

-- BcdTreeDefs.Mesa Edited by Johnsson on April 12, 1978 4:33 PM

DIRECTORY

BcdTabDefs: FROM "bcdtabdefs",  
TableDefs: FROM "tabledefs";

BcdTreeDefs: DEFINITIONS = BEGIN

TreeLink: TYPE = RECORD [  
  SELECT tag: \* FROM  
  hash => [index: BcdTabDefs.HTIndex],  
  symbol => [index: BcdTabDefs.STIndex],  
  string => [index: TableDefs.TableIndex],  
  subtree => [index: TreeIndex],  
  ENDCASE];

TreeNode: TYPE = MACHINE DEPENDENT RECORD [  
  free: BOOLEAN, -- reserved for allocator  
  name: NodeName,  
  nsons: [0..MaxNSons],  
  codelinks: BOOLEAN,  
  fill: [0..377B],  
  sourceindex: CARDINAL,  
  son1: TreeLink,  
  son2: TreeLink,  
  son3: TreeLink,  
  son4: TreeLink,  
  son5: TreeLink]; -- and as many more as needed

MaxNSons: CARDINAL = 7; -- without null terminator  
TreeNodeSize: CARDINAL = 2; -- minimum tree node size

endTreeIndex: TreeIndex = LAST[TreeIndex]-1;  
nullTreeIndex: TreeIndex = LAST[TreeIndex];  
TreeIndex: TYPE = POINTER [0..TableDefs.TableLimit) TO TreeNode;

empty: TreeLink = TreeLink[subtree[index: nullTreeIndex]];  
endmark: TreeLink = TreeLink[subtree[index: endTreeIndex]];  
nullid: TreeLink = TreeLink[hash[index: BcdTabDefs.HTNull]];

TreeSonField: TYPE = RECORD [soni: TreeLink];  
TreeXIndex: TYPE = POINTER [0..TableDefs.TableLimit) TO TreeSonField;

NodeName: TYPE = {list, item, source, config, assign, plus, then, module};

-- tree construction interface

m1push: PROCEDURE [v: TreeLink];  
m1pop: PROCEDURE RETURNS [TreeLink];  
maketree: PROCEDURE [name: NodeName, count: INTEGER] RETURNS [TreeLink];  
makelist: PROCEDURE [size: INTEGER] RETURNS [TreeLink];

pushtree: PROCEDURE [name: NodeName, count: INTEGER];  
pushlist: PROCEDURE [size: INTEGER];  
pushhash: PROCEDURE [hti: BcdTabDefs.HTIndex];

Attribute: TYPE = {links};

setsourceindex: PROCEDURE [source: CARDINAL];  
setattribute: PROCEDURE [attr: Attribute, value: BOOLEAN];

-- tree deallocation

freenode: PROCEDURE [node: TreeIndex];  
freetree: PROCEDURE [t: TreeLink] RETURNS [TreeLink];

-- tree testing

testtree: PROCEDURE [t: TreeLink, name: NodeName] RETURNS [BOOLEAN];

listlength: PROCEDURE [t: TreeLink] RETURNS [CARDINAL];  
listhead: PROCEDURE [t: TreeLink] RETURNS [TreeLink];  
listtail: PROCEDURE [t: TreeLink] RETURNS [TreeLink];

-- tree manipulation

```
TreeScan: TYPE = PROCEDURE [t: TreeLink];
TreeMap: TYPE = PROCEDURE [t: TreeLink] RETURNS [v: TreeLink];

scanlist: PROCEDURE [root: TreeLink, action: TreeScan];
reversescanlist: PROCEDURE [root: TreeLink, action: TreeScan];
updatelist: PROCEDURE [root: TreeLink, action: TreeMap] RETURNS [TreeLink];

-- administrative procedures

treeinit, treeerase: PROCEDURE;

END.
```