

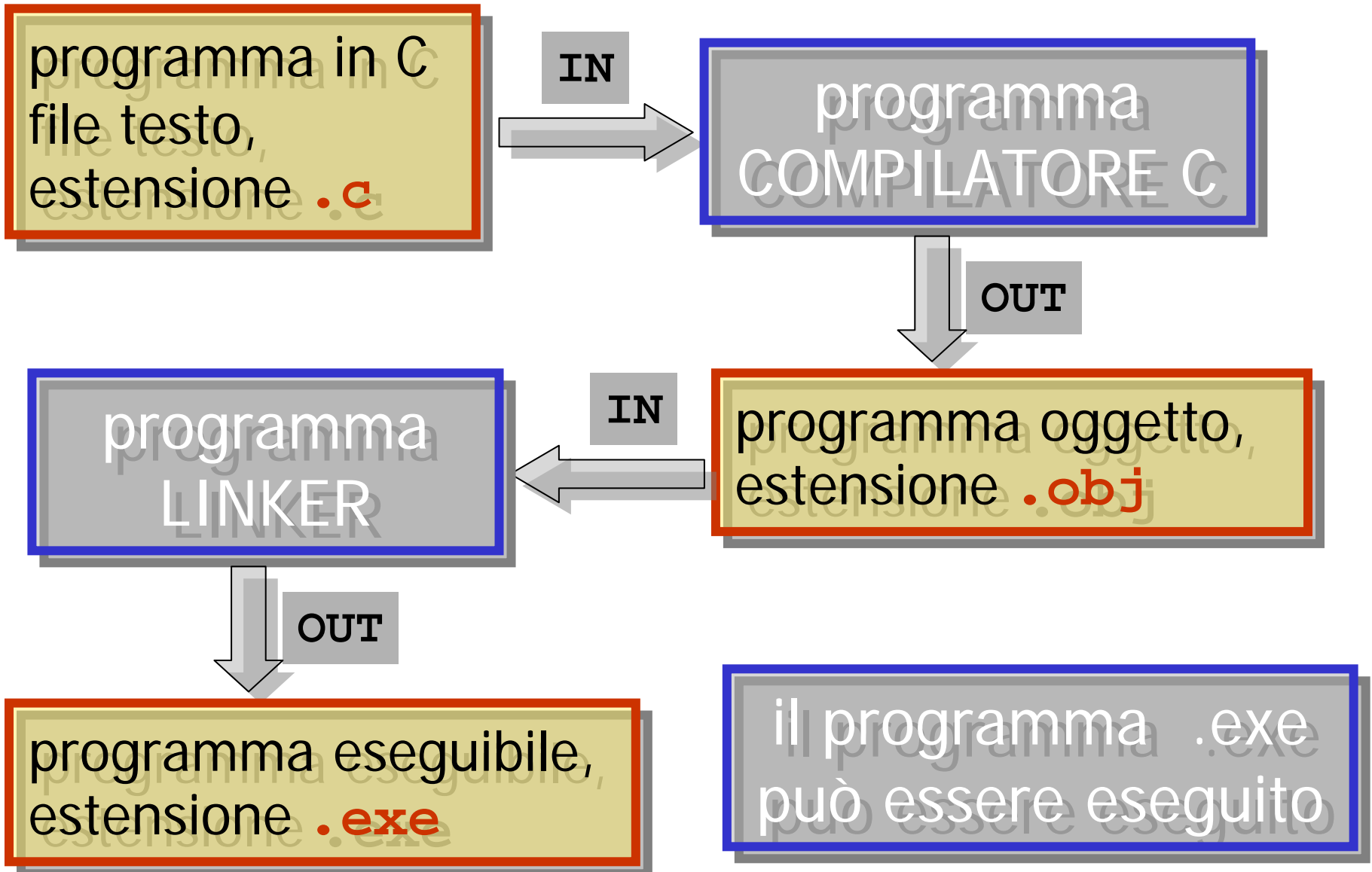
# Tecniche e strumenti per l'organizzazione del software in ambiente Microsoft Visual C++

librerie C di utente

librerie dinamiche

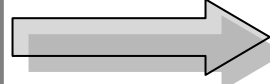
progetti multipli

# processo di **compilazione-linking-esecuzione**



programma in C  
file testo,  
estensione **.c**

IN



programma  
COMPILATORE C

OUT

Esercizio1.c

```
#include <stdio.h>

int Somma(int X, int Y);
```

programma  
LINKER

IN



programma oggetto,  
estensione **.obj**

OUT

programma  
eseguibile,  
estensione  
**.exe**

```
printf("\nLa somma di %d e %d e': %d\n", A, B, S);
```

il programma .exe  
può essere eseguito

Name	Size	Type	Date Modified	Attributes
Esercizio1.obj	4 KB	Object File	07/12/2006 21.44	AC
Esercizio1.pch	172 KB	Precompiled Hea...	07/12/2006 21.44	AC
vc60.idb	33 KB	VC++ Minimum R...	07/12/2006 22.28	AC
vc60.pdb	44 KB	Program Debug ...	07/12/2006 21.44	AC
Esercizio1.exe	161 KB	Application	07/12/2006 22.28	AC
Esercizio1.ilink	168 KB	Incremental Link...	07/12/2006 22.28	AC
Esercizio1.pdb	329 KB	Program Debug ...	07/12/2006 22.28	AC

# Scomponiamo il sorgente

```
#include <stdio.h>
```

Esercizio1.c

```
int Somma(int X, int Y);
```

```
void main()
```

```
{
```

```
    int A, B, S;
```

```
    printf("Inserire il primo valore: ");
```

```
    scanf("%d", &A);
```

```
    printf("Inserire il secondo valore: ");
```

```
    scanf("%d", &B);
```

```
    S = Somma(A, B);
```

```
    printf("\nLa somma di %d e %d e': %d\n", A, B, S);
```

```
}
```

Funzioni.c

```
int Somma(int X, int Y)
```

```
{
```

```
    return X + Y;
```

```
}
```

# Creiamo un file **include**

```
#include <stdio.h>
#include "Funzioni.h"
```

**Esercizio1.c**

```
void main()
{
    int A, B, S;

    printf("Inserire il primo valore: ");
    scanf("%d", &A);

    printf("Inserire il secondo valore: ");
    scanf("%d", &B);

    S = Somma(A, B);

    printf("\nLa somma di %d e %d e': %d\n", A, B, S);
}
```

**Funzioni.c**

```
#include <stdio.h>
#include "Funzioni.h"

int Somma(int X, int Y)
{
    return X + Y;
}
```

**Funzioni.h**

```
int Somma(int X, int Y);
```

# Creiamo un file **include**



```
#include <stdio.h>
#include "Funzioni.h"
```

**Esercizio1.c**

**Funzioni.c**

```
#include <stdio.h>
...
int Y)
```

Name	Size	Type	Date Modified	Attributes
Esercizio1.obj	3 KB	Object File	08/12/2006 0.16	AC
Esercizio1.pch	172 KB	Precompiled Hea...	08/12/2006 0.16	AC
vc60.idb	33 KB	VC++ Minimum R...	08/12/2006 0.26	AC
vc60.pdb	44 KB	Program Debug ...	08/12/2006 0.20	AC
Funzioni.obj	2 KB	Object File	08/12/2006 0.20	AC
Esercizio1.exe	161 KB	Application	08/12/2006 0.26	AC
Esercizio1.ilk	168 KB	Incremental Link...	08/12/2006 0.26	AC
Esercizio1.pdb	337 KB	Program Debug ...	08/12/2006 0.26	AC



```
printf("\nLa somma di %d e %d e': %d\n", A, B, S);
```

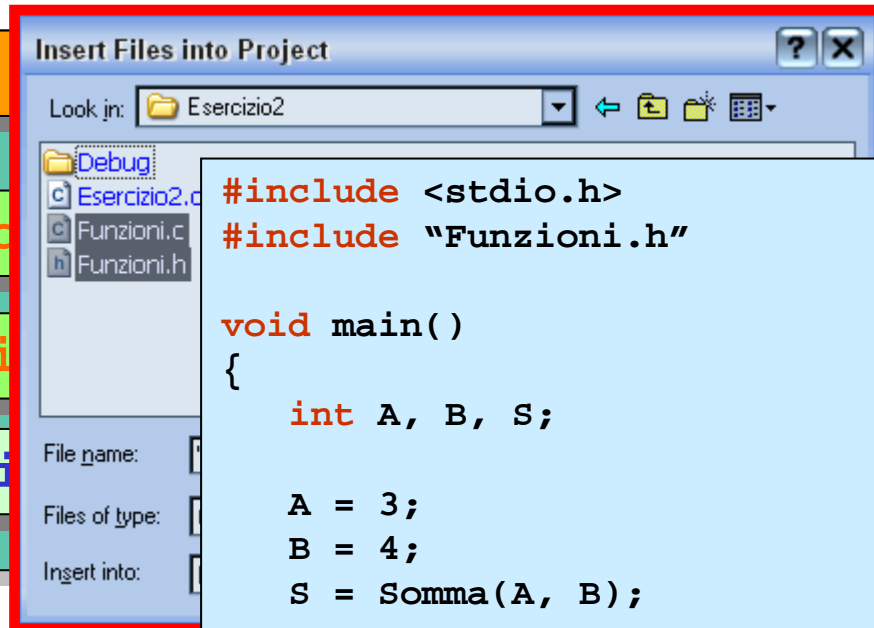
```
int Somma(int X, int Y);
```



# Riutilizzo del software

- tramite i file sorgenti (.c e .h)
- tramite i file oggetto (.obj)
- tramite le librerie
  - Statiche (.lib) (Library)
  - Dinamiche (.dll) (Dynamic Link Library)

# Riutilizzo tramite sorgenti



Progetto2

Esercizio2.c

Main()

Somma()

```
#include <stdio.h>
#include "Funzioni.h"

void main()
{
    int A, B, S;

    A = 3;
    B = 4;
    S = Somma(A, B);

    printf("\nLa somma di %d e %d e': %d\n", A, B, S);
}
```

Soluzione 1

- Copiare i sorgenti (**Funzioni.c** e **Funzioni.h**) nella directory di Progetto2 ed **aggiungerli** al progetto





# Riutilizzo tramite sorgenti

The screenshot shows a Visual Studio IDE with a project named "Progetto3". A source file named "Esercizio3.c" is open, containing the following C code:

```
#include <stdio.h>
#include "Funzioni.h"

void main()
{
    int A, B, S;

    A = 3;
    B = 4;
    S = Somma(A, B);

    printf("\nLa somma di %d e %d e': %d\n", A, B, S);
}
```

The code includes a header file "Funzioni.h" which is not present in the project. A compilation error is shown in the Output window:

```
-----Configuration: Esercizio3 - Win32 Debug-----
Compiling...
Esercizio3.c
c:\programmazione\esercizio\esercizio3.c(2) : fatal error C1083: Cannot open include file: 'Funzioni.h': No such file or directory
Funzioni.c
Error executing cl.exe.

Esercizio3.exe - 1 error(s), 0 warning(s)
```

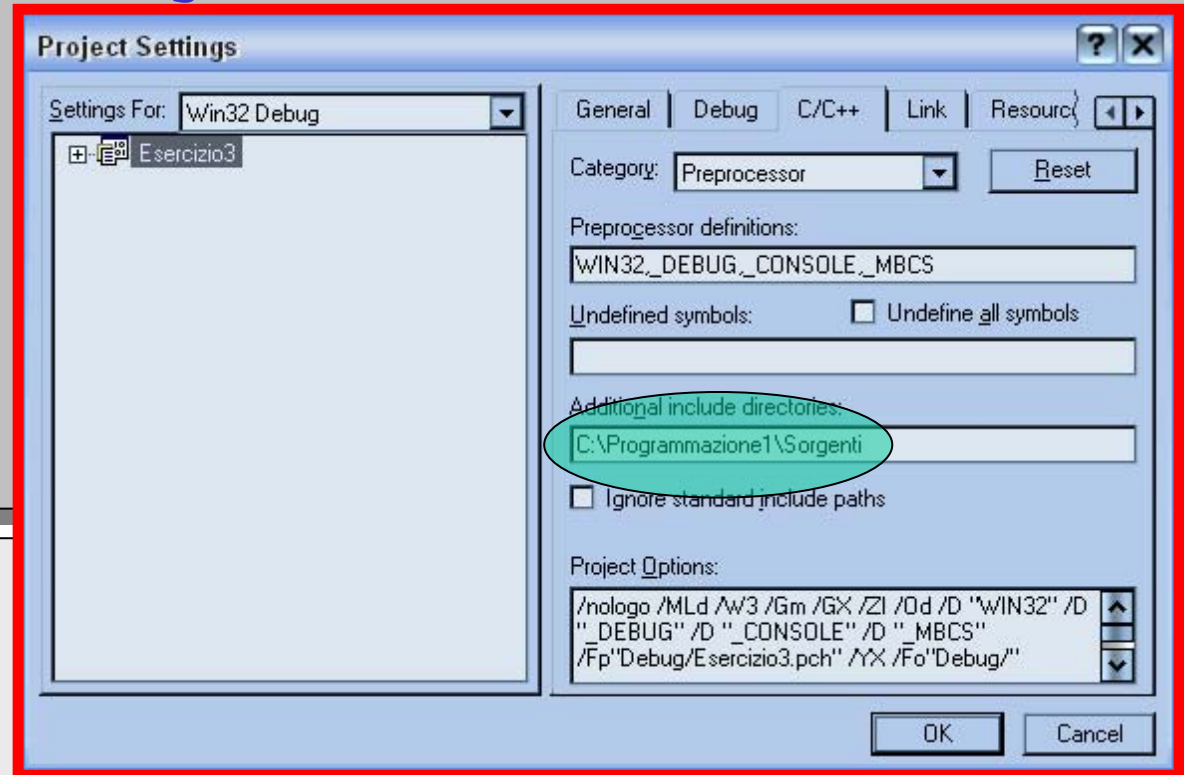
A red box highlights the "Insert Files into Project" dialog box, which is currently empty. Another red box highlights the "ERRORE!!!" message in the Output window. The "Funzioni.h" file is listed in the "Look in:" field of the dialog box.

- Aggiungere i file al progetto



# Riutilizzo tramite sorgenti

Bisogna modificare i **settings** di progetto!  
Dal menù **Projects\Settings**



## Soluzione 2

- Accedere ai file **un'altra** directory

- Creare la directory **C:\Programmazione1\Sorgenti**
- Copiarci i file sorgenti (**Funzioni.c** e **Funzioni.h**)
- Aggiungere i file al progetto



# Riutilizzo tramite sorgenti

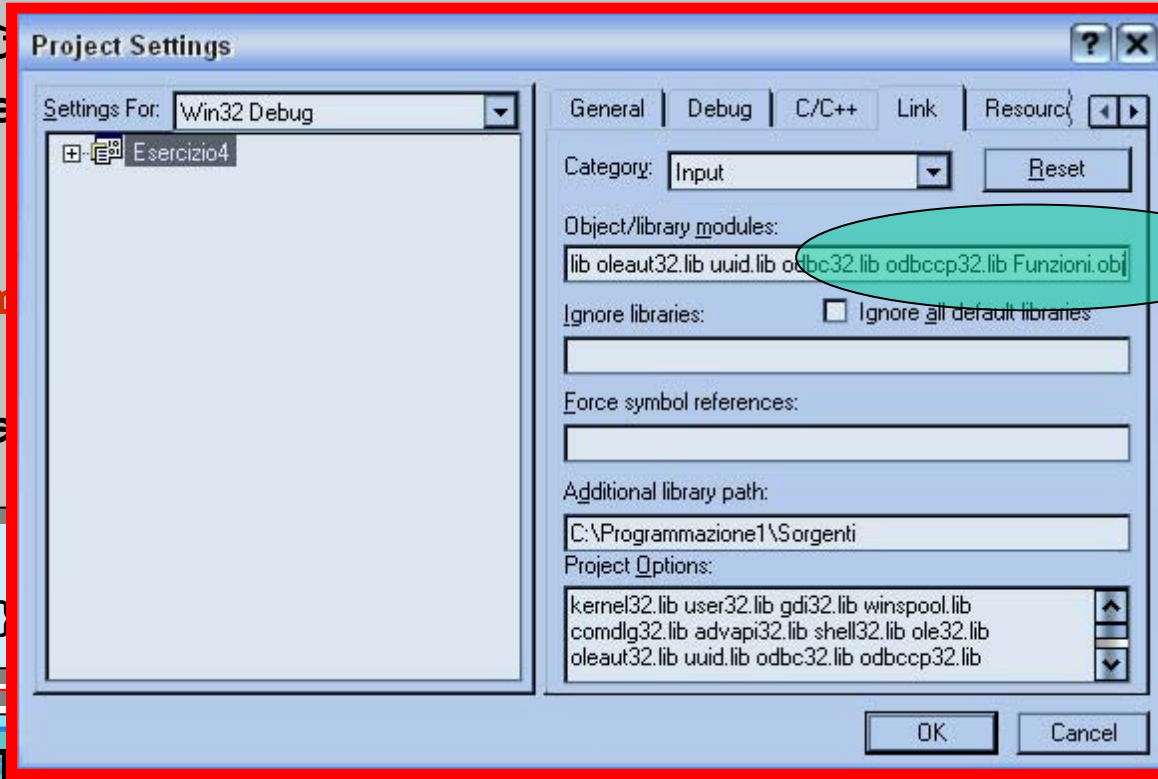
Confronti tra le due soluzioni

- **In entrambi i casi:**
  - vengono prodotti due file **.obj** nella directory Debug
  - il file **Funzioni.c** deve essere **ricompilato** in ogni progetto
  - i sorgenti sono **visibili** ad ogni utilizzatore
  - si lavora a livello del **compilatore**
- **Soluzione1:**
  - c'è una **duplicazione** dei sorgenti che, in presenza di **bug**, richiede la **sostituzione in ogni progetto** dei file **Funzioni.c** e/ o **Funzioni.h**

# Riutilizzo tramite file oggetto

Bisogna modificare i **settings** di progetto!

- Dal menù poter accedere
- Copiare C:\Program
- Modifica



Esercizio1.obj

Funzioni.obj

Esercizio1.LIB

Esercizio1.LIB



# Riutilizzo tramite file oggetto

## Sintesi

- si lavora a livello del **compilatore**
- vengono riutilizzati i file **.obj** prodotti da altri progetti
- il file **Funzioni.c** viene **compilato** solo una volta
- i sorgenti sono **visibili solo** a chi li ha **creati**
- **non c'è duplicazione** dei sorgenti
- **solo**, in presenza di **bug** o modifiche, richiede la **ricompilazione**
- c'è sempre bisogno di un **progetto principale** che **generi** il file oggetto

Ri



he

- creare un nuovo progetto
- aggiungere un file sorgente
- **SENZA** aggiungere librerie
- vuole esportare il progetto
- compilare il progetto
- dalla directory del progetto nella directory di lavoro
- dalla directory di lavoro
- Creare un nuovo progetto con un main
- collegarsi al compilatore sia del compilatore che del linker
- Compilare il progetto

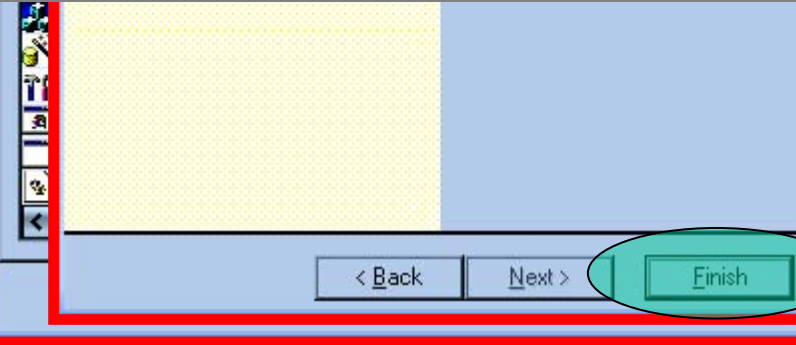
```
#include <stdio.h>
#include "Libreria.h"

void main()
{
    int A, B, S;

    A = 3;
    B = 4;
    S = Somma(A, B);

    printf("\nLa somma di %d e %d e': %d\n", A, B, S);
}
```

Esercizio5.c



```
Libreria.h
che si
c
h>
ia.h"
int Y)
}
```

Libreria.h

```
int Somma(int X, int Y);
```

# Riutilizzo tramite librerie...dinamiche

**DLL**

**Solo per pochi!**

**Dynamic Library**

**Dynamic Link Library**

# Conosciamolo meglio !!

Member of Microsoft Visual Studio 



**Microsoft Visual C++ 6.0**  
Enterprise Edition

**L' IDE di Microsoft**

 **Microsoft**  
**VISUAL**  
**Studio**

**This product is licensed to:**  
Black Skulls  
Pirates

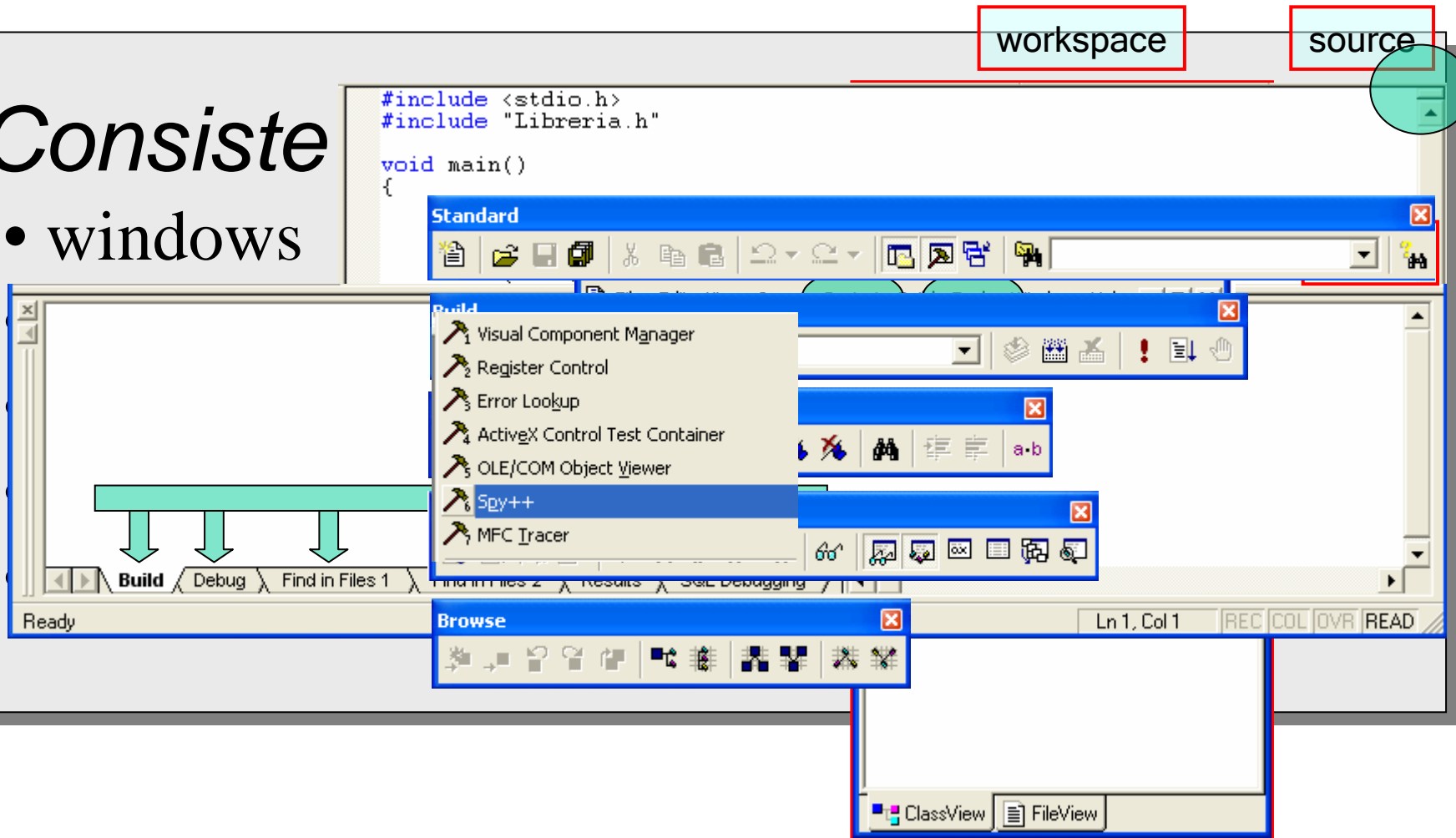
This program is protected by U.S. and international  
copyright laws as described in Help About.  
Copyright 1985-1998 Microsoft Corporation.

**I**ntegrated **D**evelopment **E**nvironment



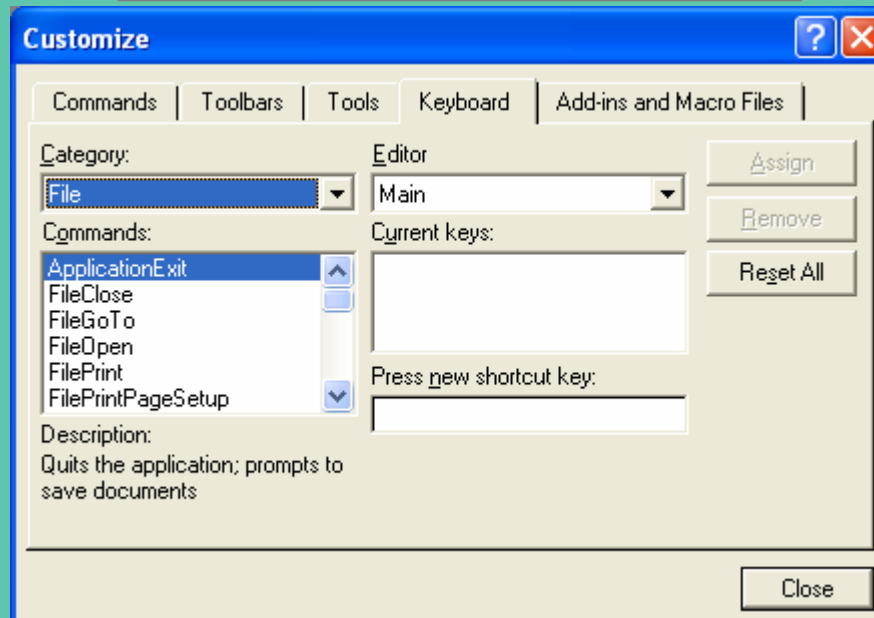
# USER Interface

- *Consiste*
- windows



# USER Interface

**Completamente personalizzabile**



# I menu principali: Project\Settings-Tools\Options

- *Project\Settings*

- Permette la **personalizzazione** del **progetto** attuale

- Debug
- C/C++
- Link
- ...
- Post-build step

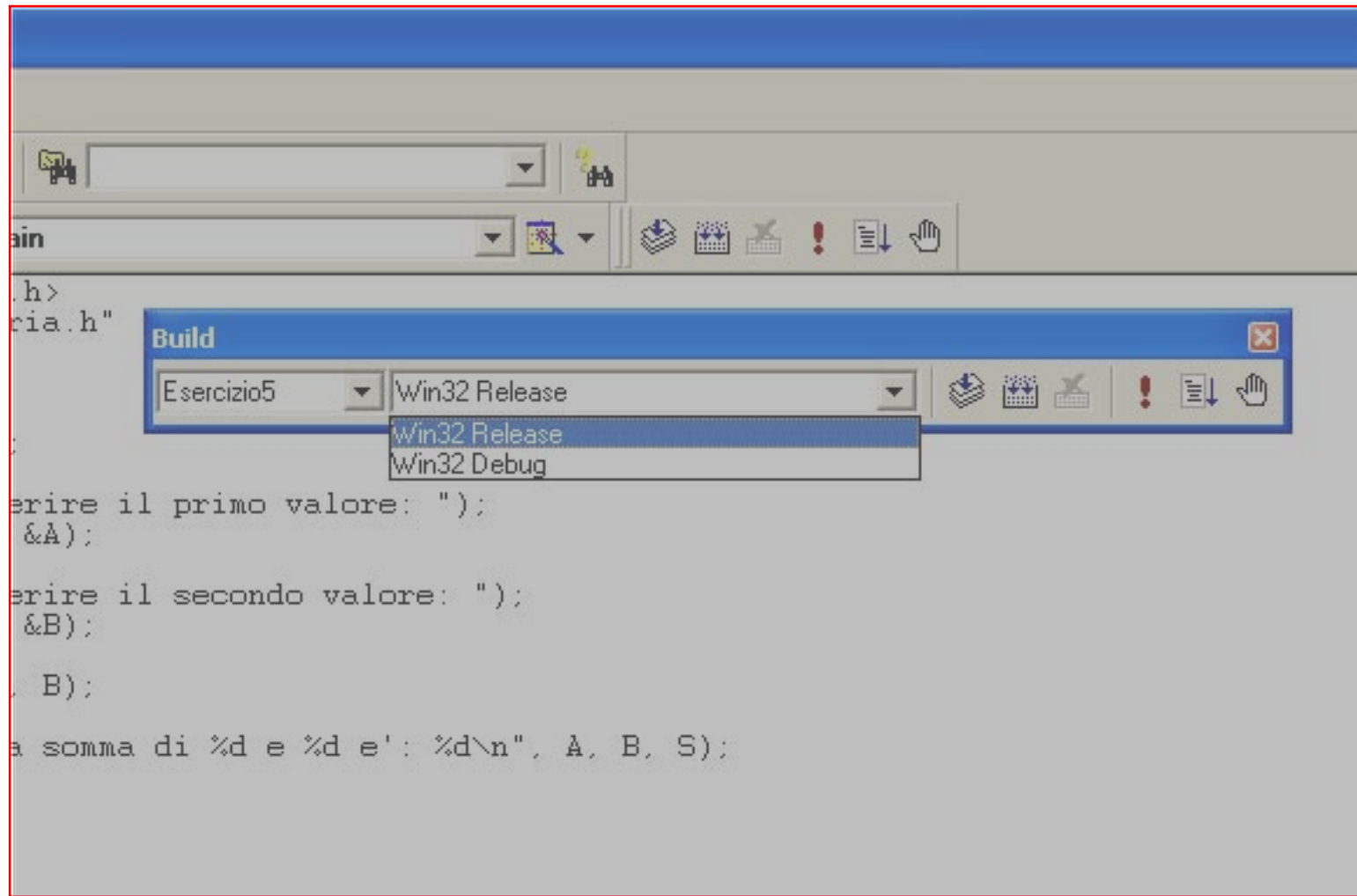
- *Tools\Options*

- Permette la **personalizzazione** dell' **IDE**

Influenza ogni progetto

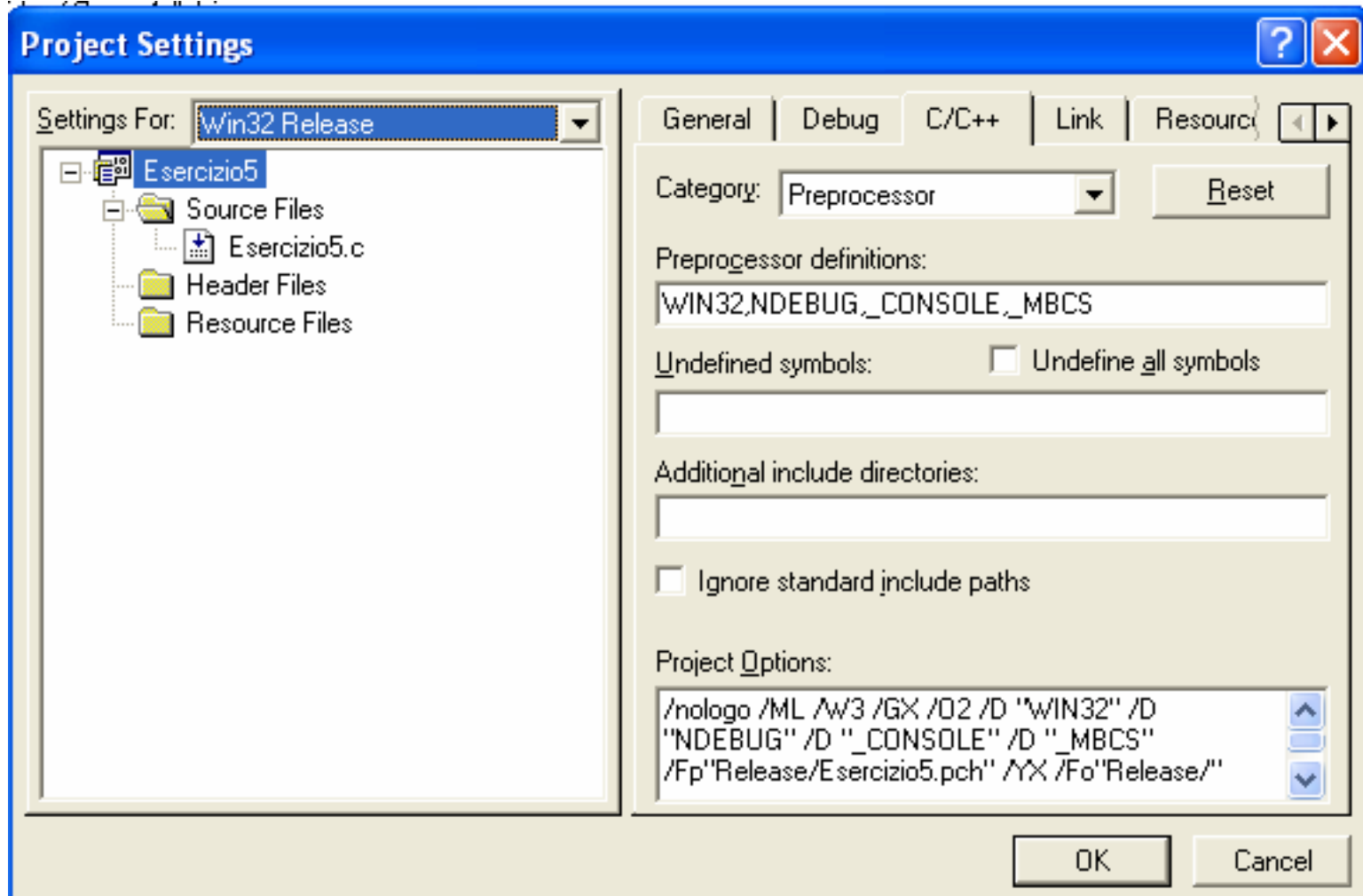
- Editor
- Debug
- Font e colori
- ...
- Directory

# Configurazioni multiple



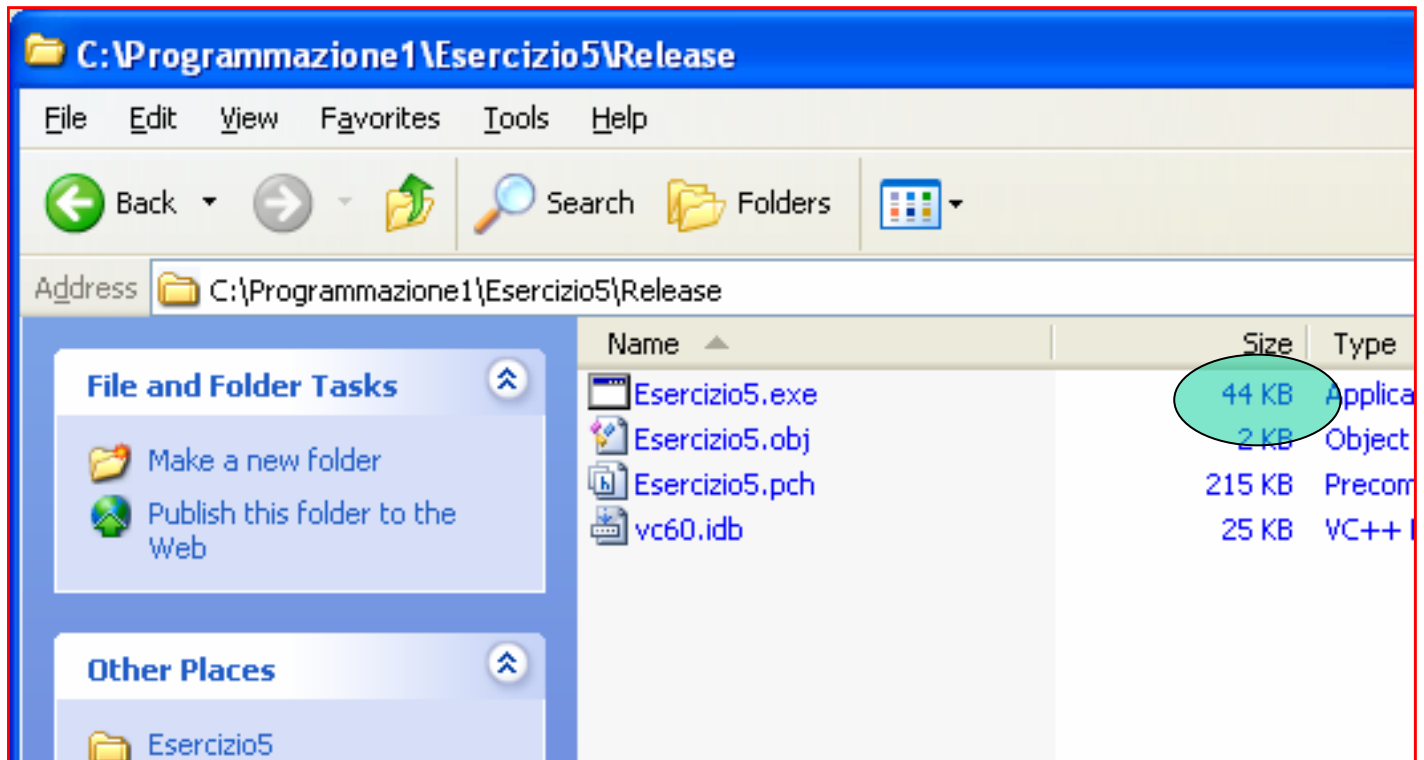
# Configurazioni multiple

Sono indipendenti tra di loro e permettono di generare eseguibili o librerie differenti



# Configurazioni multiple

I risultati della compilazione si trovano, nella directory del progetto, in sottocartelle che hanno il nome della configurazione

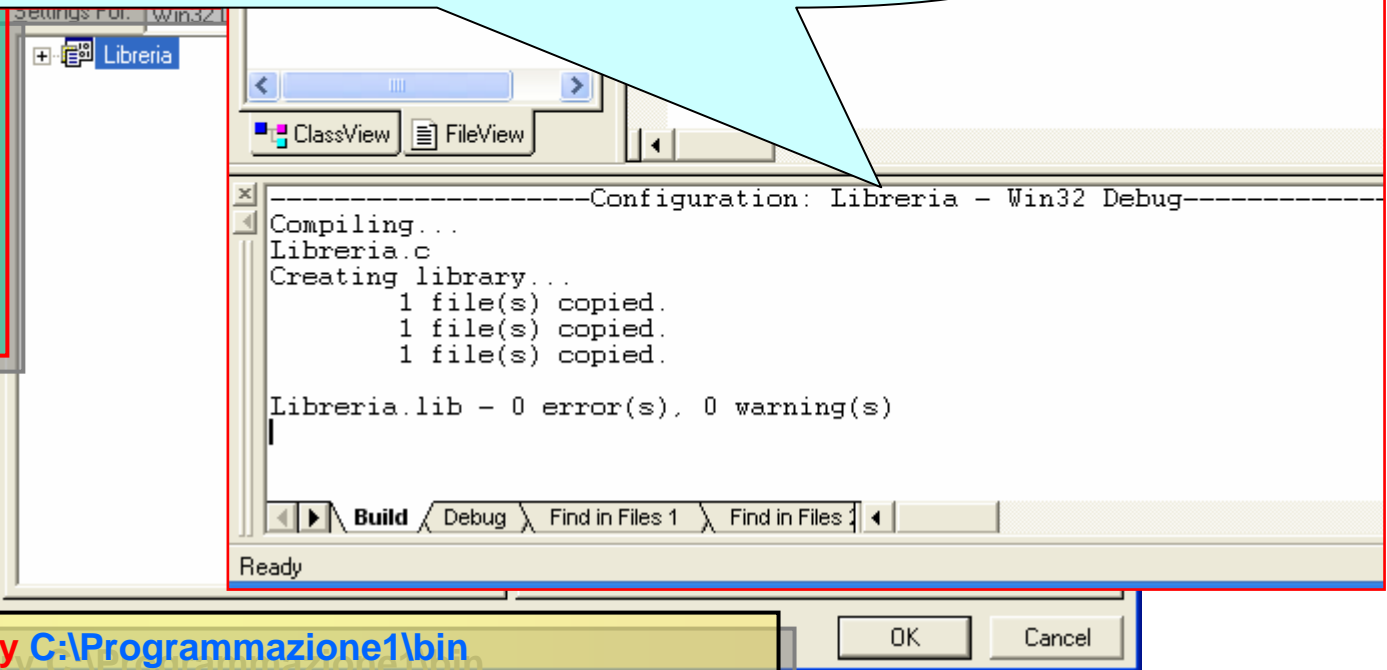


# Configurazioni multiple

È possibile far eseguire, dopo una compilazione avvenuta con successo, alcune operazioni *automaticamente*

```
copy $(TargetPath) C:\Programmazione1\bin
copy $(TargetDir)\$(TargetName).lib C:\Programmazione1\sorgenti
copy libreria.h C:\Programmazione1\Sorgenti
```

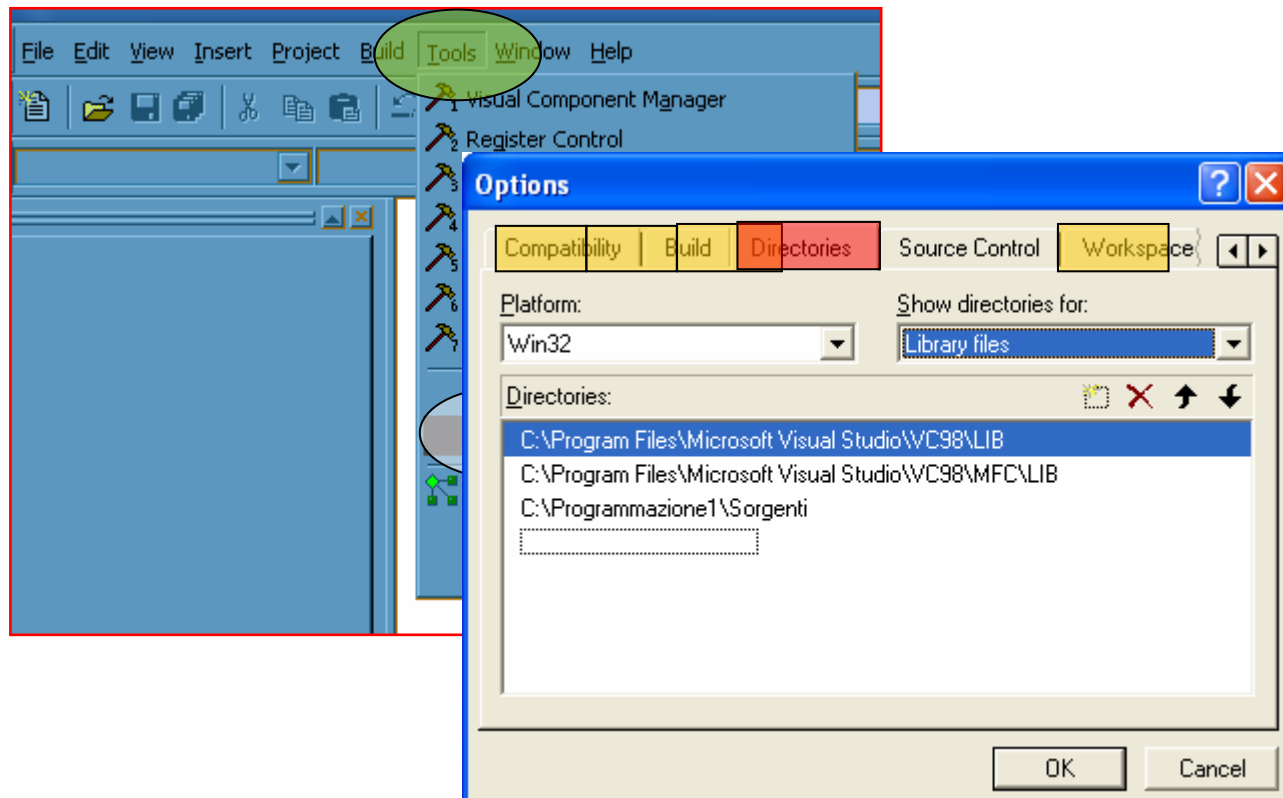
`$(TargetPath)`  
`$(TargetDir)`  
`$(TargetName)`  
`$(TargetExt)`  
`$(TargetArgs)`  
`$(WkspDir)`  
`$(WkspName)`



- Creiamo la directory `C:\Programmazione1\bin`
- Utilizzando *Post-build step* copiare:
  - Libreria.lib nella directory `C:\Programmazione1\bin`
  - Libreria.h nella directory `C:\Programmazione1\Sorgenti`

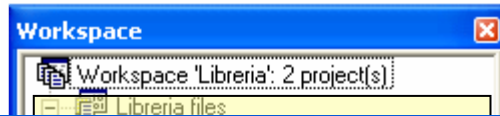


# Le options dell'IDE

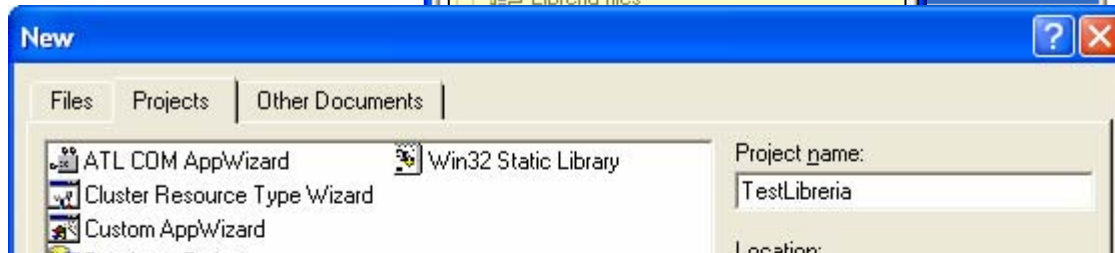




# Progetti multipli

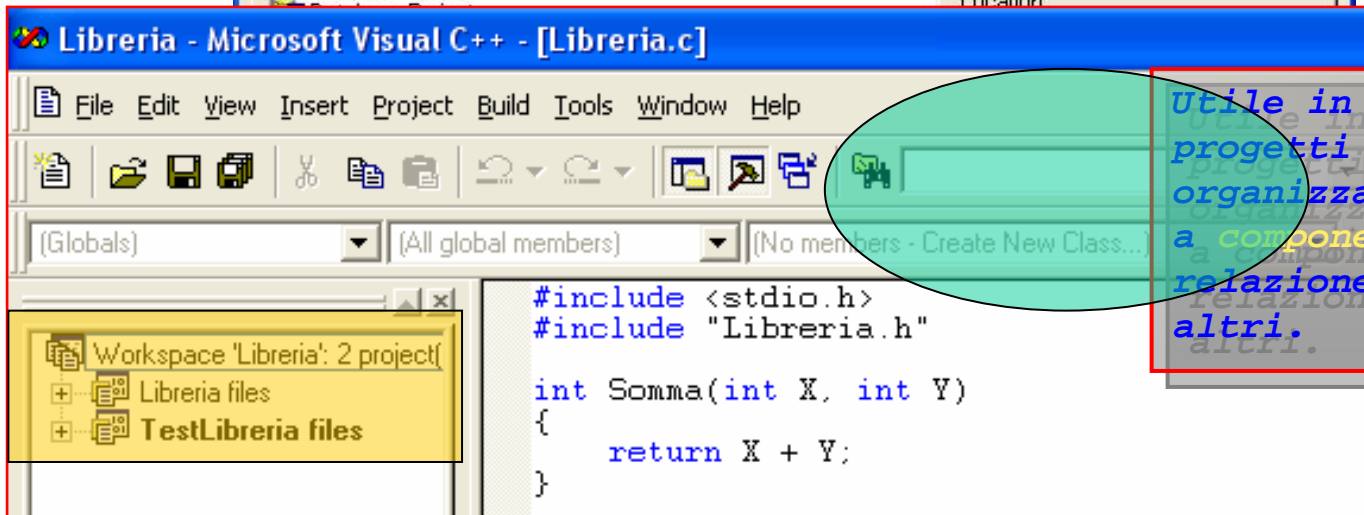


Unico workspace



rimo progetto

econdo progetto



Utile in presenza di  
progetti complessi con  
organizzazione modulare ed  
a componenti fortemente in  
relazione gli uni con gli  
altri.

