

GoDiagramTM Express for .NET Windows 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 applications using Windows Forms in the full .NET Framework.

GoDiagramTM Express 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.

This installation is initially licensed under the terms of the evaluation license agreement, `ExpressEvalLicense.rtf` in the `docs` subdirectory. Upon purchasing a valid binary development license for GoDiagram Express on a particular computer, and after installing it with the `LicenseManager` application, you may compile a license for `Northwoods.GoExpress.dll` into your application and distribute `Northwoods.GoExpress.dll`. The binary development license agreement is `docs\ExpressBinaryLicense.rtf`. There are three redistributable assemblies:

- `GoDiagram Express: Northwoods.GoExpress.dll`
- `GoXml Express: Northwoods.GoExpress.Xml.dll`
- `GoSvg Express: Northwoods.GoExpress.Svg.dll`

Please read the **GoDiagram Express Introduction** document, `docs\GoExpressIntro.doc`. It includes 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\GoExpressFAQ.doc`.

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.GoExpress.dll`, containing the GoDiagram Express controls and classes
 - `Northwoods.GoExpress.Xml.dll`, the library for the customizable reading and writing of XML files
 - `Northwoods.GoExpress.Svg.dll`, the library for the customizable generation of SVG files
 - `*.xml`, XML documentation for code editor tooltips in Visual Studio
- `\docs`
 - `GoExpressIntro.doc`, introduction to GoDiagram Express
 - `GoExpressFAQ.chm`, Frequently Asked Questions list
 - `GoExpressUserGuide.doc`, User Guide for GoDiagram Express
 - `GoExpress.chm`, API Reference Manual for GoDiagram Express
 - `ExpressEvalLicense.rtf`, evaluation license agreement file
 - `ExpressBinaryLicense.rtf`, binary development license agreement file

.\bin

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

.\Samples, sample applications written in C#

- *.exe, sample applications using GoDiagram Express
- 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
 - **MinimalApp**, a minimal GoDiagram application
 - **OrgCharter**, a simple organization chart editor
 - **ProtoApp**, prototype application
 - **StateCharter**, a simple state diagram editor
 - **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 Express samples translated from C# to VB.NET

- `BuildSamplesVB.bat`, a command file to recompile the sample applications

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**

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.GoExpress.dll` version 2.4 assembly, not an older version.

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

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

Version=2.4.1.1

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.GoExpress 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.
- 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**, 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**, **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**
- 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

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 **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

Added Sample Classes:

- **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.