

Guida pratica all'uso di *SharpDevelop*

SharpDevelop è uno strumento di sviluppo integrato e visuale (IDE, *Integrated Development Environment*) rilasciato con licenza GNU-GPL, quindi utilizzabile liberamente.

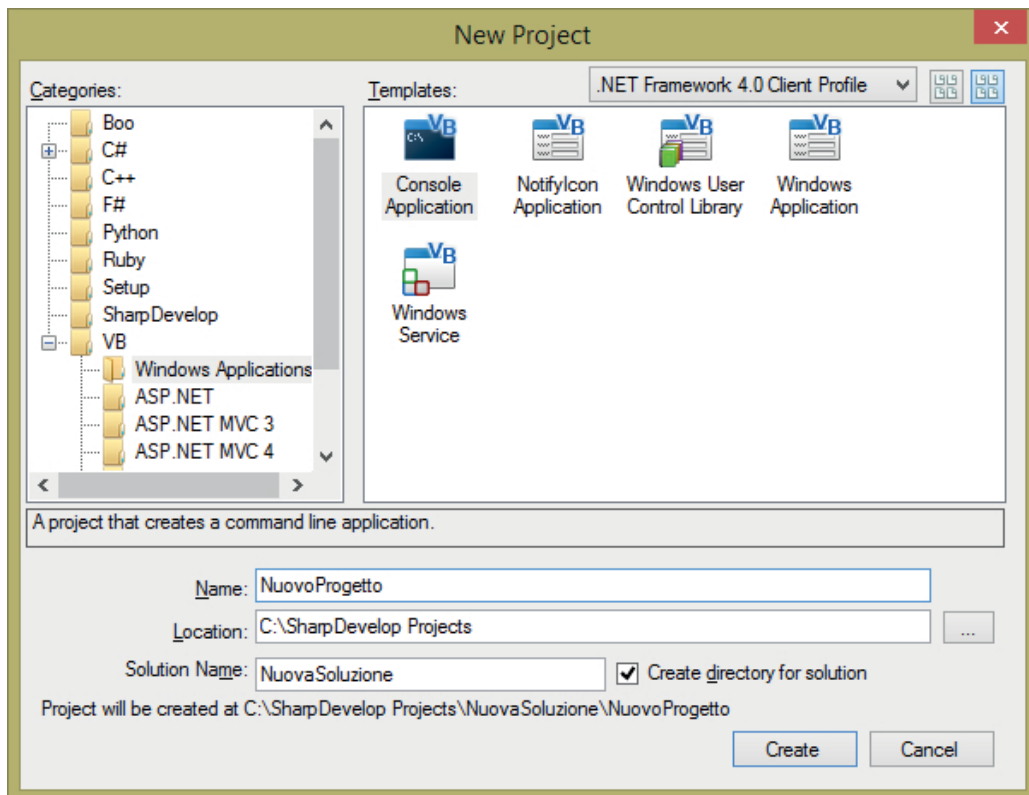
È un'alternativa all'ambiente *Microsoft Visual Studio*, che è l'IDE di riferimento per lo sviluppo di applicazioni in ambiente Microsoft.

La modalità di utilizzo è analoga a quella della versione Microsoft, quindi ha le seguenti caratteristiche:

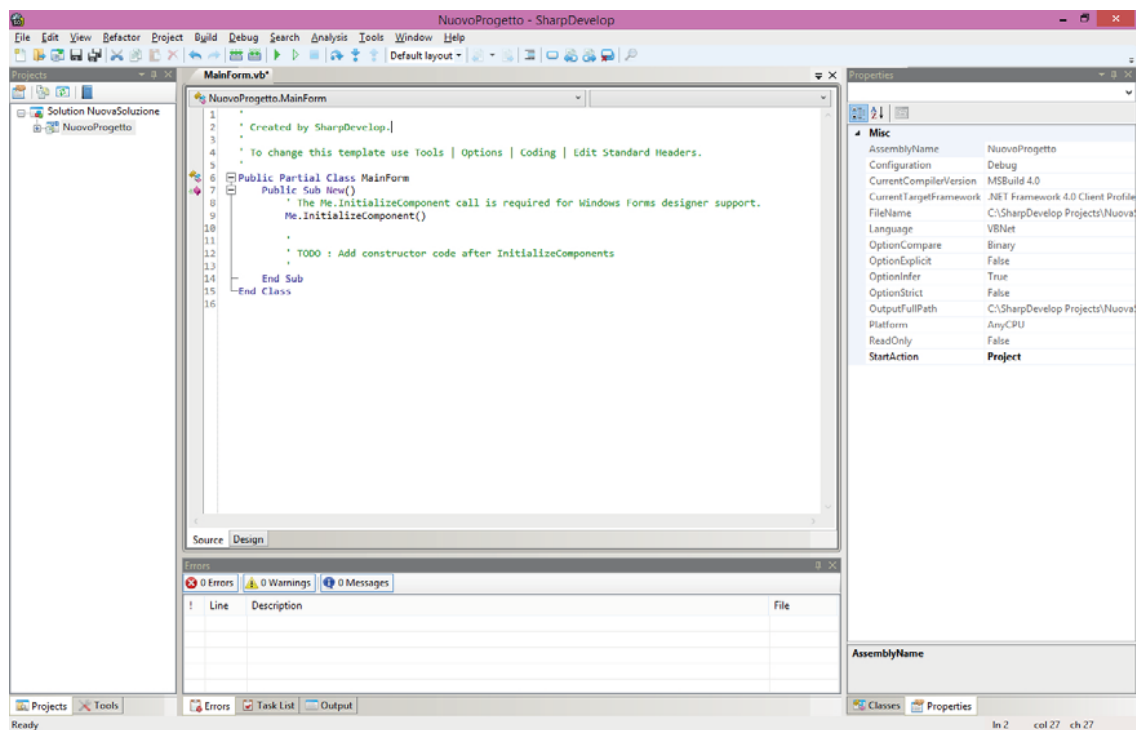
- **Soluzione:** la soluzione è il contenitore di progetti che formano un programma completo. Le soluzioni create con *SharpDevelop* possono essere *compilate* e *generate* anche da strumenti Microsoft, quindi sono *portabili*.
- **Progetto:** è l'unità minima che produce un risultato compilato ed eseguibile. Ad una soluzione possono essere aggiunti progetti di varie tipologie. A partire dalla versione 4.4, è possibile aggiungere le seguenti tipologie di progetto:
 - *Applicazioni Windows:* è possibile creare progetti per applicazioni Windows *a console*, *a finestre* o *servizi*.
 - *Applicazioni Web:* è possibile creare applicazioni Web con *ASP.NET*, utilizzando il nuovo framework *ASP.NET MVC* o creare *servizi Web* sia con tecnologia *ASP.NET* che con il nuovo framework *WCF*.

I progetti possono essere creati utilizzando tutti i linguaggi supportati dal framework *Microsoft.NET*, in particolare C# (si legge *C sharp*), che è il linguaggio ufficialmente supportato per la piattaforma, oltre a Visual Basic (spesso indicato anche con VB.NET).

Per creare un progetto è necessario creare prima una soluzione; dal menu *File > New > Solution* è possibile attivare la schermata di selezione della tipologia di progetto.



Nella fase di progettazione, la finestra principale di *SharpDevelop* si presenta come in figura:



I pannelli hanno le seguenti funzioni:

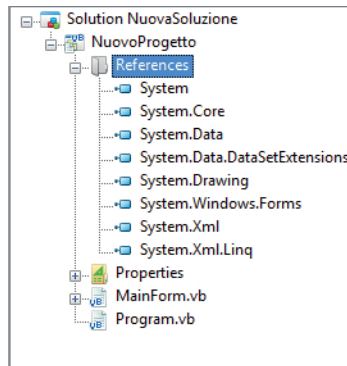
- Pannello *Projects*: consente la selezione dei file da includere o meno nel progetto e ne consente l'apertura nell'Editor principale del programma. Facendo clic con il tasto destro del mouse sugli elementi nel pannello è possibile attivare le funzioni principali su questi elementi come la compilazione, il debug, l'apertura o la modifica dei riferimenti.
- Pannello *Properties*: visualizza le proprietà dell'elemento correntemente selezionato e consente di impostarne i valori.
- Pannello inferiore: contiene la scheda *Errors*, che ad ogni compilazione riporta l'elenco di tutti gli errori sintattici presenti nell'intera soluzione.
- Editor principale: può visualizzare il contenuto di diversi file sorgenti contemporaneamente, suddivisi in schede. Ogni file sorgente può essere visualizzato in diverse modalità:
 - *Design*: se il componente lo prevede, *SharpDevelop* è in grado di attivarne la progettazione in modalità visuale.
 - *Source*: consente di visualizzare il codice sorgente memorizzato in un file.



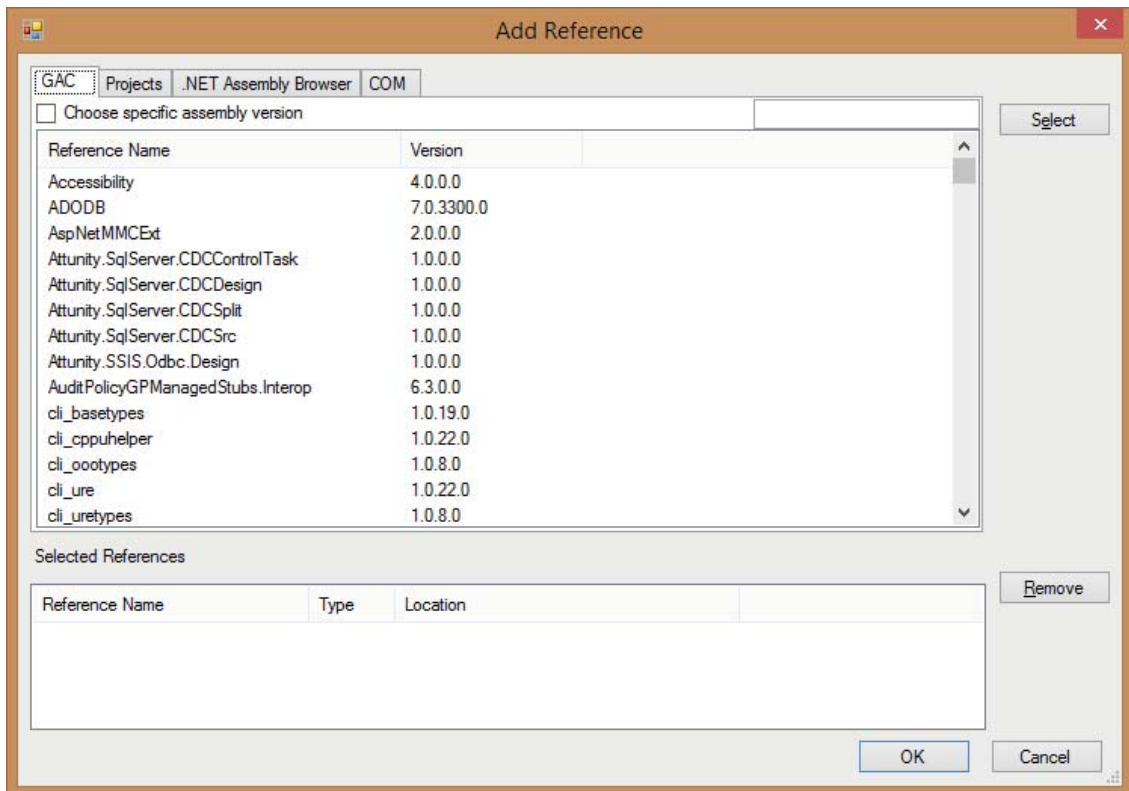
- Barra dei pulsanti di *default*: è la barra degli strumenti visualizzata nella modalità di funzionamento predefinita e riporta i comandi per svolgere le funzioni più frequenti; tra queste:
 - compilazione di tutti i progetti o del progetto correntemente selezionato.
 - avvio in modalità debug, senza debug e arresto del programma di debug. L'avvio del programma, ne comporta anche la compilazione.

Ogni progetto in *SharpDevelop* come in *Microsoft Visual Studio*, per essere compilato ha bisogno di definire i riferimenti alle librerie o ai moduli utilizzati, che siano presenti in altri progetti della stessa soluzione o installati nel sistema in uso.

Questa configurazione è consentita dalla funzione *References* visualizzata nel pannello *Projects* come una sottodirectory virtuale di ogni progetto.

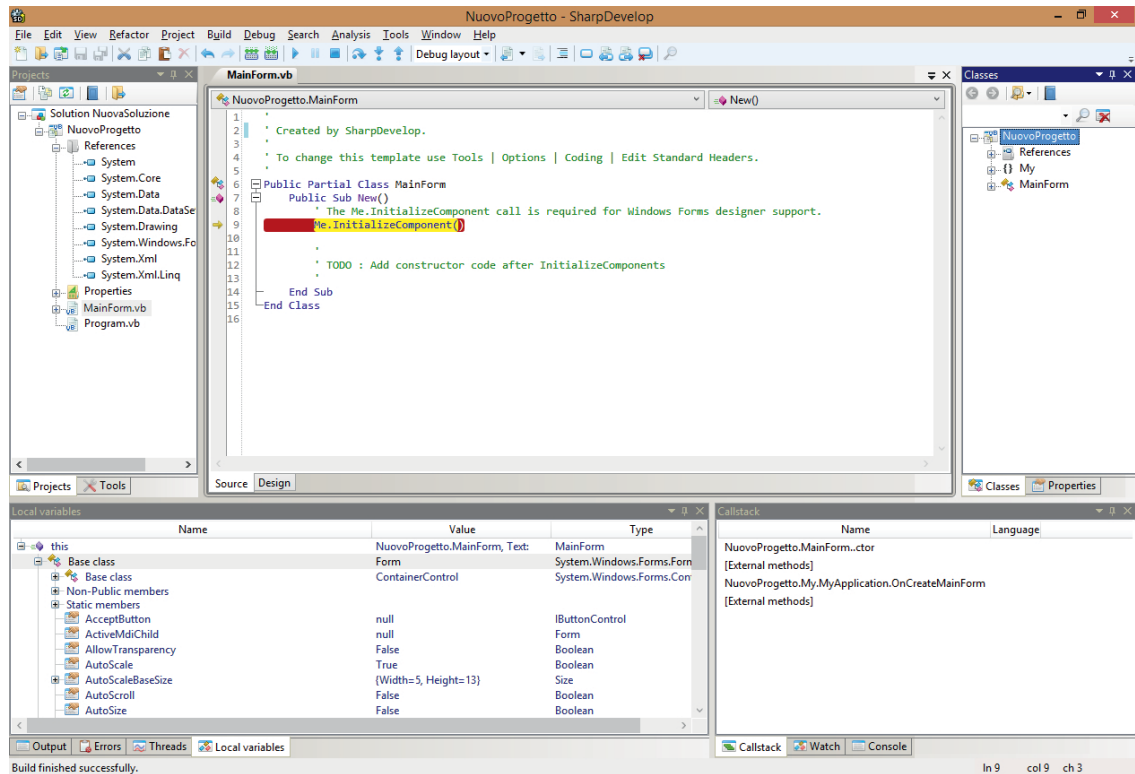


Facendo clic con il tasto destro sulla sottodirectory virtuale *References*, viene attivato il pannello di gestione dei riferimenti a librerie esterne al progetto, come si vede in figura.



Durante la compilazione dei progetti, *SharpDevelop* tiene conto di tutti i riferimenti espressi nella directory virtuale *References*, oltre che del codice sorgente scritto nei vari file del progetto.

Quando un progetto è avviato in modalità di debug, *SharpDevelop* attiva automaticamente la visualizzazione di debug, mostrata nella seguente figura:



Nell'area in basso, lo spazio di lavoro è stato arricchito di ulteriori pannelli utili nell'attività di debug, tra cui:

- Pannello *Local variables*: mostra le variabili valide nel contesto corrente, quello in cui il codice è interrotto nel punto selezionato tramite un *Punto di interruzione*, con i relativi valori.
- Pannello *Call stack*: indica qual è la sequenza delle chiamate che ha portato l'esecuzione nel punto di interruzione correntemente attivato.