

```
-- file DIActionDefs.Mesa
-- last modified by Barbara, July 20, 1978 4:42 PM

DIRECTORY
  DebuggerDefs: FROM "debuggerdefs" USING [SOPointer],
  DIDefs: FROM "didefs" USING [
    ESPointer, hereESPointer, Operator, predefinedType, thereESPointer,
    TIPointer],
  DILitDefs: FROM "dilitletdefs" USING [LTIndex, STIndex],
  SymbolTableDefs: FROM "symboltabledefs" USING [SymbolTableBase],
  SymDefs: FROM "symdefs";

DIActionDefs: DEFINITIONS SHARES DIDefs =
  BEGIN OPEN DIDefs;

--using grammar version 7

--eval stack manipulation
EvalStackOverflow: SIGNAL;
EvalStackEmpty: SIGNAL;
NILesp: SIGNAL;
NotOnEvalStack: SIGNAL;
pushevalstack: PROCEDURE [esp: ESPointer];
popevalstack: PROCEDURE RETURNS [esp: ESPointer];
popNevalstack: PROCEDURE [n: CARDINAL] RETURNS [esp: ESPointer];

--type stack manipulation
TypeStackOverflow: SIGNAL;
TypeStackEmpty: SIGNAL;
pushtypestack: PROCEDURE [tip: TIPointer];
poptypestack: PROCEDURE RETURNS [tip: TIPointer];

assignvalue: PROCEDURE [rhs: ESPointer, lhs: thereESPointer];
TypesDontMatch: SIGNAL [esp1, esp2: ESPointer];
performAddOp: PROCEDURE [es2, es1: ESPointer, op: Operator]
  RETURNS [result: hereESPointer];
performMultOp: PROCEDURE [es2, es1: ESPointer, op: Operator]
  RETURNS [result: hereESPointer];

--perform an action on an eval stack item
qualifyItem: PROCEDURE [esp: ESPointer, id: DILitDefs.STIndex,
  locals: BOOLEAN] RETURNS [ESPointer];
dereferenceItem: PROCEDURE [esp: ESPointer] RETURNS [thereESPointer];
loopholeItem: PROCEDURE [esp: ESPointer, tip: TIPointer]
  RETURNS [ESPointer];
loopholeUnspecItem: PROCEDURE [esp: ESPointer] RETURNS [ESPointer];
minusItem: PROCEDURE [esp: ESPointer] RETURNS [new: hereESPointer];
TooManyIndirections: SIGNAL;
addressofItem: PROCEDURE [tesp: thereESPointer] RETURNS [new: hereESPointer];
IncorrectType: SIGNAL [esp: ESPointer];

--built in calls
lengthItem: PROCEDURE [esp: ESPointer] RETURNS [new: hereESPointer];
baseItem: PROCEDURE [esp: ESPointer] RETURNS [new: hereESPointer];
desc1Item: PROCEDURE [name: thereESPointer] RETURNS [new: hereESPointer];
desc2Item: PROCEDURE [length, base: ESPointer] RETURNS [new: hereESPointer];
memItem: PROCEDURE [esp: ESPointer] RETURNS [new: hereESPointer];

--type operations
typeOp: PROCEDURE [typeop: Operator, tip: TIPointer]
  RETURNS [new: hereESPointer];
setPredefined: PROCEDURE [type: predefinedType] RETURNS [tip: TIPointer];
SearchFileForType: PROCEDURE [file, id: DILitDefs.STIndex]
  RETURNS [tip: TIPointer];
SearchForType: PROCEDURE [id: DILitDefs.STIndex] RETURNS [tip: TIPointer];
SearchForVariantType: PROCEDURE [var: DILitDefs.STIndex, tip: TIPointer]
  RETURNS [TIPointer];
pointertoType: PROCEDURE [tip: TIPointer] RETURNS [TIPointer];
InvalidType: SIGNAL [tip: TIPointer];

--interval notation
InvalidInterval: SIGNAL [b1, b2: UNSPECIFIED];
InvalidLongInterval: SIGNAL [left, right: LONG INTEGER];
setIntervalBit: PROCEDURE [esp: ESPointer] RETURNS [ESPointer];
printOctal: PROCEDURE [n, start: ESPointer];
printInterval: PROCEDURE [n, start, exp: ESPointer];
```

```
--expression lists
InvalidExpression: SIGNAL;
evaluateExpList: PROCEDURE RETURNS [ESPointer];
startList: PROCEDURE [size: CARDINAL] RETURNS [new: hereESPointer];
incrementList: PROCEDURE;

--handle literals
litType: TYPE = {num, lnum, char, sr};
getLiteral: PROCEDURE [type: litType, value: DILitDefs.LTIndex]
  RETURNS [new: hereESPointer];
getLongLiteral: PROCEDURE [value: DILitDefs.LTIndex]
  RETURNS [new: hereESPointer];
getStringLiteral: PROCEDURE [value: DILitDefs.STIndex]
  RETURNS [new: hereESPointer];

--symboltable manipulation
LookupId: PROCEDURE [id: DILitDefs.STIndex] RETURNS [ESPointer];
SearchFileForId: PROCEDURE [file, id: DILitDefs.STIndex] RETURNS [ESPointer];
SearchFrameForId: PROCEDURE [num: DILitDefs.LTIndex, id: DILitDefs.STIndex]
  RETURNS [ESPointer];

--utilities
espToSop: PROCEDURE [esp: ESpOinter, sop: DebuggerDefs.SOPointer];
sopToEsp: PROCEDURE [sop: DebuggerDefs.SOPointer, tesp: thereESPointer];
AllocateHereStackItem: PROCEDURE RETURNS [hesp: hereESPointer];
AllocateThereStackItem: PROCEDURE RETURNS [tesp: thereESPointer];
FreeStackItem: PROCEDURE [esp: ESpOinter];
Transfer: PROCEDURE [esp: ESpOinter] RETURNS [newesp: hereESPointer];
ActualValue: PROCEDURE [hesp: hereESPointer]
  RETURNS [value: UNSPECIFIED];
LongValue: PROCEDURE [hesp: hereESPointer]
  RETURNS [LONG INTEGER];
NotImplemented: SIGNAL;

--initialization and reset
GetSetUp: PROCEDURE;
GetCurrentST: PROCEDURE RETURNS [SymbolTableDefs.SymbolTableBase];
CleanUp: PROCEDURE;
ResetStacks: PROCEDURE;
ResetTypeStack: PROCEDURE;

END...
```