

# Dev C++

## Note di utilizzo

M. Malatesta - Dev C++ Note di utilizzo-08

1  
03/11/2010

## Prerequisiti

- Utilizzo di Windows
- Effettuare il download da Internet
- Compilazione di un programma

M. Malatesta - Dev C++ Note di utilizzo-08

2  
03/11/2010

# Introduzione

Lo scopo di queste note è quello di diffondere la conoscenza di **Dev C++**, ambiente di sviluppo integrato (**IDE**, *Integrated Development Environment*) per i linguaggi C/C++, disponibile sia per Windows, che per Linux.

E' possibile creare applicazioni con finestre grafiche o con modalità testuale (es. finestra MS-DOS).

Si danno le caratteristiche fondamentali per:

- Installare
- Configurare

**Dev C++** per usarlo e per realizzare applicazioni.

# Il DevC++

**DevC++** è un **IDE** che funziona da interfaccia (*front-end*) verso il compilatore GCC che è pienamente compatibile con lo standard C89 o ANSI C.

Grazie alla sua interfaccia grafica, un ambiente IDE consente di utilizzare in modo semplice gli strumenti offerti dal compilatore senza dover ricorrere necessariamente alle istruzioni da riga di comando (command-line).

La versione testata oggi disponibile è la 4.9.9.2 che può essere scaricata liberamente dai seguenti siti:

## Dove reperirlo...

- Documentazione per **Dev C++**  
[http://www.bloodshed.net/dev/docs/doc\\_tutorials.html](http://www.bloodshed.net/dev/docs/doc_tutorials.html)
- Programmazione C++  
[http://www.bloodshed.net/dev/docs\\_c\\_cpp.html](http://www.bloodshed.net/dev/docs_c_cpp.html)
- Programmazione Windows (Win 32)  
[http://www.bloodshed.net/dev/docs/doc\\_win32.html](http://www.bloodshed.net/dev/docs/doc_win32.html)
- **Dev C++** si può scaricare anche come file .zip dal sito  
<http://www.bloodshed.net/devcpp.html>
- Sito da cui scaricare l'IDE e il compilatore:
- <http://sourceforge.net/projects/dev-cpp/>

M. Malatesta - Dev C++ Note di utilizzo-08

5  
03/11/2010

## Dove reperirlo...

Finestra informazioni in  
**DevC++**  
(versione **4.9.8.10**)



M. Malatesta - Dev C++ Note di utilizzo-08

6  
03/11/2010

# Requisiti di sistema

Per poter eseguire **DevC++** sono necessari i seguenti requisiti minimi:

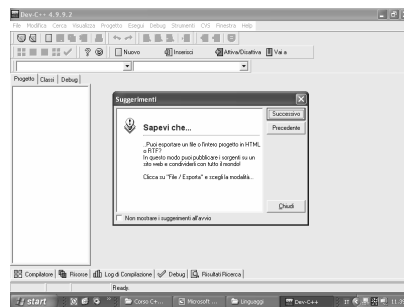
- **Sistema Operativo:** Windows 98/2000/ME/NT, XP, Vista
- **RAM:** minimo 8Mb (consigliati almeno 32 Mb)
- **CPU:** Intel Pentium 100 MHz (consigliati almeno 233 MHz)
- **Spazio su disco:** 30 MB (consigliati almeno 45 Mb)

# Procedura di installazione

- Creare una cartella temporanea (*temp*) in cui porre il **file di installazione** scaricato **devcpp-Xsetup.exe** (X sta ad indicare la versione, ad esempio devcpp-4.9.9.2\_nomingw\_setup.exe)
- Eseguire il **file di installazione** e seguire le indicazioni della procedura guidata, accettando, possibilmente, le scelte consigliate (si consiglia l'installazione in C:\Dev-Cpp)
- Cancellare la cartella temporanea *temp*.
- Creare sul *desktop*, se lo si desidera, una icona di collegamento al file C:\.....\devcpp.exe per facilitarne il successivo utilizzo

# Configurazione di DevC++

Lanciando **DevC++** si ottiene un ambiente con un aspetto simile a quello mostrato. La finestra centrale offre spiegazioni su alcune particolarità di **DevC++** che possono essere molto utili.



M. Malatesta - Dev C++ Note di utilizzo-08

9  
03/11/2010

# Configurare il compilatore

Selezionare il menu **Strumenti** e l'opzione **Opzioni di compilazione**

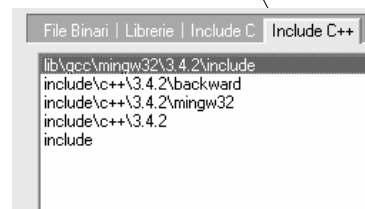
- Selezionare la scheda **Compilatore**, selezionare con un segno di spunta la casella **Aggiungi i comandi seguenti in fase di compilazione** e nella casella relativa digitare  
“-Wall”
- Nella scheda **Generazione del codice** selezionare la voce **Linker**. Assicurarsi che la voce **Genera le informazioni per il debug** sia impostata a **Yes** (in caso contrario impostarla)

M. Malatesta - Dev C++ Note di utilizzo-08

10  
03/11/2010

# Configurare il compilatore

- Sempre nella scheda **Generazione del codice**, selezionare la voce **Ottimizzazione** ed aprire il sottomenu relativo facendo click sul "+". Selezionare **Ulteriori ottimizzazioni** ed assicurarsi che la voce **Ottimizzazione migliore** sia impostata a **Yes** (in caso contrario impostarla)
- Aprire la scheda **Cartelle** e verificare (nel caso aggiungere) che nella linguetta:
  - **File binari** compaia **C:\Dev-Cpp\Bin**
  - **Librerie** compaia **lib**
  - **Include C** compaia **include**
  - **Include C++** compaiano le scritte indicate a fianco**Si ricorda che DevC++ deve essere installato in C:\Dev-Cpp**



M. Malatesta - Dev C++ Note di utilizzo-08

11  
03/11/2010

# Configurare il compilatore

- Nella scheda **Programmi** deve apparire quanto segue:



- Fare click su **OK** per confermare le impostazioni immesse.

M. Malatesta - Dev C++ Note di utilizzo-08

12  
03/11/2010

# Configurare l'ambiente

- Selezionare il menu **Strumenti** e l'opzione **Opzioni di ambiente**
- Selezionare la scheda **File & Cartelle**
- Impostare nella casella **Cartella predefinita dell'utente** il percorso dove si scriveranno i propri programmi in C++. I percorsi vanno indicati nelle caselle di testo in uno dei seguenti modi:
  - digitando in questa casella il percorso della cartella richiesta
  - navigando sul disco mediante il pulsante di destra per individuare la cartella richiesta

M. Malatesta - Dev C++ Note di utilizzo-08

13  
03/11/2010

# Applicazioni con DevC++

**DevC++** comprende alcuni schemi standard (**template**) che contengono lo scheletro già predisposto per creare:

- **Applicazioni DOS**, ossia programmi con i quali l'utente può interagire mediante la sola modalità testuale (finestra DOS)
- **Applicazioni Windows**, ossia programmi facenti uso di grafica e che interagiscono con l'utente mediante finestre, icone, ecc, tipiche dell'ambiente Windows (*non contenuto in queste note*)

Normalmente, dopo la prima installazione, sono presenti i primi 5 template



M. Malatesta - Dev C++ Note di utilizzo-08

14  
03/11/2010

# Applicazioni DOS

Per creare un'applicazione DOS occorre svolgere le seguenti fasi:

- 1. Creare un nuovo progetto.** Un progetto è una specie di contenitore che conterrà tutti gli elementi richiesti per compilare un programma
- 2. Scrivere il codice sorgente** del programma in C++
- 3. Compilare e linkare** il codice sorgente. In caso di errori tornare al passo 2
- 4. Eseguire l'applicazione.** Se sono presenti errori, correggerli tornando al passo 2

# Applicazioni DOS

## 1. Creare un nuovo progetto

Per creare un nuovo progetto

1. Selezionare **File/New Project** e scegliere **Console Application**
2. Assicurarsi che sia impostato il pulsante di opzione su **C++**
3. Digitare il nome del progetto e confermare facendo click su **OK**
4. Creare una nuova cartella assegnandole un nome (tipicamente lo stesso assegnato al progetto). E' bene creare una cartella separata per ciascun progetto



# Applicazioni DOS

## 1. Creare un nuovo progetto

5. Il **DevC++** suggerisce il nome **main.cpp** del programma principale del progetto. Confermare, o modificarlo, e fare click su **Salva**.
6. Viene salvato il file **main.cpp** avente la seguente struttura (**template per tutte le applicazioni DOS**)

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main (int argc, char *argv[])
{
    system ("PAUSE");
    return EXIT_SUCCESS;
}
```

M. Malatesta - Dev C++ Note di utilizzo-08

17  
03/11/2010

# Applicazioni DOS

## 2. Scrivere il codice sorgente

1. Modificare il sorgente standard come segue:

```
#include <cstdlib>
#include <iostream>
using namespace std;
int main (int argc, char *argv[] )
{
    int a,b;
    cout<<"Immetti il primo numero:"<<endl;
    cin>>a;
    cout<<"Immetti il secondo numero:"<<endl;
    cin>>b;
    cout<<"La somma e' "<<a+b<<endl;
    system ("PAUSE");
    return EXIT_SUCCESS;
}
```

2. Salvare il codice

M. Malatesta - Dev C++ Note di utilizzo-08

18  
03/11/2010

# Applicazioni DOS

## 3. Compilare e linkare

1. Selezionare **Esegui/Compila (Ctrl F9)**.
2. Se il sistema segnala errori nella finestra in basso, procedere alla loro correzione e ripetere il salvataggio e la compilazione
3. Chiudere la finestra di compilazione

# Applicazioni DOS

## 4. Eseguire l'applicazione

1. Selezionare **Esegui/Esegui** o **Ctrl F10** per eseguire il programma.  
L'esecuzione può anche essere invocata con doppio click sull'icona del file eseguibile creato da **DevC++**.

Per eseguire una applicazione utilizzando i parametri della linea di comando:

- selezionare il menu **Esegui**
- scegliere l'opzione **Parametri**
- immettere, separandoli con spazi, i parametri della linea di comando

## La libreria **conio.h** (Borland®)

Per usare le funzioni tipiche della libreria **conio.h** TurboC della **Borland** © è necessario:

- Disporre dei file **libconio.a** e **conio.h** (reperendo su Internet il file `winbgim_devcpp4990_20041220.zip`)
- Copiare il file **libconio.a** nella cartella `lib` di Dev-Cpp (in base al percorso ad es. `C:\Dev-Cpp\lib`)
- Copiare il file **conio.h** nella cartella `include` di Dev-Cpp (ad es. `C:\Dev-Cpp\include`)
- Configurare il compilatore e il linker come indicato di seguito

## La libreria **conio.h** (Borland®)

- In **Strumenti** selezionare **Opzioni di compilazione**
- Nella casella "Aggiungi questi comandi alla linea di comando del **Linker**"
  - Selezionare la casella di spunta
  - Scrivere nella casella di testo `C:\Dev-Cpp\lib\libconio.a` (supponendo il DevCpp installato in `C:\Dev-Cpp`)
  - Digitare **OK**
- Includere nei propri file la libreria **conio.h**
- A questo punto dovrebbero essere disponibili tutte le funzioni della libreria **conio.h** del TurboC

## La libreria **winbgim.h** (Borland<sup>©</sup>)

Il package `winbgim_devcpp4990_20041220.zip` utilizzato per installare la libreria **conio.h** contiene anche due importanti librerie per l'utilizzo delle funzionalità grafiche di **DevC++**.

- **libbgi.a**
- **winbgim.h**

Installando queste librerie, si può predisporre lo schermo in modalità grafica per tracciare grafici e disegni.

- Copiare il file **libbgi.a** nella cartella lib di Dev-Cpp (ad es. `C:\Dev-Cpp\lib`)
- Copiare il file **winbgim.h** nella cartella include di Dev-Cpp (ad es. `C:\Dev-Cpp\include`)

M. Malatesta - Dev C++ Note di utilizzo-08

23  
03/11/2010

## La libreria **winbgim.h** (Borland<sup>©</sup>)

- In **Strumenti** selezionare **Opzioni di compilazione**
- Nella casella "Aggiungi questi comandi alla linea di comando del **Linker**"
  - Selezionare la casella di spunta
  - Scrivere nella casella di testo **-lbgi -lgdi32 -luser32** (attenzione ai segni "-" davanti ai nomi)
  - Digitare **OK**
- Includere nei propri file la libreria **winbgim.h**
- A questo punto dovrebbero essere disponibili tutte le funzioni della libreria **winbgim.h** del TurboC

M. Malatesta - Dev C++ Note di utilizzo-08

24  
03/11/2010

## La libreria **winbgim.h** (Borland<sup>©</sup>)

Per utilizzare in modo efficace la grafica, si può (ma non è indispensabile) creare un apposito **template**, in modo da disporre già di uno scheletro iniziale per le applicazioni grafiche.

Per creare il template per la grafica occorre:

- posizionarsi nella directory **C:\Dev-Cpp\Templates**
- copiare il file "2-ConsoleApp.template" e rinominarlo come "6-ConsoleAppGraphics.template"
- mediante un text editor (blocco note o altro) applicare le seguenti modifiche al file "6-ConsoleAppGraphics.template"

## La libreria **winbgim.h** (Borland<sup>©</sup>)

- nella sezione [**Template**], sostituire
  - *Name=Console Application* con *Name=Console Graphics Application*
  - *Description=A console application (MSDOS window)* con *Description=A console application (MSDOS window) with graphics*
- nella sezione [**Unit0**], sostituire
  - *Cpp=consoleapp\_cpp.txt* con *Cpp=consoleapp\_cpp\_gr.txt*
- nella sezione [**Project**], aggiungere la seguente riga:
  - *Linker=-lbgdi -lgdi32 -luser32*  
("l" è la lettera "l" minuscola e non il numero "1")
- uscire e salvare il file

## La libreria **winbgim.h** (Borland<sup>©</sup>)

- sempre nella directory **C:\Dev-Cpp\Templates**, mediante un test editor (Blocco note) creare il file "*consoleapp\_cpp\_gr.txt*" nel quale caricare il template riportato a fianco.
- A questo punto quando si crea un nuovo progetto, sarà possibile utilizzare questo template se si vuole creare una applicazione grafica

```
/* Titolo: */
/* File..: */
/* Data..: */
/* Autore: */
#include <cstdlib>
#include <iostream>
#include <winbgim.h>
using namespace std;
int main(int argc, char *argv[])
{   initwindow (400,300);
    while (!kbhit());
    closegraph();
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

M. Malatesta - Dev C++ Note di utilizzo-08

27  
03/11/2010

## Creazione di nuovi moduli

E' possibile aggiungere **nuovi moduli** al progetto corrente in due modi:

- 1a. Selezionare **Progetto/Nuova Unità**.
- 1b. Selezionare **File/Nuovo file sorgente**

Successivamente:

2. Scrivere il codice sorgente relativo e salvare.
3. Compilare (Ctrl F9). **DevC++** automaticamente compila sempre tutte le unità presenti in un progetto

M. Malatesta - Dev C++ Note di utilizzo-08

28  
03/11/2010

# Aggiunta di moduli esistenti

E' possibile aggiungere **moduli preesistenti** al progetto corrente in due modi:

- 1a. Selezionare **Progetto/Aggiungi al progetto**
- 1b. Fare click con tasto destro sul nome del progetto (finestra di sinistra) e selezionare **Aggiungi al progetto**
  
2. Compilare (Ctrl F9). DevC++ automaticamente compila sempre tutte le unità presenti in un progetto

# Argomenti

- Il DevC++
- Dove reperirlo...
- Requisiti di sistema
- Procedura di installazione
- Configurazione di **DevC++**
- Configurare il compilatore
- Configurare l'ambiente
- Applicazioni con **DevC++**
- Applicazioni DOS
  1. Creare un nuovo progetto
  2. Scrivere il codice sorgente
- 3. Compilare e linkare
- 4. Eseguire l'applicazione
- La libreria **conio.h** (Borland®)
- La libreria **winbgim.h** (Borland®)
- Creazione di nuovi moduli
- Aggiunta di moduli esistenti