



Programming *STATISTICA* from .NET

Last Update: 2009

U.S. Headquarters: StatSoft, Inc. • 2300 E. 14th St. • Tulsa, OK 74104 • USA • (918) 749-1119 • Fax: (918) 749-2217 • info@statsoft.com • www.statsoft.com

Australia: StatSoft Pacific Pty Ltd.
Brazil: StatSoft South America
Bulgaria: StatSoft Bulgaria Ltd.
Czech Rep.: StatSoft Czech Rep. s.r.o
China: StatSoft China

France: StatSoft France
Germany: StatSoft GmbH
Hungary: StatSoft Hungary Ltd.
India: StatSoft India Pvt. Ltd.
Israel: StatSoft Israel Ltd.

Italy: StatSoft Italia srl
Japan: StatSoft Japan Inc.
Korea: StatSoft Korea
Netherlands: StatSoft Benelux BV
Norway: StatSoft Norway AS

Poland: StatSoft Polska Sp. z.o.o.
Portugal: StatSoft Ibérica Lda
Russia: StatSoft Russia
Spain: StatSoft Ibérica Lda

S. Africa: StatSoft S. Africa (Pty) Ltd.
Sweden: StatSoft Scandinavia AB
Taiwan: StatSoft Taiwan
UK: StatSoft Ltd.

Programming *STATISTICA* from .NET

Virtually every aspect of *STATISTICA* is exposed as a set of COM interfaces that are registered on a machine when *STATISTICA* is installed.

Since .NET-based languages cannot communicate with COM directly, a wrapper class called the COM Interop can be utilized to integrate the *STATISTICA* libraries into your .NET project. The COM Interop layer is created automatically by the Visual Studio .NET IDE when you import a COM interface. The COM Interop layer handles all of the details regarding interacting with the COM libraries in .NET. With the COM Interop layer in place, the *STATISTICA* COM interfaces behave like any other .NET object.

This white paper addresses the following topics:

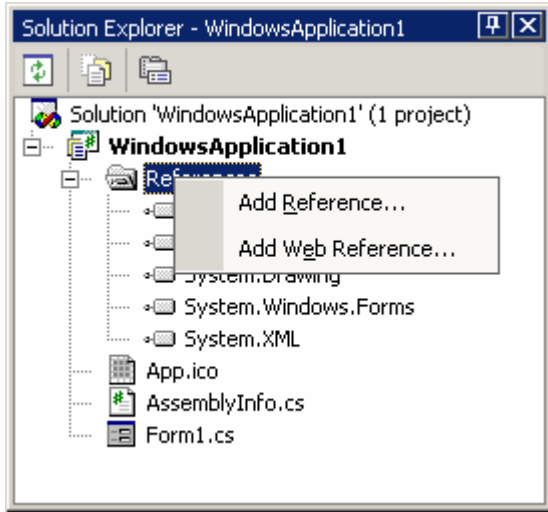
- Adding the *STATISTICA* Object Library into Your .NET Project
- Manually Creating the COM Interop Library
- Supporting Multiple Versions of *STATISTICA*
- Instantiating *STATISTICA*
- The Library Version of *STATISTICA*

Adding the STATISTICA Object Library into Your .NET Project

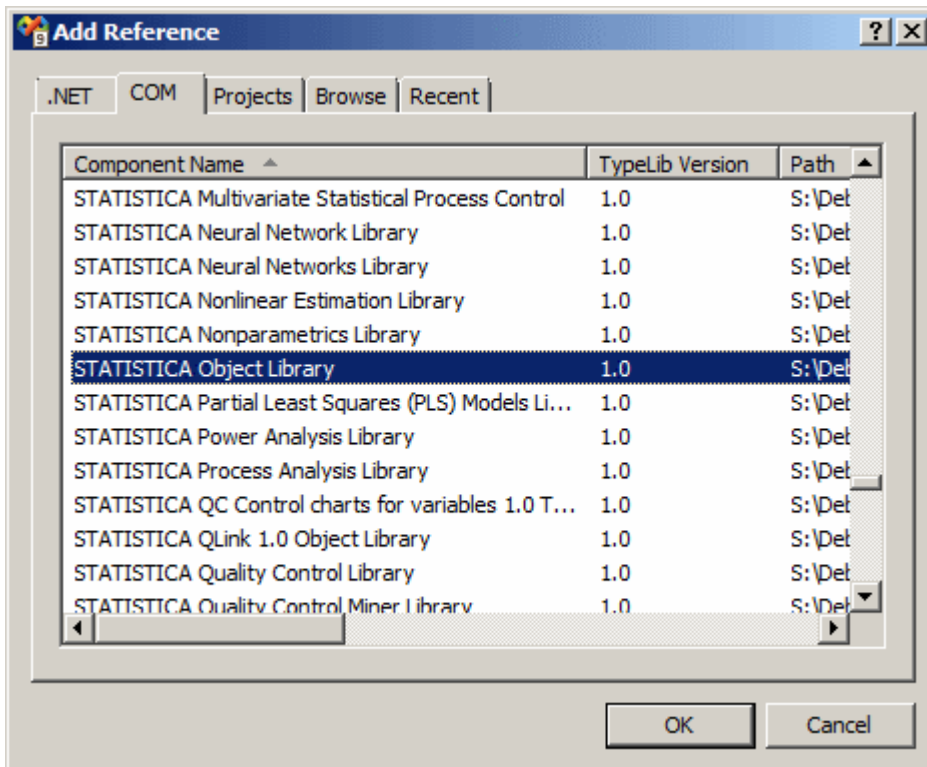
The .NET Interop layer is created automatically by adding the desired *STATISTICA* COM interfaces into your .NET project. *STATISTICA Object Library* is the base *STATISTICA* COM library.

To add the *STATISTICA* Object Library to a .NET project, first select the desired .NET project in *Solution Explorer*, and then select **Add References** from the shortcut menu (accessed by right clicking on the .NET project).

The **Add Reference** dialog will be displayed.

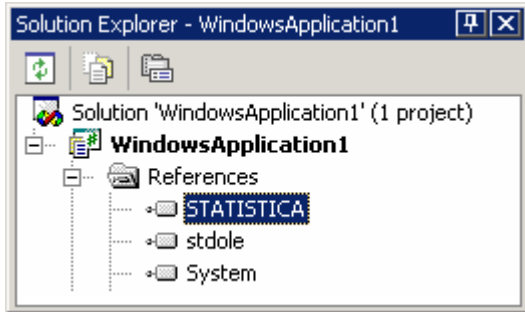


In the **Add Reference** dialog, select the **COM** tab. From the **Component Name** list, select **STATISTICA Object Library**, and click **OK**.



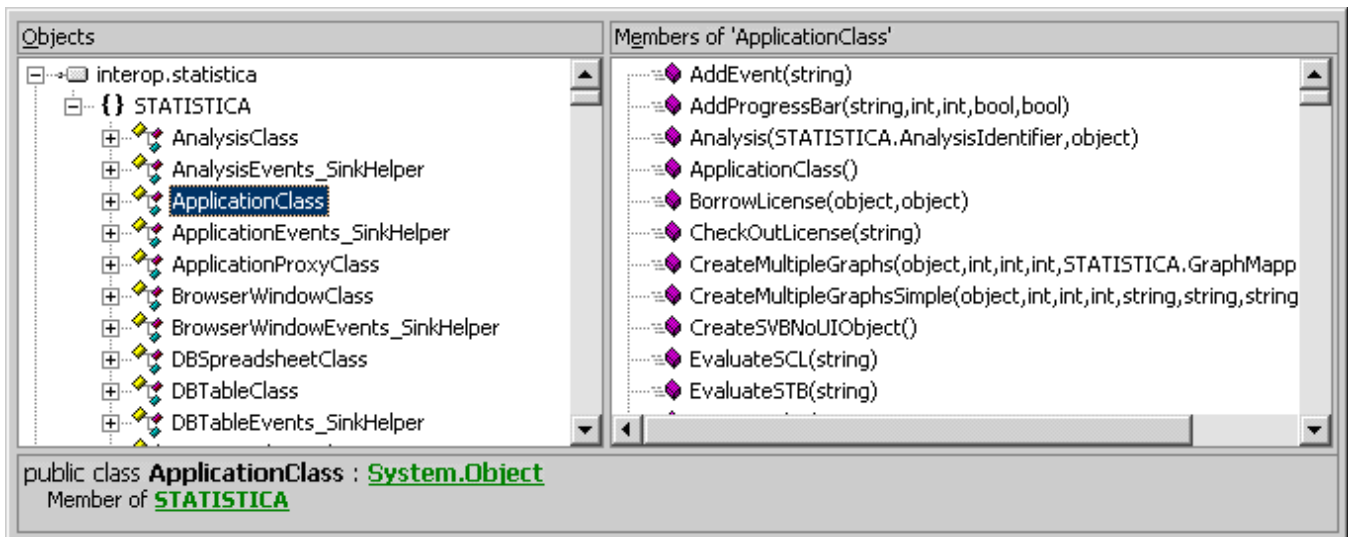
At this point, the necessary COM Interop library is created automatically.

Under the project *References* node, you will now see the entry *STATISTICA*.



The file *Interop.STATISTICA.dll* is also added to the project output directory. The *STATISTICA* COM Interop library is stored in this file.

To view the *STATISTICA* object library from your .NET project, right-click on the *STATISTICA* reference, and from the shortcut menu, select **View in Object Browser**.



Manually Creating the COM Interop Library

It is also possible to create the COM Interop library manually and import it into your .NET project. This gives you the ability to specify a different name for the Interop DLL as well as define a custom namespace. The program that enables you to create an Interop is *TLBIMP.EXE*. From a Visual Studio command prompt, execute *TLBIMP* with an initial parameter of the type library source.

In the example below, the output DLL name and namespace are also specified.

```

C:\erase>tlbimp "c:\program files\statsoft\statistica 9\statist.exe"
/out:Interop.STATISTICA.dll /sysarray /namespace:STATISTICA
Microsoft (R) .NET Framework Type Library to Assembly Converter 1.1.4322.573
Copyright (C) Microsoft Corporation 1998-2009. All rights reserved.

TlbImp warning: Interface ISWESSDocInfo is marked as [dual], but does not derive from
IDispatch. It will be converted as an IUnknown-derived interface.
TlbImp warning: At least one of the arguments for 'StaWorkbookSite.GetWorkbookInfo' can
not be marshaled by the runtime marshaler. Such arguments will therefore be passed as a
pointer and may require unsafe code to manipulate.
TlbImp warning: At least one of the arguments for 'StaWorkbookSite.NotifyChange' can not
be marshaled by the runtime marshaler. Such arguments will therefore be passed as a
pointer and may require unsafe code to manipulate.
Type library imported to C:\erase\Interop.STATISTICA.dll

C:\erase>_

```

In this example, we reference the file *STATIST.EXE* since that executable contains the *STATISTICA Object Library* type library. Once the Interop DLL is generated, you can add it to your .NET project by selecting **Add Reference** from the *Solution Explorer* as before, but this time click the **Browse** button to select the newly created Interop DLL.

Supporting Multiple Versions of *STATISTICA*

To support multiple version numbers of *STATISTICA*, it is necessary to maintain separate *STATISTICA Object Library* Interop DLLs for each version number of *STATISTICA* you want to support. You can use the *TLBIMP* command to generate Interop DLLs against specific versions of *STATIST.EXE* and other DLLs. When distributing the application, make sure the correct version of the *STATISTICA* Interop DLL is deployed with your .NET application.

Instantiating *STATISTICA*

Because of its COM architecture, *STATISTICA* can be incorporated into many different development environments. When using *STATISTICA* from an external development environment, it is necessary to have a top-level object called the application object. The application object is the application itself and will contain other objects (for example, spreadsheets and graphs), but access to these other objects is restricted unless the application object is running.

Assuming you are using the default namespace *STATISTICA*, the interface you should declare your variable as is *STATISTICA.Application*. To create an instance of *STATISTICA*, set your variable equal to `new STATISTICA.ApplicationClass()`.

```
STATISTICA.Application pApp = (STATISTICA.Application)
    new STATISTICA.ApplicationClass();

pApp.Visible = true;
```

When an instance of the *STATISTICA.ApplicationClass* is created, a *STATIST.EXE* process will be launched. This is equivalent to launching *STATISTICA* from the **Start** menu. The *STATISTICA* instance is initially hidden but can be made visible. Since it is a separate process, all calls to this instance are made out of process.

The Library Version of STATISTICA

In addition to the *STATISTICA.Application* object, there is also a lighter-weight, higher-performance version of the object called *STATISTICA.Library*. The *Library* version is licensed separately and therefore may not be available with your installation. It contains identical interfaces as the *STATISTICA.Application* library. Any existing code that uses the *Application* object can be replaced with the *Library* object.

The main restriction is that the *STATISTICA* user interface features are not available from the *Library* version. Therefore, in the example above, if the *Application* object was instantiated as a new *STATISTICA.LibraryClass*, it would not be possible to make the object visible (and show the *STATISTICA* interface).

The *Library* version of *STATISTICA* is loaded in-process, which means accessing its COM interfaces is more efficient than using the *Application* version of the object (which is loaded out of process). Since it is loaded in-process, multiple versions of the library cannot be instantiated. Normally, you would only instantiate one *Library* object or one *Application* object in your program.