

-- ListInterface.mesa; modified by Sandman, April 15, 1978 12:28 PM

DIRECTORY

```

AltoDefs: FROM "altodefs",
BcdDefs: FROM "bcddefs",
CommanderDefs: FROM "commanderdefs",
IODefs: FROM "iodefs",
ListerDefs: FROM "listerdefs",
OutputDefs: FROM "outputdefs",
SegmentDefs: FROM "segmentdefs",
StringDefs: FROM "stringdefs",
SymbolTableDefs: FROM "symboltabledefs",
SymDefs: FROM "symdefs";

```

DEFINITIONS FROM ListerDefs, OutputDefs, SymDefs;

ListInterface: PROGRAM IMPORTS CommanderDefs, IODefs, ListerDefs, OutputDefs, SegmentDefs, StringDefs,

\*\*SymbolTableDefs

EXPORTS ListerDefs =

BEGIN

FileSegmentHandle: TYPE = SegmentDefs.FileSegmentHandle;

FileHandle: TYPE = SegmentDefs.FileHandle;

symbols: SymbolTableDefs.SymbolTableBase;

PrintInterface: PROCEDURE =

BEGIN OPEN symbols;

sei: ISEIndex;

number: NumberFormat = NumberFormat[10, FALSE, FALSE, 4];

FOR sei ← FirstCtxSe[stHandle.outerCtx], NextSe[sei]

UNTIL sei = ISENull DO

IF (seb+UnderType[(seb+sei).idtype]).typetag = transfer THEN

SELECT XferMode[(seb+sei).idtype] FROM

procedure, signal, error, program =>

BEGIN

PutNumber[LOOPHOLE[(seb+sei).idvalue], number];

PutString[" -- "];

PrintSei[sei];

PutString[": "];

PutModeName[XferMode[(seb+sei).idtype]];

PutChar[IODefs.CR];

END;

ENDCASE;

ENDLOOP;

RETURN

END;

ModePrintName: ARRAY TransferMode OF STRING = [

"PROCEDURE", "PORT", "SIGNAL", "ERROR", "PROCESS", "PROGRAM", "NONE"];

PutModeName: PROCEDURE[n: TransferMode] =

BEGIN

PutString[ModePrintName[n]]; RETURN

END;

Interface: PROCEDURE[root: STRING] =

BEGIN OPEN StringDefs, SegmentDefs;

i: CARDINAL;

defs: BOOLEAN ← FALSE;

bcdFile: STRING ← [40];

sseg: FileSegmentHandle;

AppendString[bcdFile,root];

FOR i IN [0..bcdFile.length) DO

IF bcdFile[i] = '.' THEN EXIT;

REPEAT FINISHED => AppendString[bcdFile, ".bcd"];

ENDLOOP;

BEGIN

[symbols: sseg] ← Load[bcdFile |

NoFGT => RESUME;

NoCode => BEGIN defs ← TRUE; RESUME END;

NoSymbols, IncorrectVersion, MultipleModules => GOTO badformat;

SegmentDefs.FileNameError => GOTO badname];

IF ~defs THEN GOTO badformat;

symbols ← SymbolTableDefs.AcquireSymbolTable[

SymbolTableDefs.TableForSegment[sseg]];

```
ListerDefs.SetRoutineSymbols[symbols];
OpenOutput[root, ".il"];
WriteFileID[];
PrintInterface[];
SymbolTableDefs.ReleaseSymbolTable[symbols];
SegmentDefs.DeleteFileSegment[sseg];
CloseOutput[];
EXITS
  badformat => IODefs.WriteString["Bad Format!"];
  badname => IODefs.WriteString["File Not Found!"];
END;
END;
```

```
command: CommanderDefs.CommandBlockHandle;
```

```
command ← CommanderDefs.AddCommand["Interface", LOOPHOLE[Interface], 1];
command.params[0] ← [type: string, prompt: "Filename"];
```

```
END...
```