

```

/*+1=====*/
/*  MODULE                                BOOL.H                                */
/*=====*/
/*  FUNCTION      Headerfile for module BOOL.C (and user modules).
 *
 *  SYSTEM        Standard (ANSI/ISO) C.
 *                Tested on PC/MS DOS 5.0 (MSC 600A).
 *
 *  SEE ALSO     Modules : ERROR.C, BOOL.C
 *
 *  PROGRAMMER   Allan Dystrup.
 *
 *  COPYRIGHT    (c) Allan Dystrup, 1991
 *
 *  VERSION      $Header: d:/cwk/kf/bool/RCS/bool.h 1.1 92/10/25 16:51:29
 *                Allan_Dystrup Exp Locker: Allan_Dystrup $
 *
 *                -----
 *                $Log:bool.h $
 *                Revision 1.1 92/10/25 16:51:29 Allan_Dystrup
 *                Initial revision
 *
 *=====*/

/*****
/*****  HEADER FILES *****/
/*****
#ifndef _BOOL_H
/* Make sure this header is included only once */
#define _BOOL_H
/* Matching #endif is at end-of-file */

/* Include header with general definitions */
#ifndef _GENERAL_H
/* Bool.h depends on defs. in General.h */
#include "general.h"
#endif

/*-----*/
/* Defines to handle declaration, allocation and initialization of global */
/* data in a common header file. */
/* The _BOOL_INIT(x) macro won't work with aggregates because it interprets*/
/* a comma as indicating a new macro parameter (not part of the current); - */
/* Aggregates are initialized inside a #ifdef _BOOL_ALLOC ... #endif */
/*-----*/
#ifdef _BOOL_ALLOC
/* BOOL.C: Allocate space for globals -- */
#define _BOOL_CLASS
/* ignore (no extern class) */
#define _BOOL_INIT(x) x
/* initialize */
#else
/* EXTERN: No allocation in user modules */
#define _BOOL_CLASS extern
/* declare as extern */
#define _BOOL_INIT(x) /*ignore*/
/* don't initialize */
#endif
/* END _BOOL_ALLOC ----- */

/*****
/*****  DATA STRUCTURES *****/
/*****
#define SYMMAX 125+2
/* Size of symbol table, max. 125 entries */
#define STRMAX (SYMMAX*10)
/* Size of lexeme array, - arbitrary long */
#define OUTMAX (SYMMAX*2)
/* Size of outputstring, - arbitrary long */

/* Structure for symbol table */
_BOOL_CLASS struct entry {
/* Structure of entry in symbol table : */

```

```

    BYTE      *pzLexptr;          /* Ptr. to symbol string (ie. lexeme) */
    FLAG      fValue;            /* Attribute value : boolean [T | F] */
} symtable[SYMMAX];            /* Symbol table is Array of sych entries */

/*****
/***** FUNCTION PROTOTYPES *****/
/*****/

_BOOL_CLASS BYTE*
    pzParse      P((BYTE * pzStr));

_BOOL_CLASS FLAG
    fInterpret  P((BYTE * pzStr));

_BOOL_CLASS int
    iSymLookup  P((BYTE * pbStr, int len));

_BOOL_CLASS void
    vSymReset   P((void));

#endif /* #ifndef _BOOL_H */
/* END module bool.h */
/*-1=====*/

```