

# GoDiagram Differences, Win and Pocket

Copyright © 2002-2008 Northwoods Software Corporation

GoDiagram™ Win for Microsoft® .NET Windows Forms is a .NET class library containing a set of Windows Forms controls for easily building interactive diagrams in stand-alone .NET-based applications.

GoDiagram™ Pocket for Microsoft® .NET Compact Framework is a .NET class library containing a set of Windows Forms controls for easily building interactive diagrams that run on a Pocket PC.

The two class libraries share nearly all of the design and implementation with each other. The User Guide provides details about GoDiagram, most of which apply to both products. You will need to read that document before you can really make good use of either GoDiagram product.

## Differences Between the Assemblies

The principal difference between the two products is that GoDiagram Pocket depends on the .NET Compact Framework whereas GoDiagram Win depends on the standard .NET Framework.

Nearly all of the GoDiagram classes are almost identical, so much of your source code can be transferred between projects with minimal change. You just need to refer to the `Northwoods.GoPocket.dll` assembly instead of the `Northwoods.Go.dll` assembly. The namespaces for the classes are the same: **Northwoods.Go**.

The `Northwoods.Go.Draw.dll` and `Northwoods.Go.Svg.dll` assemblies are not supported in GoDiagram Pocket.

In general the differences are due to types and concepts that were taken out or do not apply in a .NET Compact Framework application:

- Differences in **System.Windows.Forms.Control**
- Notions of mouse-over or hovering, and Windows drag-and-drop between windows
- No mouse buttons, except that a long click is treated as a context-click
- No mouse wheel
- Missing **System.Windows.Forms** types: **Cursor**, **DragEventArgs**
- No printing
- No **GoDrawing** class or **GoFigure** enumerated type
- No non-solid brushes or pens, or non-zero pen widths for shapes, except that **GoStroke** has additional Pocket-specific properties for controlling the pen width
- No drawing or bitmaps with alpha-blending
- Some differences in text formatting
- No XOR drawing mode, including when using the rubber band tool
- No **GoPartInfo**, due to no need for JavaScript generation

Although a GoDiagram Pocket application may in certain circumstances be able to run on your desktop using the standard .NET Framework (that is, without using the PocketPC emulator), such use is not supported.

For those of you who are converting a GoDiagram Win application into a GoDiagram Pocket application, here is a detailed list of differences. (This list might not be complete or accurate.)

- **GoView** changes:
  - **Control** changes:
    - implemented **BackgroundImage** property
    - changed **HandleScroll** method
    - added **OnResize** method override
    - added **OnPaintBackground** override (that does nothing)
    - removed **OnSizeChanged**, **OnCreateControl**, **OnVisibleChanged** method overrides
    - removed **OnBackColorChanged**, **OnBackgroundImageChanged**, **OnStyleChanged**, **OnSystemColorsChanged** method overrides
  - Mouse/pointer changes:
    - removed **OnDoubleClick**, **OnMouseWheel** method overrides
    - removed **DoMouseWheel**, **DoMouseHover**, **DoHover**, **DetectHover** methods
    - removed **HoverDelay** property
    - removed **ObjectHover** event and **OnObjectHover** and **RaiseObjectHover** methods
    - removed **BackgroundHover** event and **OnBackgroundHover** and **RaiseBackgroundHover** methods
    - removed **ObjectEnterLeave** event and **OnObjectEnterLeave** and **RaiseObjectEnterLeave** methods
    - removed **DoObjectEnterLeave** method
    - removed **OnMouseLeave** method override
    - removed **DoMouseOver**, **DoBackgroundMouseOver** methods
  - No drag-and-drop between windows:
    - removed **OnDragOver**, **OnDragEnter**, **OnDragLeave**, **OnQueryContinueDrag**, **OnDragDrop** method overrides
    - removed **DoInternalDrag**, **DoExternalDrag**, **GetExternalDragImage** methods
    - removed **DoInternalDrop**, **DoExternalDrop** methods
    - removed **ExternalObjectsDropped** event and **OnExternalObjectsDropped**, **RaiseExternalObjectsDropped** methods
    - changed **AllowDragOut** property to always be false
    - implemented **AllowDrop** property, which is always false
  - No printing:
    - removed **Print**, **PrintShowDialog**, **PrintPreview**, **PrintPreviewShowDialog** methods
    - removed **PrintDocumentPage**, **PrintView**, **PrintDecoration** methods
    - removed **PrintDocumentSize**, **PrintDocumentTopLeft**, **PrintScale**, **PrintsViewObjects** properties
  - GDI+ differences:
    - removed **DrawXorLine**, **DrawXorRectangle** methods
    - removed **SmoothingMode**, **TextRenderingHint** properties
    - removed **GridPenWidth**, **GridPenDashStyle** properties
    - implemented **PointF** structure
  - Other Windows Forms differences:
    - removed **Border3DStyle** property
    - added **ContextClickTime** property

- removed **DefaultCursor**, **Cursor** properties
- implemented **SortOrder** enum
- implemented **ToolTip** (as **GoObject** shown in **GoView**, upon a single click since there's no mouse-over)
- **GoObject**: removed **OnHover**, **OnEnterLeave**, **OnMouseOver** methods
- **GoShape**: removed **Fill...** methods, defining gradient brushes
- **GoStroke**: added **PenWidth**, **HighlightPenWidth** properties
- **GoText**: removed **StringTrimming** property
- **GoLabeledLink**: added **PenWidth**, **HighlightPenWidth** properties (from **.RealLink**)
- **GoInputEventArgs**: removed **EventArgs** property

## Differences in the Sample Applications

Because the non-**GoView** code for both platforms is nearly identical, all of the instructive example code would be the same. Thus only the MinimalApp application has been ported from GoDiagram Win.