



Adding Custom Code To Reporting Services

Introduction

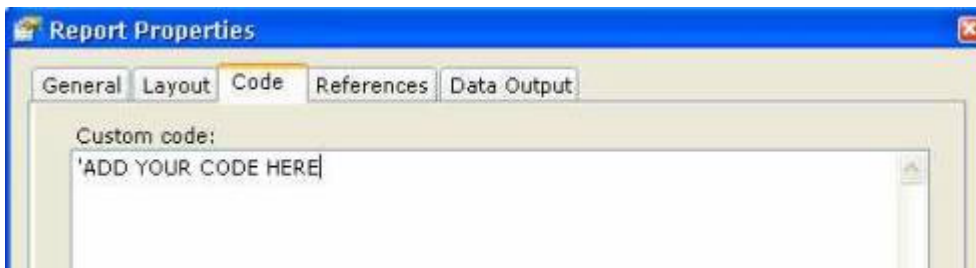
SQL Server 2005 Reporting Services has techniques for developing reports that can have their own code and also can access code from custom libraries i.e., the reports can have complex logic of their own or from a library. Earlier, if we look at any report, it will have a designer for creating the layout and some method through which we can assign the data for the report. [For example: DataSource for a report]

So, when a report is executed, the data is populated and assigned to the report for display. Reporting services add one more step to this process i.e., now the reports can now think and perform operations depending on the data.

In this article, we will see first look at the option that allows adding code and creating objects for accessing the DLL code.

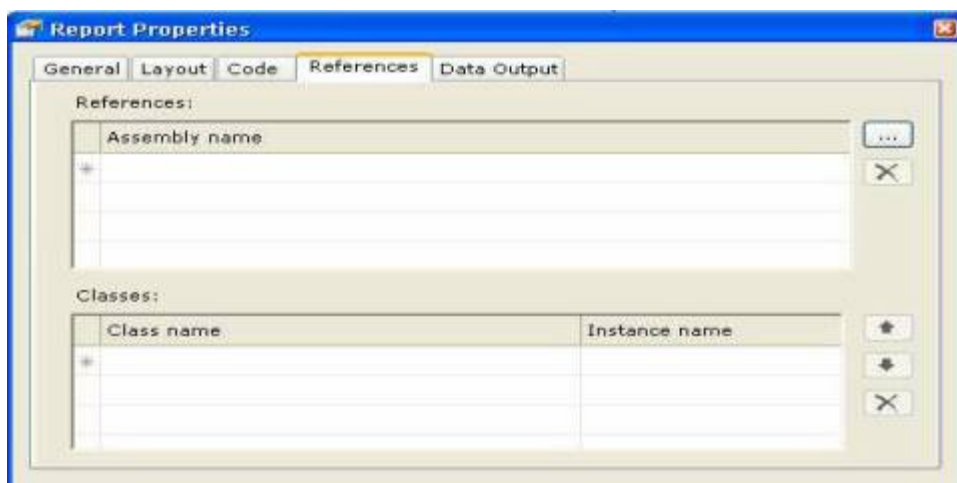
Adding code

- Create a report or open an existing report.
- Open the Report Properties option from the Report menu. Dialog box opens. [Figure 1]



[Figure 1]

- The Code tab is used to add the code. This code can contain a class or functions. Only **VB .NET** can be used.
- The References tab is used to specify the external DLLs to be used. This DLL can be created in any language. [Figure 2]



[Figure 2]

- The important step in the References tab is the creation of the objects. The DLL functions can be accessed using the object name specified.

To Remember:

- If a DLL is using some additional libraries, it has to be added as a reference. For example, System.Xml has to be referenced if any XML functions are used.
- Only the functions returning basic data-types [like string, integer] are supported.
- Custom DLL has to be copied to the \Common7\IDE\PrivateAssemblies folder of VS 2005 installation. [Example: "C:\Program Files\Microsoft Visual Studio 8\Common7\IDE\PrivateAssemblies"]. The DLL has to be referenced in the References tab from this path only.
- Each time modifications are done to the DLL, the new version has to be copied to the above folder.
- Keyword code to be used while accessing the custom code.
- The function/property has to be fully qualified including the object name.
- The functions can also take values as parameters, which can be any field from a dataset or user specified value.

Example:

This example will show how to add and use the code in the reports. Also, the use of DLL will be shown. The sample DLL for this example is created in C# 2005.

Create a new report [name of the report is TestReport.rdl] and open the Report Properties from the Report menu. Add the following code in the Code tab:

```
PUBLIC CLASS TestVB
    FUNCTION MyFunction AS STRING
        RETURN "Value returned through a FUNCTION from Code Tab."
    END FUNCTION

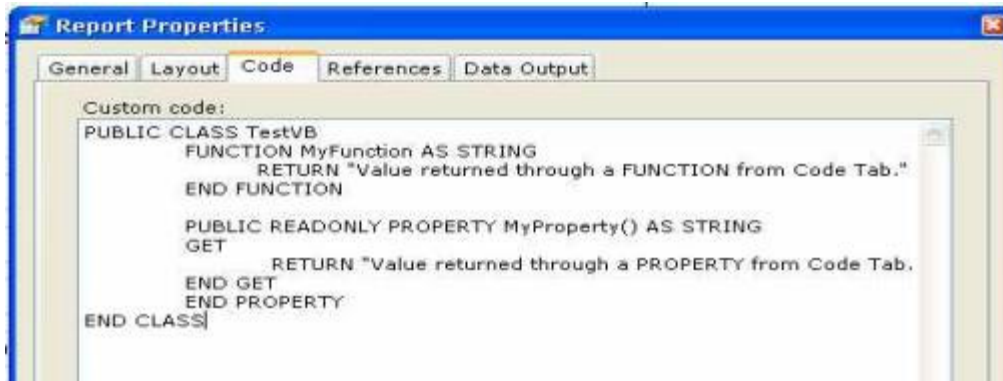
    PUBLIC READONLY PROPERTY MyProperty() AS STRING
    GET
        RETURN "Value returned through a PROPERTY from Code Tab."
    END GET
    END PROPERTY
END CLASS
```

```

FUNCTION MyFunction1 AS STRING
    RETURN "Value returned through a FUNCTION from Code Tab [No class]."
```

END FUNCTION

The code contains a CLASS having a function and a property in VB 2005. Remember that you can add as many functions/properties. This functions/properties can be part of a class or can exist without class. If a class is present, then an object is required to be created for accessing the functionality. If no class, then it can be accessed directly. Add this code in the Code tab. [Figure 3]



[Figure 3]

Create a simple DLL having properties and functions that are returning basic data-types values and taking parameters. The code for the DLL is in C# 2005, but it can be in any language.

```

namespace TestDLL
{
    public class Test
    {
        public string MyProperty
        {
            get
            {
                return "Data accessed through PROPERTY from a DLL.";
            }
        }

        public string MyFunction()
        {
            return "Value returned by a FUNCTION in the DLL.";
        }

        public string MyNewFunction(string MyString)
        {
            return "Parameter specified in FUNCTION was : " + MyString;
        }

        public bool MyBoolFunction()
        {
            return false;
        }

        public int MyIntFunction()
        {
            return 9999;
        }
    }
}
```

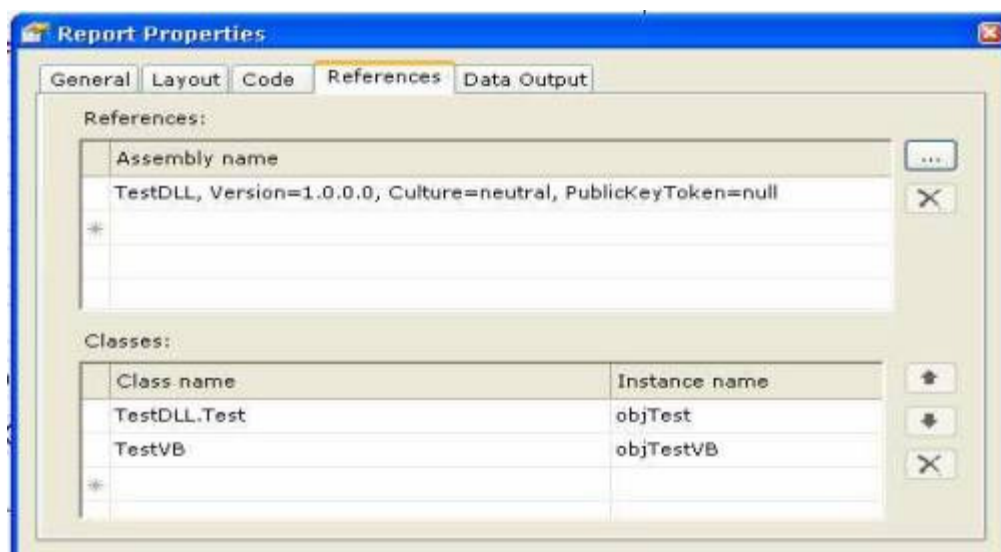
```

    }
}
}

```

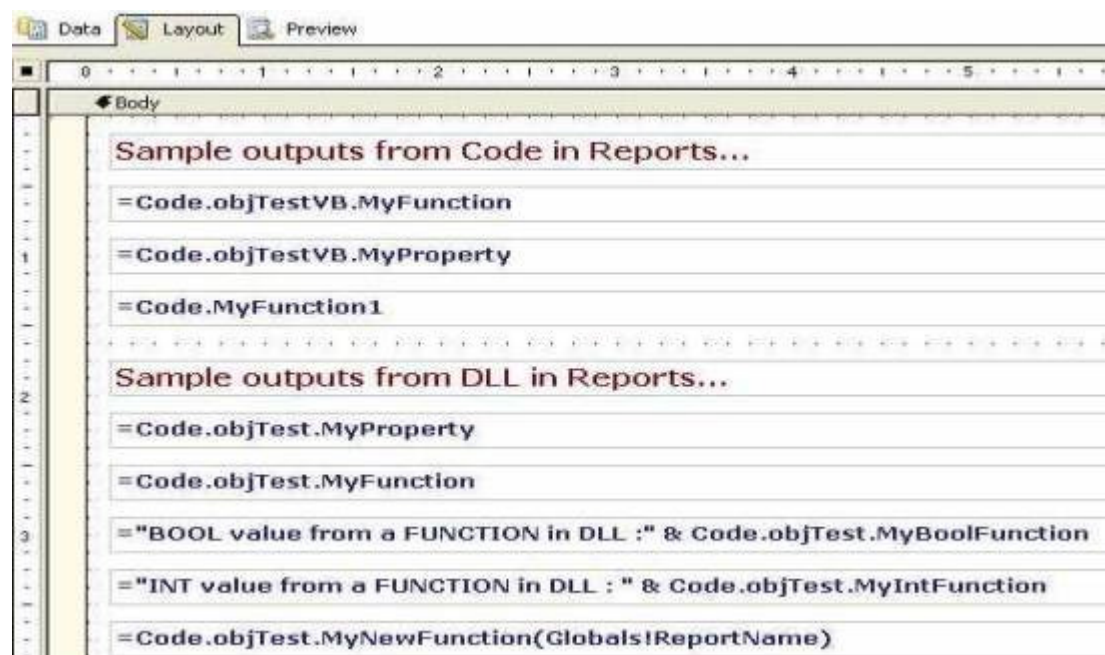
Name of the library having this class is TestDLL. After successfully compilation, copy the DLL to the \Common7\IDE\PrivateAssemblies folder of VS 2005 installation.

Add reference to the DLL by browsing to the PrivateAssemblies folder. Also, create objects for the classes. The objects are required for accessing the functionality. [Figure 4]



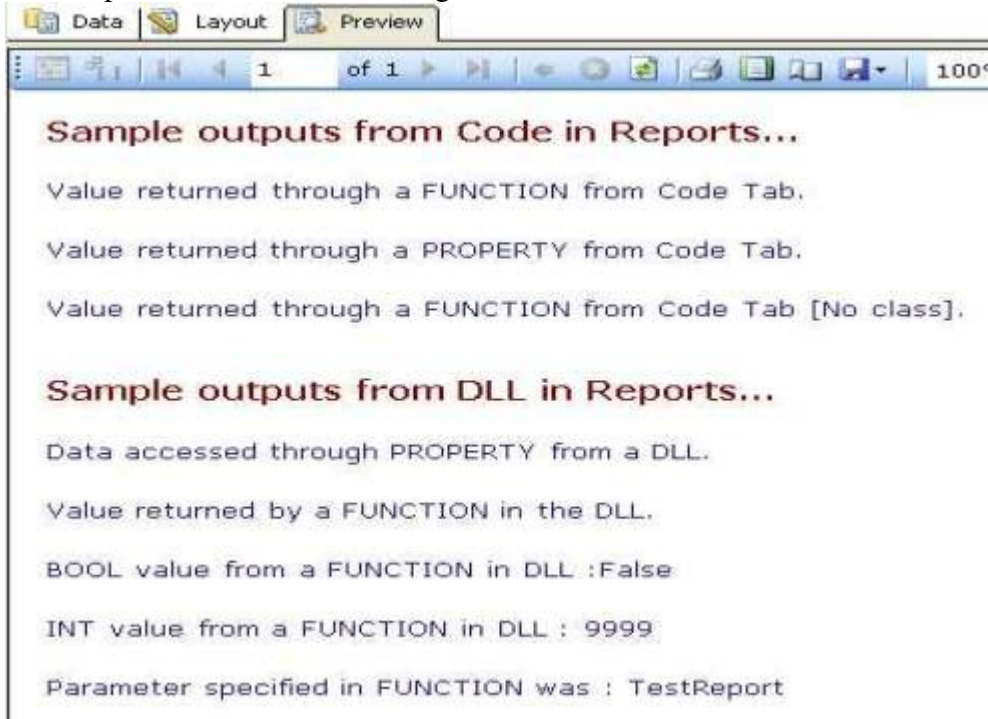
[Figure 4]

Add textboxes to the report layout and specify expressions to each of them, which will be accessing the functions/properties from the code and DLL. Keyword Code is used for accessing any function/property. The parameter value for MyNewFunction is the name of the report. [Figure 5]



[Figure 5]

The output will be as shown in Figure 6.



[Figure 6]

Conclusion

As we have seen that it is very easy to access code from DLLs and also add our own code, this can be used for performing complex operations and conditional processing of the data. The reports will no longer be used only for displaying data. Using this functionality, the reports can now figure out what to do with the data and be more powerful i.e., thinking and displaying data.

Copyright © 2002-2007 Red Gate Network. All Rights Reserved.