MIRACL 大数运算库使用手册

游贵荣

一. MIRACL 简介

MIRACL(Multiprecision Integer and Rational Arithmetic C/c++ Library)是一套由 Shamus Software Ltd.所开发的一套关于大数运算函数库，用来设计与大数运算相关的密码学之应用，包含了 RSA 公开密码学、Diffie-Hellman 密钥交换(Key Exchange)、AES、DSA 数字签名，还包含了较新的椭圆曲线密码学(Elliptic Curve Cryptography)等等。运算速度快，并提供源代码。MIARCL 是当前使用比较广泛的基于公钥加密算法保护实现的大数库之一，据说要使用该库用于商业软件，需要交纳一笔昂贵的授权费——1000$。

二. MIRACL 常用函数调用手册

声明：此处只列出和大数相关的简单运算函数，以及产生大数随机数的函数调用手册，具体请查看 manual.doc 文档。不当之处，请大家批评指正！

函数原型: void absol(big x, big y);
功能说明: 取 x 的绝对值，y=|x|

函数原型: void add(big x, big y, big z);
功能说明: 两个大数相加，z=x+y

```
Example: add(x,x,x);  // This doubles the value of x.
```

函数原型: void bigbits(int n, big x);
功能说明: 产生一个 n 位的大整数，初始化随机种子由 irand 函数实现

```
Example: bigbits(100,x); //This generates a 100 bit random number
```

函数原型: int cinstr(big x, char *s);
功能说明: 将大数字符串转换成大数
返回值: 输入字符数的个数

Example: mip->IOBASE=16; // input large hex number into big x
cinstr(x,” AF12398065BFE4C96DB723A”);

函数原型: int compare(big x, big y);
功能说明: 比较两个大数的大小
返回值: x>y 时返回+1, x=y 时返回 0, x<y 时返回-1

函数原型: void convert(int n, big x);
功能说明: 将一个整数 n 转换成一个大数 x

函数原型: void copy(big x, big y);
功能说明: 将一个大数赋值给另一个大数，y=x

函数原型: int cotstr(big x, char *s);
功能说明: 将一个大数根据其进制转换成一个字符串
返回值: 字符串长度

函数原型: void decr(big x, int n, big z) ;
功能说明: 将一个大数减去一个整数， z=x-n.

函数原型: void divide(big x, big y, big z);
功能说明: 两个大数相除，z=x/y; x=x mod y，当变量 y 和 z 相同时，x 为余数，商不返回（即 y 的值不变）；当 x 和 z 相同时，x 为商，余数不返回。
Example: divide(x,y,y);//x 为余数，y 值不变

函数原型: BOOL divisible(big x, big y)
功能说明: 测试 x 能否整除 y
返回值: y 除 x 余数为 0，返回 TRUE，否则返回 FALSE

函数原型: int igcd(int x, int y) ;

功能说明：返回两个整数的最大公约数

函数原型：void incr(big x, int n, big z);
功能说明：将一个大数加上一个整数， z=x+n
Example:　　incr(x,2,x);　/* This increments x by 2. */

函数原型：void mirkill(big x);
功能说明：释放内存大数所占的内存

函数原型：miracl *mirsys(int nd, int nb);
功能说明：初始化 MIRACL 系统，该函数必须在调用 MIRACL 库函数之前先执行
Example:　　miracl *mip=mirsys(500,10);//初始化 500 位的 10 进行制数

函数原型：void mirexit();
功能说明：清除 MIRACL 系统，释放所有内部变量

函数原型：void multiply(big x, big y, big z);
功能说明：两个大数相乘，z=x.y

函数原型：void negify(big x, big y);
功能说明：大数取负号，y=-x.

函数原型：int numdig(big x);
功能说明：返回大数 x 中数字的个数

函数原型：void premult(big x, int n, big z);
功能说明：一个大数乘以一个整数，z=n.x

函数原型：int subdiv(big x, int n, big z);
功能说明：一个大数除以一个整数，z=x/n.

返回值：余数

函数原型：BOOL subdivisible(big x, int n)
功能说明：测试 n 能否整除 x
返回值：x 除以 n 余数为 0，返回 TRUE，否则返回 FALSE

函数原型：void bigdig(int n, int b, big x);
功能说明：产生一个指定长度的进制的随机数，该函数使用内置的随机数发生器，初始化种子调用 irand 函数
Example：bigdig(100,10,x); //产生一个 100 位的 10 进制随机数

函数原型：void bigrand(big w, big x);
功能说明：使用内置的随机数发生器，产生一个小于 w 的大数随机数，x<w

函数原型： int egcd(big x, big y, big z);
功能说明:计算两个大数的最大公约数，z=gcd(x,y)

函数原型：void expb2(int n, big x)
功能说明：计算 2 的 n 次方的大数
Example： expb2(1398269,x); //2^1398269
```
           decr(x,1,x);       //x = x - 1
           mip->IOBASE=10;    //使用 10 进制
           cotnum(x,stdout);  //输出到屏幕
```
This calculates and prints out the largest known prime number (on a true 32-bit computer with lots of memory!)

函数原型： void expint(int b, int n, big x);
功能说明：计算 b 的 n 次方的大数

函数原型：void fft_mult(big x, big y, big z);
功能说明：使用 Fast Fourier 算法计算两个大数乘积，z=x.y

函数原型:　unsigned int invers(unsigned int x, unsigned int y);
功能说明:计算两个无符号整数（要求互素）的模逆，返回 $x^{-1} \mod y$

函数原型:　BOOL isprime(big x);
功能说明:判断一个大数是否为素数，使用概率测试算法
返回值: x 为素数返回 TRUE，否则返回 FALSE

函数原型: void powmod(big x, big y, big z, big w);
功能说明: 模幂运算，$w=x^y \mod z$

函数原型: void sftbit(big x, int n, big z);
功能说明:将一个大数左移或右移 n 位，n 为正数时左移，负数时右移

函数原型: int xgcd(big x, big y, big xd, big yd, big z);
功能说明: 计算两个大数的扩展最大公约数，也可以用来计算模逆，这个函数比 mad 函数运算速度稍慢。$z=gcd(x, y)=x. xd+y. yd$
Example:　xgcd(x,p,x,x,x);　//计算 x^-1 mod p
/* $x = 1/x \mod p$　(p is prime) */

三. MIRACL 函数库调用举例

1.使用微软的 VS.NET 2003 中文版

（1）启动 Microsoft Visual Studio .NET 2003，选择“文件”→“新建”→“项目”命令，如图 1-1 所示；

图 1-1　新建项目

（2）打开“新建项目”对话框，选择“Win32 控制台项目”模板，在“名称”文本框中输入“TestMircal”，如图 1-2 所示，单击“确定”按钮；

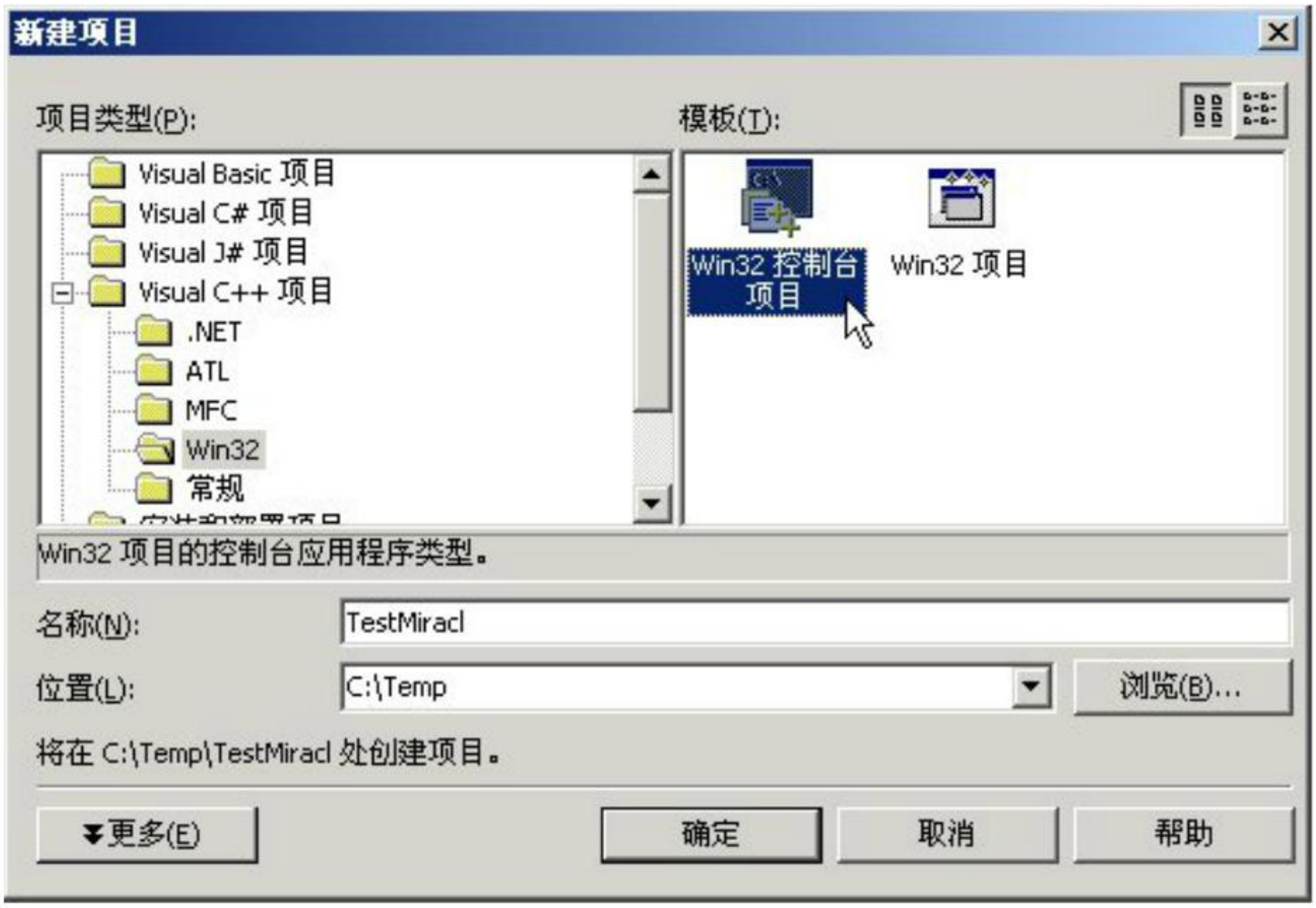


图 1-2　选择模板

（3）单击“完成”按钮，完成新建项目；

（4）将大数运算静态库文件 ms32.lib 和头部文件 miracl.h 和 mirdef.h 拷贝到项目所在文件夹，本例中为“C:\Temp\TestMiracl”，如图 1-3 所示；

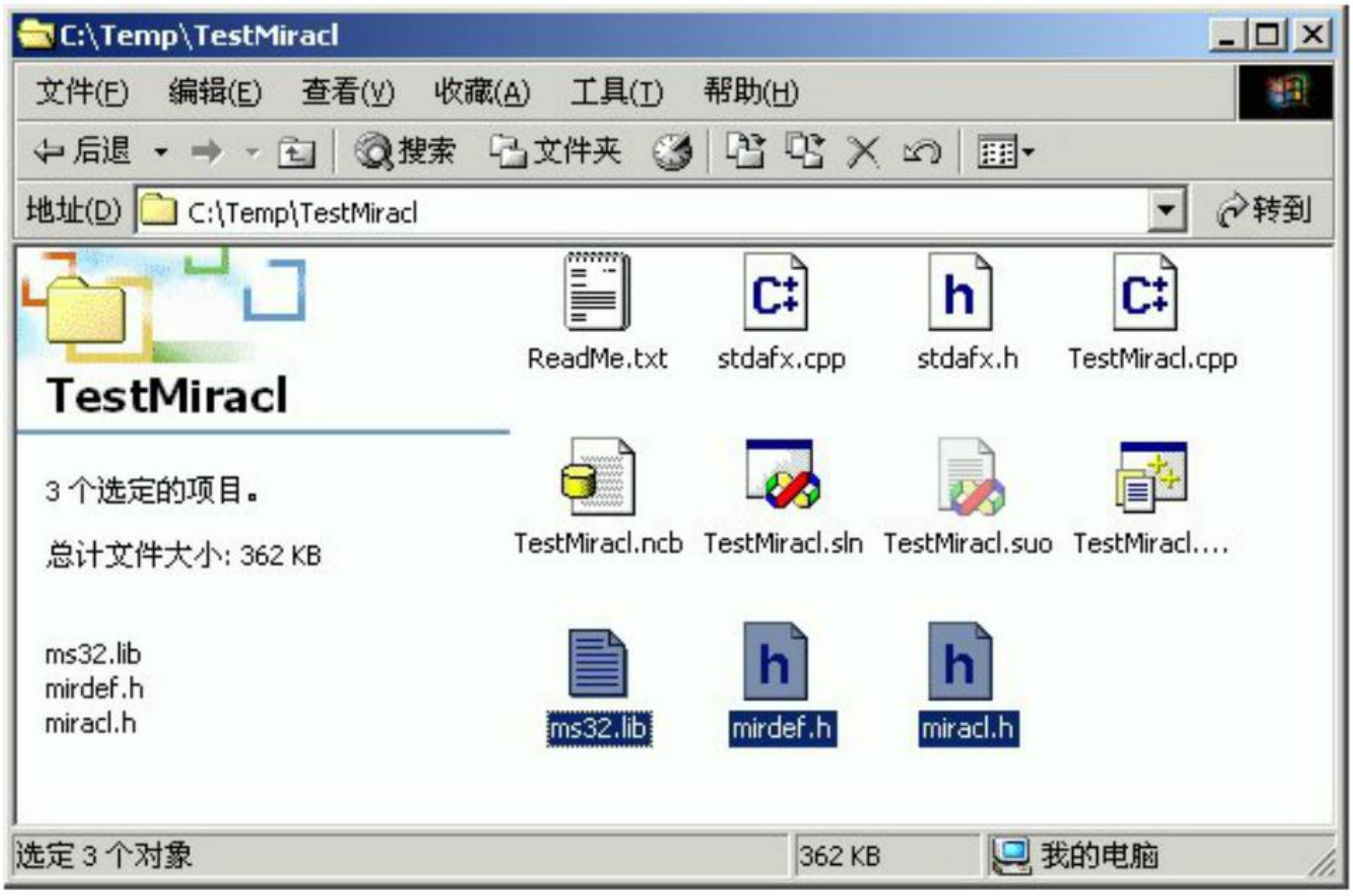


图 1-3　拷贝大数运算库所需文件

（5）将大数运算静态库文件 ms32.lib 文件添加到项目中，操作方法是：右击“TestMircal”,选择快捷菜单中的“添加”→“添加现有项”命令，如图 1-4 所示；

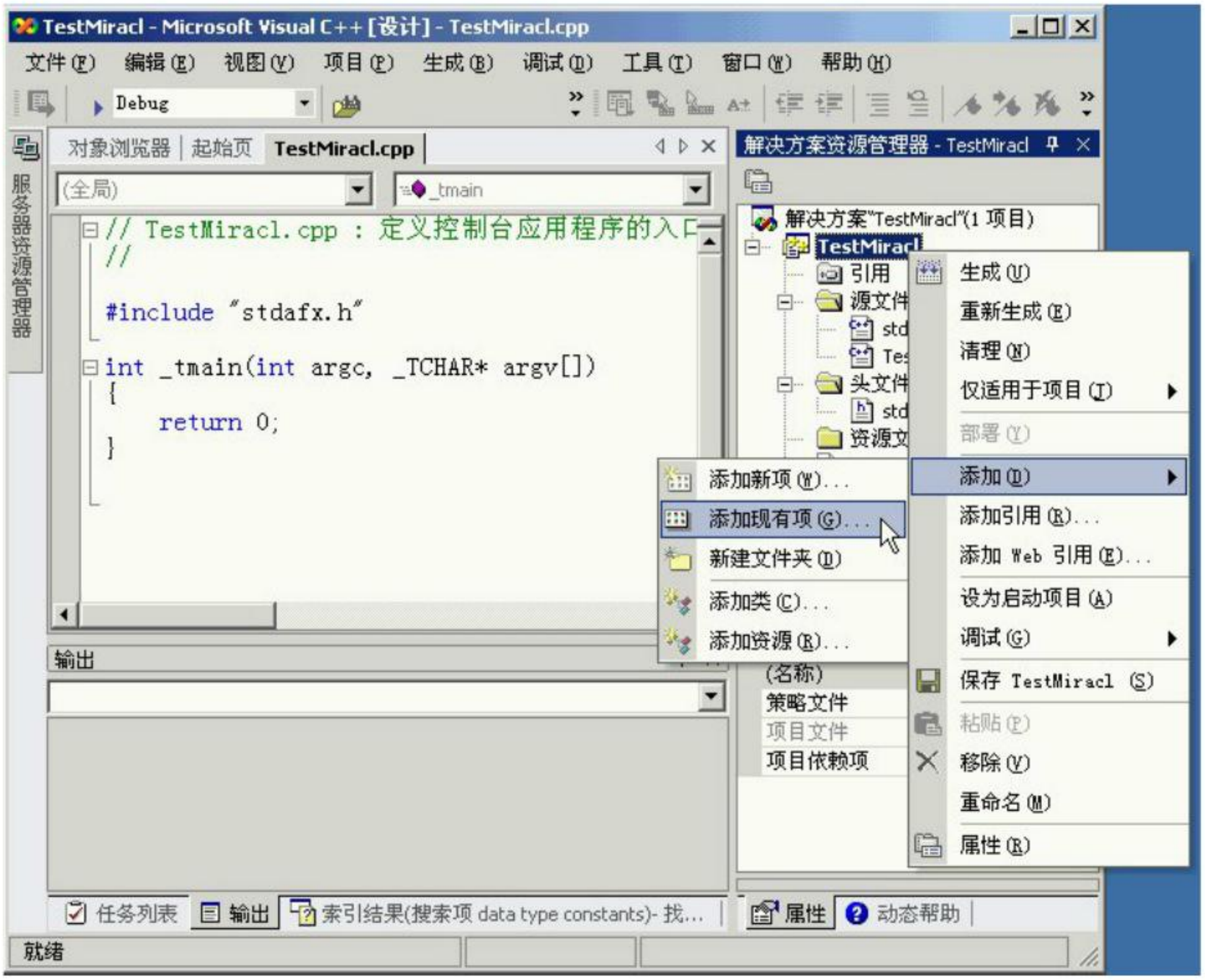


图 1-4　打开添加现有项对话框

（6）打开“添加现有项-TestMircal”对话框，选择文件类型为“所有文件(*.*)”，双击“ms32.lib”文件，将其添加到项目中，如图 1-5 所示；

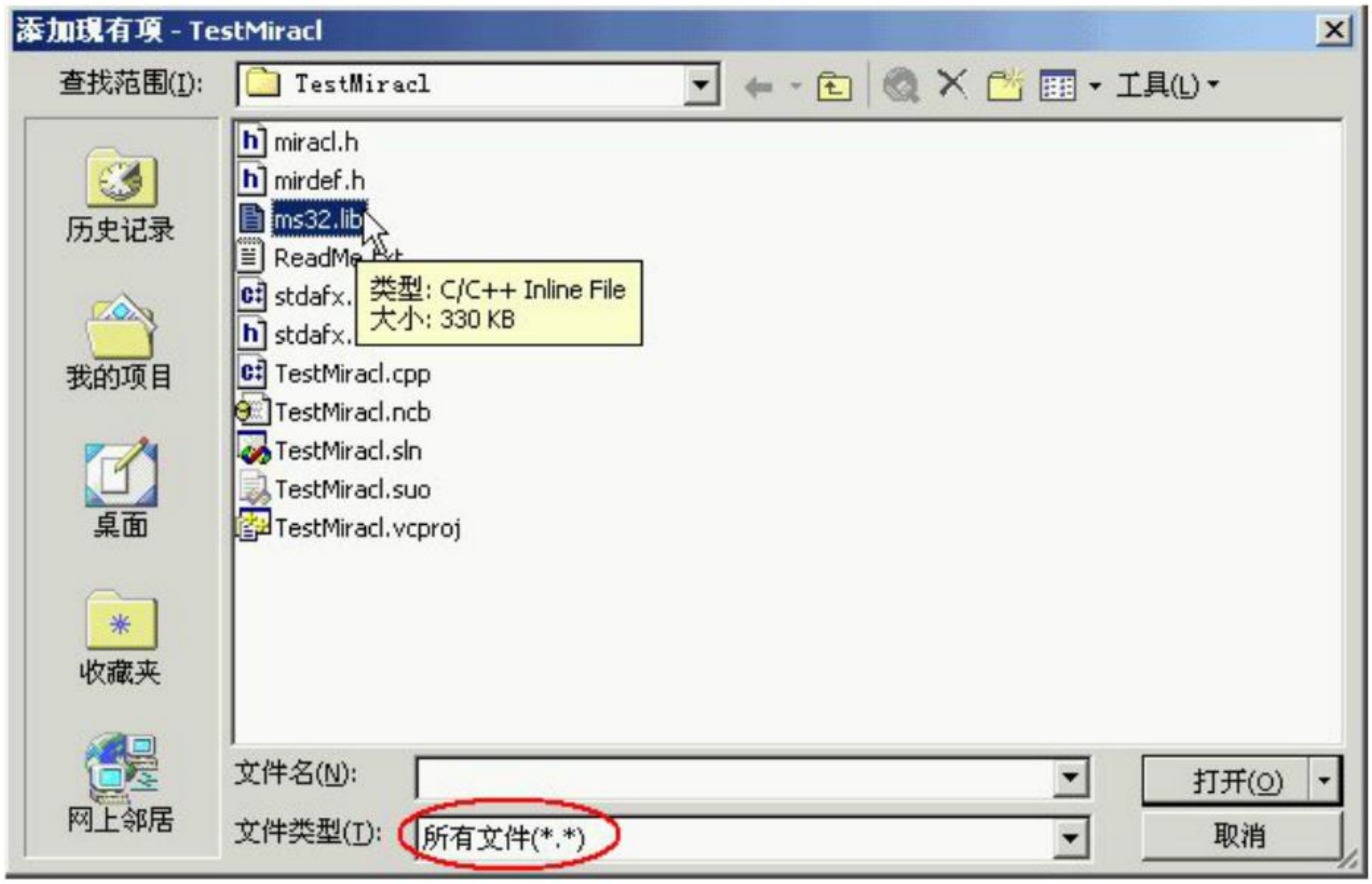


图 1-5　添加 ms32.lib 库文件

（7）右击“TestMircal”,选择快捷菜单中的“添加”命令，打开“属性页”对话框，单击“C/C++”配置属性，选择“预编译头”选项，设置为“不使用编译头”，如图 1-6 所示，单击“确定”按钮；

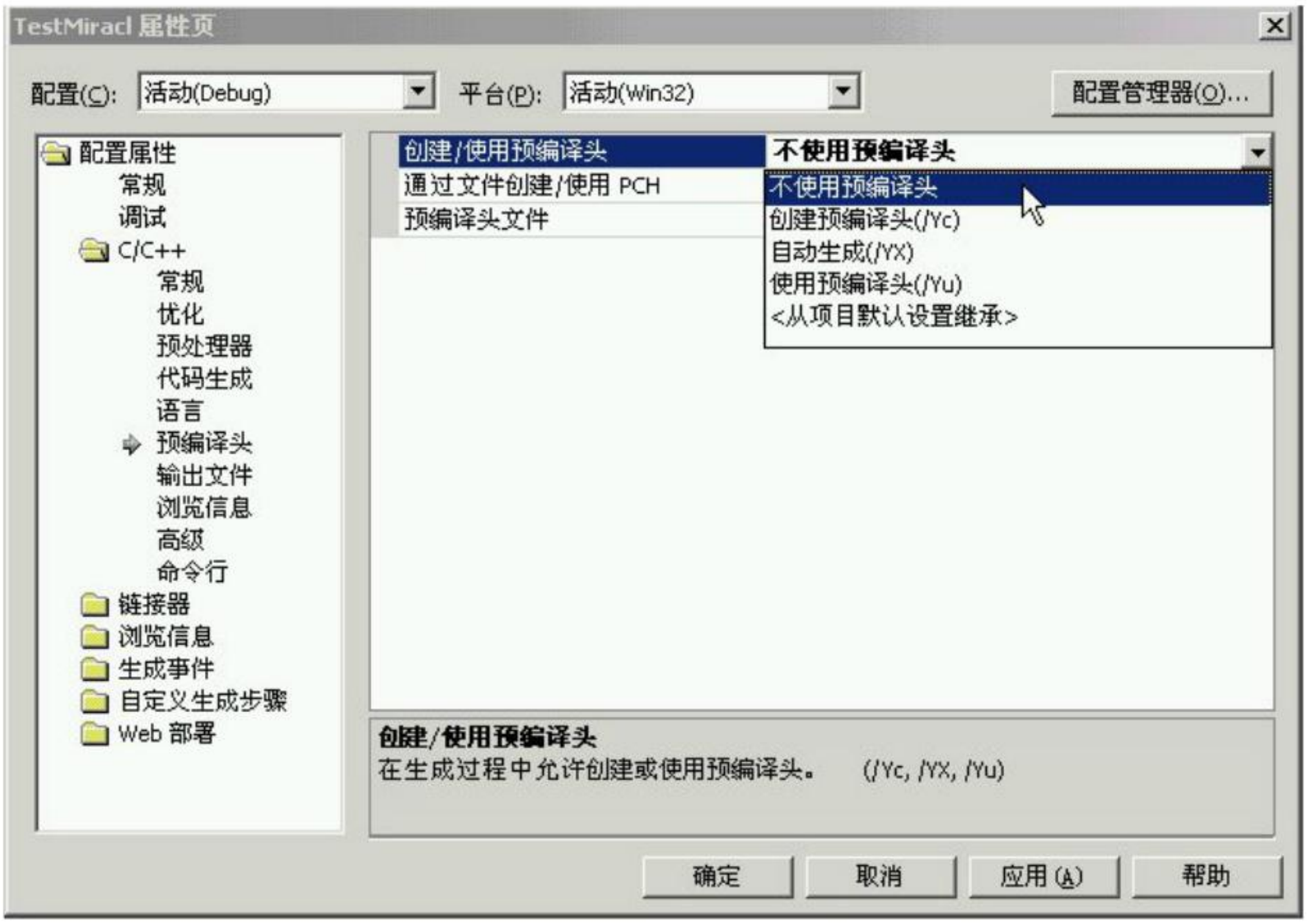
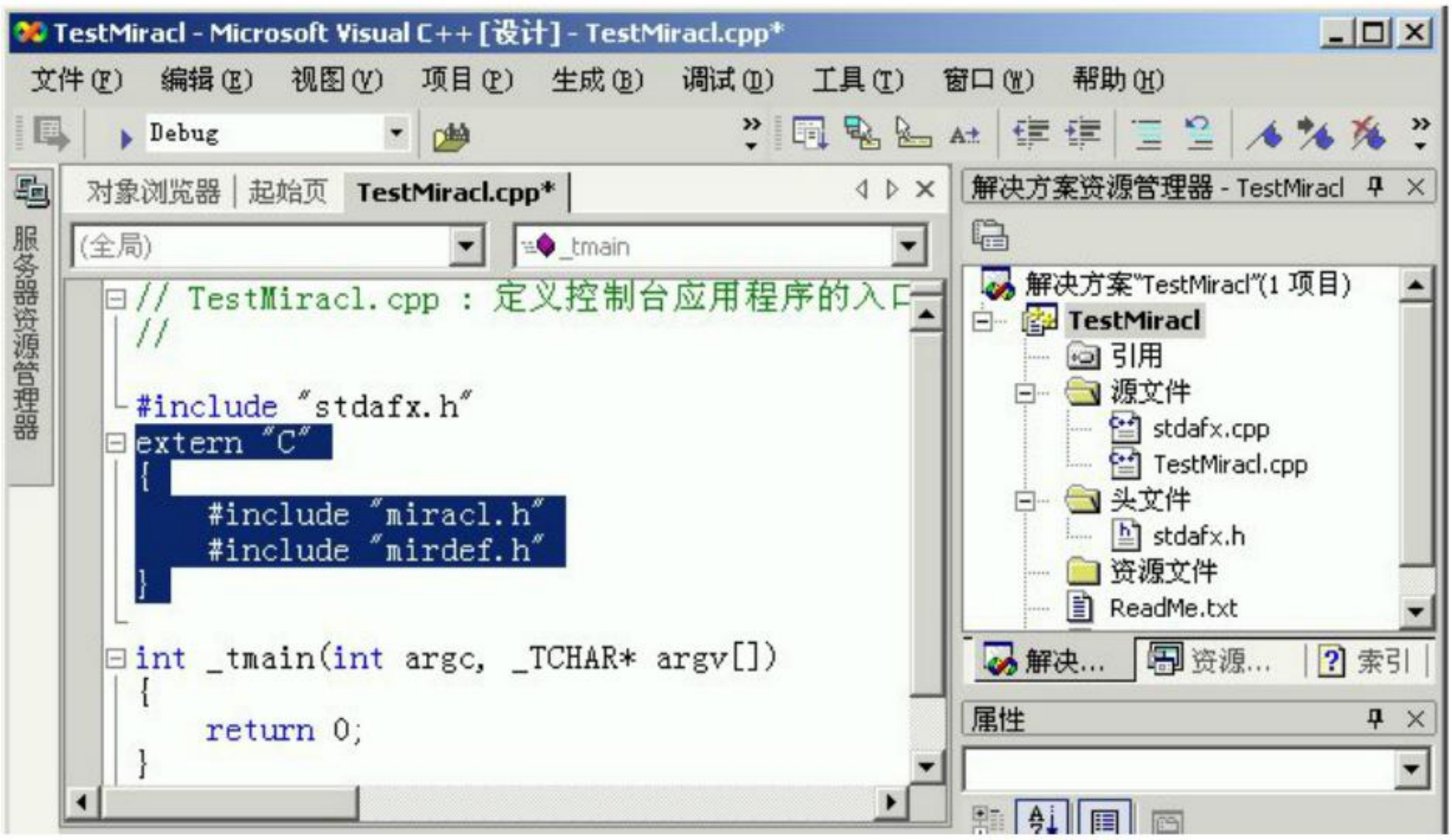


图 1-6　不使用预编译头

（7）为项目添加如下头文件的包含，此处使用 extern "C"是表示用 C 的方式编译，因为 ms32.lib 是 C 的库，不是 C++的库，如图 1-7 所示；

```
extern "C"
{
    #include "miracl.h"
    #include "mirdef.h"
}
```



（8）在_tmain 函数中插入如下代码，以测试大数运算情况；

```
miracl *mip = mirsys(400,10);  //初始化一个400位10进制的大数系统
big x,y,z;


x = mirvar(177);
y = mirvar(79);
z = mirvar(0);
divide(x, y, z);    //x=x mod y, z=x/y

cotnum(x,stdout);   //x=19
cotnum(y,stdout);   //y=79
cotnum(z,stdout);   //z=2

    multiply(x, y, z); //z=x*y
    mip->IOBASE=16;     //将原来的10进制改为16进制模式
    cotnum(z, stdout); //5DD

/*  测试13^-1 mod 2436 = 937
    x = mirvar(13);
    y = mirvar(2436);
```

```
    xgcd(x, y, z, z, z);
    std::cout<<"z=";
    cotnum(z, stdout);
*/

    mirkill(x);          //释放大数变量
    mirkill(y);
    mirkill(z);

    //=============================================================
    //下面进行RSA算法加密和解密运算
char OutStr[500];
char mStr[]="Computer";

    big m=mirvar(0); //m 明文
    big c=mirvar(0); //c 密文

    big p=mirvar(0); //大素数p
    big q=mirvar(0); //大素数q

    big n=mirvar(0); //n 模数
    big pn=mirvar(0); //欧拉函数值pn = (p - 1)(q - 1)

    big d=mirvar(0); //d 私钥
    big e=mirvar(0); //e 公钥


    mip->IOBASE=10;     //将原来的16进制改为10进制模式

    expb2(500, p);      //计算2的500次方, 2^1024 ~= 1.8 * 10^308
    nxprime(p, p);      //找一个比2的500次方大的素数
    std::cout<<"p=";
    cotnum(p, stdout);

    //还是测试一下是否为素数
    if ( isprime(p) ) std::cout<<"p is a prime!"<<"\n";
```

```
premult(p, 2, q);  //q=p*2
nxprime(q, q);     //找一个比p*2大的素数
std::cout<<"q=";
cotnum(q, stdout);
//还是测试一下是否为素数
if ( isprime(q) ) std::cout<<"q is a prime!"<<"\n";

multiply(p, q, n); //n = (p - 1)(q - 1)

//以下计算欧拉函数值pn
decr(p, 1, p);     //p = p - 1
decr(q, 1, q);     //q = q - 1
multiply(p, q, pn);//pn = (p - 1)(q - 1)

convert(65537, e); //取e公钥为2的16次方加1
//cinstr(e,"65537");   //取e公钥为2的16次方加1

xgcd(e, pn, d, d, d);  //计算d = e^-1 mod n
std::cout<<"d=";
cotnum(d, stdout);

bytes_to_big(8, mStr, m);  //将8个字符的明文，转换成大数
std::cout<<"m=";
cotnum(m, stdout);

//加密
powmod(m,e,n,c);           //计算c=m^e mod n
std::cout<<"c=";
cotnum(c, stdout);

//解密
powmod(c,d,n,m);           //计算m=c^d mod n
std::cout<<"m=";
cotnum(m, stdout);
big_to_bytes(256,m,OutStr,FALSE); //将m转换成数组写入temp
OutStr[8] = '\0';
std::cout<<"OutStr="<<OutStr<<"\n";
```

```
mirkill(m);          //释放大数变量
mirkill(c);

mirkill(p);
mirkill(q);

mirkill(d);
mirkill(e);

mirkill(n);
mirkill(pn);

mirexit();
```

2.使用微软的 VC++V6.0 英文版

使用 VC++V6.0 和 VS.NET2003 的区别主要在创建项目和项目属性设置有点不一样。关键就是要把 MS32.LIB 静态库文件添加到 project 中，还有就是 project 的项目属性中的编译选项不要用“预编译头文件”。详细情况请参见 msvisual.txt 文档。