

7,028,527 members and growing! (19,910 online)

lubita7052 | Settings | **Watched Items** | Bookmarks | My Articles | Sign out

Makes You Smarter*

[Home](#) [Articles](#) [Quick Answers](#) [Message Boards](#) [Job Board](#) [Catalog](#) [Help!](#)
[Lounge](#)[Desktop Development](#) » [Miscellaneous](#) » [General](#)License: [A Public Domain dedication](#)

VC6, VC7, VC7.1, Win2K, WinXP, Win2003, MFC, Dev

Posted: **4 Apr 2004**
 Updated: **23 Jun 2006**
 Views: **205,388**
 Bookmarked: **218 times**

CDiagramEditor - DIY vector and dialog editor

By [Johan Rosengren](#)

A feature rich vector editor skeleton.

Announcements

- [Post a VS2010 Tip or Trick, Win a Zune HD](#)
- [VS2010 Tech Summit Free, on-demand](#)
- [Monthly Competition](#)

The Daily Insider

[That Lost 4G Phone](#)
 Daily IT news: [Signup now.](#)

Articles

Desktop Development
 Button Controls
 Clipboard
 Combo & List Boxes
 Dialogs and Windows
 Desktop Gadgets
 Document / View
 Edit Controls
 Files and Folders
 Grid & Data Controls
 List Controls
 Menus
Miscellaneous
 Printing
 Progress Controls
 Selection Controls
 Shell and IE programming
 Smart Client
 Splitter Windows
 Static & Panel Controls
 Status Bar
 Tabs & Property Pages
 Toolbars & Docking windows
 Tree Controls
 Web Development
 Mobile Development
 Enterprise Systems
 Database
 Multimedia
 Languages
 Platforms, Frameworks & Libraries
 General Programming
 Graphics / Design
 Development Lifecycle
 General Reading
 Third Party Products
 Mentor Resources

Services

[Product Catalog](#)
[Job Board](#)
[CodeProject VS2008 Addin](#)

Feature Zones

[Product Showcase](#)
[WhitePapers / Webcasts](#)
[.NET Dev Library](#)
[ASP.NET 4 Web Hosting](#)



Search

[Print Friendly](#) [Share](#) [Bookmark](#) [Discuss](#) [Report](#)

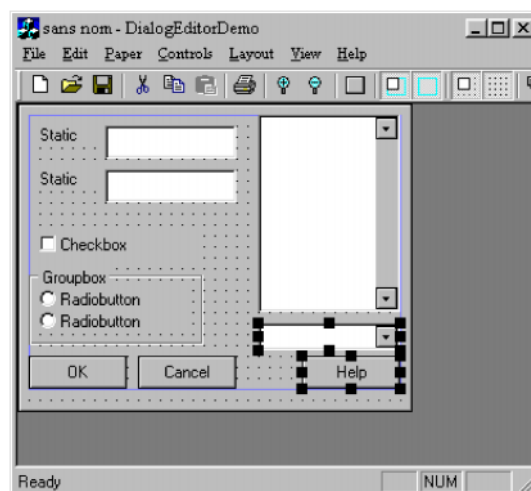
143 votes for this article. Popularity: 10.18 Rating: 4.72 out of 5

[Advanced Search](#)
[Add to IE Search](#)

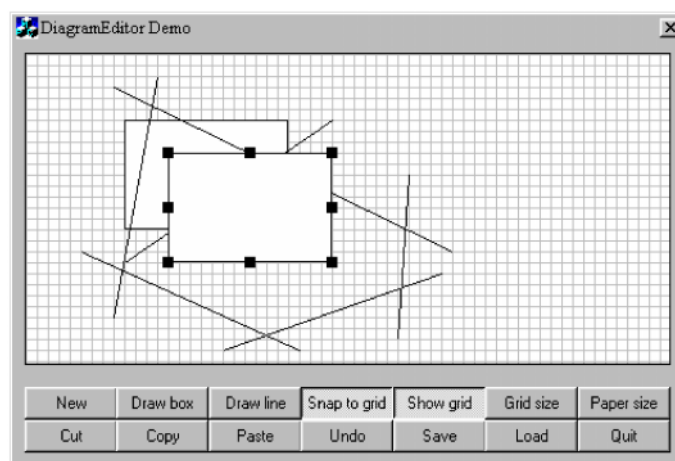


Is your email address OK? You are signed up for our [newsletters](#) but your email address is either unconfirmed, or has not been reconfirmed in a long time. Please click [here](#) to have a confirmation email sent so we can confirm your email address and start sending you newsletters again.

- [Download source files - 63.9 Kb](#)
- [Download DialogEditorDemo demo project - 116 Kb](#)
- [Download DiagramEditorDemo demo project - 80.6 Kb](#)
- [Download CDiagramEditor documentation - 67.1 Kb](#)
- [Download the demo DialogEditorDemo EXE file - 40.3 Kb](#)
- [Download the demo DiagramEditorDemo EXE file - 27.2 Kb](#)



The DialogEditorDemo app



The DiagramEditorDemo app

Introduction

So you wanted to add a forms editor to your application? A dialog editor? Something that allows drawing of HTML `<div>`s? Here is a feature rich skeleton (!) to get you started. **CDiagramEditor** is a package that gives

you a basic visual editor intended for vector objects. Although perhaps not sufficient to create a CAD-application, you'll indeed be able to create a dialog editor. The editor itself is derived from `CWnd`, with a separate class handling the data, so you'll be able to either use it in a dialog or doc/view application.

Some of the built-in features:

- Full mouse handling. Draw and drag multiple objects, drag draw, and resize single ones.
- Paper size, margin, and grid easily configured.
- Snap to grid, restrain moving and resizing to margin or paper. Or draw outside the paper, if you want.
- Configurable keyboard interface.
- Plug-in popup menu for the editor.
- Unlimited zoom.
- Unlimited undo.
- Support for saving and loading to `CStrings/CStringArrays`.
- For the programming aficionado, the different drawing functions of the editor are virtual, so if you're not satisfied with the look, derive a class and do it yourself.
- Base objects for rectangles and lines, you derive the drawing objects you need from those.
- Support for object-specific popup menus.
- Support for object-specific property dialogs.

And more. Of course, this package is fiendishly difficult to use? No, add an instance of `CDiagramEditor` to your app, derive the drawing classes you need, and you have an editor.

I've added two demo applications. The smaller one, **DiagramEditorDemo**, shows the basics, adding an editor to a dialog application.

DialogEditorDemo is a SDI doc/view application showing how to use the separate data container class, how to plug into the document loading and saving, and using a factory to create the drawing objects, how to add print and print preview, custom background redraw, exporting, and much, much more.

The package consists of the following classes:

- `CDiagramEditor` - The editor itself.
- `CDiagramEntityContainer` - A container class for the editor data.
- `CDiagramEntity` - Base class for the drawing objects.
- `CDiagramLine` - A `CDiagramEntity`-derived class that can be used as a base class for line objects.
- `CDiagramMenu` - Popup editor menu.
- `CDiagramPropertyDlg` - Pure virtual base class for object property dialogs.

and a few small help classes.

As it is quite a lot of code, I've added HTML-documentation with an overview of the classes, and a page of How-tos and class details for the bigger classes. The code is commented as well.

History

The beginning of time

Original release.

1/5 2004

Working intensely with the `CDiagramEditor`, I've made a few enhancements to the package, mainly to:

1. Facilitate deriving from `CDiagramEntityContainer`.
2. Make it possible to use the same internal clipboard for several editors.

The second modification is to be able to use `CDiagramEditor` in a MDI-application, where you might want to copy and paste objects across editors. It is accomplished by separating out the copy/paste-functionality from `CDiagramEntityContainer` and putting it in a class of its own - `CDiagramClipboardHandler`.

Especially irritating, while doing these modifications using a MDI-app as the host, was that even though property dialogs were created with the editor as the parent, the parent returned calling `GetParent` in the `OnOK`-handlers. Thus, the editor was not updated correctly when changes were made to the objects. I had to work around this by adding an explicit redraw-parent to `CDiagramPropertyDlg` and adding a new member `Redraw` that should be used instead of `GetParent() -> RedrawWindow()` in derived classes for MDI-applications.

Other than that, I must admit that refactoring **is** a major hassle in MSVC++ 6.0, something I thought only wimps complained about earlier (have I been lucky in the small amount of refactoring necessary in my life up to this point?).

The documentation is updated accordingly, and the changes are:

CDiagramEditor

- `SendMessageToObjects` sets the selected-flag to `TRUE` - sending the message only to selected objects.
- Made some message mapping functions virtual to allow enhancements in derived classes.
- Added accessors for states to allow access for derived classes.
- Set interact mode to `MODE_NONE` after Ctrl+clicking item (to avoid undesired movement of lines).
- Set scrollbar positions to zero in `Clear`. Earlier, when loading a document, the scrollbars were not reset to 0,0.
- Sending this window as a parent to the property dialog. This is to redraw the editor properly after changes in the dialog in MDI-apps.
- Changed C-style casts to `static_cast`.
- Removed `ShowProperties-const`-ness.

CDiagramEntityContainer

- Made `RemoveAt` virtual.
- Made several undo- and copy/paste functions virtual. Added array accessors for derived classes. Moved the member function `Find` to the `protected` section. All this to simplify deriving from `CDiagramEntityContainer`.
- Copy/paste-handling removed to a separate class (`CDiagramClipboardHandler`) to allow several containers to share the same clipboard.
- Changed C-style casts to `static_cast`.

CDiagramEntity

- Changed accessors for `m_type` - `SetType` and `GetType` - to `public`.
- Added colon as a replace-character for saving in `FromString` and `GetString`. Earlier, names or titles with embedded colons would have been invalid.
- Added a redraw parent to the property dialog in `ShowProperties`.

CDiagramPropertyDlg

- Added a redraw parent, `m_redrawWnd`, to redraw the proper window in MDI-applications. Derived classes can now use `Redraw` if the editor needs to be updated.

If you are using `CDiagramEditor` in an existing project, the things affecting you will be *Set interact mode...*, *Set scrollbar positions...*, and *Added colon as a replace-character...* above. To use the new package, download the source code again and copy the files over the existing ones, and add `CDiagramClipboardHandler.cpp` and `CDiagramClipboardHandler.h` to your project.

15/5 2004

Bug fix in the dialog editor demo

The `Export` function of the `CDiagramEntity`-derived objects in the demo wasn't updated to `const`, and so was not called by the export-mechanism.

13/6 2004

This is definitely not an essential update. Mainly, I'm updating the framework because of an upcoming UML-editor, where a few more things have been made `virtual` or `const` to ease derived editors. Anyhow, on with the list:

Virtual size checks when scrolling

Graham suggested this (see below). Although the scroll position functionality ought to have reported correct values, it's better safe than sorry, so I've included page check in the `VScroll` and `HScroll` routines - not unnecessarily trying to scroll an editor window with no scroll bars.

Modifications for derived editors

- `CDiagramEditor`
 - Added access to `m_subMode` in `SetInteractMode`.
 - Made `OnObjectCommand` virtual to allow interception in derived classes.
 - Made `GetBackgroundColor` `const`.
 - Made `SelectAll` virtual.
- `CDiagramEntityContainer`
 - Made `GetAt` virtual.
 - `RemoveAll`, added check to see if there are any items in the object array.
 - Made `RemoveAll` access data container objects directly, to avoid chained deletes in derived classes. Applies if a `delete` should delete other items automatically.

Who should update the editor code? Anyone experiencing problems while trying to scroll outside the paper area?

8/7 2004

Time for another update! I've added some bug corrections, some enhancements to further assist derived editors, and added some new functionality. The corrections are:

- Corrected a bug in `SetZoom`, where minimum size was not honored (**pgrohs**).
- Zeroing `m_internalData` after a `delete` in `SetDiagramEntityContainer` to avoid crashes (**pgrohs**).
- Corrected bug in alignment-handling in `LeftAlignSelected` et. al. - all, not only selected items where aligned.
- Correction in `VirtualToScreen`, not deducting scrollbar positions any longer (**Wolfgang Busch**).
- Added `\\newline` as a replace-character to allow newlines in titles and names when saving/loading (**Unruled Boy**).

Enhancements:

- Made `AddObject` virtual and added the virtual `MoveObject` function. This is to allow trapping of added or moved objects in the editor in derived classes (**sunmoon9898**).
- Made clipboard-functions and `DeleteAllSelected` virtual.
- Added a virtual `GetCursor` function to simplify the addition of customized cursors..
- Coordinate conversion functions made public (**Wolfgang Busch**).
- Made `Add` and `Remove` in the container virtual.
- Added a `GetSelectCount` to the container.

New functionality:

Groups

Suggested by **Unruled Boy**

The selected objects can now be grouped by right-clicking the editor and selecting Group. Grouped objects can be moved, copied, and

pasted as a single unit. By right-clicking and selecting Ungroup, all grouping will be removed for the selected items.



Programmatically, this can be accomplished by calling the public functions `CDiagramEditor::Group` and `CDiagramEditor::Ungroup`. `CDiagramEditor::UpdateGroup(CCmdUI* pCmdUI)` and `CDiagramEditor::UpdateUngroup(CCmdUI* pCmdUI)` can be called as command enablers from - for example - the view.

Grouping is implemented as a new attribute in `CDiagramEntity`.

Caveat

Old data files for the demo applications based on `CDiagramEditor` are no longer compatible with the framework due to the new base attribute. I've considered this an acceptable tradeoff, and hope it is not an inconvenience. Derived works will have to be updated to load/save this attribute, and I've added functionality in `CDiagramEntity` to simplify this, `LoadFromString`, `GetDefaultGetString`, `GetHeaderFromString`, and `GetDefaultFromString`.

A typical `FromString` in a hypothetical `CDiagramEntity`-derived class, `CSomeEntity`, with an extra `CString` attribute. `m_someattribute` can now be written as:

```

BOOL CSomeEntity::FromString( const CString& str )
{
    BOOL result = FALSE;
    CString data( str );

    if( LoadFromString( data ) )
    {
        CTokenizer tok( data );

        CString someattribute;
        tok.GetAt( 0, someattribute);



        SetSomeAttribute( someattribute );
        result = TRUE;
    }

    return result;
}

```

where `CDiagramEntity::LoadFromString` is used to set the base class attributes, and the rest of the attributes are parsed from the modified input-string.

`GetString` can be written as:

```

CString CSomeEntity::GetString() const
{
    str.Format( _T( "%s;" ), // Note the starting comma
        GetSomeAttribute()
    );

    str = GetDefaultGetString() + str;
    return str;
}

```

Mouse-wheel support

Suggested by **John A. Johnson**.

By handling `WM_MOUSEWHEEL` in `CDiagramEditor`, the paper can now be scrolled with the mouse wheel. I've thought about making this a bit more flexible. Configurable scroll steps and zoom comes to mind, but this will have to wait. `OnMouseWheel` is made virtual to allow personal attempts in this direction :-)

Panning

Suggested by **John A. Johnson**.

Panning is implemented. By pressing the middle-button, a panning marker is displayed at the point of the cursor. If the mouse is moved, the editor is scrolled in the direction of the cursor. The scroll-amount is relative to the distance between the marker and the current cursor. The cursor changes to imply the current cursor direction. I've used built-in cursors for this, to avoid dependencies on resources.

I checked P J Arends' excellent class for this, `CWindowScroller`, and it was indeed possible to plug it in directly in the editor, but as I want to have a completely free license for the `CDiagramEditor`, I based the work on MSDN (The Visual Programmer article on panning by George Shepherd and Scot Wingo) instead.

Key functions for this are `OnMButtonDown`, `SetPanning`, and `DrawPanning`.

Zoom to fit

Suggested by **John A. Johnson**.

This function will set the zoom level so that all objects in the current diagram are visible. The function to call is the public function `ZoomToFitScreen`.

Finally, a **very big thanks** to all those giving feedback (especially those attributed above - but **all** feedback is welcome) on the

framework.

4/8 2004

A handful of minor fixes and additions this time.

- Added scroll wheel mode. By calling `SetScrollWheelMode(WHEEL_SCROLL/WHEEL_ZOOM)`, the scroll wheel will either scroll or zoom.
- Added virtual to more message handlers, among them the scrollbar ones.
- Added virtual to `SetZoom`.
- Added virtual functions to set the scrollbar positions to get one single point where this is made.
- **Bug:** Checking if the cursor is outside of restraint when placing an item in `OnLButtonDown`.
- Added `ScrollIntoView` commands. The function scrolls the currently selected object(s) into view.
- **Bug:** Fixed bug with selection markers - virtualization of the mouse coordinates was in discord with the selection marker rectangles (**Marc G**).
- Added `SelectAll` and `UnselectAll` to the container.

28/8 2004

Far too early, here comes, nevertheless, a bug correction round. Once again, a big, fat **thanks** to you all for the feedback!

- `CDiagramEditor`: added check for non-normalized rect when hit-testing in `OnLButtonDown`, as lines have them.
- `CDiagramEntity`: setting `m_parent` to `NULL` in the `ctor` (**Marc G**).
- `CTokenizer`: changed a `char` to `TCHAR` to allow UNICODE builds (**Enrico Detoma**).

25/3 2005

A maintenance update, with a few additions:

- Made `UnselectAll` virtual (**Grisha Vinevich**).
- Added the minimum step size one pixel in `OnKeyDown` to avoid 0-pixel movement for arrow keys (**Graham G Pearson**).
- Made `UnselectAll` virtual (**Grisha Vinevich**).
- Added the `PopUndo` function to pop the latest undo item from the stack.
- Made `IsUndoPossible` const.
- Made `SetParent/GetParent` public.

as well as a correction to the line-selection mechanism (**Graham G Pearson**).

15/5 2005

Yet another maintenance update:

- Routing all object selections through a virtual member function to allow customization of the selections (**Janiv Ratson**).
- Added a Redo-command. This command will undo the last undo. Handlers and enablers as for Undo, but the message handled is, of course, `ID_EDIT_REDO`. Key-mappings to Ctrl+Y (**Janiv Ratson**).
- Added message mapping for `ID_EDIT_UNDO` (**Graham G Pearson**).
- Added message mapping for `ID_EDIT_REDO`.

Note: I've only updated the source code for the [Download source files](#)-link - not the demo projects. So, make sure you use this download when you incorporate the code into your own project!

23/6 2006

Long overdue, another update:

- Added the `PasteToPosition` function (**Janiv Ratson**)
- **Bug:** Corrected bug in `ZoomToFit` (**DanMoshe**)
- **Bug:** Corrected bug when resizing several grouped objects with the mouse (**obeea**)
- **Bud:** Added dirty-parameter to `SendMessageToObjects` for the dirty handling - not all commands should set the document as changed (**Janiv Ratson**)
- **Bug:** Corrected `CDiagramEntityContainer` undo-stack handling to take stack size 0 into consideration (**David Hoos**)
- **Bug:** Added the group to the data copied in `CDiagramEntity::Copy` (**JeffBean**)
- Minor layout changes, and removal of redundant scrollbar handling (**Alexey Shalnov**)

Note: Once again, I've only updated the source code for the [Download source files](#)-link - not the demo projects. So, make sure you use this download when you incorporate the code into your own project!

License

This article, along with any associated source code and files, is licensed under [A Public Domain dedication](#)

About the Author

Johan Rosengren

45 years old, married, three kids.

Started with computers more than 20 years ago on a CBM-64.

Read Theoretical Philosophy at the University of Lund.

Working as a C++ consultant developer.

Science-fiction freak. Enjoy vintage punkrock.

Occupation: Software Developer (Senior)



Company: Abstrakt Mekanik AB
Location: Sweden

Member

Other popular Miscellaneous articles:

- [MFC Grid control 2.26](#)
A fully featured MFC grid control for displaying tabular data. The grid is a custom control derived from CWnd
- [Themed Windows XP style Explorer Bar](#)
A fully customizable Windows XP style Explorer Bar that supports Windows XP themes and animated expand/collapse with transparency.
- [TaskbarNotifier, a skinnable MSN Messenger-like popup in C# and now in VB.NET too](#)
The TaskbarNotifier class allows to display an MSN Messenger-like animated popup with a skinned background
- [CPPToolTip v2.1](#)
A class that allows you to display your data for a control as tooltip
- [Formula Editor](#)
Formula-editor for editing and exporting mathematical content



Article Top

Rate this article for us! Poor ☐ ☐ ☐ ☐ ☐ Excellent



FAQ

Noise Tolerance Layout Per page

New Message

Msgs 1 to 25 of 246 (Total in Forum: 246) ([Refresh](#))
First Prev Next

Great Work!!!! Very Useful for SCADA IDE development	gsheladia	23:22 17 Aug '09
Bezier curves handling?	jasperlotus	9:03 20 May '09
Great work! I am wondering how to keep the virtual page just the same size of the client window?	tiger54910	18:36 19 Mar '09
Re: Great work! I am wondering how to keep the virtual page just the same size of the client window?	Johan Rosengren	1:22 22 Mar '09
This is a great work !!	arif.setiawan	21:36 25 Nov '08
Re: This is a great work !!	Johan Rosengren	0:00 26 Nov '08
Draw Text??	Member 4496270	4:46 30 May '08
Re: Draw Text??	Johan Rosengren	23:20 30 May '08
How to draw an Activex object?	kstone	6:19 28 May '08
Re: How to draw an Activex object?	Johan Rosengren	12:00 28 May '08
Thanks ,I using your code in my SCADA project	confideman2000	18:40 15 May '08
Re: Thanks ,I using your code in my SCADA project	Johan Rosengren	23:10 16 May '08
HI how to create that stand alone exe file ??	satishmanohar	4:13 24 Jan '08
Re: HI how to create that stand alone exe file ??	Johan Rosengren	18:38 24 Jan '08
Re: HI how to create that stand alone exe file ??	satishmanohar	19:36 24 Jan '08
Re: HI how to create that stand alone exe file ??	Johan Rosengren	11:35 25 Jan '08
Re: HI how to create that stand alone exe file ??	satishmanohar	22:51 25 Jan '08
Only to let you know	John A. Johnson	13:29 3 Jan '08
Re: Only to let you know	Johan Rosengren	21:08 3 Jan '08
The latest version	li0000	8:49 25 Nov '07
Re: The latest version	Johan Rosengren	10:18 25 Nov '07
Paper size	li0000	8:43 25 Nov '07
Re: Paper size	Johan Rosengren	10:17 25 Nov '07
Re: Paper size	li0000	11:04 25 Nov '07
Re: Paper size	Johan Rosengren	22:13 10 Jan '08

Last Visit: 23:56 14 Apr '10
Last Update: 0:02 2 May '10
1 2 3 4 5 6 7 8 9 10 Next »

General
 News
 Question
 Answer
 Joke
 Rant
 Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+PgUp/PgDown to switch pages.

[PermaLink](#) | [Privacy](#) | [Terms of Use](#)
 Last Updated: 23 Jun 2006
 Editor: [Smitha Vijayan](#)

Copyright 2004 by Johan Rosengren
 Everything else Copyright © [CodeProject](#), 1999-2010
 Web24 | [Advertise on the Code Project](#)

