

GoDiagram[™] Win and GoDiagram Pocket version 2.4.1 Release Notes

Copyright © 2002-2005 Northwoods Software Corporation

This kit includes sets of assemblies and samples for building interactive diagram applications using two different versions of Windows Forms: the full .NET Framework running on Windows and for the .NET Compact Framework on the Pocket PC.

GoDiagram[™] Win requires the Microsoft[®] .NET Framework 1.1 or 2.0 SDK. It also works with interactive development environments such as Visual Studio .NET 2003 or Visual Studio 2005, or with Borland C#Builder.

GoDiagram Pocket for Microsoft .NET Compact Framework requires the Microsoft .NET Compact Framework 1.0 SDK or Visual Studio .NET 2003 or Visual Studio 2005.

GoDiagram Web for ASP.NET Web Forms is a separate product in a separate installation kit. GoDiagram Web and its additional assemblies share most of the same functionality as the Windows Forms components but are designed for ASP.NET to run on web servers.

This kit also includes assemblies and samples for several optional libraries that extend GoDiagram:

- GoLayout, for automatic positioning of nodes
- GoInstruments, for displaying instruments such as dials, meters, gauges, and rulers
- GoXml, for customized reading and writing of XML documents
- GoSvg, for generating customized interactive SVG files

GoSvg is still not yet implemented for the PocketPC; if this functionality is required for your application, please contact us.

This kit also includes an unsupported assembly that previews future functionality: GoDraw. At the current time GoDraw includes a ruler control (**GoRuler**), a predefined view using two rulers (**GoDrawView**), and controls for choosing a Color (**GoColorControl** and **GoColorForm**). Preliminary documentation is provided in GoDraw.chm.

This installation is initially licensed under the terms of the evaluation license agreement, WinEvalLicense.rtf in the docs subdirectory. Upon purchasing a valid binary development license for a GoDiagram assembly on a particular computer, and after installing the license with the LicenseManager application, you may compile a license for that assembly into your application and distribute that assembly. The binary development license agreement is docs\WinBinaryLicense.rtf. Each product has its own redistributable assembly:

- GoDiagram Win: Northwoods.Go.dll
- GoLayout Win: Northwoods.Go.Layout.dll
- GoInstruments Win: Northwoods.Go.Instruments.dll
- GoDiagram Pocket: Northwoods.GoPocket.dll
- GoLayout Pocket: Northwoods.GoPocket.Layout.dll
- GoInstruments Pocket: Northwoods.GoPocket.Instruments.dll

The GoXml and GoSvg assemblies are licensed under the terms of their respective GoDiagram assemblies and can be used at no additional cost:

- GoXml Win: Northwoods.Go.Xml.dll

- GoSvg Win: Northwoods.Go.Svg.dll
- GoXml Pocket: Northwoods.GoPocket.Xml.dll

If you want to deploy into environments where only the .NET Framework 1.0 is installed, contact us for DLLs that are compiled for .NET 1.0.

If you want to deploy into a reduced-trust environment, such as part of a no-touch deployment application, or called by a DLL that is hosted by Internet Explorer, contact us for special licensing instructions and DLLs that can run with limited permissions.

For GoDiagram Win, please read the **GoDiagram Win Introduction** document, docs\GoWinIntro.doc.

For GoDiagram Pocket, please read the **GoDiagram Pocket Introduction** document, docs\GoPocketIntro.doc.

Both documents include a section about how to distribute your applications with the assemblies for which you have purchased a license.

You will find many answers to your questions in the **Frequently Asked Questions** (FAQ) list that is in docs\GoDiagramFAQ.chm.

You might also want to search the **GoDiagram Forum** at <http://www.nwoods.com/forum>.

Installation Directory Contents

- \
 - Release Notes (this README file)
- \lib *assemblies compiled for .NET Framework version 1.1 and for PocketPC*
 - Northwoods.Go.dll, containing the GoDiagram Win controls and classes
 - Northwoods.Go.Layout.dll, the automatic layout library components (GoLayout) for GoDiagram Win
 - Northwoods.Go.Instruments.dll, the instruments library (GoInstruments) for GoDiagram Win
 - Northwoods.Go.Xml.dll, the library for the customizable reading and writing of XML files
 - Northwoods.Go.Svg.dll, the library for the customizable generation of SVG files
 - Northwoods.GoPocket.dll, containing the GoDiagram Pocket controls and classes
 - Northwoods.GoPocket.Layout.dll, the automatic layout library components (GoLayout) for GoDiagram Pocket
 - Northwoods.GoPocket.Instruments.dll, the instruments library (GoInstruments) for GoDiagram Pocket
 - Northwoods.GoPocket.Xml.dll, the library for the customizable reading and writing of XML files
 - *.xml, XML documentation for code editor tooltips in Visual Studio
- \lib2.0 *GoDiagram Pocket assemblies compiled for .NET Compact Framework version 2.0 for Pocket PC*
 - Northwoods.GoPocket.dll, containing the GoDiagram Pocket controls and classes
 - Northwoods.GoPocket.Layout.dll, the automatic layout library components

(GoLayout) for GoDiagram Pocket

- `Northwoods.GoPocket.Instruments.dll`, the instruments library (GoInstruments) for GoDiagram Pocket
- `Northwoods.GoPocket.Xml.dll`, the library for the customizable reading and writing of XML files
- `*.xml`, XML documentation for code editor tooltips in Visual Studio

`.\docs`

- `GoWinIntro.doc`, introduction to GoDiagram Win
- `GoPocketIntro.doc`, introduction to GoDiagram Pocket
- `GoDiagramFAQ.chm`, Frequently Asked Questions list for GoDiagram (Win, Pocket, and Web)
- `GoUserGuide.doc`, User Guide for GoDiagram (Win, Pocket, and Web)
- `GoLayoutUserGuide.doc`, User Guide for GoLayout (Win, Pocket, and Web)
- `GoInstrumentsUserGuide.doc`, User Guide for GoInstruments (Win, Pocket, and Web)
- `GoWin.chm`, API Reference Manual for GoDiagram, GoXml, GoSvg, GoLayout, and GoInstruments (Win and Pocket)
- `GoWinWebDiffs.doc`, a listing of the differences in GoDiagram between Windows Forms and Web Forms
- `GoWinPocketDiffs.doc`, a listing of the differences in GoDiagram between Win and Pocket
- `WinEvalLicense.rtf`, evaluation license agreement file
- `WinBinaryLicense.rtf`, binary development license agreement file

`.\bin`

- `LicenseManager.exe`, to install developer's licenses for the GoDiagram products

`.\Samples`, sample GoDiagram Win applications written in C#

- `*.exe`, sample applications using GoDiagram Win
- Sample source code, in subdirectories:
 - **BasicApp**, a simple editor using basic nodes
 - **Classier**, GoDiagram class hierarchy browser and property/method viewer
 - **Demo1**, GoDiagram demo application, built by modifying prototype app
 - **FamilyTree**, displays the relationships amongst some of the English Tudors
 - **FlowCharter**, a simple flow chart editor
 - **IconicApp**, a simple editor using nodes displaying icons
 - **InstrumentDemo**, demonstrates various dials and meters from the `Northwoods.Go.Instruments` library
 - **LayoutDemo**, demonstrates some of the features of the automatic layout library, `Northwoods.Go.Layout`
 - **MinimalApp**, a minimal GoDiagram application
 - **MovableLinkApp**, a simple editor that supports disconnected or partly connected links
 - **ObjectBrowser**, GoDiagram part hierarchy browser
 - **OrgCharter**, a simple organization chart editor
 - **Planogrammer**, a simple application for designing/organizing racks of items in a store
 - **Processor**, an editor for a model of a work-process system
 - **ProtoApp**, prototype application

- **StateCharter**, a simple state diagram editor
- **SubGraphApp**, an editor that supports three different kinds of subgraphs
- **TreeApp**, a two-dimensional tree that supports expand/collapse
- **WebWalker**, shows references between HTML web pages
- **BuildSamples.bat**, a command file to recompile the sample applications
- **.\SamplesVB**, the same GoDiagram Win samples translated from C# to VB.NET
 - **BuildSamplesVB.bat**, a command file to recompile the sample applications
- **.\PocketSamples**, sample GoDiagram Pocket applications written in C#, for .NET CF 1.0
 - *.exe, sample applications using GoDiagram Pocket
 - Sample source code, in subdirectories:
 - **MinimalApp**, a minimal GoDiagram Pocket application
 - **Shaper**, a simple app that lets you create different shapes (including Bezier strokes and polygons), animates **GoPie** shapes, rotates the whole document including rotating strokes and polygons, and includes a sample meter from GoInstruments Pocket
- **.\PocketSamplesVB**, just the same MinimalApp GoDiagram Pocket sample translated from C# to VB.NET

Summary of Significant New Features since 2.3

The complete list of changes and new features is included later in this document.

- Added **GoGrid** class and **IGoDragSnapper** interface
- Added **GoSheet** class and **GoView.BackgroundHasSheet** and **GoView.SheetStyle** properties
- Added support for arrow keys in **GoView**
- Added **GoView.SelectionStarting** and **SelectionFinished** events
- Added **GoView.ObjectSelectionDropReject** and **BackgroundSelectionDropReject** events
- Added **GoView.ObjectSelectionDropped** and **BackgroundSelectionDropped** events
- Added support for Z-order within a **GoLayer**
- Added **GoCollapsibleHandleStyle** enum, for drawing a **GoCollapsibleHandle** in different styles
- Added new sample: **Planogrammer**, an editor for designing/organizing racks of items in a store
- Added new example class: **ClassDiagramNode**, like a UML Class node with a single **GoBoxPort**, that supports collapsing sections
- Added new example class: **CollapsingRecordNode**, like **RecordNode**, but with collapsing sections, that supports ports on each side of each item
- Added new example class: **ScrollingMultiTextNode**, inherits from **GoMultiTextNode**, but adds two buttons to support resizing and scrolling the **GoListGroup** inside the **GoMultiTextNode**
- Added preview version of unsupported Northwoods.Go.Draw assembly (GoDiagram Win only)

Upgrading to 2.4

If you are upgrading from a version earlier than 2.3.1, we suggest that you first upgrade to version 2.3.1. GoDiagram supports simultaneous installations of multiple major/minor versions. Even if you do not want to take the time to migrate your code through

intermediate versions, we strongly suggest reading the release notes for releases you want to skip.

You will need to make sure your projects now refer to the new Northwoods.Go.* version 2.4 assemblies, not older versions.

You will need to change all of your licenses.licx files so that references to GoDiagram Win components of:

Version=1.2.3.4 (or the version you have been using until now)
are replaced with:
Version=2.4.1.1

For PocketPC applications, if you want to target .NET Compact Framework 2.0 with Visual Studio 2005, you will need to change your GoDiagram assembly references to use the DLLs that are in the lib2.0 subdirectory, and you will need to invoke Visual Studio 2005's "Upgrade Project" command on each of your projects to convert them from .NET CF 1.0 to .NET CF 2.0. In addition, you may need to make other code changes, such as upgrading additional resource files.

Known Bugs and Misfeatures

- There are a number of Microsoft bugs that you may encounter:
 - License keys are only recognized when they are linked into an executable assembly (.EXE), not a library (.DLL); you need to have your executable's project refer to Northwoods.Go and include the desired component references in the executable project's LICENSES.LICX file (you can leave them in your library that uses GoDiagram, but they are ineffective there)
 - SOAP serialization
 - text clipping
 - when setting properties on **Controls** created by **GoControls**, and the real control needs to be re-created

Detailed List of Changes since 2.3.1

All assemblies are now compiled for .NET 1.1 (at least), and all Visual Studio projects have been upgraded to VS2003. The assemblies work well with .NET 2.0, and you can upgrade any of the sample projects using Visual Studio 2005. For .NET Compact Framework 2.0, you will need to use the new GoDiagram Pocket assemblies compiled for .NET CF 2.0.

GoView

- improved Forms designer support for **GoView**'s grid and sheet properties and a few other properties; some values that had been stored in the RESX file as "binary" have now been replaced with designer generated code. However, there is still no Forms designer support for GoDiagram Pocket controls.
- fixed **GoView.MoveSelection** to heed **GoObject.CanMove**
- improved **GoView.CopySelection** to maintain Z-order
- added **GoView.DisableKeys**, affecting **GoToolManager.DoKeyDown** by disabling certain behaviors, specified by **GoViewDisableKeys** enum

- added **GoView.ArrowMoveLarge** and **ArrowMoveSmall** properties, to specify how far arrow keys move the **Selection**
- added **GoView.DrawsXorMode** (Win & Express only), to control behavior of **DrawXorBox**; default is false; automatically set to false when setting **GoToolRubberBanding.AutoScrolling** to true
- added **GoView.SelectionStarting** and **SelectionFinished** events, with corresponding methods; these events surround mass changes in selection, such as in calls to **GoView.SelectAll**, **SelectInRectangle**, **CopySelection**, **DeleteSelection**, **EditPaste**, **DoExternalDrop**; in your application, event handlers for **ObjectGotSelection** and **ObjectLostSelection** are still needed, since individual changes to **GoView.Selection** might not raise these **SelectionStarting** and **SelectionFinished** events
- fixed some cases of updating selection handles too soon or too often when an object has been changed
- fixed some bugs refreshing view for certain shapes when the Pen changed
- added **GoView.ObjectSelectionDropReject** and **BackgroundSelectionDropReject** events, with corresponding methods, to customize whether the user may drop the **Selection** on a document object or in the background
- added **GoView.ObjectSelectionDropped** and **BackgroundSelectionDropped** events, with corresponding methods, to customize the behavior when the user drops the **Selection** onto a document object or onto the background
- added **GoView.DoSelectionDropReject** method, to either call **RaiseObjectSelectionDropReject** and **GoObject.OnSelectionDropReject**, or to call **RaiseBackgroundSelectionDropReject**, depending on whether there is a document **GoObject** at the mouse point
- added **GoView.DoSelectionDropped** method, to either call **RaiseObjectSelectionDropped** and **GoObject.OnSelectionDropped**, or to call **RaiseBackgroundSelectionDropped**, depending on whether there is a document **GoObject** at the drop point
- changed **GoView.PrintDocumentPage** access from protected to public, for easier use by custom **PrintControllers**

GoDocument, Layers, Collections

- added **GoDocument.Copy()** method, for making a copy of a **GoDocument**
- fixed **GoDocument.IsUnoccupied** for rectangles extending beyond the document bounds
- changed **GoChangedEventArgs.CanUndo** to be true even when **IsBeforeChanging** is true--this is only an issue if you have extended a class to make use of the **GoDocument.RaiseChanging** or **GoObject.Changing** methods to record a **GoChangedEventArgs** before making any undoable state changes
- added **GoLayerCollection.NextLayer** method
- extended **GoLayerCollection.InsertDocumentLayer[After/Before]** to work with layers that are already present in collection
- added **GoLayerCollection.SortByZOrder** method, to sort an array of objects by their Z-order
- fixed bug in **GoView.OnDocumentChanged** handling change in Z-order of **GoLayers** in a **GoLayerCollection**
- added **GoLayer.MoveBefore**, **MoveAfter**, **NextObject** methods for Z-order support within layers
- added **GoCollection.AddRange** method, for **Add**'ing all **GoObjects** in a given **IGoCollection** (and **GoSelection.AddRange** raises **GoView.SelectionStarting** and

SelectionFinished events)

GoGrid

- added **IGoDragSnapper** interface, to allow objects to control dragging behavior
- added **GoGrid**, implementing **IGoDragSnapper**
- added **GoView.BackgroundGrid** property and **GoView.CreateGrid** method
- reimplemented **GoView** grid properties to delegate to the **GoView.Grid** object, that is either **GoView.BackgroundGrid** or **GoView.Sheet.Grid**
- removed **GoView.DrawGrid...** methods (now on **GoGrid**)
- added **GoView.SnapPoint** method, that looks for **IGoDragSnapper** objects
- removed **GoView.FindNearestGridPoint** method (now on **GoGrid**)

GoSheet

- added **GoSheet**, incorporating a **GoRectangle** and a **GoGrid** to present a sheet of paper
- added **GoView.SheetStyle**, **Sheet**, and related properties, and **GoView.CreateSheet** method, to support displaying a sheet of paper and how to show it as the view's size changes, with the sheet getting the document's **PaperColor**, and the rest of the view the Control's **BackColor**
- added **GoView.BackgroundHasSheet** property, when set to true creates the **Sheet** if needed, and then redirects the **GoView.Grid** property and all of the **GoView.Grid...** properties to refer to the **Sheet's Grid** rather than the view's **BackgroundGrid**
- added **GoView.UpdateExtent** method, to update the view's **DocExtent** as the view's size changes
- added **GoView.RescaleWithCenter**, to change the view's **DocScale** and **DocPosition** so that a particular document point remains centered in the view

GoObject

- added static/shared **GoObject.LargestSizeKeepingAspectRatio** method, for convenience in determining the largest size that will fit a given size while keeping the aspect ratio given by a different size

GoSubGraph

- improved **GoSubGraph.CollapseChild** to move the whole child rather than just the child's **SelectionObject**, to handle cases where the child's **LayoutChildren** method was not implemented to handle independent moves of the **SelectionObject**
- fixed some cases of undo not restoring link paths correctly
- fixed **GoSubGraph.ComputeInsideMarginsSkip** to skip child links that are connected to temporary ports, such as when reconnecting a link by **GoToolRelinking**
- added **GoCollapsibleHandleStyle** enum, for drawing a **GoCollapsibleHandle** in different styles
- added **GoCollapsibleHandle.Style** property
- **GoCollapsibleHandle** now inherits from **GoRoundedRectangle** rather than from **GoRectangle**, so you can set the **Corner** property to support rounded rectangular appearance
- added **GoCollapsibleHandle.Bordered** property, whether to draw the (rounded) rectangle
- added **GoCollapsibleHandle.FindCollapsible** method, to generalize and allow for customization of finding the **IGoCollapsible** object on which the handle needs to call

Collapse() or **Expand()**

- improved **FindCollapsible** method to search **Parent** chain, instead of only looking at immediate **Parent**, for the **IGoCollapsible** object

GoShapes, GoText, GoImage, GoControl et al.

- added static/shared **GoShape.DrawRoundedRectangle** method
- fixed a case of printing text with wrong font size

Nodes, Ports, and Links

- improved sizing of **GoButton** when there is no **Icon** and no **Label**
- added **GoButton.AutoRepeating** property, to support repeating behavior while user holds button down
- added **GoListGroup.TopIndex** property, to specify first visible item in list and allow independent sizing; defaults to -1, to show all items and not be resizable
- added **GoListGroup.ComputeMaximumItemSize**, to limit minimum size of listgroup when resizing
- added **GoMultiTextNode.FindPortIndex**, for convenience in finding the item/index for a port
- improved **GoLabeledLink.Pick** to respect the **RealLink's PickMargin**

Tools

- added **GoToolManager.DoKeyDown** handling of arrow keys, and Ctrl-Insert/Shift-Delete/Shift-Insert and Ctrl-Shift-Z (but override of **GoView.IsInputKey** is not in ReducedTrust version of Northwoods.Go.dll)
- added **GoTool.IsBeyondDragSize** predicate, for convenience in checking whether the mouse has gone far enough to be considered a drag instead of a sloppy click
- changed **IGoActionObject** by adding **OnActionActivated** method, to give access to **GoView** when activated
- added **GoToolAction.StartAutoAdjusting** and **StopAutoAdjusting** methods, and **AdjustedAutomatically** property, to support auto-repeating action objects, such as buttons that auto-repeat

Added Samples:

- **Planogrammer**, an editor for designing/organizing racks of items in a store

Added Sample Classes:

- **ClassDiagramNode**, like a UML Class node with a single **GoBoxPort**, that supports collapsing sections
- **CollapsingRecordNode**, like **RecordNode**, but with collapsing sections, that supports ports on each side of each item
- **ScrollingMultiTextNode**, inherits from **GoMultiTextNode**, but adds two buttons to support resizing and scrolling the **GoListGroup** inside the **GoMultiTextNode**
- **GradientColorLink**, a **GoLink** whose color changes gradually from one end to the other
- **LitIconicNode**, a **GoIconicNode** that supports highlighting of the Icon
- **MetafileView**, a **GoView** that can generate enhanced Windows metafiles (EMF), of the current view or of a collection of objects

Support

Northwoods Software provides e-mail support during the 30-day evaluation period and for 30 days after purchase. If you purchase the optional support subscription, you receive e-mail support for a year after purchase plus all new versions that are released during that period. For technical support, send e-mail to GoDiagram@nwoods.com.

For general sales and licensing questions, send e-mail to GoSales@nwoods.com.