

SourceInsight 应用技巧

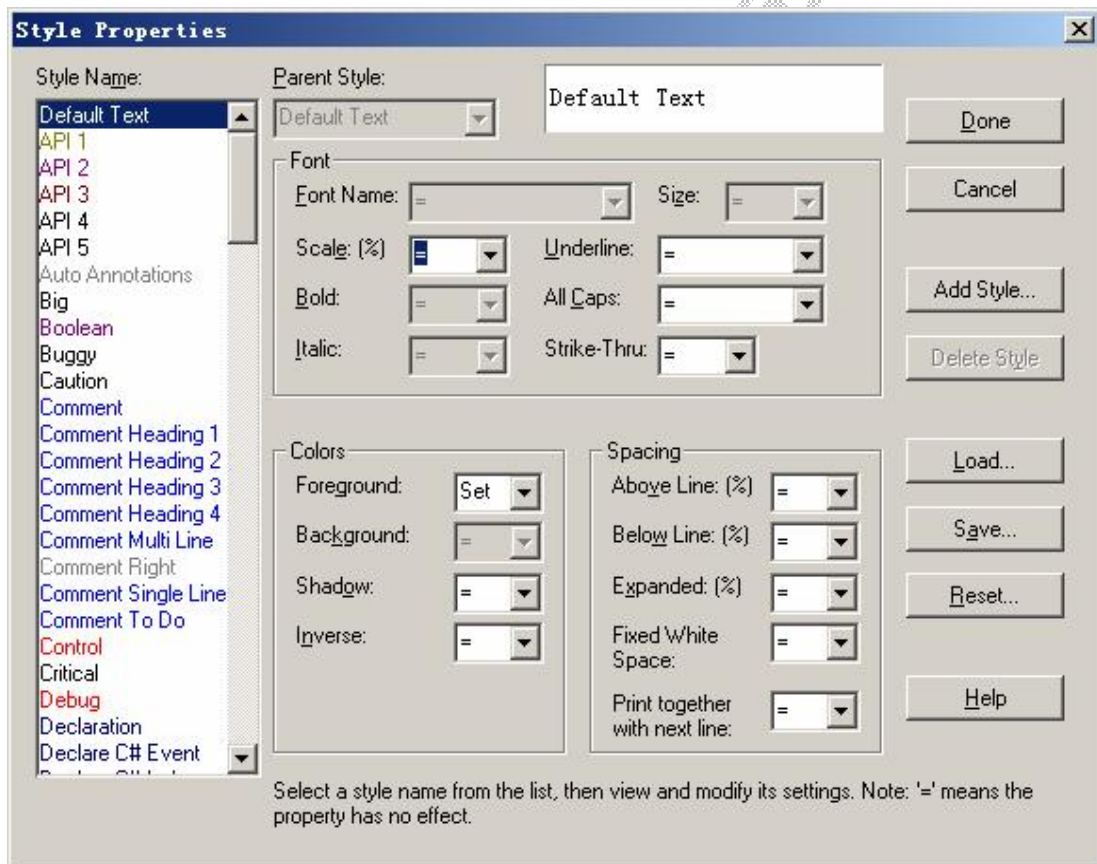
-----崔庆伟

Source Insight 实质上是一个支持多种开发语言（java,c ,c++等等）的编辑器，只不过由于其查找、定位、彩色显示等功能的强大，而被我们当成源代码阅读工具使用。

对于一些如何安装，新建、打开工程这些地球人都知道的本文就不提了，主要是跟大家讨论一些常用的使用技巧。

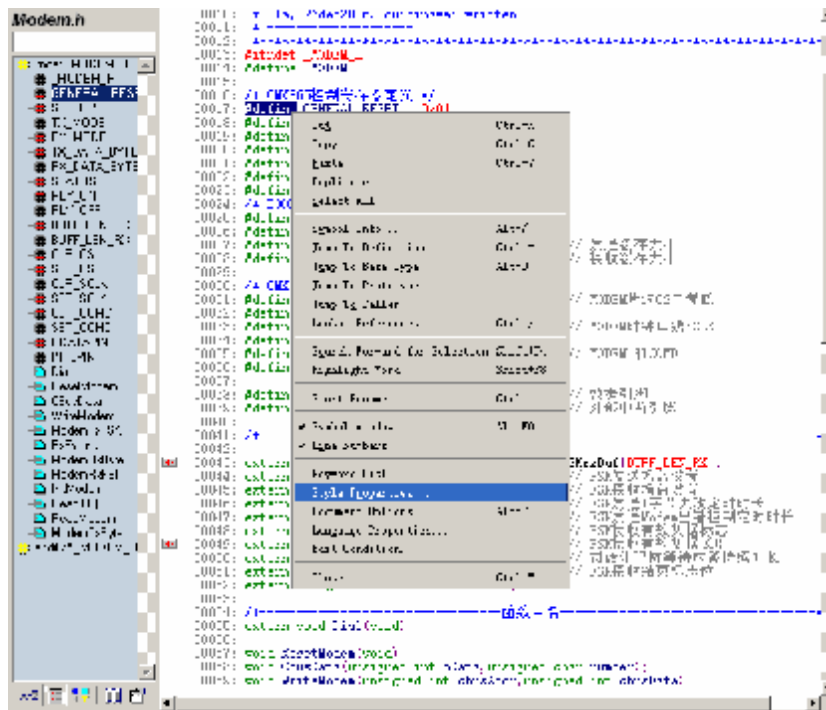
- .语法着色设置:

方法 1：Option->Style Properties 进入如下界面：



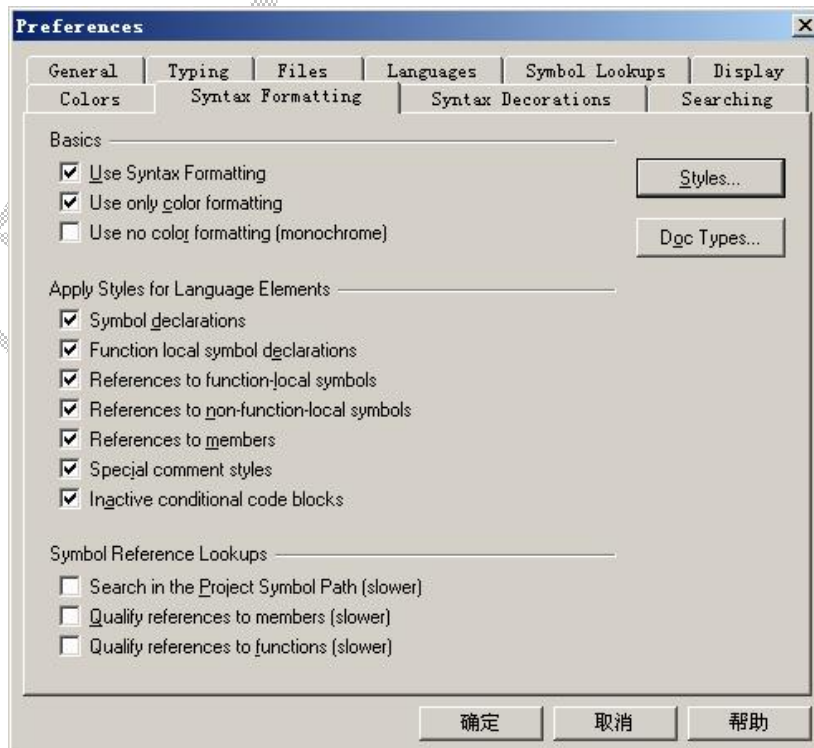
方法 2：选择要更改的语句点右击选择 Style Properties，如下图是对

#define 进行语法设置的截图。此方法是更改起来有针对性，省时省力！

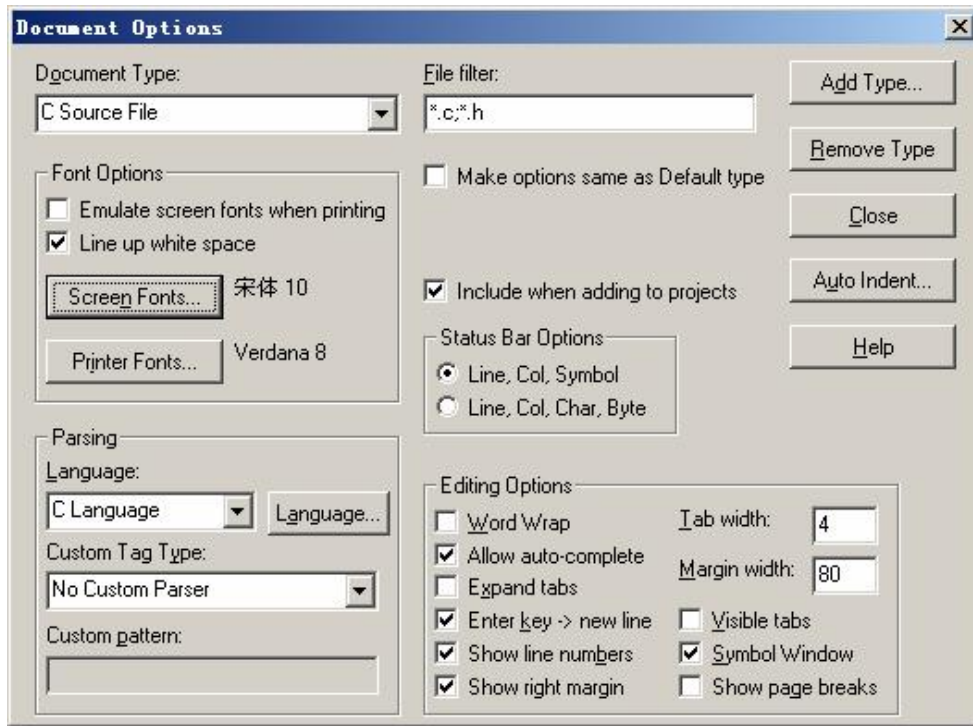


提示: 多数人喜欢 SI 除了她适于阅读程序的强大功能外, 就是她华丽的外表了。但语法格式只选择使用语法着色时, Symbol window 有时会死掉。我的方法是进行如下设置:

a) Option->Style Properties->Syntax Formatting 勾选 Use only color Formatting, 去掉字体阴影, 界面如下:

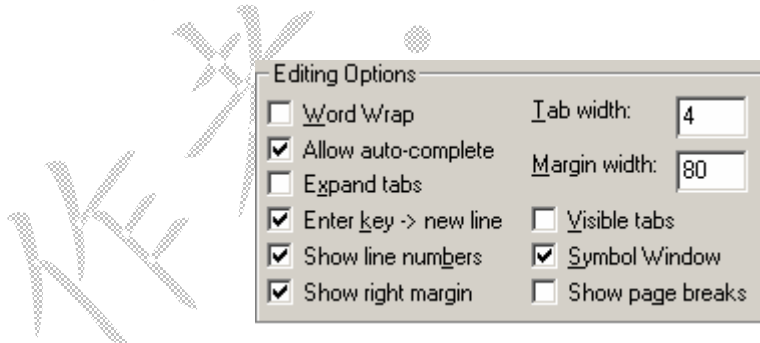


b) Option->Document Options 中 ScreenFonts 字体全设为宋体 10, 界面如下:



二. 缩进:

1. Option->Document Options->Editing Options 中 Tab Width 设置 TAB 键宽度, 界面如下:



2. Option->Document Options->Auto Indent... 中设置自动缩进类型, 界面如下:



三、常用快捷键:

1. 块缩进 **F9、F10**

提示: 也可进行单行的前缩进, 后缩进。只在光标处在目标行就可以执行, 不一定全选目标行。


2. 块选择 **ALT + 鼠标左击** 选择界面如下:

```
00001: /*****
00002:  * Define.h - Define.h
00003:  *
00004:  * Copyright 1994-2006 Routon Technology Co.,Ltd.
00005:  *
00006:  * DESCRIPTION: -
00007:  *   C语言常用宏定义
00008:  *   写好C语言, 漂亮的宏定义很重要, 使用宏定义可以防止出错, 提
00009:  *   , 方便性 等等。下面是一些成熟软件中常用的宏定义。。。。
00010:  * modification history
00011:  * -----
00012:  * 01a, 23nov2005, cuiqingwei written
00013:  * -----
00014:  *****/
00015:
00016: // 防止一个头文件被重复包含
00017: #ifndef _DEFINE_H
00018: #define _DEFINE_H
00019:
00020: // 重新定义一些类型, 防止由于各种平台和编译器的不同, 而产生的:
00021: // 方便移植。
00022: typedef unsigned char uint8; // 无符号8位整型变量
00023: typedef signed char int8; // 有符号8位整型变量
00024: typedef unsigned int uint16; // 无符号16位整型变量
00025: typedef signed int int16; // 有符号16位整型变量
00026: typedef unsigned long uint32; // 无符号32位整型变量
00027: typedef signed long int32; // 有符号32位整型变量
00028: typedef float fp32; // 单精度浮点数 (32位长度)
00029: typedef double fp64; // 双精度浮点数 (64位长度)
00030:
00031: // 位变量定义 //
00032: typedef union{
00033:     struct{
00034:         unsigned b0:1;
00035:         unsigned b1:1;
00036:         unsigned h2:1;
```



3. 查找

a) 当前文件查找 **Ctrl+F** 或点击 

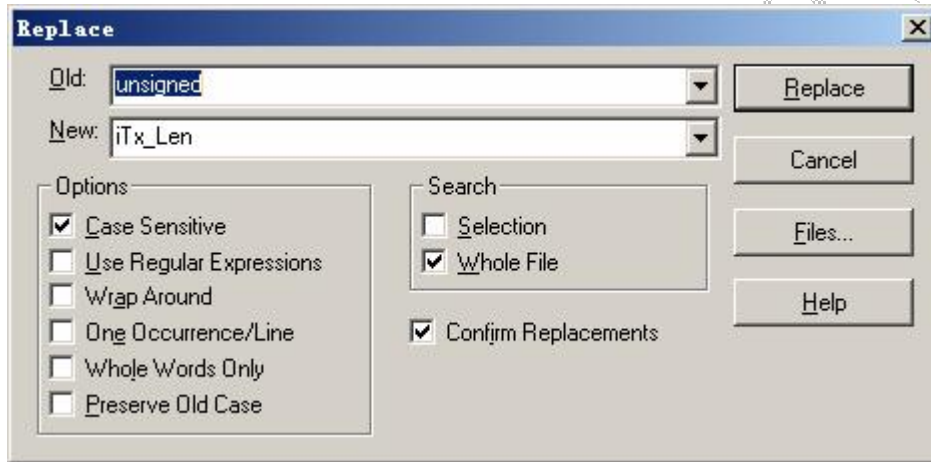
F3 或点击  上一个 **F4** 或点击  下一个

b) 工程内查找 **Ctrl+/** 或点击  进入如下界面, 选择“Apeend”追加。



 上一个 **Shift + F9** 或点击  下一个

4. 替换 **Ctrl + H** 进入如下界面，点 **Replace** 只替换当前文件，点 **Files...** 在整个工程内替换。



5. 其它:

Alt+, 前进

Alt+, 后退

F2 切换已打开文件

F5 跳到指定行

F6 平铺两窗口

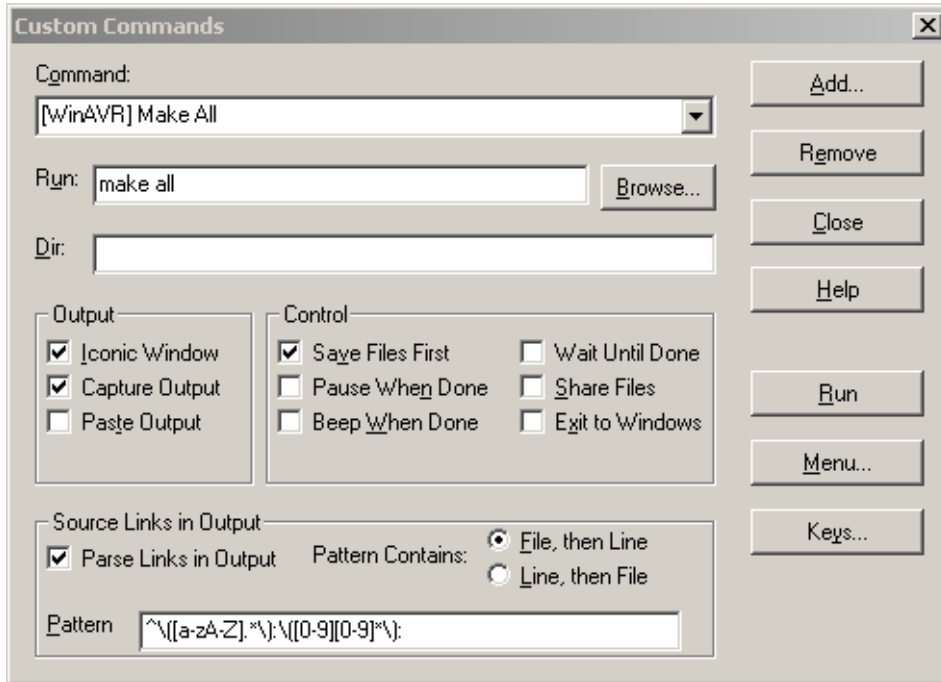
F7 浏览项目符号

F8 浏览本地文件符号

四. 自定义命令:

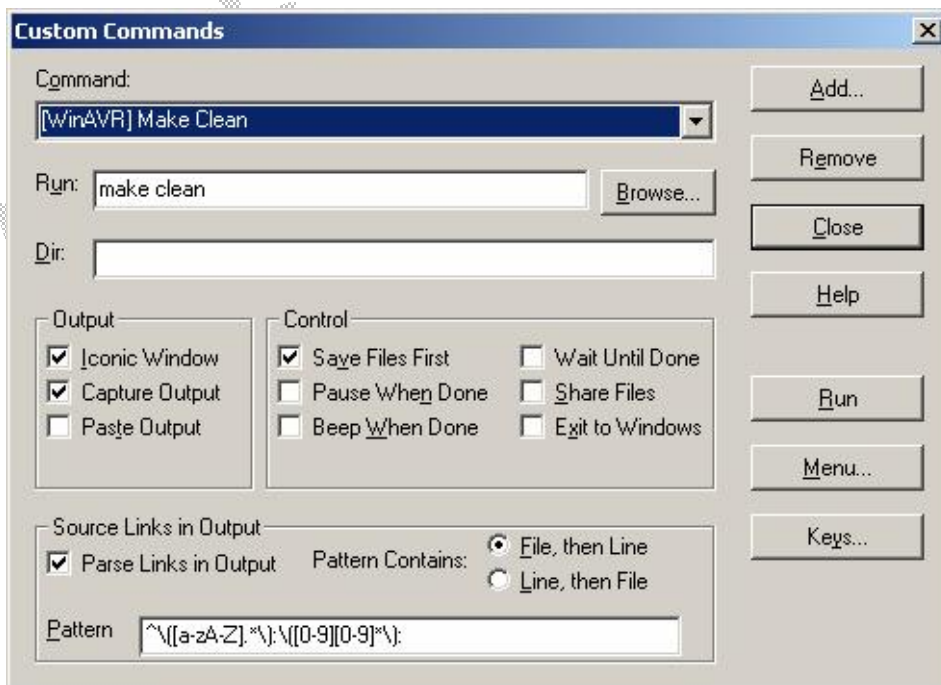
以前整理过 SourceInsight 配置 WinAVR, 此处引用以说明如何配置自定义命令^_^

1. Make All 配置界面



将 Pattern 内内容改为 `^\\([a-zA-Z].*\\):\\([0-9][0-9]*\\):`

2. Make Clean 配置界面



将 Pattern 内内容改为 `^\\([a-zA-Z].*\\):\\([0-9][0-9]*\\):`

3. Make File 配置

用 Uedit32 打开..\WinAVR\mfile\mfile.tcl 按下图所示进行修改!

```
0 10 20 30 40 50 60 70 80 90
1 #!/usr/local/bin/tixwish
2 #
3 # "THE BEER-WARE LICENSE" (Revision 42) (by Poul-Henning Kamp):
4 # Joerg Wunsch <j.gnu@uriah.heep.sax.de> wrote this file. As long as you
5 # retain this notice you can do whatever you want with this stuff. If we meet
6 # some day, and you think this stuff is worth it, you can buy me a beer
7 # in return.
8 #
9 #
10 # Simple Makefile generator for AVR-GCC
11 #
12 # $Id: mfile.tcl,v 1.46 2006/01/19 13:20:51 j Exp $
13
14 # Full path name where various files will be found
15
16 if {[info exists env(MFILE_HOME)]} {
17     set prefix $env(MFILE_HOME)
18 } else {
19     if {$tcl_platform(platform) == "unix"} {
20         set prefix "/usr/local/share/mfile"
21     } else {
22         set prefix "D:/WinAVR/mfile/"
23     }
24 }
25
26 set templatename "$prefix/makefile_template"
27 set helpfile "$prefix/help.html"
28 set mainicon "$prefix/mfile.xbm"
29 set helpicon "$prefix/htmlview.xbm"
30
31 global helpfile mainicon helpicon
32
33 # color to highlight changed items with
34 global highlight
35 set highlight "\#ffff80"
36
37 #
38 # end of configurable section
39 #
```

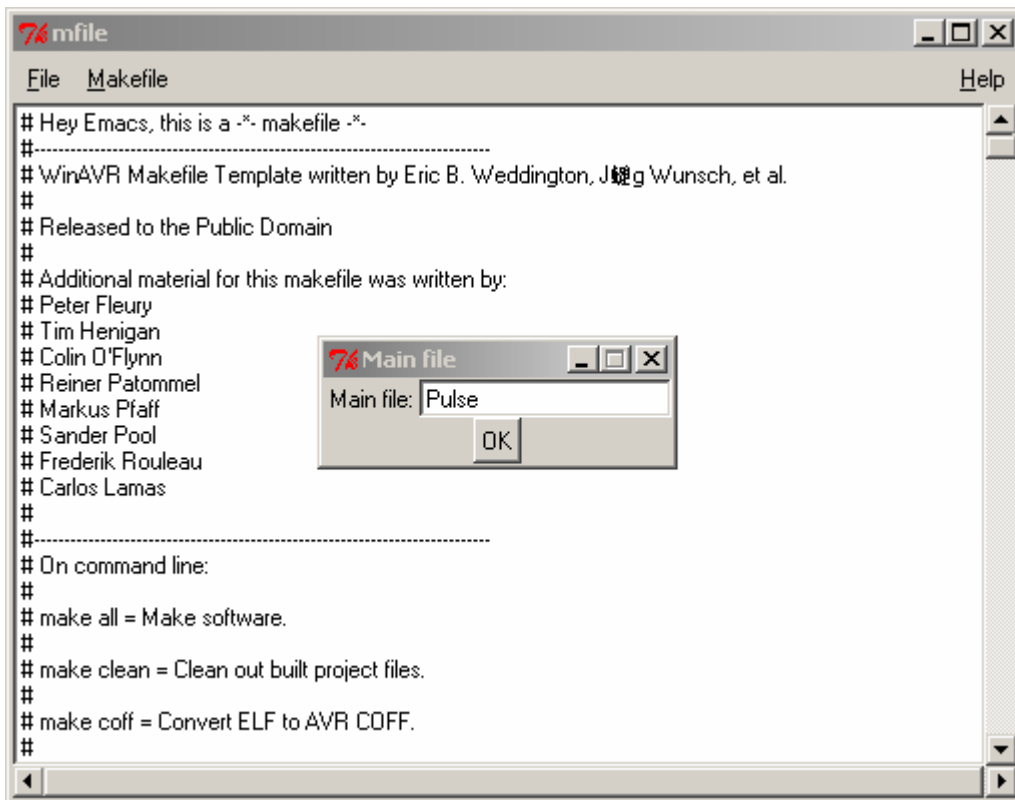
将选择处（蓝底）改为 mfile.tcl 所在路径，路径符一定要用“/”

配置界面：

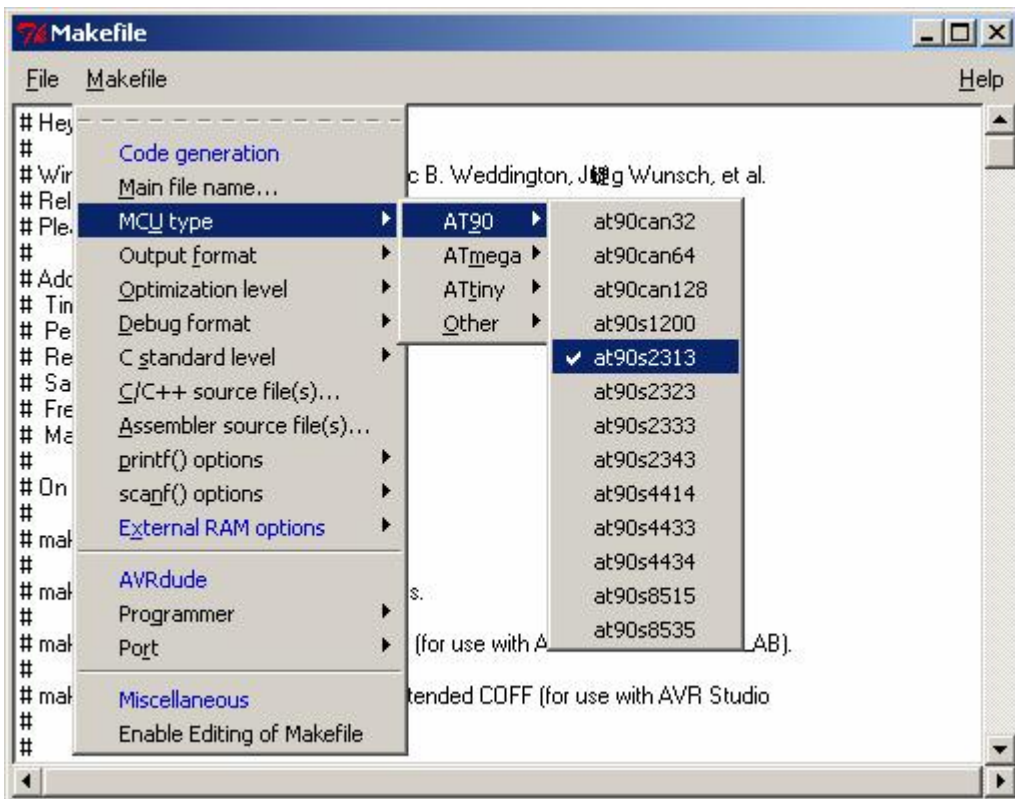


4. Make File 设置

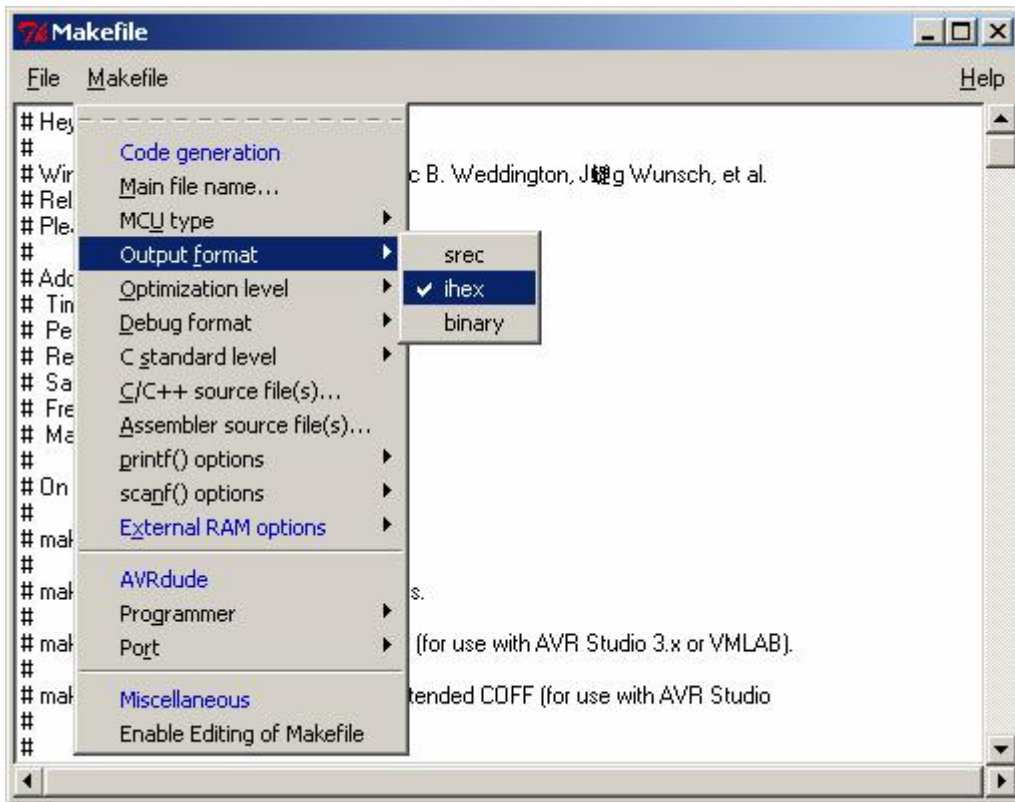
①点击菜单栏的 Makefile 的按钮，再点击“Main file name...”，在出现的对话框里填写主函数名，单击 OK。



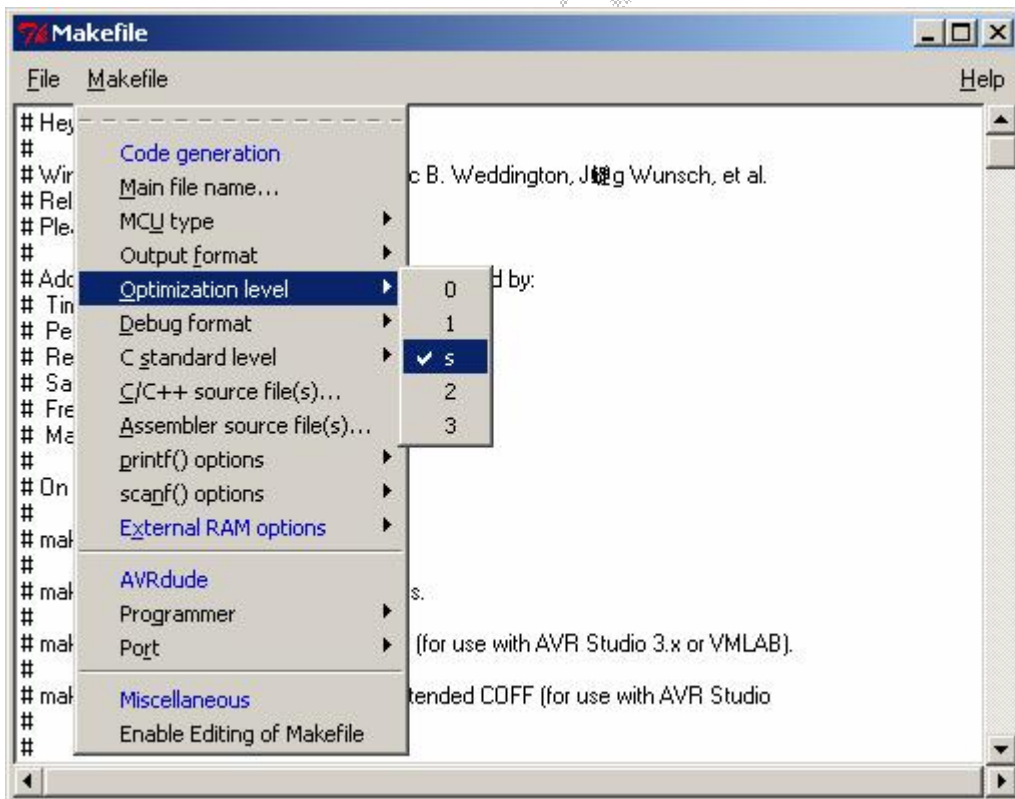
②点击 Makefile→MCU type 选择芯片类型。



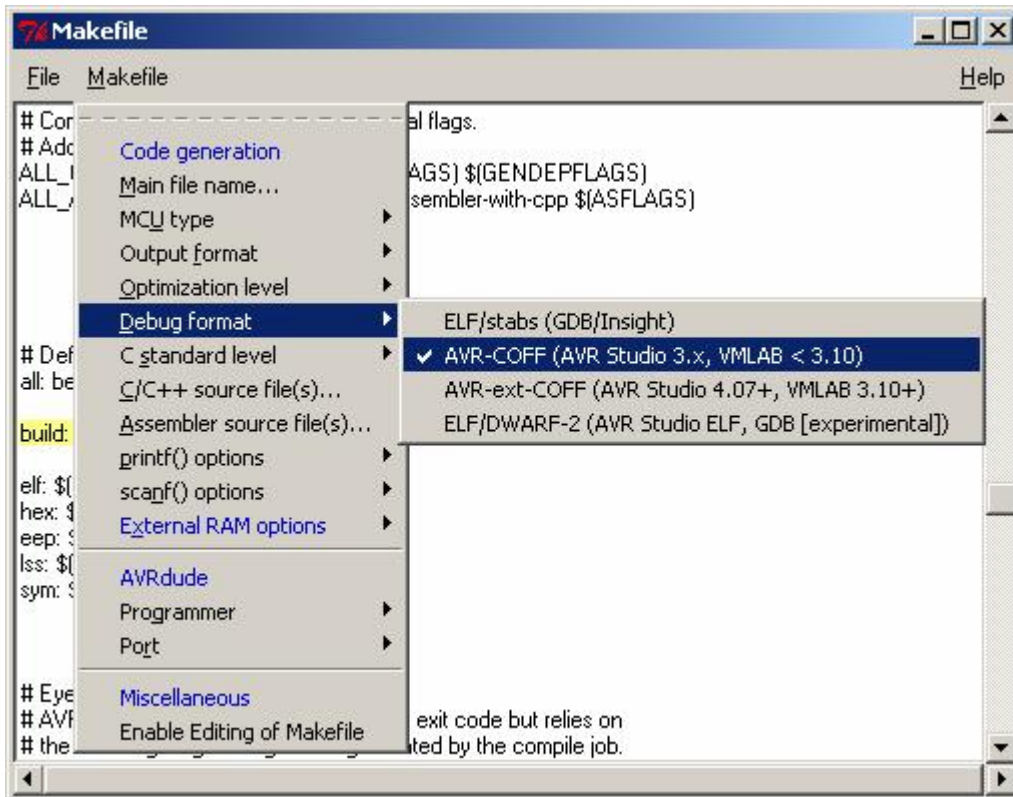
③点击 Makefile->Output format 输出格式。



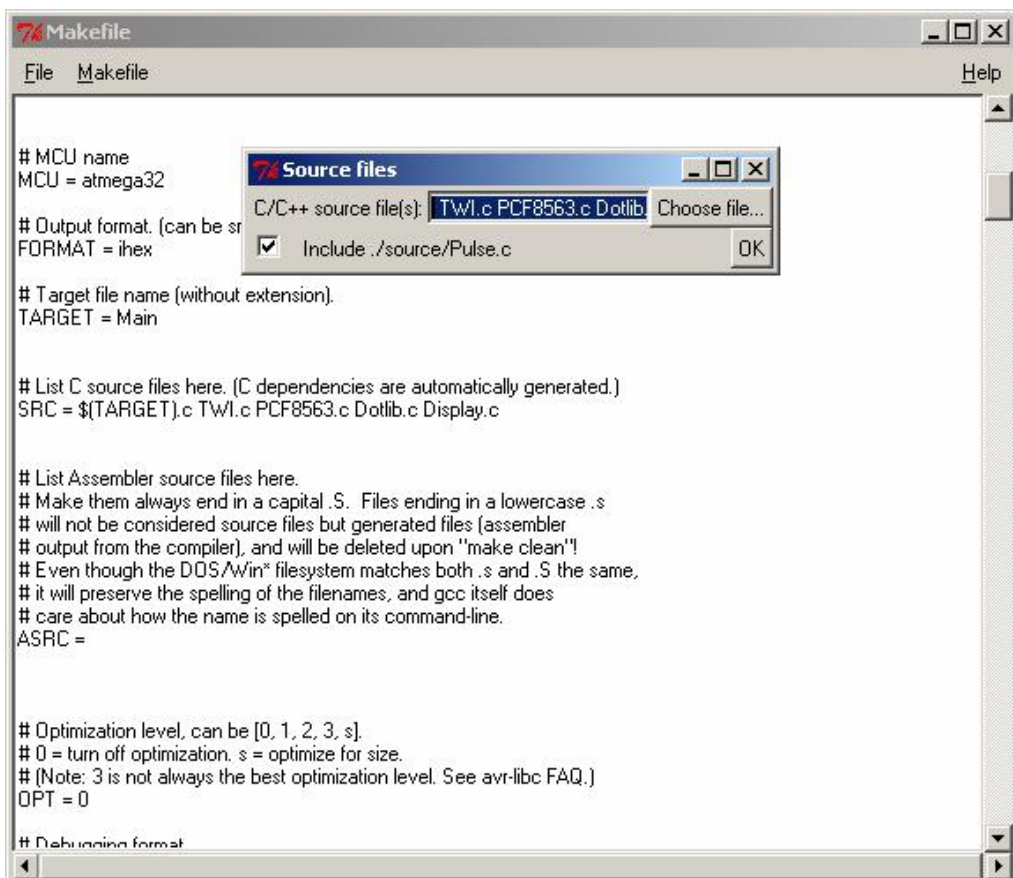
④点击 Makefile->Optimization level 优化级别。

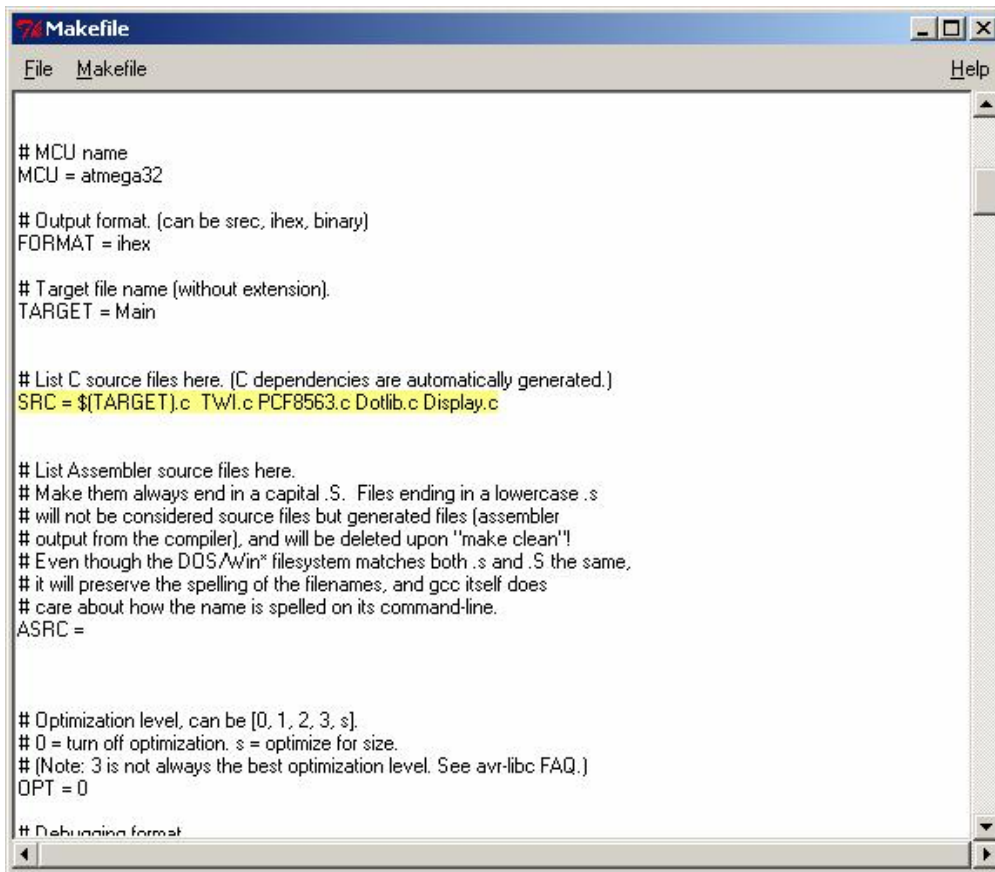


⑤点击 Makefile->Debug format 调试格式。

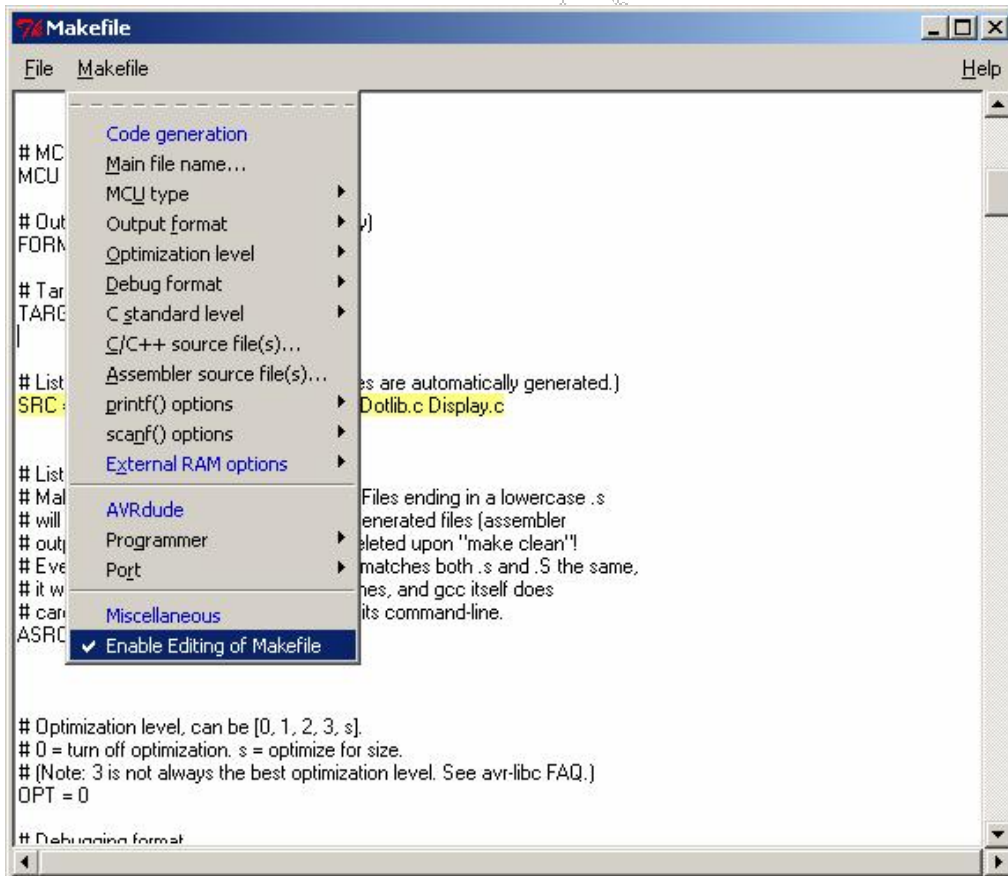


⑥点击 Makefile→C/C++source file(s)…。这一项，对只有一个程序文件可以不用管，多个的若与 Makefile 在同一目录可以直接加入其它 C 文件的文件名，不在同一目录下要加入路径。



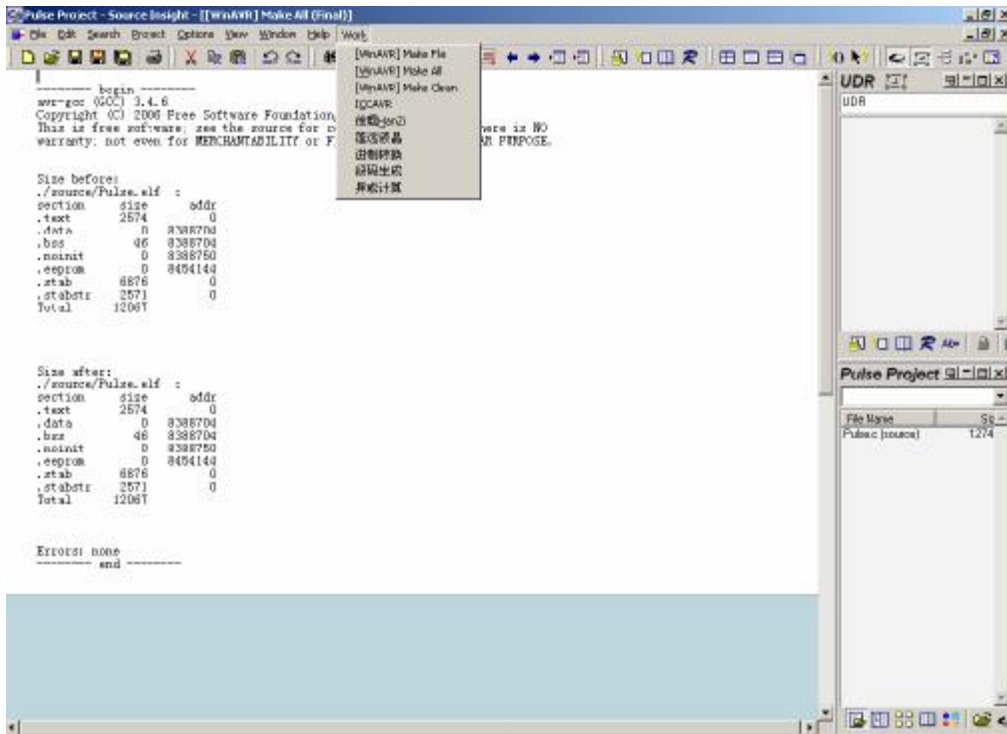


⑦点击 Makefile→C/C++source file(s) ，可手动更改 Makefile。

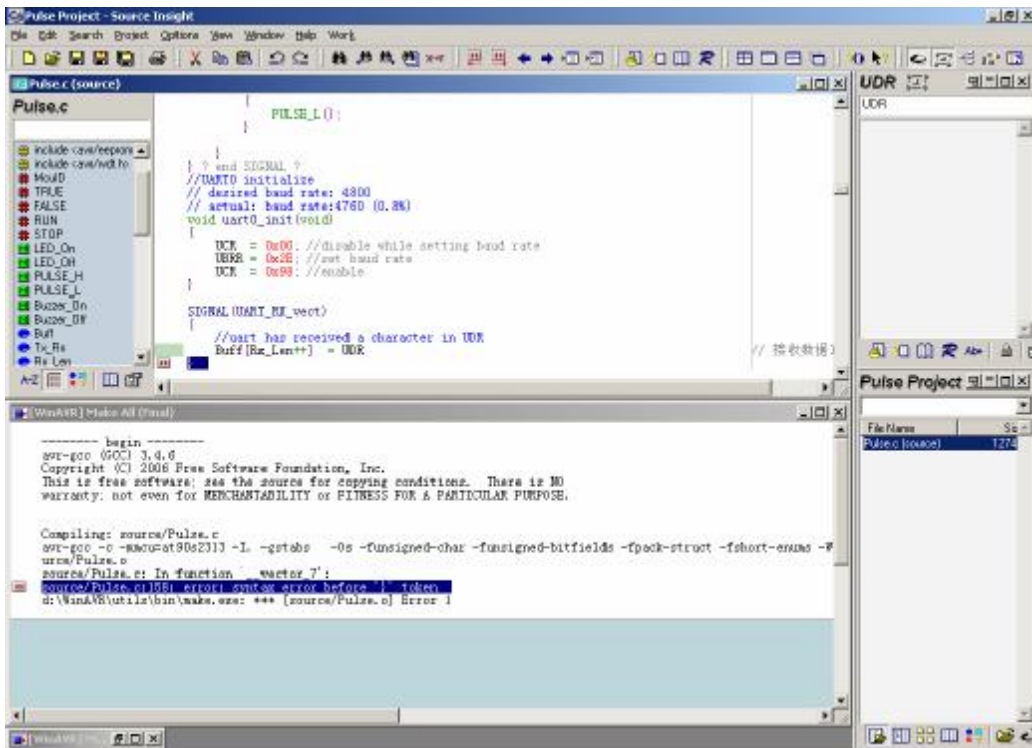


5. Make All 执行界面

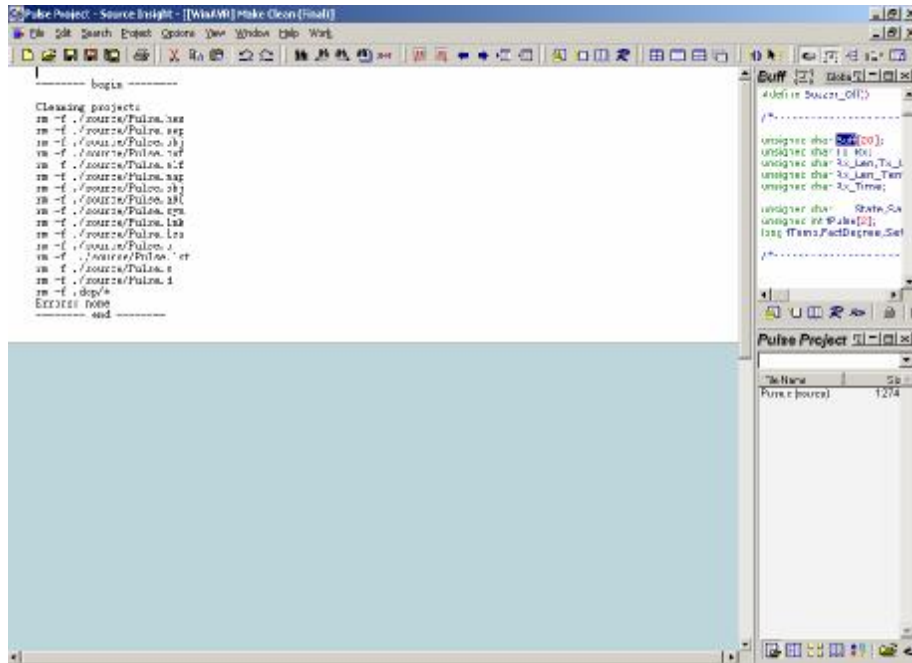
①正常



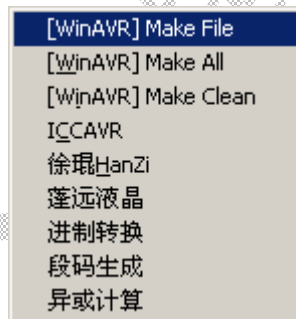
②有错



6. Make Clean 执行界面



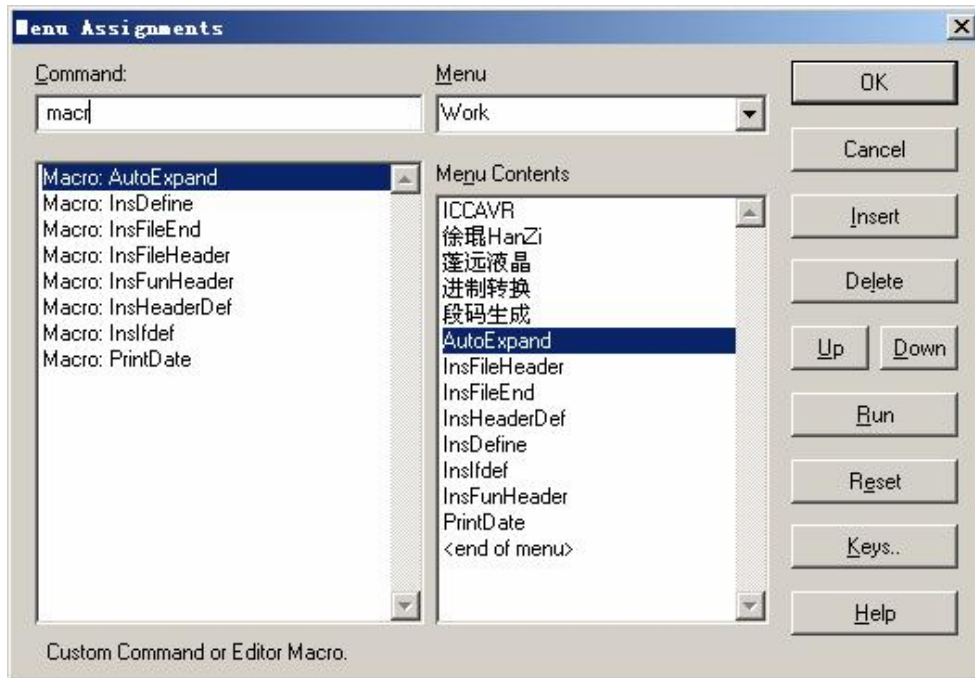
配置好后的菜单预览



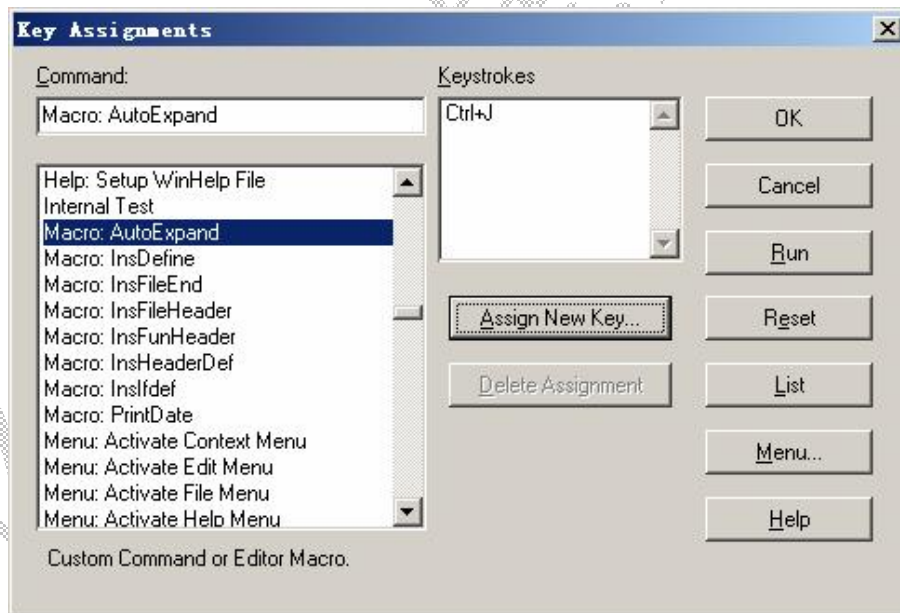
五. 常用宏:

将 `Utils.em` (此文件全部内容见附录) 替换原有文件, 进行如下设置

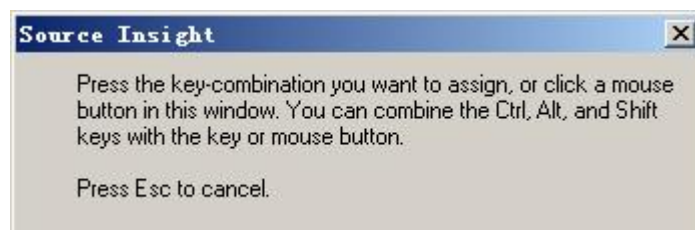
1. Option->Menu Assignments 中 command 栏键入 macro 出下如下界面, Menu 栏中选择 Work, 将所需宏插入 work.



2. 为宏指定快捷键。选择好目标宏后，点 Keys... 出现如下界面：



点击 Assign New Key... 指定快捷键，提示界面如下：



3.配置好后，Work 菜单界面如下：

<u>I</u> CCA VR	
徐 <u>珉</u> Han Zi	
蓬远液晶	
进制转换	
段码生成	
<u>A</u> utoExpand	Ctrl+J
<u>I</u> nsFileHeader	
<u>I</u> nsFileEnd	Ctrl+E
<u>I</u> nsHeaderDef	
<u>I</u> nsDefine	Ctrl+D
<u>I</u> nsIfdef	
<u>I</u> nsFunHeader	
<u>P</u> rintDate	Ctrl+P

提示：宏的具体含义，请仔细阅读附录中 `Utils.em` 文件。

附录

```
/* Utils.em - 常用宏 */

/*****
 * Utils.em - 文件头
 *
 * Copyright 1998-2003 Routon Technology Co.,Ltd.
 *
 * DESCRIPTION: -
 *   文件头注释
 * modification history
 * -----
 * 01a, 27dec2005, cuiqingwei written
 * -----
 *****/
macro InsFileHeader()
{
    szMyName = "cuiqingwei"    // 此处是作者名

    // Get current time
    szTime = GetSysTime(1)
    Day = szTime.Day
    Month = szTime.Month
    Year = szTime.Year
    if (Day < 10)
        szDay = "0@Day@"
    else
```

```

szDay = Day
szMonth = NumToName (Month)

hBuf = GetCurrentBuf ()
szpathName = GetBufName (hBuf)
szfileName = GetFileName (szpathName)
nlength = StrLen (szfileName)
//szInf = Ask ("Enter the information of file:")
//szDescription = Ask ("Enter the description of file:")
szInf = Ask ("请输入文件信息:")
szDescription = Ask ("请输入文件描述:")

hbuf = GetCurrentBuf ()
// begin assembling the title string
InsBufLine (hbuf,
"/*****")
InsBufLine (hbuf, 1, " * @szfileName@ - @szInf@")
InsBufLine (hbuf, 2, " * ")
InsBufLine (hbuf, 3, " * Copyright 1998-2003 Routon Technology Co., Ltd. ")
InsBufLine (hbuf, 4, " * ")
InsBufLine (hbuf, 5, " * DESCRIPTION: - ")
InsBufLine (hbuf, 6, " * @szDescription@")
InsBufLine (hbuf, 7, " * modification history")
InsBufLine (hbuf, 8, " * -----")
InsBufLine (hbuf, 9, " * 01a, @szDay@@szMonth@@Year@, @szMyName@ written")
InsBufLine (hbuf, 10, " * -----")
InsBufLine (hbuf, 11, "
*****/")

// put the insertion point inside the header comment
SetBufIns (hbuf, 1, nlength + strlen (szInf) + 8) // 光标指在文件名后
}

/*-----
- Name: InsFileEnd
- DESCRIPTION: -
- 插入文件结尾注释
- modification history
- -----
- 01a, 27dec2005, cuiqingwei written
* -----
-----*/

macro InsFileEnd ()
{

```



```

hwnd = GetCurrentWnd()
ln = GetWndSelLnFirst(hwnd)
hbuf = GetCurrentBuf()
sz = "/*-----END OF FILE-----*/"
InsBufLine(hbuf, ln, sz)
SetBufIns(hbuf, ln, strlen(sz)+1) // 光标指在函数名后
}

/*-----
- Name: 函数名
- DESCRIPTION: -
- 函数注释
- Input:
- Output:
- Returns:
-
- modification history
-----
- 01a, 27dec2005, cuiqingwei written
* -----
-----*/

macro InsFunHeader()
{
    szMyName = "cuiqingwei" // 此处是作者名

    // Get a handle to the current file buffer and the name
    // and location of the current symbol where the cursor is.
    // Get current time
    szTime = GetSysTime(1)
    Day = szTime.Day
    Month = szTime.Month
    Year = szTime.Year
    if (Day < 10)
        szDay = "0@Day@"
    else
        szDay = Day
    szMonth = NumToName(Month)
    //szName = Ask("Enter the name of function:")
    //szDescription = Ask("Enter the description of function:")
    szName = Ask("请输入函数名:")
    szDescription = Ask("请输入函数描述:")

    hwnd = GetCurrentWnd()
    ln = GetWndSelLnFirst(hwnd)

```

```

hbuf = GetCurrentBuf()

// begin assembling the title string
InsBufLine(hbuf,
/*-----*/
InsBufLine(hbuf, ln + 1, " - Name: @szName@")
InsBufLine(hbuf, ln + 2, " - DESCRIPTION: - ")
InsBufLine(hbuf, ln + 3, " - @szDescription@ ")
// remove by cuiqingwei CutWord(szDescription)
InsBufLine(hbuf, ln + 4, " - Input: ")
InsBufLine(hbuf, ln + 5, " - Output: ")
InsBufLine(hbuf, ln + 6, " - Returns: ")
InsBufLine(hbuf, ln + 7, " - ")
InsBufLine(hbuf, ln + 8, " - modification history")
InsBufLine(hbuf, ln + 9, " - -----")
InsBufLine(hbuf, ln +10, " - 01a, @szDay@@szMonth@@Year@, @szMyName@ written")
InsBufLine(hbuf, ln +11, " * -----")
InsBufLine(hbuf, ln +12,
/*-----*/

// put the insertion point inside the header comment
SetBufIns(hbuf, ln + 1, strlen("Name: ") + strlen(szName) + 8) // 光标指在函数名后
}

/*-----*/
- Name: InsHeaderDef
- DESCRIPTION: -
- 插入头文件条件编译
- modification history
- -----
- 01a, 27dec2005, cuiqingwei written
* -----
/*-----*/

macro InsHeaderDef()
{
hBuf = GetCurrentBuf()
szpathName = GetBufName(hBuf)
szfileName = GetFileName(szpathName)
szfileName = toupper(szfileName)
nlength = StrLen(szfileName)
i = 0 /* loop control */
szdefineName = ""
while (i < nlength)
{

```

```

        if (szfileName[i] == ".")
            szdefineName = Cat(szdefineName, "_")
        else
            szdefineName = Cat(szdefineName, szfileName[i])
            i = i + 1
    }
    szdefineName = Cat("_", szdefineName)
    IfdefineSz(szdefineName)
}

/*-----
- Name: InsIfdef
- DESCRIPTION: -
- 插入条件编译
- modification history
-----
- 01a, 27dec2005, cuiqingwei written
* -----*/

macro InsIfdef()
{
    //sz = Ask("Enter ifdef condition:")
    sz = Ask("请输入 ifdef 条件:")

    if (sz != "")
    {
        // IfdefSz(sz);
        hwnd = GetCurrentWnd()
        sel = GetWndSel(hwnd)
        hbuf = GetWndBuf(hwnd)
        // get line the selection (insertion point) is on
        szLine = GetBufLine(hbuf, sel.lnFirst - 1);
        chTab = CharFromAscii(9)
        // prepare a new indented blank line to be inserted.
        // keep white space on left and add a tab to indent.
        // this preserves the indentation level.
        i = 0 /* loop control */
        ich = ""
        while (szLine[i] == " " || szLine[i] == chTab)
        {
            ich = Cat(ich, szLine[i])
            i = i + 1
        }
    }
}

```

```

        InsBufLine(hbuf, sel.lnFirst, "")
        InsBufLine(hbuf, sel.lnFirst + 1, "@ich@#ifdef @sz@")
        InsBufLine(hbuf, sel.lnFirst + 2, "@ich@" # chTab)
        InsBufLine(hbuf, sel.lnFirst + 3, "@ich@#endif /* @sz@ */")
        SetBufIns(hbuf, sel.lnFirst + 2, StrLen(ich) + StrLen(chTab))
    }
}
// Wrap ifdef <sz> .. endif around the current selection
macro IfdefineSz(sz)
{
    hwnd = GetCurrentWnd()
    lnFirst = GetWndSelLnFirst(hwnd)
    lnLast = GetWndSelLnLast(hwnd)

    hbuf = GetCurrentBuf()
    InsBufLine(hbuf, lnFirst, "#ifndef @sz@")
    InsBufLine(hbuf, lnFirst + 1, "#define @sz@")
    InsBufLine(hbuf, lnLast + 3, "#endif /* @sz@ */")
    SetBufIns (hbuf, lnFirst + 2, 0)
}

/*-----
- Name: InsDefine
- DESCRIPTION: -
- 插入宏定义
- modification history
- -----
- 01a, 27dec2005, cuiqingwei written
*-----*/
macro InsDefine()
{
    sz = Ask("请输入 #define 条件:")

    if (sz != "")
    {
        hwnd = GetCurrentWnd()
        lnFirst = GetWndSelLnFirst(hwnd)

        hbuf = GetCurrentBuf()
        InsBufLine(hbuf, lnFirst, "#define @sz@")
    }
}

```

```
/*-----  
    Automatically expands C statements like if, for, while, switch, etc..
```

To use this macro,

1. Add this file to your project or your Base project.
2. Run the Options->Key Assignments command and assign a convenient keystroke to the "AutoExpand" command.
3. After typing a keyword, press the AutoExpand keystroke to have the statement expanded. The expanded statement will contain a ### string which represents a field where you are supposed to type more.

The ### string is also loaded in to the search pattern so you can use "Search Forward" to select the next ### field.

For example:

1. you type "for" + AutoExpand key
2. this is inserted:

```
for (###; ###; ###)  
{  
    ###  
}
```

3. and the first ### field is selected.

```
-----*/
```

```
macro AutoExpand()  
{  
    // get window, sel, and buffer handles  
    hwnd = GetCurrentWnd()  
    if (hwnd == 0)  
        stop  
    sel = GetWndSel(hwnd)  
    if (sel.ichFirst == 0)  
        stop  
    hbuf = GetWndBuf(hwnd)  
  
    // get line the selection (insertion point) is on  
    szLine = GetBufLine(hbuf, sel.lnFirst);  
  
    // parse word just to the left of the insertion point  
    wordinfo = GetWordLeftOfIch(sel.ichFirst, szLine)  
    ln = sel.lnFirst;  
  
    chTab = CharFromAscii(9)
```

```

// prepare a new indented blank line to be inserted.
// keep white space on left and add a tab to indent.
// this preserves the indentation level.
ich = 0
while (szLine[ich] == ' ' || szLine[ich] == chTab)
{
    ich = ich + 1
}

szLine = strmid(szLine, 0, ich)
sel.lnFirst = sel.lnLast
sel.ichFirst = wordinfo.ich
sel.ichLim = wordinfo.ich

// expand szWord keyword...
if (wordinfo.szWord == "if" ||
wordinfo.szWord == "while" ||
wordinfo.szWord == "elseif")
{
    SetBufSelText(hbuf, " (###)")
    InsBufLine(hbuf, ln + 1, "@szLine@" # "{");
    InsBufLine(hbuf, ln + 2, "@szLine@" # chTab);
    InsBufLine(hbuf, ln + 3, "@szLine@" # "}");
}
else if (wordinfo.szWord == "for")
{
    SetBufSelText(hbuf, " (###)")
    InsBufLine(hbuf, ln + 1, "@szLine@" # "(");
    InsBufLine(hbuf, ln + 2, "@szLine@" # chTab);
    InsBufLine(hbuf, ln + 3, "@szLine@" # "}");
}
else if (wordinfo.szWord == "switch")
{
    SetBufSelText(hbuf, " (###)")
    InsBufLine(hbuf, ln + 1, "@szLine@" # "(");
    InsBufLine(hbuf, ln + 2, "@szLine@" # "case ");
    InsBufLine(hbuf, ln + 3, "@szLine@" # chTab);
    InsBufLine(hbuf, ln + 4, "@szLine@" # chTab # "break;");
    InsBufLine(hbuf, ln + 5, "@szLine@" # "default:");
    InsBufLine(hbuf, ln + 6, "@szLine@" # chTab);
    InsBufLine(hbuf, ln + 7, "@szLine@" # "}");
}
else if (wordinfo.szWord == "do")

```

```

{
    InsBufLine(hbuf, ln + 1, "@szLine@" # "{")
    InsBufLine(hbuf, ln + 2, "@szLine@" # chTab);
    InsBufLine(hbuf, ln + 3, "@szLine@" # "} while ();")
}
else if (wordinfo.szWord == "case")
{
    SetBufSelText(hbuf, "###")
    InsBufLine(hbuf, ln + 1, "@szLine@" # chTab)
    InsBufLine(hbuf, ln + 2, "@szLine@" # chTab # "break;")
}
else
stop

SetWndSel(hwnd, sel)
LoadSearchPattern("###", true, false, false);
Search_Forward
}

/* GET WORD LEFT OF ICH */
/*-----*/
Given an index to a character (ich) and a string (sz),
return a "wordinfo" record variable that describes the
text word just to the left of the ich.
Output:
wordinfo.szWord = the word string
wordinfo.ich = the first ich of the word
wordinfo.ichLim = the limit ich of the word
-----*/
macro GetWordLeftOfIch(ich, sz)
{
    wordinfo = "" // create a "wordinfo" structure

    chTab = CharFromAscii(9)

    // scan backwards over white space, if any
    ich = ich - 1;
    if (ich >= 0)
    while (sz[ich] == " " || sz[ich] == chTab)
    {
        ich = ich - 1;
        if (ich < 0)
            break;
    }
}

```

```

// scan backwards to start of word
ichLim = ich + 1;
asciiA = AsciiFromChar("A")
asciiZ = AsciiFromChar("Z")
while (ich >= 0)
{
    ch = toupper(sz[ich])
    asciiCh = AsciiFromChar(ch)
    if ((asciiCh < asciiA || asciiCh > asciiZ) && !IsNumber(ch))
        break // stop at first non-identifier character
    ich = ich - 1;
}

ich = ich + 1
wordinfo.szWord = strmid(sz, ich, ichLim)
wordinfo.ich = ich
wordinfo.ichLim = ichLim;

return wordinfo
}

/*-----
- Name: GetFileName
- DESCRIPTION: -
- 取得文件名
- Input:  pathName
- Returns:name
- modification history
- -----
- 01a, 27dec2005, cuiqingwei written
* -----
*/

macro GetFileName(pathName)
{
    nlength = strlen(pathName)
    i = nlength - 1
    name = ""
    while (i + 1)
    {
        ch = pathName[i]
        if ("\" == "@ch@")
            break
        i = i - 1
    }
}

```



```

    }
    i = i + 1
    while (i < nlength)
    {
        name = cat(name, pathName[i])
        i = i + 1
    }

    return name
}

/*-----
- Name: NumToName
- DESCRIPTION: -
-   将月份数字转换为英文缩写
- Input: Month
- Returns:Num
- modification history
-----
- 01a, 27dec2005, cuiqingwei written
* -----*/

macro NumToName(Month)
{ /*
    if (Month == 1)
        return "jan"
    if (Month == 2)
        return "feb"
    if (Month == 3)
        return "mar"
    if (Month == 4)
        return "apr"
    if (Month == 5)
        return "may"
    if (Month == 6)
        return "jun"
    if (Month == 7)
        return "jul"
    if (Month == 8)
        return "aug"
    if (Month == 9)
        return "sep"
    if (Month == 10)
        return "oct"

```

```

    if (Month == 11)
        return "nov"
    if (Month == 12)
        return "dec"
*/
if (Month == 1)
    return ".01."
if (Month == 2)
    return ".02."
if (Month == 3)
    return ".03."
if (Month == 4)
    return ".04."
if (Month == 5)
    return ".05."
if (Month == 6)
    return ".06."
if (Month == 7)
    return ".07."
if (Month == 8)
    return ".08."
if (Month == 9)
    return ".09."
if (Month == 10)
    return ".10."
if (Month == 11)
    return ".11."
if (Month == 12)
    return ".12."
}

/*-----
- Name: PrintDate
- DESCRIPTION: -
-   计算并输入当前时间
- Output: 打印当前时间
- modification history
-----
- 01a, 27dec2005, cuiqingwei written
* -----
-----*/

macro PrintDate()
{
    szTime = GetSysTime(1)

```

```

Day = szTime.Day
Month = szTime.Month
Year = szTime.Year
if (Day < 10)
    szDay = "0@Day@"
else
    szDay = Day
szMonth = NumToName(Month)
hbuf = GetCurrentBuf()
ln = GetBufLnCur(hbuf)
szCurLine = GetBufLine(hbuf, ln)
DelBufLine(hbuf, ln)
InsBufLine(hbuf, ln, "@szCurLine@ @szDay@@szMonth@@Year@")

//SetBufIns(hbuf, ln, StrLen(szCurLine) + 10) //适用于月份英文缩写
SetBufIns(hbuf, ln, StrLen(szCurLine) + 11) //适用于月数字表示
}

/*-----
- Name: CutWord
- DESCRIPTION: -
- auto newline
- Input: ncurLine, szInf
- modification history
- -----
- 01a, 27dec2005, cuiqingwei written
* -----*/

macro CutWord(ncurLine, szInf)
{
    LENGTH = 63
    nlength = StrLen(szInf)
    i = 0 /* loop control */
    begin = 0 /* first character's index of current line */
    pre = 0 /* preceding word's index */
    hbuf = GetCurrentBuf()
    // nline = GetBufLnCur()
    while (i < nlength)
    {
        /* remove by t357
nrow = 0
sz = ""
while (nrow < 80)
{

```

```

if (nlength < 0)
break
sz = Cat(sz, szInf[nrow])
nrow = nrow + 1
nlength = nlength - 1
}
InsBufLine(hbuf, nline, sz)
szInf = szInf[nrow]
}
*/

c = szInf[i]
if (" " == @c@ && (i - b < LENGTH))
{
pre = i
}
else if (" " == @c@)
{
szOutput = ""
k = begin /* loop control */
while (k < pre)
{
szOutput = Cat(szOutput, szInf[k])
k = k + 1
}
InsBufLine(hbuf, ncurLine, sz)
ncurLine = ncurLine + 1
begin = pre
}
i = i + 1
}
if (h != i - 1)
{
szOutput = ""
k = begin /* loop control */
while (k < pre)
{
szOutput = Cat(szOutput, szInf[k])
k = k + 1
}
InsBufLine(hbuf, ncurLine, sz)
ncurLine = ncurLine + 1
}
}
/*----- END OF FILE -----*/

```