBJ:vltree.obj      } vltree,
         {$U OBJ:scan.obj        } scan,
         {$U OBJ:labscan.obj     } labscan,
         {$U OBJ:schema.obj      } schema;
      {$ENDC }

   const


      WarnIndex     = 0;
      WarnNoSchema  = 1;
      WarnOldSchema = 2;
      BadFileDesc   = 3400;
      BadFileSchema = 3401;

   PROCEDURE FileRecover(VAR ddresult: integer; VAR warnings: elemset;
                         oldprefix, newprefix, pfname, pnewname,
                         precname: ptrpathname;

                         PROCEDURE
                         ECheck(EC: ecstr; ftype: integer; VAR Check: checkinfo;
                                VAR result: integer);

                         PROCEDURE
                         indexmake(VAR ddresult: integer; ifilename, filename,
                                   tempdir1, tempdir2: ptrpathname;
                                   temp_ldsn: integer; nisort, nikeys: integer;
                                   piflds: ptridesc; duplok: boolean);

                         PROCEDURE
                         MarksRevenge(VAR result: integer;
                                      VAR warnings: elemset; lsid: integer);

                         ToolsMarketCode: integer);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 79 Characters: 2585

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : REFS
*
********************************************************************************

USES {$U+} REFS;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/PASDEFS.OBJ *) PasDefs,
         (*$U OBJ/MEMMAN.OBJ  *) MemMan,
         (*$U OBJ/LISTS.OBJ   *) Lists;
      {$ELSEC }
         (*$U OBJ:PASDEFS.OBJ *) PasDefs,
         (*$U OBJ:MEMMAN.OBJ  *) MemMan,
         (*$U OBJ:LISTS.OBJ   *) Lists;
      {$ENDC }

   TYPE


      Ref           = integer;
      RefHRow       = ARRAY [ - 128..127] OF Ref;
      RefHRowPtr    = ^RefHRow;
      RefHArr       = ARRAY [ - 128..127] OF RefHRowPtr;
      RefHPtr       = ^RefHArr;
      RefRec        = RECORD
                         refloc: integer;
                         nextref: Ref;
                      END;
      RefRow        = ARRAY [ - 128..127] OF RefRec;
      RefRowPtr     = ^RefRow;
      RefArr        = ARRAY [ - 128..127] OF RefRowPtr;
      RefArrPtr     = ^RefArr;
      RefHandle     = ^RefObject;
      RefObject     = RECORD
                         memRefs: MMHandle;
                         refHeads: RefHPtr;
                         refStore: RefArrPtr;
                         refsLH: ListHandle;
                      END;

   FUNCTION AddRef(rh: RefHandle; head: Ref; l: integer; FUNCTION
                   GetMore(n, row: integer): boolean): Ref;

   PROCEDURE EachRef(rh: RefHandle; head: Ref; PROCEDURE
                     Visit(i: Ref));

   FUNCTION MoreHeads(rh: RefHandle; n, row: integer): boolean;

   PROCEDURE NewRefList(rh: RefHandle; head: Ref);

   PROCEDURE InitRLists;

   FUNCTION InitRefs(nHeads, nRefs: longint): RefHandle;

   PROCEDURE CopyRefLists(rh: RefHandle; old, new: Ref);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 70 Characters: 1976

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SCAN
*
*******************************************************************************

USES {$U+} SCAN;


INTRINSIC;

INTERFACE

   USES
      {$IFC OSBUILT }
         {$U libsm/unitstd.obj} unitstd,
         {$U libdb/dbenv.obj } dbenv,
         {$U libin/INTRLIBp.obj } international,
         {$U libdb/dbdecl1.obj } dbdecl1,
         {$U libos/syscall.obj } syscall,
         {$U libos/psyscall.obj } psyscall,
         {$U libdb/lowlevel.obj } lowlevel,
         {$U libdb/pooler.obj } pooler,
         {$U libdb/heap.obj } heap,
         {$U libdb/czcompact.obj } czCompact,
         {$U libdb/vltree.obj } vltree;
      {$ELSEC}
         {$U OBJ:dbenv.obj } dbenv,
         {$U INTRLIB.obj } international,
         {$U OBJ:dbdecl1.obj } dbdecl1,
         {$U OBJ:syscall.obj } syscall,
         {$U OBJ:lowlevel.obj } lowlevel,
         {$U OBJ:pooler.obj } pooler,
         {$U OBJ:heap.obj } heap,
         {$U OBJ:czcompact.obj } czCompact,
         {$U OBJ:vltree.obj } vltree;
      {$ENDC}

   FUNCTION  SParmsBad(scanid: integer; which: integer; nsearch: integer;
                       pentry: ptrdata): integer;

   FUNCTION  scanidbad(scanid: integer): boolean;

   PROCEDURE pstack_validate(VAR sresult: integer);

   FUNCTION  keysize(fileid: integer; pentry: ptrdata): integer;

   PROCEDURE einsert(VAR sresult: integer; VAR offender: integer;
                     scanid: integer; pentry: ptrdata; size: integer;
                     newticket: boolean);

   PROCEDURE efetch(VAR sresult: integer; scanid: integer; which: integer;
                    nsearch: integer; pentry: ptrdata);

   PROCEDURE eupdate(VAR sresult: integer; VAR offender: integer;
                     scanid: integer; which: integer; nsearch: integer;
                     pentry: ptrdata; pnewrec: ptrdata; size: integer);

   PROCEDURE edelete(VAR sresult: integer; scanid: integer; which: integer;
                     nsearch: integer; pentry: ptrdata);

```

```
********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 67 Characters: 2224

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SCHEMA
*
********************************************************************************

USES {$U+} SCHEMA;


INTRINSIC;

INTERFACE

   USES
      {$IFC OSBUILT }
         {$U libsm/unitstd.obj} unitstd,
         {$U libdb/dbenv.obj } dbenv,
         {$U libin/INTRLIBp.obj } international,
         {$U libdb/dbdecl1.obj } dbdecl1,
         {$U libos/syscall.obj } syscall,
         {$U libos/psyscall.obj } psyscall,
         {$U libdb/lowlevel.obj } lowlevel,
         {$U libdb/pooler.obj } pooler,
         {$U libdb/heap.obj } heap,
         {$U libdb/czcompact.obj } czCompact,
         {$U libdb/vltree.obj } vltree,
         {$U libdb/scan.obj } scan,
         {$U libdb/labscan.obj } labscan;
      {$ELSEC }
         {$U OBJ:dbenv.obj } dbenv,
         {$U INTRLIB.obj } international,
         {$U OBJ:dbdecl1.obj } dbdecl1,
         {$U OBJ:syscall.obj } syscall,
         {$U OBJ:lowlevel.obj } lowlevel,
         {$U OBJ:pooler.obj } pooler,
         {$U OBJ:heap.obj } heap,
         {$U OBJ:czcompact.obj } czCompact,
         {$U OBJ:vltree.obj } vltree,
         {$U OBJ:scan.obj } scan,
         {$U OBJ:labscan.obj } labscan;
      {$ENDC }

   const


      indexfile     = 0;
      fileschema    = 1;
      queryCheckPoint = 2;
      not_needed    = 0;
      via_only      = 1;
      on_only       = 2;
      on_and_via    = 3;
      new_index     = 4;

   TYPE
      isegment      = PACKED RECORD
                         aord: 0..1;
                         field: 0..32767;
                      END;
      ptrisegment   = ^isegment;
      idesc         = ARRAY [0..0] OF isegment;
      ptridesc      = ^idesc;
      ientry        = RECORD
                         lab: labelentry;
                         iduplok: boolean;
                         isort: integer;
                         ikeys: integer;
                         iunique: uniqueid;
                         iflds: idesc;
                      END;
      ptrientry     = ^ientry;

   PROCEDURE filldesc(VAR ddresult: integer; pfile: ptrfiledesc);

   PROCEDURE makefile(VAR ddresult: integer; pname: ptrpathname;
                      pfdesc: ptrfiledesc);

   PROCEDURE openindexes(VAR ddresult: integer; VAR viafile: integer;
                         scanid: integer; ifilename: ptrpathname;
                         intent: integer; labelusage: integer);

   PROCEDURE cvt_scan(VAR ddresult: integer; scanid, sintent,
                      sviafile: integer);

   PROCEDURE initpool(VAR ddresult: integer; first_time: boolean;
                      heap_ldsn: integer; nbuff: integer; dheapsize: integer;
                      pheapname: ptrpathname; dstype: Tdstype;
                      pagesize: integer);

   PROCEDURE flushscan(VAR ddresult: integer; scanid: integer);

   PROCEDURE closescan(VAR ddresult: integer; scanid: integer);

   PROCEDURE openscan(VAR ddresult: integer; onname, vianame: ptrpathname;
                      VAR scanid: integer; sintent: integer);

   PROCEDURE makedescfile(VAR ddresult: integer; pname: ptrpathname;
                          pfdesc: ptrfiledesc);

   PROCEDURE clonefile(VAR ddresult: integer; sourcefile, filename: ptrpathname;
                       needtomakefile: boolean);

   PROCEDURE makeformfile(VAR ddresult: integer; pform: ptrdata;
                          pfname: ptrpathname; needtomakefile: boolean);

   PROCEDURE quick_label_scan(VAR ddresult: integer; scanid: integer;
                              sintent: integer; VAR lscanid: integer);

   PROCEDURE get_header(VAR ddresult: integer; scanid: integer;
                        VAR head: header);

   PROCEDURE getform(VAR ddresult: integer; scanid: integer; pdesc: ptrdata);

   PROCEDURE deletefile(VAR ddresult: integer; pfname: ptrpathname);

   PROCEDURE deleteindex(VAR ddresult: integer; piname, pfname: ptrpathname);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************


End of File -- Lines: 123 Characters: 4043
```

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SCRAP
*
********************************************************************************

USES {$U+} SCRAP;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libsm/UnitStd    } UnitStd,
      {$U libsm/UnitHz     } UnitHz,
      {$U libos/SysCall    } SysCall,
      {$U libsu/UnitFile   } UnitFile,
      {$U libqd/QuickDraw  } QuickDraw,
      {$U libfm/FontMgr    } FontMgr,
      {$U libqd/Storage    } Storage,
      {$U libwm/Events     } Events,
      {$U libwm/Folders    } Folders,
      {$U libsb/WmlStd     } WmlStd,
      {$U libsb/Wmlsb      } Wmlsb,
      {$U libsu/UnitFmt    } UnitFmt,
      {$U libpm/PmDecl     } PmDecl,
      {$U libpr/PrStdInfo  } PrStdInfo,
      {$U libsu/UnitCS     } UnitCS,
      {$U libsu/UnitFigAtom} UnitFigAtom,
      {$U libsu/UnitFf     } UnitFf;

   CONST
      ScrapNil      = 0;
      ScrapFE       = 2;
      ScrapMtx      = 3;
      ScrapBGraf    = 4;
      ScrapList     = 5;
      ScrapDwg      = 6;
      ScrapPert     = 7;
      ScrapCs       = 12;
      ScrapUG       = 13;
      SecondOpenScrapCall = 4050;
      HZINITfailed  = 4051;
      NoUndoScrap   = 4052;
      TooManyProcesses = 4053;
      ProcCalledByNonOwner = 4054;
      ProcessNotFound = 4055;
      ScrapNotPicture = 4059;
      DataSegNotOpen = - 4060;

   TYPE
      ScrapType     = 0..15;

   VAR
      icsFfScrap:   TB;
      ifilScrap:    TB;
      CurrScrapSet: SET OF ScrapType;
      Scrap1Figures_RefNum, Scrap2Figures_RefNum, Scrap1Lotus_RefNum,
      Scrap2Lotus_RefNum: TL;

   PROCEDURE InitScrap(VAR InitErr: Integer);

   PROCEDURE KillScrapOwner(Who: ProcessId; VAR KillErr: TC);

   PROCEDURE ReviveScrapOwner(Who: ProcessId);

   PROCEDURE DrawUScrap(VAR DrawErr: TC);

   PROCEDURE BackOutOfScrap;

   PROCEDURE OpenScrap(VAR OpenErr: Integer);

   FUNCTION DSegOfScrap: Integer;

   FUNCTION AddrofScrapDSeg: LongInt;

   FUNCTION HzOfScrap: THz;

   PROCEDURE ClaimScrap;

   PROCEDURE InheritScrap(SaveOld: Boolean);

   PROCEDURE UndoInheritScrap(VAR UndoErr: integer);

   PROCEDURE AcceptInheritScrap;

   PROCEDURE EraseScrapData(VAR EraseErr: integer);

   PROCEDURE StartPutScrap(VAR PutErr: integer);

   PROCEDURE PutScrap(which: ScrapType; what: TH; VAR PutErr: integer);

   PROCEDURE PutGrScrap(What: PicHandle; VAR PutErr: integer);

   PROCEDURE PutCsScrap(icsContents: TB; VAR PutCSErr: integer);

   PROCEDURE EndPutScrap(VAR PutErr: integer);

   PROCEDURE StartGetScrap(VAR GetErr: integer);

   PROCEDURE GetScrap(VAR which: ScrapType; VAR what: TH);

   PROCEDURE GetGrScrap(VAR What: PicHandle);

   PROCEDURE GetCsScrap(VAR Contents: TB);

   PROCEDURE EndGetScrap(VAR GetErr: integer);

   PROCEDURE bindUTDSeg(VAR BindErr: integer);

   PROCEDURE unBindUTDSeg(VAR UnBindErr: integer);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 119 Characters: 2953

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SHELLCOM
*
********************************************************************************

USES {$U+} SHELLCOM;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U StdUnit } StdUnit;

   CONST
      SCNone        = - 1;
      SCAny         = 0;
      SCText        = 1;
      SCBufrMax     = 1023;
      SC_SetReallyStop = 1;
      SC_GetReallyStop = 2;
      SC_SetUnSavedEdits = 6960;
      SC_GetUnSavedEdits = 8751;

   VAR
      SCRetStr:     SUStr;
      SCRunCmnd:    SUStr;
      SCBufr:       PACKED ARRAY [0..1023] OF CHAR;
      SCExecAbort:  BOOLEAN;
      SCReallyStopExec: BOOLEAN;
      SCUnSavedEdits: BOOLEAN;

   PROCEDURE SCInit;

   PROCEDURE SCSetRunCmd(RC: SUStr);

   PROCEDURE SCSetRetStr(RS: SUStr);

   PROCEDURE SCSetExecAbort(B: BOOLEAN);

   PROCEDURE SCReWrite(WriteType: INTEGER; Key: SUStr);

   FUNCTION SCReset(ReadType: INTEGER; Key: SUStr): BOOLEAN;

   FUNCTION SCClose(KillBufr: BOOLEAN; Key: SUStr): BOOLEAN;

   FUNCTION SCPutCh(Ch: CHAR): BOOLEAN;

   FUNCTION SCGetCh(VAR Ch: CHAR): BOOLEAN;

   FUNCTION SCPutLine(L: SUStr): BOOLEAN;

   FUNCTION SCGetLine(VAR L: SUStr): BOOLEAN;

   FUNCTION SCShellCmd(Cmd: INTEGER; P: SUStrP): BOOLEAN;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
```

```
*
*******************************************************************************

```

End of File -- Lines: 64 Characters: 1441

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : STDUNIT
*
*******************************************************************************

USES {$U+} STDUNIT;


INTRINSIC;

INTERFACE

   USES
      {$U libOS/SysCall.obj    } SysCall,
      {$U libPL/PasLibCall.obj } PasLibCall,
      {$U libPL/PPasLibC.obj   } PPasLibC;

   CONST
      SUMaxStrLeng  = 255;
      SUNullStr     = '';
      SUSpace       = ' ';
      SUOrdCR       = 13;
      SUMaxPNLeng   = 66;
      SUMaxVNLeng   = 33;
      SUMaxFNLeng   = 32;
      SUVolSuffix   = '-';

   TYPE
      SUSetOfChar   = SET OF CHAR;
      SUStrP        = ^SUStr;
      SUStr         = STRING[255];
      SUVolName     = STRING[SUMaxVNLeng];
      SUFile        = FILE;
      SUFileP       = ^SUFile;
      PromptState   = (SUDefault, SUEscape, SUNone, SUOptions, SUValid,
                       SUInvalid);
      ErrTextRet    = (SUOk, SUBadEFOpen, SUBadEFRead, SUErrNNotFound);
      ConvNState    = (SUValidN, SUNoN, SUBadN, SUNOverFlow);

   VAR
      SUOsBootV:    SUVolName;
      SUMyProcV:    SUVolName;
      SUBell, SUBackSpace, SUCr, SUTab, SUEsc, SUDle, SUNul: CHAR;
      SUNullS:      SUStr;
      SUKeyBoard:   INTERACTIVE;

   PROCEDURE SUInit;

   PROCEDURE SUDone;

   FUNCTION SUUpCh(Ch: CHAR): CHAR;

   FUNCTION SULowCh(Ch: CHAR): CHAR;

   PROCEDURE SUUpStr(S: SUStrP);

   PROCEDURE SULowStr(S: SUStrP);

   FUNCTION SUEqStr(S1: SUStrP; S2: SUStrP): BOOLEAN;

FUNCTION SUEq2Str(S1: SUStrP; S2: SUStr): BOOLEAN;

PROCEDURE SUTrimLeading(S: SUStrP);

PROCEDURE SUTrimTrailing(S: SUStrP);

PROCEDURE SUTrimBlanks(S: SUStrP);

PROCEDURE SUAddCh(S: SUStrP; Ch: CHAR; MaxStrLeng: INTEGER;
                  VAR OverFlow: BOOLEAN);

PROCEDURE SUConcat(S1: SUStrP; S2: SUStrP);

PROCEDURE SUAddStr(S1: SUStrP; S2: SUStrP; MaxStrLeng: INTEGER;
                   VAR OverFlow: BOOLEAN);

PROCEDURE SUSetStr(Dest: SUStrP; Src: SUStrP);

PROCEDURE SUCopyStr(Dest: SUStrP; Src: SUStrP; Start, Count: INTEGER);

FUNCTION SUIsVolName(FN: SUStrP): BOOLEAN;

PROCEDURE SUVolPart(PathN: SUStrP; VolN: SUStrP);

PROCEDURE SUAddExtension(FN: SUStrP; DefExt: SUStr; MaxStrLeng: INTEGER;
                         VAR OverFlow: BOOLEAN);

PROCEDURE SUSplitFN(PathN: SUStrP; CatN: SUStrP; FN: SUStrP; Ext: SUStrP);

PROCEDURE SUMakeFN(PathN: SUStrP; CatN: SUStrP; FN: SUStrP; Ext: SUStr;
                   VAR OverFlow: BOOLEAN);

PROCEDURE SUChkFN(FN: SUStrP; VAR PState: PromptState; DefVol: SUStr;
                  DefFN: SUStr; DefExt: SUStr);

PROCEDURE SUGetCh(VAR Ch: CHAR);

PROCEDURE SUGetLine(S: SUStrP; VAR PState: PromptState);

PROCEDURE SUGetStr(S: SUStrP; VAR PState: PromptState; DefVal: SUStr);

PROCEDURE SUGetFN(FN: SUStrP; VAR PState: PromptState; DefVol: SUStr;
                  DefFN: SUStr; DefExt: SUStr);

PROCEDURE SUGetInt(VAR I: INTEGER; VAR PState: PromptState; DefVal: INTEGER);

PROCEDURE SUWaitEscOrSp(VAR PState: PromptState);

PROCEDURE SUWaitSp;

PROCEDURE SUGetChInSet(VAR Ch: CHAR; Chars: SUSetOfChar);

FUNCTION SUGetYesNo: BOOLEAN;

FUNCTION SUGetBool(Default: BOOLEAN): BOOLEAN;

PROCEDURE SUGetErrText(ErrFN: SUStr; ErrN: INTEGER; ErrMsg: SUStrP;
                       VAR ErrRet: ErrTextRet);

PROCEDURE SUErrText(ErrFN: SUStr; ErrN: INTEGER; ErrMsg: SUStrP);

PROCEDURE SUStopExec(VAR ErrNum: INTEGER);

PROCEDURE SUCloseExec(VAR ErrNum: INTEGER);

PROCEDURE SUInitSysVols;
```

```

   PROCEDURE SUSysReset(F: SUFileP; FN: SUStr; VAR IOStatus: INTEGER);

   PROCEDURE SUIntToStr(N: INTEGER; S: SUStrP);

   PROCEDURE SULIntToStr(N: LONGINT; S: SUStrP);

   PROCEDURE SUStrToInt(NS: SUStrP; VAR N: INTEGER; VAR CState: ConvNState);

   PROCEDURE SUStrToLInt(NS: SUStrP; VAR N: LONGINT; VAR CState: ConvNState);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 144 Characters: 3939

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : STORAGE
*
*******************************************************************************

USES {$U+} STORAGE;


INTRINSIC;

INTERFACE

   USES                                {$U libsm/UnitStd } UnitStd,
      {$U libsm/UnitHz   } UnitHz;

   TYPE
      Ptr           = TP;
      Handle        = TH;
      RelHandle     = INTEGER;
      ProcPtr       = TProc;

   VAR
      theHeap:      THz;
      ordHeap:      LongInt;

   PROCEDURE InitHeap(startPtr, limitPtr: Ptr; errorProc: ProcPtr);

   PROCEDURE SetHeap(hz: Thz);

   PROCEDURE GetHeap(VAR hz: Thz);

   FUNCTION NewPtr(byteCount: INTEGER): Ptr;

   PROCEDURE DisposePtr(p: Ptr);

   FUNCTION NewHandle(byteCount: INTEGER): Handle;

   PROCEDURE DisposeHandle(h: Handle);

   PROCEDURE SetSize(h: Handle; newSize: INTEGER);

   FUNCTION GetSize(h: Handle): INTEGER;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 50 Characters: 1075

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : SYS1LOCK
*
********************************************************************************

USES {$U+} SYS1LOCK;


INTRINSIC;

INTERFACE

   USES
      {$U libsm/UnitStd        } UnitStd,
      {$U libos/SysCall.Obj    } SysCall,
      {$U libos/PSysCall.obj   } PSysCall;

   PROCEDURE LkSYSINIT(VAR err: INTEGER);

   PROCEDURE LkOPSEG1(VAR err: INTEGER);

   PROCEDURE LkOPSEG2(VAR err: INTEGER);

   PROCEDURE LkACTIVATE(VAR err: Integer);

   PROCEDURE LkWORKSET(VAR err: INTEGER);

   PROCEDURE LkGRAFSEG(VAR err: INTEGER);

   PROCEDURE LkMOVERS(VAR err: INTEGER);

   PROCEDURE LkOPEN1LW(VAR err: Integer);

   PROCEDURE LkLWWORK(VAR err: Integer);

   PROCEDURE LkOTHRWORK(VAR err: Integer);

   PROCEDURE LkNEVER(VAR err: INTEGER);

   PROCEDURE LkFmgrutil(VAR err: Integer);

   PROCEDURE LkPMMSeg(VAR err: INTEGER);

   PROCEDURE LkWMwarm(VAR err: Integer);

   PROCEDURE LkWMJrnl(VAR err: Integer);

   PROCEDURE LkWMalert(VAR err: INTEGER);

   PROCEDURE Lkalert(VAR err: INTEGER);

   PROCEDURE LkSMcold(VAR err: INTEGER);

   PROCEDURE LkSUcold(VAR err: INTEGER);

   PROCEDURE LkFEcold(VAR err: INTEGER);

   PROCEDURE LkSBcold(VAR err: INTEGER);

   PROCEDURE LkINcold(VAR err: INTEGER);

   PROCEDURE LkWMcold(VAR err: Integer);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 70 Characters: 1513

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : TEENV
*
*******************************************************************************

USES {$U+} TEENV;


INTRINSIC;

INTERFACE

   USES
      {$U libsm/UnitStd } UnitStd,
      {$U libsm/UnitHz } UnitHz,
      {$U libqd/Storage } Storage,
      {$U libqd/QuickDraw } QuickDraw,
      {$U libfm/FontMgr   } FontMgr,
      {$U libdb/dbenv } dbenv,
      {$U libfe/fedec} FEDec,
      {$U libfe/fld } fieldedit;
      {$SETC teProduction := NOT FDbgOK }
      {$SETC teSym := FSymOK}
      {$SETC TEDEBUG := NOT teProduction  }
      {$SETC doTraceTE := TRUE}
      {$SetC fTraceTE := doTraceTE AND fTRACE}

   CONST
      OK            = 0;
      OutOfMem      = 1;
      InvalidCell   = 2;
      rmDel         = 3;
      rmOk          = 4;
      rmReName      = 5;
      parseErr      = 6;
      undoErr       = 7;
      badField      = 8;
      cvnotenuff    = 9;
      crnotenuff    = 10;
      nullSnip      = 0;
      pInfSnip      = - 2;
      mInfSnip      = - 1;
      nullTimeout   = 0;
      dfltTimeout   = - 1;
      cFolderpnl    = 0;
      cCellPnl      = 1;
      cTblPnl       = 2;
      cWidePnl      = 3;
      cDialogPnl    = 4;
      cScrapPnl     = 5;
      nullPnl       = - 1;
      cBitmapH      = 720;
      cBitmapV      = 20;
      {$IFC NOT TEDEBUG}
      TraceFmgr     = FALSE;
      {$ENDC }

   TYPE
      bitptr        = ^Bitmap;
      PnlIndex      = INTEGER;
```

```
Panel         = RECORD
                   PnlPort: GrafPtr;
                   Offset: Point;
                   PnlRect: Rect;
                END;
panelptr      = ^Panel;
pnlHandle     = ^panelptr;
STR40         = string[40];
str9          = string[9];
str100        = string[100];
pixel         = integer;
idType        = integer;
rgKind        = (aNullRg, a1CellRg, aRectRg, aRowRg, aColRg, aRowGrid,
                 aColGrid, aRowHedRg, aColHedRg, aWTBoxRg, aTblHedRg);
rgJust        = (topJ, botJ, leftJ, rightJ, botLeftJ, topLeftJ, topRightJ,
                 nearJ);
rgX           = ^range;
range         = RECORD
                   rKind: rgKind;
                   loRow: idType;
                   loCol: idType;
                   hiRow: idType;
                   hiCol: idType;
                END;
fract         = INTEGER;
snipKind      = (rowKind, colKind);
SelKind       = (aNullSl, aCellTxtSl, a1CellSl, aRectRgSl, aRowRgSl,
                 aColRgSl, aRowGrdSl, aColGrdSl, aRowHedSl, aColHedSl,
                 aTblHedSl, aWTBoxSl);
TblPart       = (outOfTbl, cellArea, rowGrid, colGrid, rowHandl, colHandl,
                 rowHed, colHed, whTbBox, tblHed);
TableRecord   = RECORD
                   SplitTable: BOOLEAN;
                   RectBorder: BOOLEAN;
                   Marquee: BOOLEAN;
                   ColHandles: BOOLEAN;
                   ColGrids: BOOLEAN;
                   ColGridHandles: BOOLEAN;
                   ColRangeOk: BOOLEAN;
                   EditColTitle: BOOLEAN;
                   InsertCols: BOOLEAN;
                   EdBlankCol: BOOLEAN;
                   RowHandles: BOOLEAN;
                   RowGrids: BOOLEAN;
                   RowRangeOk: BOOLEAN;
                   RowGridHandles: BOOLEAN;
                   EditRowTitle: BOOLEAN;
                   InsertRows: BOOLEAN;
                   EdBlankRow: BOOLEAN;
                   EditTable: BOOLEAN;
                   ShoFormulas: BOOLEAN;
                   Preview: BOOLEAN;
                   RectRgOk: BOOLEAN;
                   FieldPad: INTEGER;
                   scrollincr: ARRAY [rowkind..colkind] OF INTEGER;
                END;
tmBand        = RECORD
                   tmbKind: SnipKind;
                   tmbLoP: pixel;
                   tmbHiP: pixel;
                   tmbHideP: pixel;
                   tmbLoId: idtype;
                END;
cellParType   = (cRecord);
idModeType    = (mmfirst, mmlast, mmnext, mmprior, mmfraction);
mcType        = (mmRight, mmLeft, mmUp, mmDown, mmNone, mmRange,
                    mmBegOfRow, mmBegOfCol, mmRightPeg, mmLeftPeg);
   IOModeType    = (mmRead, mmWrite, mmIONull);
   errRecord     = RECORD
                      errSpot: interval;
                      mess1: str40;
                      mess2: str40;
                      status: integer;
                   END;
   cellRecord    = RECORD
                      align: integer;
                      font: Tlfntid;
                      protected: boolean;
                      hiLite: boolean;
                      NumRuns: integer;
                      dim: boolean;
                   END;
   BGrowResult   = (BSame, BGrew, BWent, nxtBWent);
   ScrollType    = (incrScroll, pageScroll, jmpScroll);
   ScrollDir     = (ScrollBak, ScrollFwd);
   TMstate       = RECORD
                      marqsaved: boolean;
                      rg: range;
                      isSel: boolean;
                      isOn: boolean;
                      isBorder: boolean;
                   END;

VAR
   {$IFC TEDEBUG}
   traceFMGR:    BOOLEAN;
   {$ENDC }
   dfltNbrhood:  Rect;
   nullNbrhood:  Rect;
   TblPars:      TableRecord;
   FolderPnl:    PnlIndex;
   CellPnl:      PnlIndex;
   TblPnl:       PnlIndex;
   WidePnl:      PnlIndex;
   DialogPnl:    PnlIndex;
   ScrapPnl:     PnlIndex;

PROCEDURE AdjRect(Pnl: PnlIndex; VAR r: rect);

PROCEDURE AdjToPnl(Pnl: PnlIndex; P: Point; VAR relPoint: Point);

PROCEDURE allocPnl(thePnl: PnlIndex);

PROCEDURE ChgPnlSize(P: PnlIndex; Width, Height: Pixel);

PROCEDURE FreePnl(thePnl: PnlIndex);

PROCEDURE GetPnlBits(P: pnlIndex; VAR b: bitptr);

PROCEDURE GetPnloffset(Pnl: pnlIndex; VAR offset: point);

PROCEDURE GetPnlRect(PNL: PnlIndex; VAR aPnlRect: Rect);

PROCEDURE InitPnls(VAR status: integer);

PROCEDURE MovePnl(P: PnlIndex; Offset: point);
{$IFC TEDEBUG}

PROCEDURE prntClip;

PROCEDURE PRNTPORT;

```

```
   PROCEDURE prntRect(aRect: rect);

   PROCEDURE PrntRg(s: str40; rg: range);
   {$ENDC}

   FUNCTION ptInPnl(portptr: grafptr; loc: point; hipnl: pnlindex): PnlIndex;

   FUNCTION rgEq(aRg, bRg: range): boolean;

   FUNCTION SectPnl(Pnl: pnlIndex; portptr: grafptr; updrgn: rgnhandle;
                    resultrgn: rgnhandle): BOOLEAN;

   PROCEDURE SetPnlPort(P: PnlIndex);

   PROCEDURE setRg(VAR aRg: range; aKind: rgKind; aLoRow, aLoCol, aHiRow,
                   aHiCol: idType);

   PROCEDURE SetUpPnl(P: PnlIndex; portptr: grafptr; width, Height: pixel;
                      loc: point);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 219 Characters: 6969

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : TM
*
********************************************************************************

USES {$U+} TM;


INTRINSIC;

INTERFACE

   USES                                {$U libsm/UnitStd } UnitStd,
      {$U libsm/UnitHz } UnitHz,
      {$U libqd/Storage } Storage,
      {$U libqd/QuickDraw } QuickDraw,
      {$U libfm/FontMgr   } FontMgr,
      {$U libdb/dbenv } dbenv,
      {$U libfe/FEdec } FEdec,
      {$U libfe/fld } fieldedit,
      {$U libte/teenv } teenv,
      {$U libsb/WMLstd } WMLstd,
      {$U libsb/WMlsb } WMlsb;
      {$ifc teProduction }
      {$SetC tmdebug := False }
      {$elsec }
      {$SetC tmdebug := True }
      {$endc }
      {$SetC TMSDebug := tmDebug }

   CONST
      {$IFC NOT tmdebug}
      traceSMgr     = False;
      tracetmgr     = false;
      traceXtmgr    = false;
      fwriteln      = false;
      {$ENDC }
      ascCR         = 13;
      maxString     = 130;
      splitDh       = dhSkewer;
      splitDv       = dvSkewer;
      maxrun        = 5;

   TYPE
      cellSelType   = (selectCell, selectContents, PutCaretAtEnd);
      {$IFC tmdebug}

   VAR
      traceSMgr, traceTMgr, traceXTMgr, fwriteln: boolean;
      {$ENDC }

   PROCEDURE ChgRgBorders(newRg: range; shoMarquee: boolean);

   PROCEDURE DumpSel;

   PROCEDURE EndRgBorders;

   PROCEDURE FindTblPt(hv: point; VAR PrtOfTbl: tblPart; VAR rg: range;
                       VAR virtRect, screenRect: rect);

```

```
PROCEDURE getCR(VAR rg: range);

PROCEDURE getEditCell(VAR theRow: idType; VAR theCol: idType);

FUNCTION grayOfBand(aKind: snipKind; aP: pixel): fract;

PROCEDURE growBand(aKind: snipKind; aP: pixel; VAR newP: pixel;
                   VAR result: BGrowResult);

PROCEDURE LDtmgr;

PROCEDURE LDtmLoFreq;

PROCEDURE NewRgBorders(newRg: range; useMarquee: boolean);

PROCEDURE NewRgImg(rg: range);

FUNCTION PtIn1CRg(hv: point; targRg: range): boolean;

PROCEDURE ReBldTbl(initRg: range);

PROCEDURE ScrBand(aKind: snipKind; aP: pixel; scrScale: ScrollType;
                  scrDir: ScrollDir; scrFract: fract; VAR didMove: boolean);

PROCEDURE setCR(rg: range);

PROCEDURE shoNewRg(oldRg: range; newRg: range; just: rgJust);

PROCEDURE ShoNewWid(aKind: snipKind; loSnip, hiSnip: idType);

PROCEDURE ShoTbl(shoRgnH: rgnHandle);

FUNCTION snipAllInBand(aKind: snipKind; aP: pixel; snipId: idType): boolean;

PROCEDURE SplitBand(aKind: snipKind; VAR atP: pixel; VAR didSplit: boolean);

PROCEDURE tmCurVwPt(VAR vwPt: point);

PROCEDURE tmCurVwSet(vwPt: point);

PROCEDURE tmFAct(fActivate: boolean);

PROCEDURE tmInit;

PROCEDURE tmLoadBands;

PROCEDURE tmLoadNextBand(aTmBand: tmBand);

PROCEDURE tmLoadState(atmstate: tmstate);

PROCEDURE tmNewPars;

PROCEDURE tmNewSnips(aKind: snipKind; priorSnip: idType; scrollit: boolean);

PROCEDURE tmNextBandRg(VAR bandsnip: snipKind; VAR bandloId,
                       bandhiId: idType);

PROCEDURE tmReDoViews;

PROCEDURE tmReSize;

PROCEDURE tmSaveBands;

PROCEDURE tmSaveNextBand(VAR aTmBand: tmBand);

PROCEDURE tmsavestate(VAR atmstate: tmstate);
```

```

    PROCEDURE tmSetRgFB(cRg: range; useMarquee: boolean);

    PROCEDURE tmShoBreaks;

    PROCEDURE tmShoNewTbl;

    PROCEDURE TurnOffMarquee;

    PROCEDURE TurnOnMarquee;

    FUNCTION underEdit(aRow, aCol: idType): boolean;

    PROCEDURE visRgOfView(aPt: point; VAR aRg: range);

    PROCEDURE InitProcInfo(CellInfo: ProcPtr; CellRun: ProcPtr;
                           CellValue: ProcPtr; FoldedFormula: ProcPtr;
                           GetSnipId: ProcPtr; mmBreakSnip: ProcPtr;
                           mmMapSnip: ProcPtr; SetSnipWidth: ProcPtr;
                           SnipLT: ProcPtr; SnipWidth: ProcPtr;
                           FatalError: ProcPtr);

    PROCEDURE boldflds(VAR status: integer);

    FUNCTION CellChanged: Boolean;

    PROCEDURE ChangeCellFont;

    PROCEDURE ClearSel(VAR status: integer);

    PROCEDURE CopySel(VAR status: integer);

    PROCEDURE CutSel(VAR status: integer);

    PROCEDURE DoSelAct(Activate: boolean);

    PROCEDURE DoSelCR(aRg: Range; VAR timeout: INTEGER; hilite: cellseltype);

    PROCEDURE DoTblSel(P: point; VAR Nbh: Rect; VAR Timeout: INTEGER);

    PROCEDURE DoWavSel(P: Point; VAR Nbh: Rect; VAR Timeout: Integer);

    PROCEDURE EndCR;

    PROCEDURE EnterSelection(VAR Status: Integer);

    PROCEDURE EraseWav;

    PROCEDURE ExitSel;

    PROCEDURE ExtendSel(P: point; VAR nbh: Rect; VAR delta: INTEGER);

    PROCEDURE GetSelKind(VAR theSelKind: selKind);

    PROCEDURE InitSelMgr(wavActFlg: boolean; maxLenWav: integer;
                         growdata: integer; growrun: integer);

    PROCEDURE InKey(ch: char; shiftflag: boolean; cmdflag: Boolean;
                    resetrun: boolean; VAR Timeout: INTEGER;
                    VAR status: INTEGER);

    PROCEDURE LDsmgr;

    PROCEDURE MovePointer(P: Point; VAR pRg: range; VAR nbh: rect);

    PROCEDURE MseUp(P: Point; VAR delta: INTEGER);
```

```

   PROCEDURE NewNbh(P: point; VAR nbh: Rect; VAR delta: INTEGER);

   PROCEDURE NewNbhWav(P: Point; VAR nbh: Rect; VAR delta: Integer);

   PROCEDURE NewTimOut(VAR delta: INTEGER);

   PROCEDURE NewWSel(P: Point; Pnl: PnlIndex; VAR Nbh: Rect;
                     VAR timeout: integer);

   PROCEDURE NullCR;

   PROCEDURE NullSel;

   PROCEDURE PackSMGR(hndsusdata: hnddata; offset: integer; curlen: integer;
                      VAR leninstalled: integer; VAR newlen: integer);

   PROCEDURE PasteSel(pasteruns: boolean; PasteLimit: integer;
                      VAR status: integer);

   PROCEDURE PointAtRg(P: Point; VAR pRg: range; VAR nbh: rect);

   PROCEDURE reDrawSel(showselect: boolean);

   PROCEDURE ReInitSelMgr;

   PROCEDURE restoreRgFB(VAR timeout: integer);

   PROCEDURE ReSelCR(VAR timeout: INTEGER; hilite: cellseltype);

   PROCEDURE restoreflds(VAR curtimeout: integer);

   PROCEDURE resizeflds;

   PROCEDURE SelAllTxt(VAR timeout: INTEGER);

   PROCEDURE SelNewCell(P: point; hilite: cellseltype; VAR nbh: Rect;
                        VAR timeout: integer);

   PROCEDURE SetAnchorCell(RowId, colId: idtype);

   PROCEDURE SetCellStr(str: str40);

   PROCEDURE UNDOSEL(VAR status: integer);

   FUNCTION ValidCoord(rg: range): Boolean;

   PROCEDURE WhTblSel(rg: range; VAR nbh: Rect; VAR timeout: integer);

   PROCEDURE unboldflds(VAR status: integer);

   PROCEDURE UnpackSMGR(HndSdata: Hnddata; offset: integer);
   {$IFC tmDebug }

   PROCEDURE startTMtime;

   PROCEDURE stopTMtime;
   {$ENDC}

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************
```

End of File -- Lines: 258 Characters: 6533

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : TREES
*
*******************************************************************************

USES {$U+} TREES;


{$IFC isIntrin} INTRINSIC; {$ENDC}

INTERFACE

   USES
      {$IFC SrcOnOS }
         (*$U OBJ/PASDEFS *) PasDefs,
         (*$U OBJ/MEMMAN  *) MemMan;
      {$ELSEC }
         (*$U OBJ:PASDEFS *) PasDefs,
         (*$U OBJ:MEMMAN  *) MemMan;
      {$ENDC }

   {$SETC DEBUGF = 0 }

   CONST


      NullNode      = - 32768;

   TYPE
      NRRowPtr      = ^NRRow;
      NRRow         = ARRAY [ - 128..127] OF NameString;
      NRHandle      = ^NameArr;
      NameArr       = ARRAY [ - 128..127] OF NRRowPtr;
      CompResult    = (ToLeft, Equal, ToRight);
      TreeType      = (NameT, Future1, Future2, Future3, Future4);
      Node          = integer;
      NodeRec       = RECORD
                         lLink, rLink: Node;
                      END;
      NodeRow       = ARRAY [ - 128..127] OF NodeRec;
      NodeRowPtr    = ^NodeRow;
      NodeArr       = ARRAY [ - 128..127] OF NodeRowPtr;
      NodeArrPtr    = ^NodeArr;
      TreeHandle    = ^TreeRec;
      TreeRec       = RECORD
                         nodMem: MMHandle;
                         nodes: NodeArrPtr;
                         maxNodes, curNodes: longint;
                         root: Node;
                         compare: ProcPtr;
                         store: ProcPtr;
                         CASE TRType: TreeType OF
                            NameT:
                               (names: NRHandle);
                            Future1, Future2, Future3, Future4:
                               (FutPtr: longint);
                      END;

   FUNCTION EnterNode(t: TreeHandle; VAR newN: NameString;
                      PROCEDURE duplicate
```

```
                        (t: TreeHandle; VAR np: NameString; n: Node);
                      PROCEDURE newNode
                        (t: TreeHandle; VAR x: NameString; VAR newN: Node)): Node;

   FUNCTION TLookUp(t: TreeHandle; VAR x: NameString): Node;

   PROCEDURE InOTraverse(t: TreeHandle; PROCEDURE
                         Visit(i: Node));

   FUNCTION InitNameTree(n: longint): TreeHandle;

   FUNCTION GrowTree(t: TreeHandle; n, row: integer): boolean;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 80 Characters: 2435

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITCS
*
*******************************************************************************

USES {$U+} UNITCS;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libsm/UnitStd    } UnitStd,
      {$U libsm/UnitHz     } UnitHz,
      {$U libos/SysCall    } SysCall,
      {$U libqd/Storage    } Storage,
      {$U libqd/QuickDraw  } QuickDraw,
      {$U libfm/FontMgr    } FontMgr,
      {$U libpm/PmDecl     } PmDecl,
      {$U libpr/PrStdInfo  } PrStdInfo,
      {$U libsu/UnitFmt    } UnitFmt;

   CONST
      {$IFC NOT fDbgOk}
      fTstCs        = FALSE;           {$ENDC}
      clpdDft       = 4;
      icsLst        = 10;
      ointNil       = 0;
      orecNil       = MAXINT;
      lpMax         = 2000000000;
      iprocSeqLpd   = 0;
      iprocFindLpFixed = 1;
      iprocHilight  = 2;
      iprocCrd      = 3;
      iprocFreeIcs  = 4;
      iprocFSelLpBounds = 5;
      iprocPxHcs    = 6;
      iprocLstCs    = 6;
      iimgpScrn     = 0;
      iimgpQume     = 1;
      iimgpCIto     = 2;
      iimgpLst      = 2;

   TYPE
      TLp           = TL;
      TArglp        = ARRAY [0..0] OF TLp;
      TRglp         = ^TArglp;
      TInt          = RECORD
                         ointNxt: TC;
                         ointPrv: TC;
                         ointParent: TC;
                         lpFst: TLp;
                         lpLim: TLp;
                         ics: TB;
                         fValid: TF;
                      END;
      TPint         = ^TInt;
      TArgint       = ARRAY [0..0] OF TInt;
      TRgint        = ^TArgint;
   THpic          = PicHandle;
   TPpic          = PicPtr;
   TFigd          = RECORD
                       n: TN;
                       ppic: TPpic;
                    END;
   TTyset         = RECORD
                       fRce: TF;
                       fParBnds: TF;
                       fRpe: TF
                    END;
   TAlpd          = RECORD
                       ics: TB;
                       ilpd: TB;
                       fParSt: TF;
                       lp: TLp;
                       lpLim: TLp;
                       lpSon: TLp;
                       icsSon: TB;
                       tyset: TTyset;
                       rce: TRce;
                       lpFstPar: TLp;
                       lpLimPar: TLp;
                       rpe: TRpe;
                       arce: TArce;
                       arpe: TArpe;
                       figd: TFigd;
                    END;
   TLpd           = ^TAlpd;
   TTycs          = (tycsNil, tycsUs, tycsEd, tycsFf, tycsUf, tycsFm, tycsPg,
                     tycsRuler, tycsLst, tycsFld, tycsMtrx, tycsFig);
   TCspd          = RECORD
                       argproc: ARRAY [0..iprocLstCs] OF TProc;
                       argimgp: ARRAY [0..iimgpLst] OF TProc;
                    END;
   TTyxy          = (tyxyScrn, tyxyIP, tyxyPgFract, tyxyMica, tyxyDotMx,
                     tyxyHiDotMx, tyxyLDotMx, tyxyLHiDotMx, tyxyNew1,
                     tyxyNew2, tyxyNew3, tyxyNew4, tyxyNew5, tyxyNew6,
                     tyxyNew7, tyxyNew8, tyxyNil);
   TCs            = RECORD
                       cspd: TCspd;
                       hz: THz;
                       tycs: TTycs;
                       tyxy: TTyxy;
                       ointFst: TC;
                       orecConFst: TC;
                       cRef: TC;
                       prprf: TPrRec;
                    END;
   TPcs           = ^TCs;
   THcs           = ^TPcs;
   TArghcs        = ARRAY [0..0] OF THcs;
   TRghcs         = ^TArghcs;
   TPglp          = RECORD
                       icsHdr: TB;
                       icsFtr: TB;
                       yTopHdr: TY;
                       dyHdr: TY;
                       dyHtBody: TY;
                       dyBotBody: TY;
                       tyxy: TTyxy;
                    END;
   TAmpicsprcs    = ARRAY [0..0] OF TL;
   TMpicsprcs     = ^TAmpicsprcs;

VAR
```

```
      {$IFC fDbgOk}
      fTstCs:       TF;                {$ENDC}
      tysetRpe:     TTyset;
      tysetBnds:    TTyset;
      rghcs:        TRghcs;
      mpicsprcs:    TMpicsprcs;
      icsMac:       TC;
      argalpd:      ARRAY [0..clpdDft] OF TAlpd;
      arglpd:       ARRAY [0..clpdDft] OF TLpd;
      lpdStd:       TLpd;
      tysetStd:     TTyset;
      lldInt:       TLld;
      tyxyLst:      TTyxy;
      tyxyCs:       TTyxy;
      pglpStd:      TPglp;
      fNoInvalidate: TF;

   FUNCTION LpMinP(lp1: TLp; lp2: TLp): TLp;

   FUNCTION LpMaxP(lp1: TLp; lp2: TLp): TLp;

   PROCEDURE PxRgcs;

   PROCEDURE PxAchad(VAR achad: TAchad);

   PROCEDURE PxCs(ics: TB);

   PROCEDURE FreeInt(ics: TB; oint: TC);

   FUNCTION OintMark(oint: TC; ics: TB; lpFst: TLp; lpLim: TLp; fValid: TF;
                     ointParent: TC): TC;

   FUNCTION IcsCreate(tycs: TTycs; cbCs: TC; hz: THz): TB;

   PROCEDURE FreeIcs(ics: TB);

   PROCEDURE PxLpd(lpd: TLpd; VAR achad: TAchad);

   PROCEDURE SetLpd(lpd: TLpd; ics: TB; lp: TLp; tyset: TTyset;
                    VAR achad: TAchad);

   PROCEDURE SeqLpd(lpd: TLpd; VAR achad: TAchad);

   FUNCTION FParStart(ics: TB; lp: TLp): TF;

   PROCEDURE FindLpFixed(ics: TB; lp: TLp; VAR lpFixed: TLp);

   PROCEDURE FindLpFstPar(ics: TB; lp: TLp; VAR lpFixed: TLp);

   PROCEDURE InitCs(hz: THz);

   PROCEDURE PxInt(oint: TC);

   PROCEDURE AdjustLps(ics: TB; lpFstInv: TLp; lpLimInv: TLp; dlpAdjust: TLp);

   FUNCTION FSelLpBounds(lp: TLp; ics: TB; VAR lpFstSel, lpLimSel, lpFstAtom,
                         lpLimAtom: TLp): TF;

   FUNCTION TrueStdSelLpBounds(lp: TLp; ics: TB; VAR lpFstSel, lpLimSel,
                               lpFstAtom, lpLimAtom: TLp): TF;

   FUNCTION FTextIcs(ics: TB): TF;

   PROCEDURE PurgeIcsPrcs;

*******************************************************************************
```

```
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 198 Characters: 5836

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITFF
*
*******************************************************************************

USES {$U+} UNITFF;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libsm/UnitStd    } UnitStd,
      {$U libsm/UnitHz     } UnitHz,
      {$U libos/SysCall    } SysCall,
      {$U libsu/UnitFile   } UnitFile,
      {$U libqd/QuickDraw  } QuickDraw,
      {$U libfm/FontMgr    } FontMgr,
      {$U libqd/Storage    } Storage,
      {$U libsu/UnitFmt    } UnitFmt,
      {$U libpm/PmDecl     } PmDecl,
      {$U libpr/PrStdInfo  } PrStdInfo,
      {$U libsu/UnitCS     } UnitCS;

   CONST
      {$IFC NOT fDbgOk}
      fTstFf        = FALSE;           {$ENDC}
      pwFf          = - 13142;
      pwNotFf       = 0;
      cbPgff        = 512;
      dxMinTab      = 6;
      ffVersion     = 5;
      ffVerPrPrf    = 2;
      ffVerNewRgfbcs = 3;
      ffVerMsgFst   = 3;
      ffVerCommaDecimal = 4;
      ffVerPrRec    = 5;

   TYPE
      TTyRlU        = (tyrluEnglish, tyrluMetric, tyrluPica, tyrluElite);
      TFfLeader     = RECORD
                           password: TW;
                           version: TC;
                           lpHdrFst: TLp;
                           lpFtrFst: TLp;
                           lpTxtFst: TLp;
                           dummyLpLimFf: TB;
                           tyrlu: TTyRlU;
                           pglp: TPglp;
                           prprf: TPrRec;
                        END;

   VAR
      cspdFf:       TCspd;
      {$IFC fDbgOk}
      fTstFf:       TF;
      {$ENDC}

   PROCEDURE PxHcsff(hcs: THcs);

   PROCEDURE PxLeader(VAR leader: TFfleader);

   PROCEDURE SeqLpdFf(lpd: TLpd; VAR achad: TAChad);

   FUNCTION IcsFfCreate(ifil: TC; icsFig: TB; hz: THz;
                        VAR ffleader: TFfleader): TB;

   PROCEDURE FreeFfIcs(icsFf: TB);

   PROCEDURE WriteIcsFf(icsDoc: TB; pglp: TPglp; tyrluFf: TTyRlU; ifil: TB;
                        ifilFig: TB; hzDoc: THz);

   PROCEDURE InitFF;

   FUNCTION LpMacFf(icsFf: TB): TLp;

   PROCEDURE SetLeaderOfIcsFfFromIFil(ics: TB; ifil: TC; hz: THz);

   FUNCTION IfilOfFf(icsFf: TB): TB;

   PROCEDURE GetLeaderOfIcsFf(icsFf: TB; VAR ffleader: TFfLeader);

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 90 Characters: 2363

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITFIGA
*
*******************************************************************************

USES {$U+} UNITFIGA;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libsm/UnitStd    } UnitStd,
      {$U libsm/UnitHz     } UnitHz,
      {$U libos/SysCall    } SysCall,
      {$U libsu/UnitFile   } UnitFile,
      {$U libqd/QuickDraw  } QuickDraw,
      {$U libfm/FontMgr    } FontMgr,
      {$U libqd/Storage    } Storage,
      {$U libsu/UnitFmt    } UnitFmt,
      {$U libpm/PmDecl     } PmDecl,
      {$U libpr/PrStdInfo  } PrStdInfo,
      {$U libsu/UnitCS     } UnitCS;

   CONST
      {$IFC NOT fDbgOk}
      fTstFig       = FALSE;           {$ENDC}
      tyfilFig      = - 13143;
      verFigCur     = 1;
      cErrFilNotValid = 5100;

   TYPE
      TCsfig        = RECORD
                           cs: TCs;
                           ifil: TB;
                        END;
      TPcsfig       = ^TCsfig;
      THcsfig       = ^TPcsfig;

   VAR
      cspdFig:      TCspd;
      {$IFC fDbgOk}
      fTstFig:      TF;
      {$ENDC}

   PROCEDURE PxHcsfig(hcsfig: THcsfig);

   FUNCTION  IcsFigCreate(VAR cError: TC; path: Pathname; setaccess: TSetaccess;
                          hz: THz; password: E_name): TB;

   PROCEDURE FreeFigIcs(ics: TB);

   PROCEDURE FigFindLpFixed(ics: TB; lp: TLp; VAR lpFixed: TLp);

   FUNCTION  LpFigAdd(ics: TB; hpic: THpic): TLp;

   PROCEDURE SeqLpdFig(lpd: TLpd; VAR achad: TAChad);

   FUNCTION  IfilOfFig(icsFig: TB): TB;

   PROCEDURE InitFig;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 70 Characters: 1729

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

```
******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITFILE
*
******************************************************************************

USES {$U+} UNITFILE;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libsm/UnitStd} UnitStd,
      {$U libsm/UnitHz} UnitHz,
      {$U libos/SysCall} SysCall,
      {$U libos/PSysCall} PSysCall;

   CONST
      ifilNil       = - 1;
      ifilRangeWarn = - 4056;
      noMFileErr    = 4057;
      ifilNotOpenWarn = - 4058;
      ivodNil       = MAXINT;
      ivodArgvod    = 0;
      ivodFst       = 1;
      cbVofilOnFile = 128;
      cvodMore      = 48;
      cbBlkOfVofil  = 512;
      {$IFC NOT fDbgOk}
      fTstFile      = FALSE;
      {$ENDC}

   TYPE
      String255     = STRING[255];
      TVod          = RECORD
                         ib: TL;
                         cb: TL;
                      END;
      TVofil        = RECORD
                         tyfil: TW;
                         ver: TC;
                         ivodMac: TC;
                         vodArgvod: TVod;
                      END;
      TArgvod       = ARRAY [1..1] OF TVod;
      TRgvod        = ^TArgvod;
      TSetaccess    = MSet;

   VAR
      Scrap_RefNum: TL;
      {$IFC fDbgOk}
      fTstFile:     TF;
      {$ENDC}

   PROCEDURE InitFiles(hz: THz);

   FUNCTION IfilOpen(VAR cError: TC; pathIn: String255; cbBlk: TC;
                     setaccess: TSetaccess; hzWrite: THz;
                     password: E_name): TB;

   PROCEDURE CloseIfil(VAR cError: TC; ifil: TB);

   FUNCTION FilCbOfN(n: TN): TC;

   FUNCTION FilFSwapInN(hz: THz; n: TN): TF;

   PROCEDURE FilSwapOutN(hz: THz; n: TN);

   FUNCTION NFromIfilIb(ifil: TB; ib: TL): TN;

   FUNCTION CBlkOfIfil(ifil: TC): TC;

   PROCEDURE RenameFile(VAR cError: TC; pathIn: String255; enameIn: TSp;
                        password: E_name);

   PROCEDURE KillFile(VAR cError: TC; pathIn: String255; password: E_name);

   FUNCTION FMonitorFile(path: String255): TF;

   PROCEDURE PathOfIfil(ifil: TC; VAR path: String255; VAR password: e_name);

   PROCEDURE SetIbLimOfIfil(ifil: TB; ibLim: TL);

   FUNCTION RefnumOfIfil(ifil: TB): TC;

   PROCEDURE CleanIfil(VAR cError: TC; hz: THz; ifil: TB; fIgnoreDirty: TF);

   FUNCTION CErrOfIfil(ifil: TB): TC;

   PROCEDURE GetVofil(ifil: TB; VAR vofil: TVofil);

   FUNCTION IvodAdd(ifil: TB; hSrc: TH; cb: TL): TC;

   PROCEDURE SetTyfilVer(ifil: TB; tyfil: TW; ver: TC);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 103 Characters: 2579

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

```
******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITFMT
*
******************************************************************************

USES {$U+} UNITFMT;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libsm/UnitStd} UnitStd,
      {$U libsm/UnitHz} UnitHz,
      {$U libqd/QuickDraw} QuickDraw,
      {$U libfm/FontMgr} FontMgr;
      {$SETC doTraceSU := TRUE}
      {$SetC fTraceSU := doTraceSU AND fTRACE}

   CONST
      {$IFC NOT fDbgOk}
      fTstFmt       = FALSE;           {$ENDC}
      ifldQuad      = 0;
      ifldXLFst     = 1;
      ifldXLBody    = 2;
      ifldXRt       = 3;
      ifldXtb       = 4;
      ifldTbQuad    = 5;
      ifldRgtbd     = 6;
      ifldDxtb      = 7;
      ifldBold      = 8;
      ifldItalic    = 9;
      ifldUnderline = 10;
      ifldFont      = 11;
      ifldVan       = 12;
      ifldSuperscript = 13;
      ifldYLine     = 14;
      ifldYLd       = 15;
      ifldMovTb     = 16;
      ifldClrTb     = 17;
      ifldFilTb     = 18;
      ifldKeep      = 19;
      ifldLpFig     = 20;
      ifldIcsFig    = 21;
      ifldTyfam     = 22;
      ifldFsp       = 23;
      ifldLst       = 23;
      fopcEnd       = - 1;
      ibRgtbd       = 14;
      cbRceVan      = 2;
      ifnt12Tile    = 0;
      ifntp10Tile   = 1;
      ifntp12Tile   = 2;
      ifnt18Tile    = 3;
      ifnt24Tile    = 4;
      ifnt12Cent    = 5;
      ifnt18Cent    = 6;
      ifnt24Cent    = 7;
      ifntFBold     = 8;
      ifntp10Cent   = 9;
      ifntp12Cent   = 10;
      ifntelite     = 11;
      ifntcourier   = 12;
      ifntp20Tile   = 13;
      ifntp15Tile   = 14;
      ifntp20Cent   = 15;
      ifntp15Cent   = 16;
      ifntLst       = 16;
      ibXTbd        = 0;
      ibChTbd       = 2;
      ibRgfbc       = 1;
      cbHfccInit    = 10;
      itbdLst       = 58;
      icsNil        = 0;
      lpNil         = - 2;
      tyfamModern   = 0;
      tyfamClassic  = 1;
      tyfamLst      = 1;
      tyfam1Fill    = 2;
      tyfam2Fill    = 3;
      fsp8P20       = 0;
      fsp8P15       = 1;
      fsp12P12      = 2;
      fsp12P10      = 3;
      fsp12         = 4;
      fsp14         = 5;
      fsp18         = 6;
      fsp24         = 7;
      fspLst        = 7;

   TYPE
      TFcc          = RECORD
                           cref: TB;
                           argfbc: TArgb;
                        END;
      TPfcc         = ^TFcc;
      THfcc         = ^TPfcc;
      TQuad         = (quadL, quadC, quadR, quadJ);
      TTyfill       = (tyfillNil, tyfillDots, tyfillHyph, tyfillUL);
      TTbdOld       = PACKED RECORD
                           x: TX;
                           fill4: 0..15;
                           quad: TQuad;
                           tyfill: TTyfill;
                           chLdr: TCh;
                        END;
      TTbd          = PACKED RECORD
                           x: TX;
                           fill3: 0..7;
                           fDecimalComma: TF;
                           quad: TQuad;
                           tyfill: TTyfill;
                           chLdr: TCh;
                        END;
      TArgtbd       = ARRAY [0..0] OF TTbd;
      TRgtbd        = ^TArgtbd;
      TArcpe        = PACKED RECORD
                           cb: TB;
                           b1: TB;
                        END;
      TRcpe         = ^TArcpe;
      TArce         = PACKED RECORD
                           cb: 0..255;
                           fVan: TF;
                           fBold: TF;
```

```
                      fItalic: TF;
                      fUnderline: TF;
                      fill4: 0..15;
                      cbSuperscript: TB;
                      ifnt: TB;
                      fKeep: TF;
                      fOutLine: TF;
                      fShadow: TF;
                      fFillB: TF;
                      fFillC: TF;
                      fFillD: TF;
                      fFillE: TF;
                      fFillF: TF;
                      icsFig: TB;
                      lpFig: TC;
                   END;
   TRce          = ^TArce;
   TArpe         = PACKED RECORD
                      cb: 0..255;
                      sy: TB;
                      xLftFst: TX;
                      xLftBody: TX;
                      xRt: TX;
                      yLd: TY;
                      fill1: TB;
                      yLine: 0..63;
                      quad: TQuad;
                      itbLim: TW;
                      argtbd: ARRAY [0..itbdLst] OF TTbd;
                   END;
   TRpe          = ^TArpe;
   TTyfam        = 0..3;
   TFsp          = TB;
   TFstyle       = RECORD
                      tyfam: TTyfam;
                      fsp: TFsp;
                   END;
   TTydst        = (tydstRce, tydstRpe, tydstTbd, tydstNil);

VAR
   {$IFC fDbgOk}
   fTstFmt:      TF;                {$ENDC}
   argfam:       ARRAY [0..ifntLst] OF TFam;
   tbdNil:       TTbd;
   arceStd:      TArce;
   arpeStd:      TArpe;
   ampifntfstyle: ARRAY [0..ifntLst] OF TFstyle;

PROCEDURE PxArce(VAR arce: TArce);

PROCEDURE PxArpe(VAR arpe: TArpe);

FUNCTION CfbcApplyIfld(ifld: TC; wsval: TW; pval: TPc; rpe: TRpe;
                       rce: TRce): TC;

FUNCTION IfbcLstApplyRgfbc(rgfbc: TRgb; rpe: TRpe; rce: TRce): TC;

PROCEDURE AddRgfbc(VAR hfcc: THfcc; hfccSrc: THfcc; hz: THz);

PROCEDURE AddFop(VAR hfcc: THfcc; ifld: TB; sval: TW; pval: TPc; hz: THz);

PROCEDURE GenFopsDiff(hfcc: THfcc; tydst: TTydst; rcpeStd, rcpe: TP;
                      hz: THz);

PROCEDURE zzGenParFops(hfcc: THfcc; rpe: TRpe; hz: THz);

```

```
   FUNCTION ItbFromX(rgtbd: TRgtbd; itbLst: TC; x: TX): TC;

   PROCEDURE InitFmt;

   PROCEDURE PxRgfbc(rgfbc: TRgb);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 205 Characters: 5721

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITHZ
*
********************************************************************************

USES {$U+} UNITHZ;


INTRINSIC SHARED;

INTERFACE
   {$SETC fhz  := FALSE}
   {$SETC fhzP := FALSE}

   USES
      {$U libsm/UnitStd} UnitStd,
      {$U libos/SysCall} SysCall;

   CONST
      tybkFree      = 0;
      tybkStd       = 1;
      tybkN         = 2;
      tybkNrel      = 3;
      {$IFC NOT fhz}
      fTstHz        = FALSE;           {$ENDC}
      {$IFC NOT fhz}
      fTstHzV       = FALSE;           {$ENDC}
      {$IFC NOT fhzP}
      fTstHzP       = FALSE;           {$ENDC}
      ipPoolMax     = 32000;
      cbMinAsk      = 512;
      MaxBlkSize    = 32767;

   TYPE
      TH            = ^TP;
      TArgh         = ARRAY [0..0] OF TH;
      TRgh          = ^TArgh;
      TTybk         = 0..3;
      THz           = ^TAhz;
      TBp           = PACKED RECORD
                         fRelPBase: TF;
                         CASE TF OF
                            FALSE:
                               (ip: 0..32000);
                            TRUE:
                               (iwP: 0..32000);
                      END;
      TPpn          = ^TPn;
      TStn          = PACKED RECORD
                         tybk: TTybk;
                         fDirty: TF;
                         fLock: TF;
                         fill: 0..15;
                         ubt: TB;
                      END;
      TNob          = RECORD
                         n: TN;
                         stn: TStn;
                         bp: TBp;
                         data: TW;
```

```
                   END;
   TPnob          = ^TNob;
   TPpnob         = ^TPnob;
   TArgpnob       = ARRAY [0..0] OF TPnob;
   TRgpnob        = ^TArgpnob;
   THrgpnob       = ^TRgpnob;
   TBk            = ^TAbk;
   TAbk           = RECORD
                      CASE TF OF
                         FALSE:
                            (hdr: PACKED RECORD
                                     tybk: TTybk;
                                     cw: 0..16000;
                                  END;
                             CASE TTybk OF
                                tybkNrel:
                                   (dataNrel: TW);
                                tybkStd:
                                   (bp: TBp;
                                    dataStd: TW);
                                tybkN:
                                   (SpaceForCompactRoutine: Integer;
                                    nob: TNob); );
                         TRUE:
                            (cwFree: TL;
                             bkfNxt: TBk;
                             bkfPrv: TBk; );
                   END;
   TArgwBase      = ARRAY [0..0] OF TW;
   TRgwBase       = ^TArgwBase;
   TAhz           = RECORD
                      bkFst: TBk;
                      bkLst: TBk;
                      bkfFst: TBk;
                      rgwBase: TRgwBase;
                      ipPoolMac: TC;
                      hFstFree: TH;
                      cbFree: TL;
                      hrgpnob: THrgpnob;
                      mskIpnLst: TW;
                      ipnCur: TC;
                      cpnAvail: TC;
                      ubtCur: TB;
                      procCbMore: TProc;
                      procCbOfN: TProc;
                      procFSwapInN: TProc;
                      procSwapOutN: TProc;
                      fScramble: TF;
                      hScramble: TH;
                      fUpScramble: TF;
                      fCheck: TF;
                      cCompact: TC;
                      HasConcrete: TF;
                      argpPool: ARRAY [0..0] OF TP;
                   END;

VAR
   hNil:         TH;
   pNil:         TP;
   hzNil:        THz;
   {$IFC fhz}
   fTstHzV:      TF;                {$ENDC}
   {$IFC fhz}
   fTstHz:       TF;                {$ENDC}
   {$IFC fhzP}
   fTstHzP:      TF;                {$ENDC}

   FUNCTION HAllocate(hz: THz; cb: TC): TH;

   FUNCTION HzInit(pFst: TP; pLim: TP; pBase: TP; ipPoolMac: TC; logIpnLim: TC;
                   procCbMore: TProc; procCbOfN: TProc; procFSwapInN: TProc;
                   procSwapOutN: TProc): THz;

   PROCEDURE PxHz(hz: THz);

   PROCEDURE ChangeSizeH(hz: THz; h: TH; cbNew: TC);

   PROCEDURE ChangeNSize(hz: THz; n: TN; cbNew: TC);

   PROCEDURE FreeBk(hz: THz; h: TH; tybk: TTybk);

   PROCEDURE AllocBk(hz: THz; hDst: TH; cb: TC; tybk: TTybk);

   FUNCTION PMapN(hz: THz; nSrc: TN): TP;

   FUNCTION PCreateNob(hz: THz; nSrc: TN; cbData: TC): TP;

   PROCEDURE SetFDirty(hz: THz; n: TN; fDirty: TF);

   PROCEDURE SetCbFree(hz: THz; cbFree: TL; fEnlargeHz: TF);

   FUNCTION CbDataOfH(hz: THz; h: TH): TC;

   FUNCTION PAllocate(hz: THz; cb: TC): TP;

   PROCEDURE FreeH(hz: THz; h: TH);

   PROCEDURE FreeP(hz: THz; p: TP);

   PROCEDURE FreeN(hz: THz; n: TN);

   PROCEDURE ReleaseBkNrel(hz: THz; pFstRelease: TP);

   FUNCTION PLstFree(hz: THz): TP;

   FUNCTION HzFromH(h: TH): THz;

   FUNCTION FCheckHzOk(hz: THz; VAR cBkStd: TC): TF;

   FUNCTION HLockN(hz: THz; n: TN; fNeedH: TF): TH;

   PROCEDURE UnlockN(hz: THz; n: TN);

   PROCEDURE LockHandleInConcrete(h: TH);

   PROCEDURE UnlockHandleFromConcrete(h: TH);

   PROCEDURE EnlargeHz(hz: THz; cbMore: TL);

   FUNCTION CbShrinkHz(hz: THz; cbLess: TL): TL;

   FUNCTION CbOfHz(hz: THz): TL;
   {$IFC fOS}

   FUNCTION CbMoreGrowSeg(hz: THz; cbNeed: TC): TC;
   {$ENDC}

   FUNCTION PnobFromN(hz: THz; n: TN): TPnob;

   PROCEDURE Scramble(hz: THz);

*******************************************************************************
```

```
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 198 Characters: 5673

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

***************************************************************************
*
*              APPLE LISA DESKTOP 3.0 LIBRARIES UNIT TABLE
*
***************************************************************************

UnitName Unit# File# Type   DataSize
-------- ----- ----- ------ --------
PASLIB      1    3  Intrin   000000
UNITSTD     2    4  Intrin   000002
GRAFUTIL    3    4  Intrin   000000
UNITHZ      4    4  Shared   000024
STORAGE     5    4  Intrin   000008
QUICKDRA    6    4  Intrin   0000CE
HWINT       7    4  Intrin   000000
FEDEC       8    4  Intrin   000000
FONTMGR     9    4  Shared   000054
EVENTS     10    4  Shared   000210
WINDOWS    11    4  Shared   0010F2
FOLDERS    12    4  Shared   00000A
MENUS      13    4  Shared   00009C
FLDUT      14    4  Intrin   000000
WMLSTD     15    4  Intrin   000018
WMLCRS     16    4  Intrin   0003FE
WMLGROW    17    4  Intrin   00000E
WMLSB      18    4  Intrin   000038
INTERNAT   19    4  Intrin   000000
FPLIB      20    5  Intrin   000006
ALERTMGR   21    4  Intrin   000300
MATHLIB    22    5  Intrin   000016
UNITFF     23    4  Shared   00004A
UNITCS     24    4  Shared   000644
UNITFILE   25    4  Shared   000156
UNITFMT    26    4  Shared   0001F2
BGENV      27    8  Intrin   000058
MATMGR     28    8  Intrin   00188E
PARSER     29    8  Intrin   0006FC
COMPUTE    30    8  Intrin   000084
STUBS      31    8  Intrin   000000
UUNIVERS   32    4  Intrin   000222
FEGLOBAL   33    8  Intrin   000398
COMGR      34    8  Intrin   00033C
USTDTERM   35    9  Intrin   0002CE
UQPGRAPH   36    9  Intrin   000056
TEEXEC     37    8  Intrin   000004
UQPPRINT   38    9  Intrin   000188
LCFEXEC    39    8  Intrin   0001A6
TEENV      40    6  Intrin   00080E
TM         41    6  Intrin   000BCA
FMGRUTIL   42    4  Intrin   000000
PMM        43    4  Shared   000042
PMDECL     44    4  Intrin   000000
UNITFIGA   45    4  Shared   00002A
SYS1LOCK   46    4  Intrin   000000
UCLASCAL   47    3  Intrin   00004E
UTKUNIVE   49   11  Intrin   0001CA
UOBJECT    50   10  Intrin   000164
UTEXT      51   11  Intrin   00027C
UDRAW      52   10  Intrin   0001B2
UABC       53   10  Intrin   0009F2
UDIALOG    54   11  Intrin   000564
```

```
LCUT       55    6  Intrin   00002C
IOPRIMIT   56   12  Intrin   00057E
SHELLCOM   57   12  Shared   00070C
PROGCOMM   58   12  Intrin   000006
RECOVERY   59    6  Intrin   000000
LOWLEVEL   60    6  Intrin   000126
DBDECL1    61    6  Intrin   000000
POOLER     62    6  Intrin   000052
DBENV      63    4  Intrin   000000
HEAP       64    6  Intrin   000000
VLTREE     65    6  Intrin   000000
CZCOMPAC   66    6  Intrin   00001C
LABSCAN    68    6  Intrin   000000
SCHEMA     69    6  Intrin   000000
SCAN       70    6  Intrin   000000
FIELDEDI   71    4  Intrin   0000CA
SCRAP      73    4  Shared   000250
FILERCOM   75    4  Shared   0002E8
PRPUBLIC   76    7  Intrin   000000
PRSTDINF   77    4  Shared   0005D0
PRSTDPRO   78    7  Intrin   000002
PRFILEPR   79    7  Intrin   000000
PRBUF      80    7  Intrin   000010
PRSPOOL    81    7  Intrin   000346
QUEUES     82    7  Intrin   000000
PREVENTS   83    7  Intrin   000000
PRDLGMGR   84    7  Intrin   0000BA
PRMGR      85    7  Intrin   000026
UVT100     86    9  Intrin   00001C
USOROC     87    9  Intrin   00001C
STDUNIT    88   12  Intrin   0005B0
IUMAN      90    1  Intrin   0000BC
OBJIO      91    1  Intrin   00011C
FILEIO     92    1  Intrin   000000
GRAPHS     94    1  Intrin   000008
TREES      95    1  Intrin   000000
REFS       96    1  Intrin   000004
PARTS      97    1  Intrin   000004
LISTS      98    1  Intrin   000000
MEMMAN     99    1  Intrin   000000
PASDEFS   100    1  Intrin   00025E
MPASLIB   101    2  Intrin   000000
BLKIOINT  102    3  Intrin   0000DE
BLOCKIO   103    3  Shared   0005C4
PASHEAP   104    3  Intrin   000022

***************************************************************************

                               THE END

***************************************************************************


End of File -- Lines: 113 Characters: 3628
```

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UNITSTD
*
********************************************************************************

USES {$U+} UNITSTD;


INTRINSIC;

INTERFACE
   {$Setc fDbgOK    := FALSE}
   {$Setc fOS       := TRUE}
   {$Setc fSymOK    := FALSE}
   {$Setc fstd1     := FALSE}
   {$Setc fTrace    := FALSE}
   {$SETC doTraceSM := TRUE}
   {$SETC fTraceSM  := doTraceSM AND fTrace}
   {$IFC fTrace}

   USES
      {$U libos/syscall.obj} Syscall;
      {$ENDC}
      (*$R-*)

   CONST
      chLst         = 127;
      chSp          = 32;
      chCr          = 13;
      chTab         = 9;
      olleNil       = 0;
      procNil       = 1;
      {$IFC NOT fDbgOk}
      fTstStd       = FALSE;           {$ENDC}
      {$IFC NOT fstd1}
      fTstStd1      = FALSE;           {$ENDC}

   TYPE
      TC            = INTEGER;
      TB            = - 128..127;
      TA            = 0..127;
      TP            = ^TB;
      TPc           = ^TC;
      TPa           = LONGINT;
      TF            = BOOLEAN;
      TX            = TC;
      TY            = TC;
      TL            = LONGINT;
      TW            = INTEGER;
      TCh           = TB;
      TCd           = TC;
      TN            = TL;
      TPn           = ^TN;
      TFd           = RECORD
                           iFil: TB;
                           bhi: TB;
                           iblo: TW
                        END;
      TPFd          = ^TFd;
      TNOrFd        = PACKED RECORD
```

```
                      CASE TC OF
                         1:
                            (fd: TFd);
                         2:
                            (n: TN);
                   END;
   TArgch        = ARRAY [0..0] OF TCh;
   TRgch         = ^TArgch;
   TAchad        = RECORD
                      rgch: TRgch;
                      ichFst: TC;
                      ichLim: TC;
                   END;
   TChad         = ^TAchad;
   TArgb         = ARRAY [0..0] OF TB;
   TRgb          = ^TArgb;
   THrgb         = ^TRgb;
   TAbad         = RECORD
                      rgb: TRgb;
                      ichFst: TC;
                      ichLim: TC;
                   END;
   TBad          = ^TAbad;
   TArgc         = ARRAY [0..0] OF TC;
   TRgc          = ^TArgc;
   TArgx         = ARRAY [0..0] OF TX;
   TRgx          = ^TArgx;
   TCmp          = (cmpLs, cmpEq, cmpGr);
   TAlld         = RECORD
                      olleFreeFst: TC;
                      olleLim: TC;
                   END;
   TLld          = ^TAlld;
   TAlle         = RECORD
                      olleNxt: TC;
                      ollePrv: TC;
                   END;
   TLle          = ^TAlle;
   TSp           = STRING[80];
   TPsp          = ^TSp;
   THsp          = ^TPsp;
   TARgCHAR      = PACKED ARRAY [0..0] OF CHAR;
   TRgCHAR       = ^TARgCHAR;
   TProc         = TP;

VAR
   Temp:         Integer;
   {$IFC fDbgOk}
   fTstStd:      TF;
   {$ENDC}
   {$IFC fstd1}
   fTstStd1:     TF;
   {$ENDC}
   {$IFC NOT fSymOk}                {$D-} {$ENDC}

FUNCTION CMin(c1: TC; c2: TC): TC;

FUNCTION CMax(c1: TC; c2: TC): TC;

PROCEDURE MoveAchad(VAR achadDst: TAchad; VAR achadSrc: TAchad);

PROCEDURE MoveRgch(rgchDst: TRgch; rgchSrc: TRgch; cb: TC);

PROCEDURE MoveOvRgch(rgchDst: TRgch; rgchSrc: TRgch; cb: TC);

PROCEDURE DivMod(num: TC; denom: TC; VAR quotient: TC; VAR rem: TC);
```

```

   FUNCTION IBinSearch(rgc: TRgc; icLst: TC; c: TC): TC;

   PROCEDURE Break(bkcd: TC);

   PROCEDURE MoveOlle(lld: Tlld; olle: TC; VAR olleDstFst: TC;
                      VAR olleSrcFst: TC);

   PROCEDURE PxLld(lld: TLld);

   PROCEDURE InitOlleFree(lld: TLld; olleFreeFst: TC; cbAlle: TC);

   FUNCTION IchCr(VAR achad: TAchad; fFwd: TF): TC;

   FUNCTION WAnd(w1, w2: TW): TW;

   FUNCTION WXor(w1, w2: TW): TW;

   FUNCTION WNot(w: TW): TW;

   FUNCTION WShLft(w: TW; cbitSh: TW): TW;

   FUNCTION WShRt(w: TW; cbitSh: TW): TW;

   PROCEDURE AsgnField(p: TP; ib: TW; mskNotFld: TW; val: TW; cbitSh: TC);

   FUNCTION ValOfField(p: TP; ib: TW; mskNotFld: TW; cbitSh: TC): TW;

   FUNCTION UpperCh(ch: CHAR): CHAR;

   PROCEDURE MakeSpUpper(VAR sp: TSp);
   {$IFC fTrace}

   PROCEDURE InitTrCalls;

   PROCEDURE LogCall;

   PROCEDURE ResetTrace(LogNamesAndSegs: Boolean);

   PROCEDURE StartCallLog;

   PROCEDURE StopCallLog;

   PROCEDURE ListCalls;

   PROCEDURE SetCallTrace(Traceit: Boolean);
   {$ENDC}

******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
******************************************************************************

```

End of File -- Lines: 181 Characters: 4379

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

*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : USOROC
*
*******************************************************************************

USES {$U+} USOROC;


INTRINSIC;

INTERFACE

   USES
      {$U tklib} UObject,
      {$U sys1lib} QuickDraw,
      {$U tklib} UDraw,
      {$U tklib} UABC,
      {$U tklib} UTKUniversalText,
      {$U tklib} UText,
      {$U tklib} UDialog,
      {$U iospaslib} Blkiointr,
      {$U libqp/UStdTerm} UStdTerm;

   CONST
      maxI          = 3;

   TYPE
      states        = (q0, q1, q2, q3, q4, q6);
      Ilist         = ARRAY [1..maxI] OF CHAR;
      smallstr      = STRING[10];
      TSOROC        = SUBCLASS OF TStdTerm
                         state: states;
                         I: Ilist;
                         newline: BOOLEAN;
                         params, pos, top, bottom, rmarg: INTEGER;
                         vReadBuff: TString;
                         FUNCTION CREATE(object: TObject; heap: Theap): TSOROC;
                      END;

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

End of File -- Lines: 45 Characters: 1224
```

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : UVT100
*
*******************************************************************************

USES {$U+} UVT100;


INTRINSIC;

INTERFACE

   USES
      {$U tklib} UObject,
      {$U sys1lib} QuickDraw,
      {$U tklib} UDraw,
      {$U tklib} UABC,
      {$U tklib} UTKUniversalText,
      {$U tklib} UText,
      {$U tklib} UDialog,
      {$U iospaslib} Blkiointr,
      {$U libqp/UStdTerm} UStdTerm;

   CONST
      maxp          = 5;
      maxI          = 5;
      bmarg         = 23;
      tmarg         = 0;

   TYPE
      states        = (q0, q1, q2, q3, q4, q5, q6, q7);
      Plist         = ARRAY [1..maxp] OF Tstr255;
      Ilist         = ARRAY [1..maxI] OF CHAR;
      padtype       = (ansiAppl, vt52Appl, normal);
      curskeymodes  = (ansiSet, ansiReset, vt52set);
      smallstr      = string[10];
      typeStyle     = (GR, US, UK);
      fontStuff     = RECORD
                         vcursorXY: TCoordinate;
                         charInfo: TRAttributes;
                         currType: typestyle;
                      END;
      TVT100        = SUBCLASS OF TStdTerm
                         state: states;
                         P: Plist;
                         I: Ilist;
                         newline, ansi, vt52, LetHerScroll, ukPound,
                         origabs: BOOLEAN;
                         params, pos, top, bottom, rmarg: INTEGER;
                         charAttr: fontStuff;
                         vReadBuff: TString;
                         S1Font, S0Font, currentFont: typeStyle;
                         TILE12VTS, TILE20VTS: INTEGER;
                         CursorKeyMode: curskeymodes;
                         keyPad: padtype;
                         FUNCTION CREATE(object: TObject; heap: Theap): TVT100;
                      END;

*******************************************************************************
*
```

```
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 65 Characters: 2028

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : VLTREE
*
*******************************************************************************

USES {$U+} VLTREE;


INTRINSIC;

INTERFACE

   USES
      {$IFC OSBUILT }
         {$U libsm/unitstd.obj} unitstd,
         {$U libdb/dbenv.obj } dbenv,
         {$U libin/INTRLIBp.obj } international,
         {$U libdb/dbdecl1.obj } dbdecl1,
         {$U libos/syscall.obj } syscall,
         {$U libos/psyscall.obj } psyscall,
         {$U libdb/lowlevel.obj } lowlevel,
         {$U libdb/pooler.obj } pooler,
         {$U libdb/heap.obj } heap;
      {$ELSEC}
         {$U OBJ:dbenv.obj } dbenv,
         {$U INTRLIB.obj } international,
         {$U OBJ:dbdecl1.obj } dbdecl1,
         {$U OBJ:syscall.obj } syscall,
         {$U OBJ:lowlevel.obj } lowlevel,
         {$U OBJ:pooler.obj } pooler,
         {$U OBJ:heap.obj } heap;
      {$ENDC}

   Const


      noflow        = 0;
      treeempty     = 1;
      rotate        = 2;
      coallesce     = 3;
      pageempty     = 4;
      isplit        = 5;

   FUNCTION entrysize(firstfld, lastfld, doitfld, isize: integer;
                      pedesc: ptrrecptr; prec: ptrdata): integer;

   FUNCTION quickinsert(fileid: integer; pagetype: integer; pnewrec: ptrdata;
                        size: integer; pfptr: ptrfileptr; canmove: boolean;
                        index, buff1: integer; VAR ipage: fileptr;
                        VAR iindex: integer): boolean;

   FUNCTION quickDelete(VAR kresult: integer; onfile: integer; qpage: fileptr;
                        qindex: integer): boolean;

   FUNCTION quickUpdate(VAR kresult: integer; onfile: integer; pnewrec: ptrdata;
                        size: integer; qpage: fileptr;
                        qindex: integer): boolean;

   FUNCTION splitORrotate(VAR kresult: integer; fileid: integer;
                          pagetype: integer; pnewrec: ptrdata; size: integer;
                          pfptr: ptrfileptr; canmove: boolean; index,
                          buff1: integer; leftsib, rightsib: fileptr;
                          VAR buff2: integer; VAR ipage: fileptr;
                          VAR iindex: integer; ps: ptrstackrec;
                          VAR sibpage: fileptr): integer;

   PROCEDURE firstleaf(VAR kresult: integer; fileid: integer; pagetype: integer;
                       pnewrec: ptrdata; size: integer; VAR ipage: fileptr;
                       VAR iindex: integer);

   FUNCTION pagedelete(VAR kresult: integer; fileid: integer; pagetype: integer;
                       buffer, index: integer; closestsib: fileptr;
                       VAR rbuff: integer; VAR dpage: fileptr;
                       VAR dindex: integer): integer;

   PROCEDURE findduplicate(VAR kresult: integer; fileid: integer; pedesc,
                           psdesc: ptrrecptr; nsearch: integer; is_insert,
                           duplimpossible: boolean; pkey: ptrdata; VAR buffer,
                           index: integer; VAR stackinvalid: integer);

   PROCEDURE nextprior(VAR kresult: integer; fileid: integer; VAR buffer,
                       index: integer; which: integer);

   PROCEDURE find(VAR kresult: integer; which: integer; nsearch: integer;
                  fileid: integer; psdesc: ptrrecptr; pkey: ptrdata;
                  VAR buffer: integer);

   PROCEDURE findkeyed(VAR kresult: integer; which: integer; nsearch: integer;
                       fileid: integer; psdesc: ptrrecptr; pkey: ptrdata;
                       VAR buffer, buffindex: integer);

   PROCEDURE insertkeyed(VAR kresult: integer; fileid: integer; pkey: ptrdata;
                         size: integer; VAR ipage: fileptr;
                         VAR iindex: integer);

   PROCEDURE deletekeyed(VAR kresult: integer; fileid: integer; pkey: ptrdata;
                         VAR dpage: fileptr; VAR dindex: integer);

*******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 105 Characters: 4054

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WINDOWS
*
********************************************************************************

USES {$U+} WINDOWS;


INTRINSIC SHARED;

INTERFACE

   USES
      {$U libsm/UnitStd     } UnitStd,
      {$U libsm/UnitHz      } UnitHz,
      {$U libqd/Storage     } Storage,
      {$U libqd/QuickDraw   } QuickDraw,
      {$U libfm/FontMgr     } FontMgr,
      {$U libos/SysCall     } SysCall,
      {$U libwm/Events      } Events;

   CONST
      lastWindow    = 24;

   TYPE
      StringPtr     = ^Str255;
      StringHandle  = ^StringPtr;
      WindowPeek    = ^WindowRecord;
      WindowRecord  = RECORD
                         port: GrafPort;
                         strucRgn: RgnHandle;
                         contRgn: RgnHandle;
                         updateRgn: RgnHandle;
                         windowPic: PicHandle;
                         windowProc: ProcPtr;
                         dataHandle: Handle;
                         refCon: LongInt;
                         process: LongInt;           { window process owner }
                         titleHandle: StringHandle;
                         titleWidth: INTEGER;
                         nextWindow: WindowPeek;
                         closeIcon: INTEGER;
                         iconOverlay: INTEGER;
                         available: BOOLEAN;
                         visible: BOOLEAN;
                         hilited: BOOLEAN;
                         hasTab: BOOLEAN;
                         okMove: BOOLEAN;
                         okGrow: BOOLEAN;
                         okFront: BOOLEAN;
                         okClose: BOOLEAN;
                         okActivate: BOOLEAN;
                         okDispose: BOOLEAN;
                      END;
      WindowMessage = (draw, hit, calcRgns);

   VAR
      buttonWindow: WindowPeek;
      grayRgn:      RgnHandle;
      wmPort:       GrafPtr;
      saveUpdate:   BOOLEAN;
      paintWhite:   BOOLEAN;
      windowArray:  ARRAY [0..lastWindow] OF WindowRecord;
      dPort, wPort: GrafPort;

   PROCEDURE BringToFront(window: WindowPeek; reDraw: BOOLEAN);

   PROCEDURE CalcVis(window: WindowPeek);

   PROCEDURE CalcVisBehind(startWindow: WindowPeek; clobbered: RgnHandle);

   PROCEDURE ChangeProcess(window: WindowPtr; newProcess: ProcessID);

   PROCEDURE CheckWindow(VAR event: EventRecord);

   PROCEDURE ClipAbove(window: WindowPeek);

   PROCEDURE DeleteWindow(window: WindowPeek);

   PROCEDURE DisposeString(sh: StringHandle);

   PROCEDURE DrawNew(window: WindowPeek; fUpdate: BOOLEAN);

   PROCEDURE GetWindTitle(window: WindowPeek; VAR title: Str255);

   PROCEDURE HideWindow(window: WindowPeek);

   PROCEDURE HiLiteWindow(window: WindowPeek; fHiLite: BOOLEAN);

   PROCEDURE HitContent(window: WindowPeek; event: EventRecord);

   PROCEDURE InsertWindow(window, behind: WindowPeek);

   PROCEDURE MakeActive(window: WindowPeek; event: EventRecord);

   PROCEDURE MoveWindow(windPeek: WindowPeek; dh, dv: INTEGER;
                        bringFront: BOOLEAN);

   FUNCTION NewString(str: Str255): StringHandle;

   PROCEDURE PaintBehind(startWindow: WindowPeek; clobbered: RgnHandle);

   PROCEDURE PaintOne(window: WindowPeek; clobbered: RgnHandle);

   FUNCTION PrevWindow(window: WindowPeek): WindowPeek;

   PROCEDURE SaveOld(window: WindowPeek);

   PROCEDURE SendUpdate;

   PROCEDURE SetActWindow(window: WindowPeek);

   PROCEDURE SetString(sh: StringHandle; str: Str255);

   PROCEDURE SetWindTitle(window: WindowPeek; title: Str255);

   PROCEDURE ShowWindow(window: WindowPeek);

   PROCEDURE TopActive;

   FUNCTION UpShift(ch: CHAR): CHAR;

   PROCEDURE WindowSize(window: WindowPeek; width, height: INTEGER;
                        fUpdate: BOOLEAN);

*******************************************************************************
```

```
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 132 Characters: 3916

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WMLCRS
*
*******************************************************************************

USES {$U+} WMLCRS;


INTRINSIC;

INTERFACE

   USES
      {$U libsm/Unitstd   } UnitStd,
      {$U libsm/UnitHz    } UnitHz,
      {$U libqd/Storage   } Storage,
      {$U libqd/QuickDraw } QuickDraw,
      {$U libfm/FontMgr   } FontMgr,
      {$U libfm/FmgrUtil  } FmgrUtil,
      {$U libsb/WmlStd    } WmlStd;

   CONST
      icrsFirst     = - 1;
      icrsHidden    = - 1;
      icrsEscape    = 0;
      icrsInactive  = 1;
      icrsUpDwn     = 2;
      icrsLftRt     = 3;
      icrsGrow      = 4;
      icrsIbeam     = 5;
      icrsMenu      = 6;
      icrsVSkewer   = 7;
      icrsHSkewer   = 8;
      icrsLCcross   = 9;
      icrsXIBeam    = 10;
      icrsHrGlass   = 11;
      icrsCheck     = 12;
      icrsGECross   = 13;
      icrsLFinger   = 14;
      icrsGEIbeam   = 15;
      icrsLast      = 15;

   PROCEDURE InitWmlCrs(VAR cError: TC);

   PROCEDURE SetStdCursor(icrs: TC);

   FUNCTION IcrsCurrent: TC;

*******************************************************************************
*
*                              THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 55 Characters: 1283

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

```
********************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WMLGROW
*
********************************************************************************

USES {$U+} WMLGROW;


INTRINSIC;

INTERFACE

   USES                                {$U libsm/UnitStd } UnitStd,
      {$U libsm/UnitHz        } UnitHz,
      {$U libqd/Storage       } Storage,
      {$U libqd/QuickDraw     } QuickDraw,
      {$U libfm/FontMgr       } FontMgr,
      {$U libsb/WmlStd        } WmlStd,
      {$IFC FWMLOS }
      {$U libos/SysCall  } SysCall,
      {$ELSEC }
      {$U obj:OSStub   } SysCall,
      {$ENDC}
      {$U libwm/Events        } Events,
      {$U libwm/Folders       } Folders;

   CONST
      dhGrow        = 24;
      dvGrow        = 16;

   PROCEDURE GetGrowRect(VAR rGrow: TR);

   PROCEDURE PaintGrow;

   FUNCTION FGrowHit(pt: TPt): TF;

   PROCEDURE DragFrame(ptMouse: TPt; fDrawScrolls: TF; VAR ptNewBR: TPt);

   PROCEDURE InitXorFrame;

   PROCEDURE XorFrame(ptFrameBR: TPt; fDrawScrolls: TF);

   PROCEDURE EndXorFrame;

********************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
********************************************************************************

```

End of File -- Lines: 51 Characters: 1218

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

```
*******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WMLSB
*
*******************************************************************************

USES {$U+} WMLSB;


INTRINSIC;

INTERFACE

   USES
      {$U libsm/UnitStd   } UnitStd,
      {$U libsm/UnitHz    } UnitHz,
      {$U libqd/Storage   } Storage,
      {$U libqd/QuickDraw } QuickDraw,
      {$U libfm/FontMgr   } FontMgr,
      {$U libsb/WmlStd    } WmlStd,
      {$IFC FWMLOS}
      {$U libos/Syscall   } Syscall,
      {$ELSEC}
      {$U obj:OsStub    } Syscall,
      {$ENDC}
      {$U libwm/Events    } Events;

   CONST
      dhSbox        = 24;
      dvSbox        = 16;
      dhThumb       = 24;
      dvThumb       = 16;
      dhSkewer      = 6;
      dvSkewer      = 4;

   TYPE
      TIcon         = (iconSkewer, iconArwA, iconPagA, iconGryA, iconThumb,
                       iconGryB, iconPagB, iconArwB);
      TSicon        = SET OF TIcon;
      THSb          = ^TPSb;
      TPSb          = ^TSb;
      TSb           = RECORD
                         r: TR;
                         tyvh: TTyvh;
                         sicon: TSicon;
                         siconVis: TSicon;
                         cThumb: TC;
                         refCon: TL;
                         fHead: TF;
                         hsbPrv: THSb;
                         hsbNxt: THSb;
                      END;
      TSbList       = RECORD
                         hz: THz;
                         hsbFst: THSb;
                      END;
      TPosts        = ARRAY [iconSkewer..iconArwB] OF TCd;

   VAR
      hsbNil:       THSb;
      dptSbox:      TPt;
   dptThumb:     TPt;
   dptSkewer:    TPt;
   TyvhNOT:      ARRAY [TTyvh] OF TTyvh;
   fTstSb:       TF;

PROCEDURE InitWmlSb;

PROCEDURE InitSbList(VAR sbList: TSbList; hz: THz);

FUNCTION SbCreate(VAR sbList: TSbList; hsbBefore: THSb; tyvhSb: TTyvh;
                  ptTopLeft: TPt; cdBotRt: TCd): THSb;

PROCEDURE KillSb(VAR sbList: TSbList; hsbKill: THSb);

PROCEDURE GetSbRect(hsb: THSb; VAR r: TR);

PROCEDURE SetSbRect(hsb: THSb; r: TR);

FUNCTION RefconSb(hsb: THSb): TL;

PROCEDURE SetSbRefcon(hsb: THSb; refcon: TL);

FUNCTION HsbFirst(VAR sbList: TSbList): THSb;

FUNCTION HsbNext(hsb: THSb): THSb;

FUNCTION HsbPrev(hsb: THSb): THSb;

FUNCTION TyvhOfSb(hsb: THSb): TTyvh;

PROCEDURE SetSbIcons(hsb: THSb; siconNew: TSicon);

PROCEDURE GetSbIcons(hsb: THSb; VAR sicon: TSicon);

PROCEDURE GetVisIcons(hsb: THSb; VAR siconVis: TSicon);

FUNCTION CThumbPos(hsb: THSb): TC;

PROCEDURE SetThumb(hsb: THSb; cNewThumb: TC);

PROCEDURE MoveSb(hsb: THSb; ptNewTL: TPt);

PROCEDURE AdjSbCorner(hsb: THSb; dcdAdjust: TCd; fTopLeft: TF);

PROCEDURE PaintSb(hsb: THSb);

PROCEDURE EraseSb(hsb: THSb);

PROCEDURE PaintArw(hsb: THSb; iconWhichArw: TIcon; fBlack: TF);

FUNCTION HsbFromPt(VAR sbList: TSbList; pt: TPt): THSb;

FUNCTION FSbHit(VAR sbList: TSbList; pt: TPt; VAR hsbHit: THSb;
                VAR iconHit: TIcon): TF;

PROCEDURE FlushRects(VAR rH, rV: TR);

PROCEDURE MkPosts(hsb: THSb; VAR posts: TPosts);

PROCEDURE SplitSb(VAR sbList: TSbList; hsbOld: THSb; VAR hsbNew: THSb;
                  cdCut: TCd);

PROCEDURE AdjSplitBetween(hsbBefore, hsbAfter: THSb; dcdAdjust: TCd);

FUNCTION HsbNextOnSbar(hsb: THSb): THSb;

```

```
   FUNCTION HsbPrevOnSbar(hsb: THSb): THSb;

   FUNCTION HsbSbarHead(hsb: THSb): THSb;

   FUNCTION HsbSbarTail(hsb: THSb): THSb;

   FUNCTION FSbarHead(hsb: THSb): TF;

   PROCEDURE SlideSbar(hsb: THSb; dcdAdjust: TCd);

   PROCEDURE PaintSbar(hsb: THSb);

   PROCEDURE EraseSbar(hsb: THSb);

   PROCEDURE SetupMvThumb(hsb: THSb);

   PROCEDURE MoveThumb(cNewThumb: TC);

   PROCEDURE DragThumb(hsb: THSb; ptMouse: TPt; VAR cThumbUp: TC);

   PROCEDURE FixRLimits(hsb: THSb; VAR rLimits: TR);

   PROCEDURE DragSkewer(hsbDwn: THSb; ptMouse: TPt; rLimits: TR;
                        VAR ptSkewerUp: TPt);

   PROCEDURE HideSBorder(r: TR);
   {$IFC WmlDebug }

   PROCEDURE PxSbList(VAR sbList: TSbList);

   PROCEDURE PxSicon(sicon: TSicon);

   PROCEDURE PxSb(hsb: THSb);

   PROCEDURE PxSbar(hsb: THSb);
   {$ENDC }

*******************************************************************************
*
*                            THAT'S ALL FOLKS ...
*
*******************************************************************************

```

End of File -- Lines: 170 Characters: 4238

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

******************************************************************************
*
* APPLE LISA DESKTOP LIBRARY INTERFACES (Version 3.0) : WMLSTD
*
******************************************************************************

USES {$U+} WMLSTD;


INTRINSIC;
   {$SETC FWMLOS := TRUE }

INTERFACE

   USES                                {$U libsm/UnitStd } UnitStd,
      {$U libsm/UnitHz    } UnitHz,
      {$U libqd/Storage   } Storage,
      {$U libqd/QuickDraw } QuickDraw,
      {$U libfm/FontMgr   } FontMgr;
      {$SETC WmlDebug   := FDbgOK }
      {$SETC WmlSymbols := FSymOk }
      {$SETC doTraceSB := TRUE}
      {$SetC fTraceSB := doTraceSB AND fTRACE}

   CONST
      tyvhV         = V;
      tyvhH         = H;

   TYPE
      TAport        = GrafPort;
      TPort         = GrafPtr;
      TBmp          = BitMap;
      TPt           = Point;
      TTyvh         = VHSelect;
      TR            = Rect;

   PROCEDURE PenSave;

   PROCEDURE PenRestore;

   PROCEDURE WmlTxtSave;

   PROCEDURE WmlTxtRestore;

******************************************************************************
*
*                             THAT'S ALL FOLKS ...
*
******************************************************************************


End of File -- Lines: 50 Characters: 1110

SUMMARY:

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