

GoDiagram Pocket Introduction

Copyright © 2002-2008 Northwoods Software Corporation

GoDiagram™ Pocket for Microsoft® .NET Compact Framework (“Go”) is a .NET class library containing a set of Windows Forms controls for easily building interactive diagrams in .NET-based applications running on the Pocket PC and similar devices.

GoDiagram Pocket shares nearly all of its design and implementation with GoDiagram Win, which you use to build standard Windows Forms desktop applications. The User Guide provides details about Go, most of which apply to both products. You will need to read this before you can really make good use of Go.

If you are interested in using GoDiagram Win for building regular .NET Windows Forms applications, please read the `GoWinIntro.doc` file instead.

Installation kits

The installation kits are Windows Installer files, one for .NET version 1.1, the other for .NET version 2.0. They serve as both an evaluation kit as well as the full binary product kit both for GoDiagram Pocket and for GoDiagram Win, including optional libraries. When you have purchased and installed a full binary development license for an assembly, you will be able to compile and distribute applications using that assembly without getting evaluation reminder messages or watermarks.

Both the base Go assembly and the optional GoLayout and GoInstrument assemblies are included in the binary kit. GoDiagram Win assemblies are also provided in this kit.

Before you install Go, you should already have installed the version of the .NET Compact Framework SDK that the kit depends upon.

GoDiagram Pocket Files

GoDiagram Pocket consists of four assemblies:

- **Northwoods.GoPocket.dll**, holding the **Northwoods.Go** namespace
- **Northwoods.GoPocket.Instruments.dll**, holding the **Northwoods.Go.Instruments** namespace
- **Northwoods.GoPocket.Layout.dll**, holding the **Northwoods.Go.Layout** namespace
- **Northwoods.GoPocket.Xml.dll**, holding the **Northwoods.Go.Xml** namespace

The four assemblies are in the **libpocket** subdirectory of the Go installation. They only depend on the Microsoft .NET Compact Framework System, Windows Forms, and Drawing assemblies. They do not include any unmanaged code.

Detailed documentation on the types in these libraries is provided in the **GoWin.chm** compiled HTML help file. This file, along with other documentation, is in the **docs** subdirectory of the Go installation. You may find it instructive to see a listing of the differences between Pocket and Win; this list is maintained in **GoWinPocketDiffs.doc**.

It also places some example code in the **PocketSamples** and **PocketSamplesVB** subdirectories. Both sample executable applications and the sample application source code are included.

Initial Experiences

If you haven't already run the sample applications, just to get a feel for what Go can do, please try them. Reading the source code for the applications will really help you understand how easily you can implement different kinds of features. Remember that these are sample applications. Sometimes functionality is implemented just for the sake of demonstration—no real application would want to have that combination of features, or so many different ways to achieve the same kind of functionality.

Because GoDiagram Pocket shares so much of the functionality that GoDiagram Win provides, you will also want to run the regular desktop sample applications to see what Go can do.

If you have certain features you know you want to implement, but are not sure how to do so, it might help to read the Frequently Asked Questions (FAQ) document, **GoDiagramFAQ.chm**, in the **docs** subdirectory. Another source of inspiration can be the GoDiagram forum at <http://www.nwoods.com/forum>.

It might also help to read the entire User Guide, **GoUserGuide.doc**, because it discusses much of the programming model embodied in Go. If you don't have that much time, at least read the *Go Concepts* chapter in the User Guide.

Licensing and Unlock Codes

GoDiagram is licensed per developer. There are no additional fees for distributing or running applications that incorporate our GoDiagram products. Any developer who programs using the GoDiagram application programming interface (API) must be licensed.

Each developer machine must have an installed unlock code. A single paid developer license may have multiple unlock codes associated with it, allowing that developer to work on multiple machines (home machines, laptops, etc.). By default, each GoDiagram license has 2 unlock codes associated with it. If required, a reasonable number of additional unlock codes can be requested by sending e-mail to gosales@nwoods.com.

If no unlock code is installed on a machine, GoDiagram will run in "evaluation mode" on that machine. You will see a watermark in each GoDiagram window and you may see message boxes reminding you that you that this is an evaluation version. In all other respects the functionality is identical to a machine with a valid unlock code installed. You may build and test your applications in evaluation mode, but you will not be able to distribute applications built in evaluation mode to other machines (see Deployment).

Requesting and Installing Unlock Codes with the License Manager

Unlock codes are managed using the License Manager. To license your development machine, you need to run the License Manager application (on the Start menu under Northwoods Software, GoDiagram) from that machine and follow the instructions for requesting and installing unlock codes.

If you have already purchased the software, simply select your product in the product list. This will automatically select the different assemblies that come with that product.

Click on "Request Unlock Codes" while connected to the web and fill out the requested information, which includes your e-mail address and the order number and buyer e-mail address. Note that your order number will have been sent to the buyer as the e-mail subject line when GoDiagram was purchased. Click on Submit and the unlock codes will be automatically e-mailed to you. If you have lost or forgotten your order number, click on "Get Order Information" in the License Manager, or if your development machine is not connected to the internet, please send e-mail to gosales@nwoods.com.

Finally, enter the unlock codes by pressing "Enter Unlock Codes". You will be prompted to enter the unlock codes for each assembly. Make sure you enter the matching unlock code for each requested assembly.

Each unlock code is only effective for a few days, so you should enter it in the LicenseManager application promptly, but you can always re-request another unlock code if you need to reinstall GoDiagram in the future on the same machine.

Note that when you successfully enter an unlock code into the GoDiagram LicenseManager, a license key is installed in the registry. If you frequently wipe out your disk drive to replace it with a standard disk image, you do not need to re-request unlock codes each time. You simply have to restore the registry key:

```
HKEY_CURRENT_USER\SOFTWARE\Northwoods Software\Go.NET
```

You must have read access to this registry key in order to develop using GoDiagram or to compile and link your license information into your executable (see Deployment).

Moving or Retiring Unlock Codes

If you are no longer using a particular machine for GoDiagram development, the name of the development machine is changing, or you wish to move GoDiagram development from one machine to another, click on "Remove All Licenses" while running the License Manager from your development machine. Doing so will remove the unlock codes from your current machine, causing GoDiagram to run in evaluation mode once again. You will also be presented with a form to report the deactivation of the machine. Enter your order number and buyer e-mail and click on Submit to send this information to Northwoods. This is important, as it will return your unlock code to the available pool for that order number, allowing you to successfully request a new unlock code from another machine in the future.

Note that if you are changing machine names, it is important to "Remove All Licenses" *before* renaming your machine to allow that machine name to be deactivated.

After retiring the unlock code from the previous machine, simply install your GoDiagram kit on the new machine and run the License Manager as before to request and install unlock codes for the new machine.

Deployment

To deploy an application containing GoDiagram, you must compile and link the license(s) into your application. This will allow your application (including GoDiagram DLLs) to be run on machines other than your development machine without the need for your end users to be take any action, or even be aware of GoDiagram licensing.

You will need to add a line similar to the following into the main method for your application:

```
Northwoods.GoPocket.GoView.VersionName = "01234...56789"
```

For licensing GoLayout and GoInstruments, the lines will appear similar to:

```
Northwoods.GoPocket.Layout.GoLayout.VersionName = "01234...56789"  
Northwoods.GoPocket.Instruments.GoInstruments.VersionName = "01...56789"
```

GoXml and GoSvg just depend on the licensing of Northwoods.GoPocket.dll and therefore do not require their own unlock codes or license key assignment statements.

At run-time, the .NET licensing architecture will find these linked-in license keys and pass them to each component during the component creation process so that it can decide whether to run or throw an exception.

The GoDiagram `LicenseManager` application will generate the proper statement for you. Select the desired **Northwoods.GoPocket** or **Northwoods.GoPocket.Layout** or **Northwoods.GoPocket.Instruments** assemblies, and click the “Generate License Key” button and enter the name of your assembly that creates the Go component, without the “.exe” file extension. It will produce a line of code (an assignment statement) per assembly that it copies into the clipboard. You just need to paste it into your static/shared main method.

The license key is dependent on the name of the assembly, so you cannot rename the assembly later.

You will need to generate a new license key assignment statement for each Pocket application that you create that uses Go. You will need to generate new statements when you upgrade to a new version of Go.