C# - Dynamic Type

C# 4.0 (.NET 4.5) introduced a new type that avoids compile time type checking. You have learned about the implicitly typed variable- [var](http://www.tutorialsteacher.com/csharp/csharp-var-implicit-typed-local-variable) in the previous section where the compiler assigns a specific type based on the value of the expression. A dynamic type escapes type checking at compile time; instead, it resolves type at run time.

A dynamic type can be defined using the dynamic keyword.

Example: dynamic type variable

dynamic dynamicVariable = 1;

The compiler compiles dynamic types into object types in most cases. The above statement would be compiled as:

dynamic type at compile time:

object dynamicVariable = 1;

The actual type of dynamic would resolve at runtime. You can check the type of the dynamic variable, as below:

Example: Get the actual type of dynamic type at runtime

static void Main(string[] args)

{

dynamic dynamicVariable = 1;

Console.WriteLine(dynamicVariable.GetType().ToString());

}

Output:

System.Int32

A dynamic type changes its type at runtime based on the value of the expression to the right of the "=" operator. The following example shows how a dynamic variable changes its type based on its value:

Example: dynamic

static void Main(string[] args)

{

dynamic dynamicVariable = 100;

Console.WriteLine("Dynamic variable value: {0}, Type: {1}",dynamicVariable, dynamicVariable.GetType().ToString());

dynamicVariable = "Hello World!!";

Console.WriteLine("Dynamic variable value: {0}, Type: {1}", dynamicVariable, dynamicVariable.GetType().ToString());

dynamicVariable = true;

Console.WriteLine("Dynamic variable value: {0}, Type: {1}", dynamicVariable, dynamicVariable.GetType().ToString());

dynamicVariable = DateTime.Now;

Console.WriteLine("Dynamic variable value: {0}, Type: {1}", dynamicVariable, dynamicVariable.GetType().ToString());

}

[Try it](http://www.tutorialsteacher.com/codeeditor?cid=cs-L98yXw)

Output:

Dynamic variable value: 100, Type: System.Int32   
Dynamic variable value: Hello World!!, Type: System.String   
Dynamic variable value: True, Type: System.Boolean   
Dynamic variable value: 01-01-2014, Type: System.DateTime

Methods and Properties of Dynamic Type

If you assign class object to the dynamic type then the compiler would not check for correct methods and properties name of a dynamic type that holds the custom class object. Consider the following example.

Example: dynamic

public class Student

{

public int StudentID { get; set; }

public string StudentName { get; set; }

public int Age { get; set; }

public int StandardID { get; set; }

public void DisplayStudentDetail()

{

Console.WriteLine("Name: {0}", this.StudentName);

Console.WriteLine("Age: {0}", this.Age);

Console.WriteLine("Standard: {0}", this.StandardID);

}

}

class Program

{

static void Main(string[] args)

{

dynamic dynamicStudent = new Student();

dynamicStudent.FakeMethod();

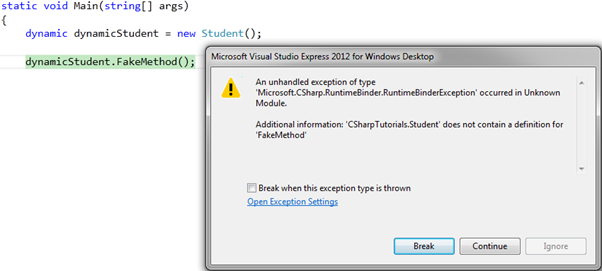
}

}

[Try it](http://www.tutorialsteacher.com/codeeditor?cid=cs-A7jGc0)

http://www.tutorialsteacher.com/Content/images/tips.pngIntellisense in Visual Studio does not give any help for the dymanic type.

In the above example, we have assigned Student object to a dynamic variable. In the second statement in Main() method, we call FakeMethod() method, which is not exists in the Student class. However, the compiler will not give any error for FakeMethod() because it skips type checking for dynamic type, instead you will get a runtime exception for it as shown below.

[](http://www.tutorialsteacher.com/Content/images/csharp/dynamic-type-error.png)Dynamic Type Fake Method Error

Dynamic Type as a Method Parameter

A method can have dynamic type parameters so that it can accept any type of parameter at run time.

Example: dynamic as a Parameter

class Program

{

static void PrintValue(dynamic val)

{

Console.WriteLine(val);

}

static void Main(string[] args)

{

PrintValue("Hello World!!");

PrintValue(100);

PrintValue(100.50);

PrintValue(true);

PrintValue(DateTime.Now);

}

}

[Try it](http://www.tutorialsteacher.com/codeeditor?cid=cs-zFTdKi)

Output:

Hello World!!   
100   
100.50   
True   
01-01-2014 10:10:50

Points to Remember :

1. The dynamic types are resolved at runtime instead of compile time.
2. The compiler skips the type checking for dynamic type. So it doesn't give any error about dynamic types at compile time.
3. The dynamic types do not have intellisense support in visual studio.
4. A method can have parameters of the dynamic type.
5. An exception is thrown at runtime if a method or property is not compatible.