

## GoDiagram<sup>™</sup> for ASP.NET Web Forms version 2.4.1 Release Notes

Copyright © 2002-2005 Northwoods Software Corporation

This kit includes sets of assemblies and samples for building interactive diagram web applications using ASP.NET WebForms.

GoDiagram<sup>™</sup> Web requires the Microsoft<sup>®</sup> .NET Framework 1.1 or 2.0 SDK, and ASP.NET. It also works with interactive development environments such as Visual Studio .NET 2003 or Visual Studio 2005.

GoDiagram Web works well with ASP.NET 2.0, but Visual Studio 2005 may introduce errors when its Conversion Wizard upgrades your web app to the recommended ASP.NET 2.0 style. See the section below, about upgrading to ASP.NET 2.0 with Visual Studio 2005.

GoDiagram Win for .NET Windows Forms is a separate product in a separate installation kit. GoDiagram Win and its additional assemblies share most of the same functionality as the ASP.NET Web Forms version of the GoDiagram components but are designed for .NET desktop/laptop applications.

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

Please note that SVG support is *not* yet integrated as an alternative rendering format by having **GoView** render SVG, potentially replacing the various kinds of images produced by the **GoWebImage** class. However, you can generate SVG files, including interactive ones, that can be viewed and manipulated by the end-user.

This installation is initially licensed under the terms of the evaluation license agreement, `WebEvalLicense.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\WebBinaryLicense.rtf`. Each product has its own redistributable assembly:

- GoDiagram Web: `Northwoods.GoWeb.dll`
- GoLayout Web: `Northwoods.GoWeb.Layout.dll`
- GoInstruments Web: `Northwoods.GoWeb.Instruments.dll`

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

- GoXml Web: `Northwoods.GoWeb.Xml.dll`
- GoSvg Web: `Northwoods.GoWeb.Svg.dll`

We suggest that you create new WebForm projects in Visual Studio and then copy the files you need from the desired sample application.

Please read the **GoDiagram Web Introduction** document, docs\GoWebIntro.doc. It includes a section about how to deploy your web 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
  - Northwoods.GoWeb.dll, containing the GoDiagram controls and classes
  - Northwoods.GoWeb.Layout.dll, the automatic layout library components (GoLayout) for GoDiagram Web
  - Northwoods.GoWeb.Instruments.dll, the instruments library (GoInstruments) for GoDiagram Web
  - Northwoods.GoWeb.Xml.dll, the library for the customizable reading and writing of XML files
  - Northwoods.GoWeb.Svg.dll, the library for the customizable generation of SVG files
  - \*.xml, XML documentation for code editor tooltips in Visual Studio
- .\docs
  - GoWebIntro.doc, introduction to GoDiagram Web
  - 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)
  - GoWeb.chm, API Reference Manual for GoDiagram, GoXml, GoSvg, GoLayout, and GoInstruments
  - GoWinWebDiffs.doc, a listing of the differences in GoDiagram between Windows Forms and Web Forms
  - WebEvalLicense.rtf, evaluation license agreement file
  - WebBinaryLicense.rtf, binary development license agreement file
- .\bin
  - LicenseManager.exe, to install developer's licenses for the GoDiagram products
  - CreateVDirs.bat, to create IIS virtual directories and applications for the sample web applications (this uses the \*.vbs files in the same directory)
- .\Samples, sample GoDiagram Web applications written in C#
  - Default.aspx, an HTML page that refers to the web applications held 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 some of the features of the instruments library
- **LayoutDemo**, demonstrates some of the features of the automatic layout library
- **MinimalApp**, a minimal GoDiagram application
- **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
- **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
- **Samples**, a short-cut to open up `Default.aspx` file in a browser
- **BuildSamples.bat**, a command file to recompile all of the samples, if they have not been converted by Visual Studio 2005
- **.\SamplesVB**, the same GoDiagram Web samples translated from C# to VB.NET
  - **BuildSamplesVB.bat**, a command file to recompile all of the samples, if they have not been converted by Visual Studio 2005

During installation, the sample web applications are installed as IIS applications under a virtual web directory named `GoDiagramWeb2.4.1Samples`. The installer uses `bin\CreateVDirs.bat` to define these sample applications. If the installation fails, or if you need to re-install the sample web applications, you can edit and run this command file. Alternatively, you can define by hand in the Internet Services Manager a virtual web directory named `GoDiagramWeb2.4.1Samples` that refers to the `.\Samples` subdirectory where you installed this kit. For each of the subdirectories you will need to edit the folder properties and Create an application.

Please note that the VB versions of the samples, and that subdirectory tree, are not installed as IIS/ASP.NET applications. You might not be able to open these projects in place using Visual Studio 2003. We recommend that you copy the sources into your own web application project.

Because the install kit executes the `bin\CreateVDirs.vbs` script to modify the IIS metabase, it is possible that your anti-virus system may warn you about running this script. If you don't trust our script, your anti-virus system might let you skip executing the script, finish the installation, and examine the script before executing `bin\CreateVDirs.bat` to install the sample web apps.

If you (re-)installed IIS after installing Visual Studio or the .NET SDK, you might be able to view the `Samples\default.aspx` page but not any of the sample web applications. The following Knowledge Base article provides a fix:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q306005>

Using Visual Studio 2003, you might not be able to start debug the web applications where they are installed. You can either attach to the process manually, or you can copy the web application into your own web site or site that Visual Studio creates. (Debugging is not a problem with Visual Studio 2005.)

When running in Visual Studio, you may need to specify the Start Page for each web application- it is always named `WebForm1.aspx`.

## 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**

## 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 version 2.4 DLLs, not an older version.

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

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

`Version=2.4.1.1`

## Upgrading to ASP.NET 2.0 with Visual Studio 2005

If you compile your application for ASP.NET 2.0 using command-line tools (not Visual Studio

2005), you shouldn't have any problems.

However, if you use Visual Studio 2005, please be aware that it will try to convert your web application to conform to the latest ASP.NET guidelines, and that the conversion may cause your web application to fail to compile or fail to run correctly.

### ***Resources***

Resource files (RESX files) may need to be moved, and code to access them may need to be altered, for them to work again. Select the project in the Solution Explorer, and either context-click to "Add ASP.NET Folder", or invoke the "Website" menu to "Add ASP.NET Folder". Choose "App\_GlobalResources". Move your RESX file into that folder. Then change code such as:

```
GoImage.DefaultResourceManager = new  
ResourceManager( typeof( WebForm1 ) );
```

to:

```
GoImage.DefaultResourceManager =  
Resources.WebForm1.aspx.ResourceManager;
```

This makes use of the new ASP.NET 2.0 feature of typed resources.

### ***Licenses***

Visual Studio 2005 and ASP.NET 2.0 have improved support for licensing, similar to that used by Windows Forms applications. If you are upgrading from ASP.NET 1.1 to ASP.NET 2.0, you will need to remove any **VersionName** assignments that you had in your Global **HttpApplication** constructor.

Instead, you should make sure your web application includes a LICENSES.LICX file. You can copy this file from one of the sample web applications that makes use of the same licensed components. Visual Studio 2005 will compile this file and produce a App\_Licenses.dll assembly in the Bin subdirectory. You should make sure the LICENSES.LICX file is kept up-to-date with lines corresponding to each kind of licensed component that you use, and that its four-part version number is also accurate.

## **Known Bugs and Misfeatures**

- There are a number of Microsoft bugs that you may encounter:
  - SOAP serialization
  - text clipping

## **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, although some code changes may be needed--see the cautions listed above regarding upgrading using Visual Studio 2005.

## 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**; 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**; 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
- 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
- GoWeb.js added -khtml-user-drag:none style for Safari support
- GoWeb.js improved mouse event position when page is scrolled in Mozilla/Firefox

## 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 **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 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

## 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](mailto:GoDiagram@nwoods.com).

For general sales and licensing questions, send e-mail to [GoSales@nwoods.com](mailto:GoSales@nwoods.com).