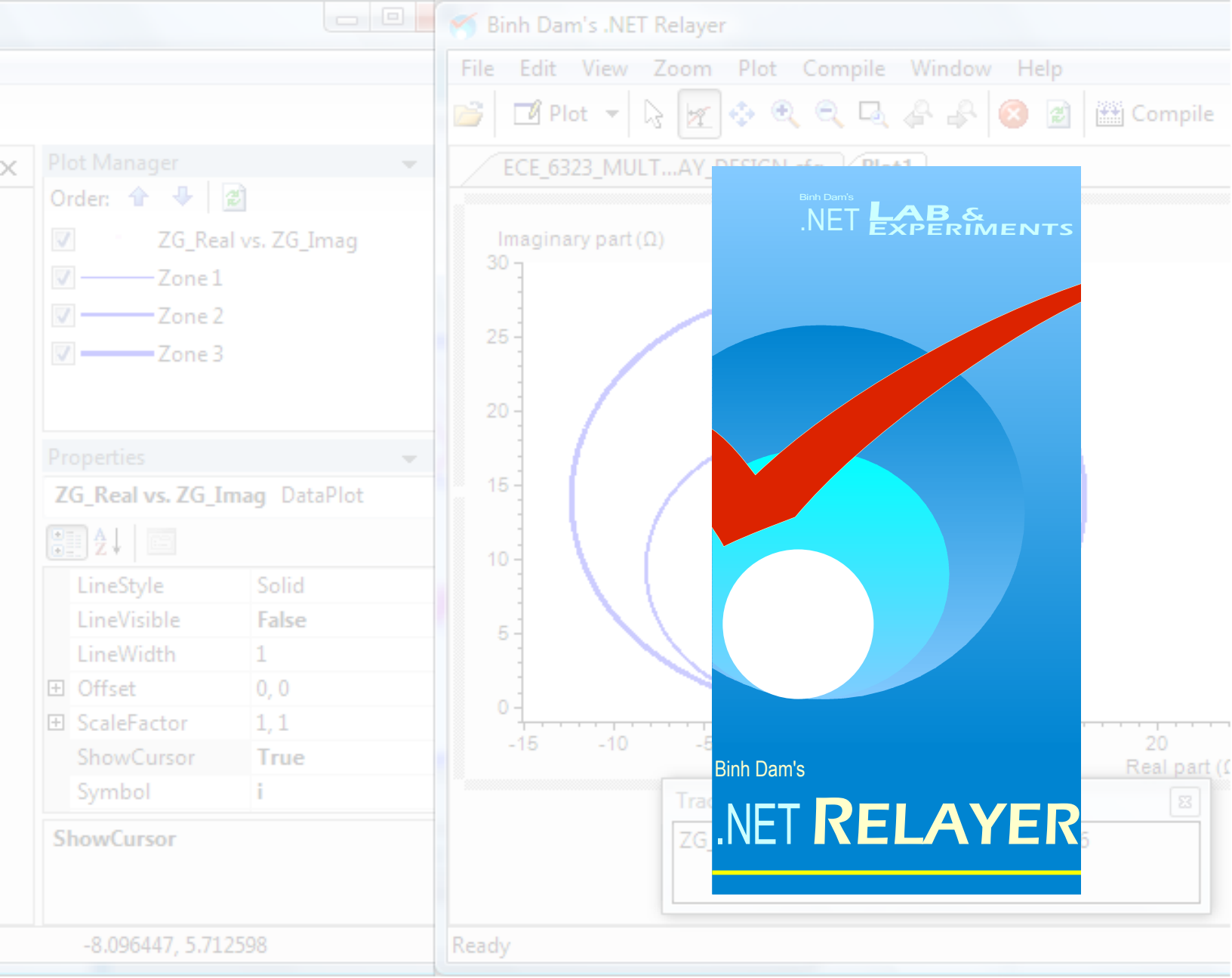


.NET Relayer Software
Quick Start Guide

IEEE COMTRADE Viewer and
Software for Protective Relay Simulation and Experiments

Applies to .NET Relayer v2.1.3381

Contents

Product Background and Overview

Thank you for your interest in the **.NET Relayer** software!
Binh Dam's .NET Relayer is an award-winning software for relay algorithm simulation and experiments. It is also a powerful viewer for IEEE COMTRADE files. The key innovation of .NET Relayer is to allow relay algorithms to be written on-the-fly and tested instantly against waveforms prerecorded in the COMTRADE format.

Power systems protection relies on advanced algorithms design and testing. The **.NET Relayer** software (formerly known as **Waveform Analyzers**) facilitates the design and testing of relay algorithms by leveraging the dynamic code generation capabilities of the .NET Framework and the features of the **.NET Lab and Experiments** software from the author.

.NET Relayer is a comprehensive viewer for IEEE COMTRADE files. Recorded voltage and current waveforms and other relay signals are easily displayed in tabular and graphical format. More information on the COMTRADE format is available from IEEE.

The **.NET Relayer** software is also suited for general waveform analysis. Relay algorithms and other signal processing methods can be implemented to analyze and respond to pre-recorded voltage and current waveforms. Users can create their own analyzers or use the built-in analyzer samples to analyze waveforms.

.NET Relayer has the following capabilities:

- IEEE COMTRADE file viewer
- Grid and graphical display of waveform data
- Easy setup of distance relay protection zones
- Mathematical plots (C# and Visual Basic expressions)
- Write and instantly test your own relay functions (unlimited arguments and parameters, C# and Visual Basic)
- Sample built-in relay functions: disk overcurrent, directional, and modified distance relay (to be updated on request).
- Support for analysis events and reports
- Sample waveforms, waveform generators, and sample waveform analyzers (QuickAnalyzer, C# DLL project, VC DLL project) included

.NET Relayer is regularly expanded to complement the capabilities of existing tools and to support relevant research activities. You may submit feedback and feature requests by contacting the author (see contact info at the end of this guide).

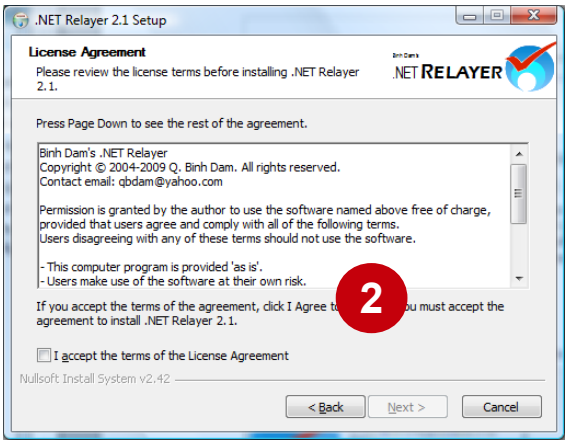
Installation

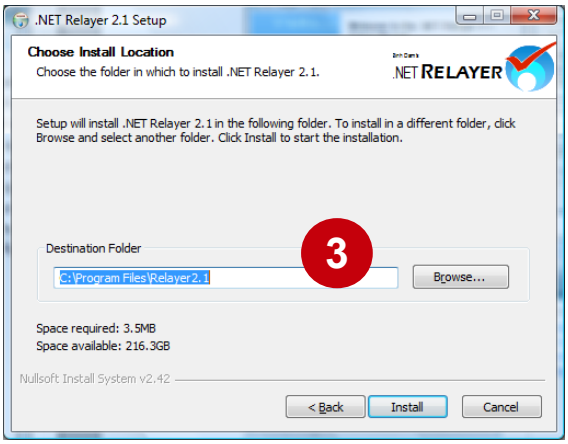
Compatible software and operating systems:

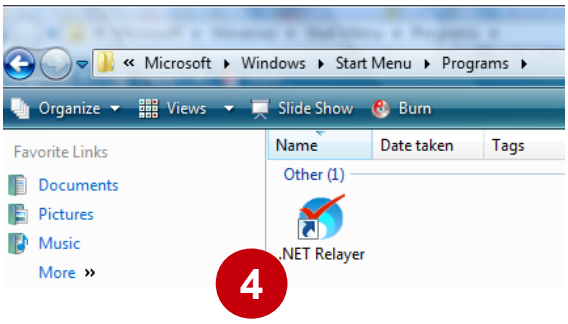
- Windows 98, ME, NT4, 2000, XP (32-bit), Vista (32-bit)
- .NET Framework versions 1.0, 1.1, 2.0, 3.0, 3.5, 4.0.

An installation program automatically configures the **.NET Relayer** software for immediate use on your computer. You may need administrator privileges to install the software on recent operating systems. If you downloaded the software in ZIP format, you need to extract the setup program from the ZIP file. The steps of the installation program are:

1. Welcome screen
2. License agreement
3. Install location selection
4. Program shortcuts (Desktop and Start Menu | Programs)







To uninstall **.NET Relayer**, use the Add/Remove Programs command from the Windows control panel.

What's New in .NET Relayer 2.1?

New Features for .NET Relayer 2.1

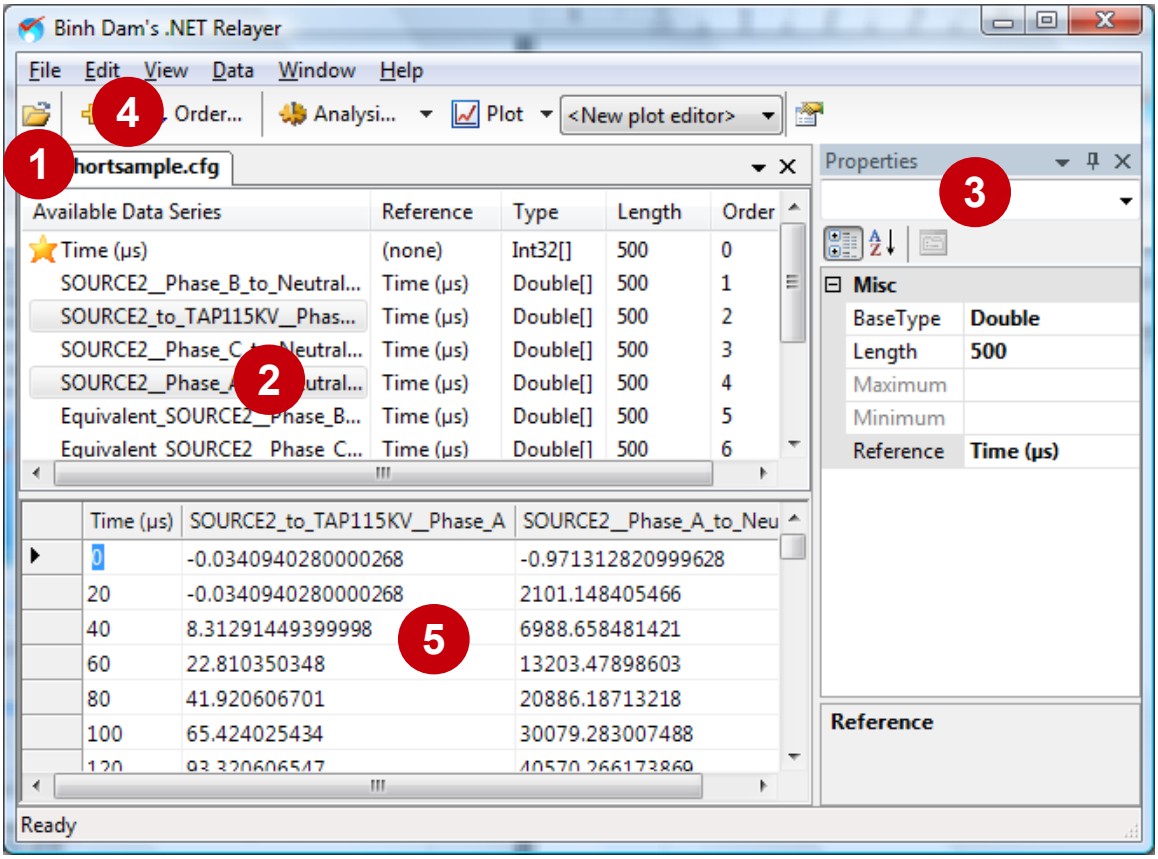
- The Plot Manager now appears automatically when plotting data for the first time. Accessible from the toolbar, it initially appears side-by-side with the Properties Window, instead of one on top of each other. The user may change the layout, and it will be retained throughout the session.
- Families of plots can be drawn and recognized by shades of the same color. Shades and curve thickness are determined according to the position of the plot to be drawn.
- Plots drawn from data series are assigned preset colors instead of random colors. Colors and curve thickness are determined according to the position of the plot to be drawn.
- Glyphs/symbols for plots are easily picked from a symbol library or imported from the Windows character map application.
- Plots can be renamed from the Plot Manager using the F2 key.
- The Send Feedback menu command links to the Software Feedback page.
- A recent file menu list is now available.

Issues Corrected in .NET Relayer 2.1

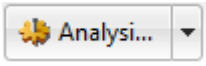
- Plot offsets and scale factors are now rendered properly, including for negative scale factors and cursor traces.
- The target plot selector in the data view toolbar shows a current updated list of open plot editors.
- Fixed an error that occurs when compiling a QuickAnalyzer with no input arguments.
- The Plot Selected Rows command is disabled when no data series is selected.
- Irrelevant objects and properties are no longer available from the Properties Window.

Navigating COMTRADE Files

1. At startup, the program automatically asks which COMTRADE file (.cfg extension) you would like to open. You may also use the File | Open command or the File | Recent Files list to access COMTRADE files.
2. The waveforms stored in the COMTRADE file are displayed in the Data Series list in the main window. When selecting waveforms from the list, their content is shown in a spreadsheet underneath the list.
3. Waveform attributes (Reference, Base Type, Length) may be changed using the **Properties Window** to the right of the screen. Waveform samples may be changed using the spreadsheet/grid display.
4. The selected waveforms may be renamed, plotted [Data | Plot], duplicated [Edit | Duplicate], reordered [Data | Reorder], transferred from one editor to the other [Edit | Move Data Series] or [Edit | Copy Data Series], or removed from the editor [Edit | Delete].
5. You may copy certain rows or samples of the selected waveforms to the clipboard for use in other applications such as spreadsheet applications.
6. Attention: all changes are lost upon closing of the waveform list. In addition, modified or deleted data cannot be recovered and must be reopened or recomputed.

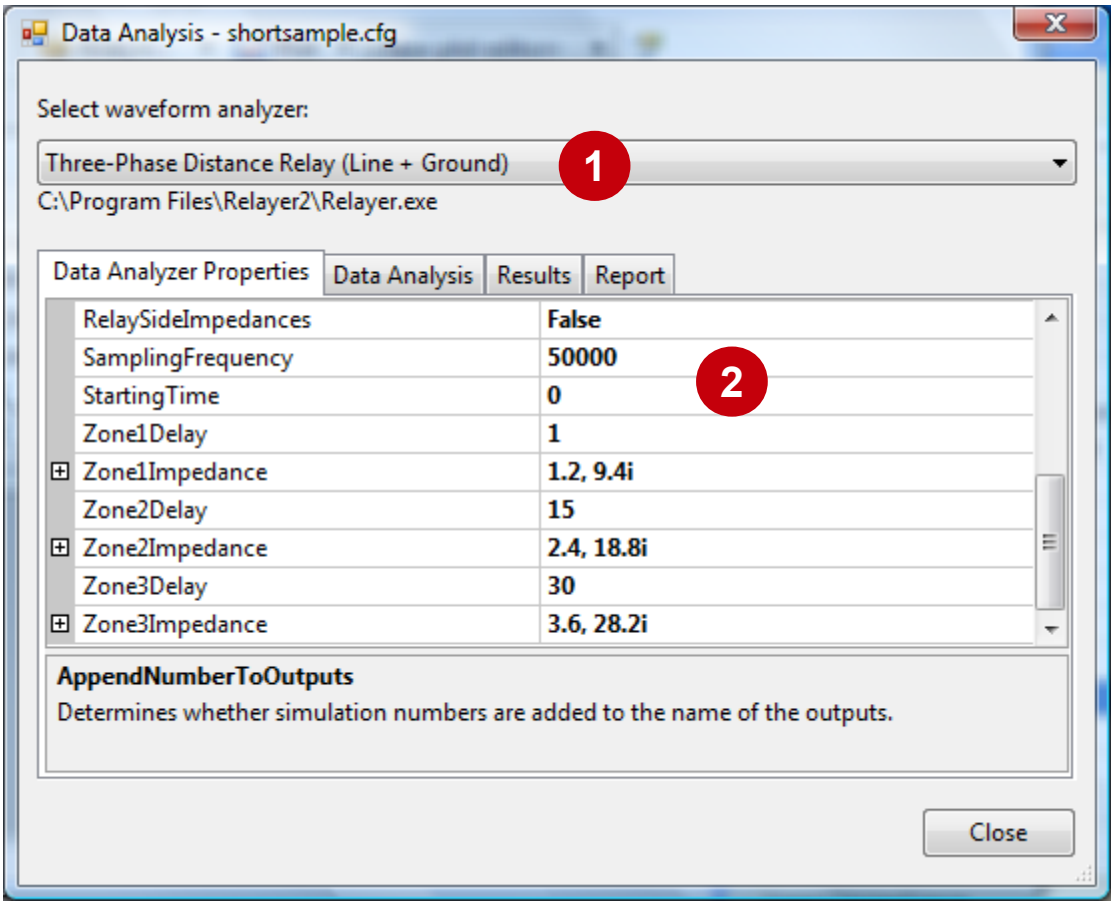


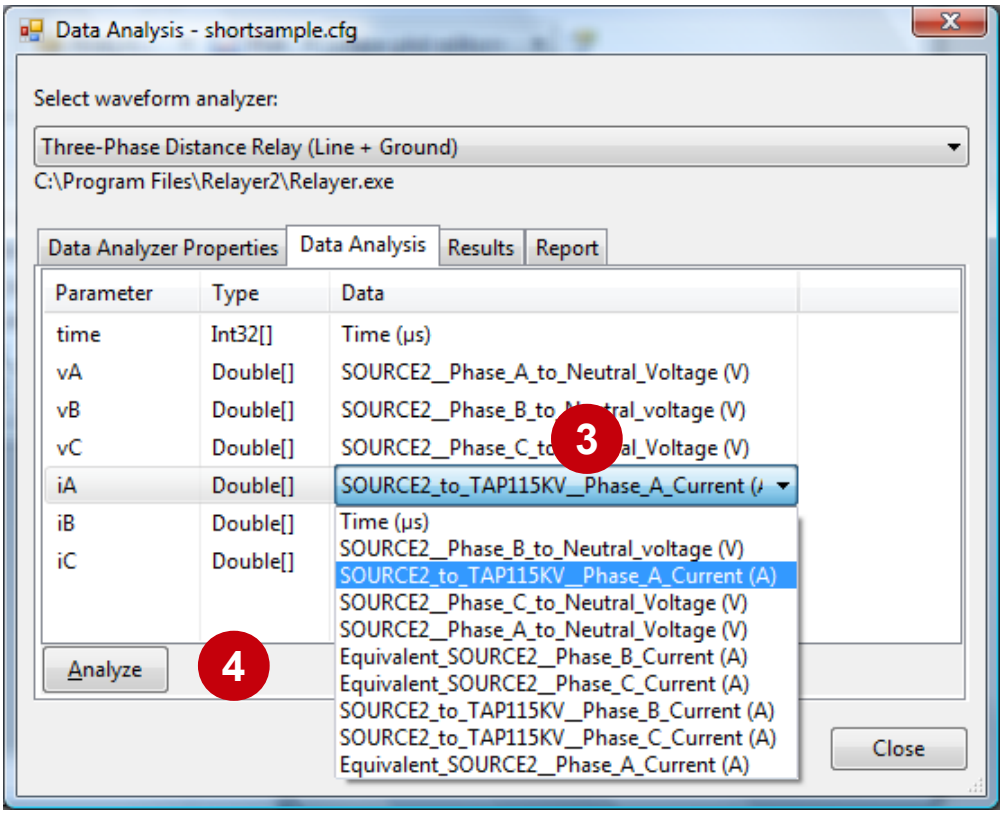
Simulating Relay Operation

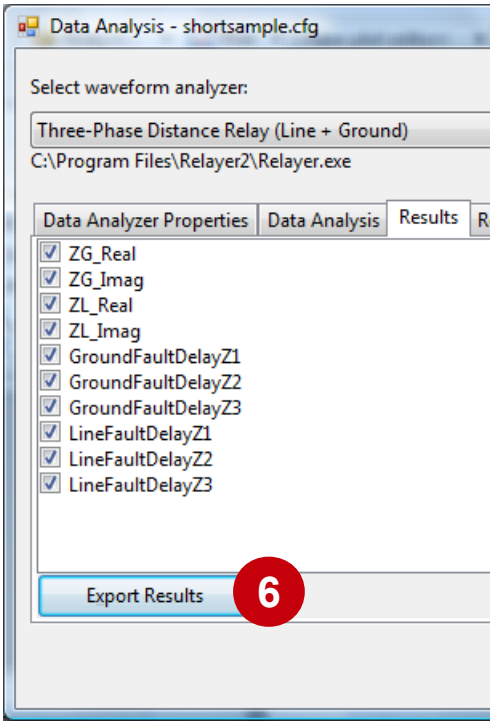
Relay simulations can be launched from the main window using Data | Analysis or using the Analysis toolbar button 

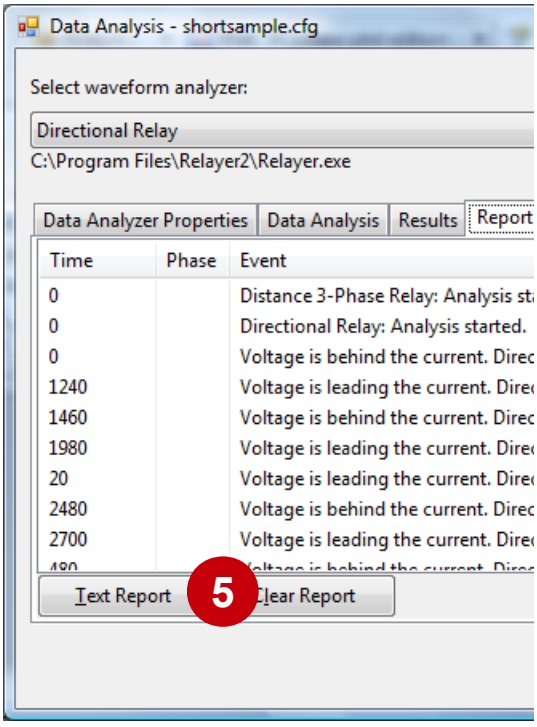
. The steps are as follows:

1. Select a relay function.
2. Configure relay properties.
3. Configure relay input parameters.
4. Launch analysis.
5. Examine reports.
6. Export results.



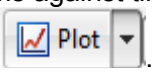




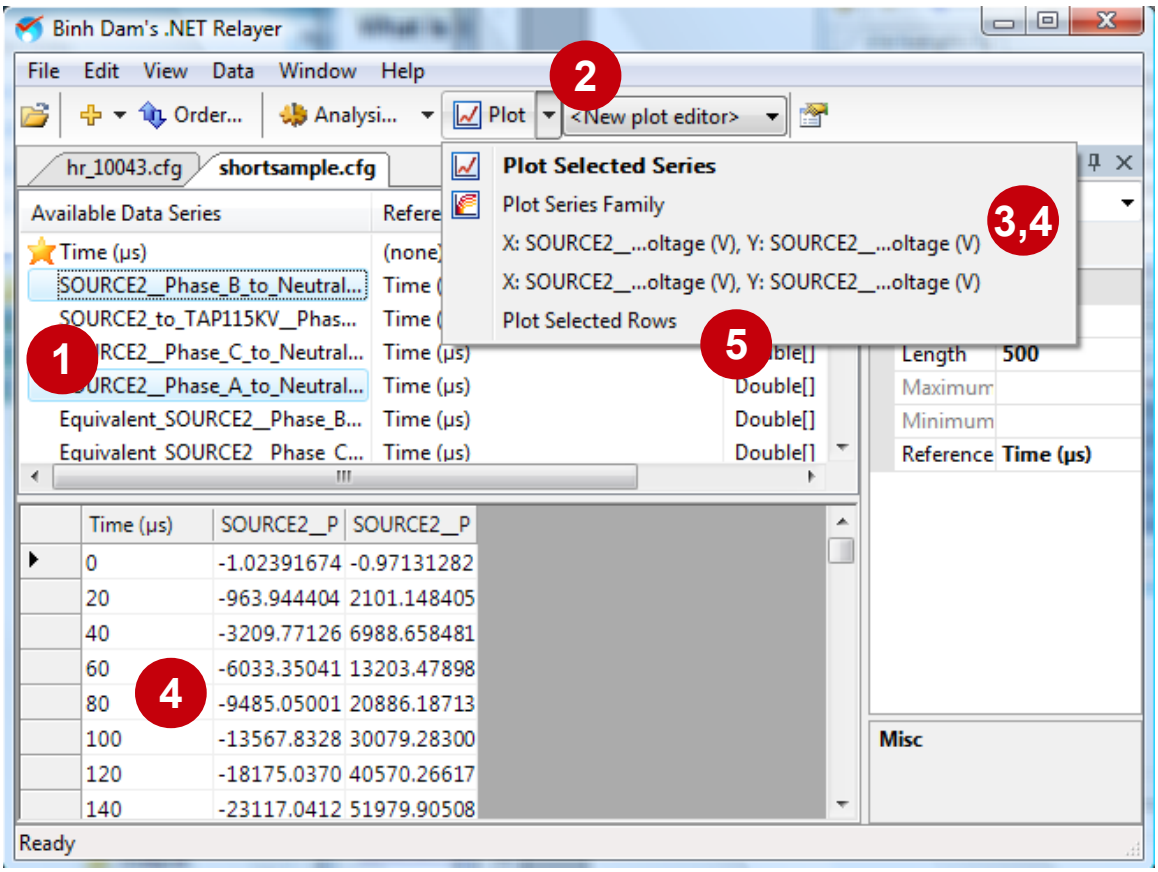


Plotting Data

The .NET Relayer software has the capability to graphically represent data from COMTRADE files and data computed from relay simulations and other analyses. The steps to plot data are as follows:

1. Select the waveforms to plot from the Data Series window
2. To plot all waveforms against time, click Data | Plot | Plot Selected Series, or use the Plot toolbar button 

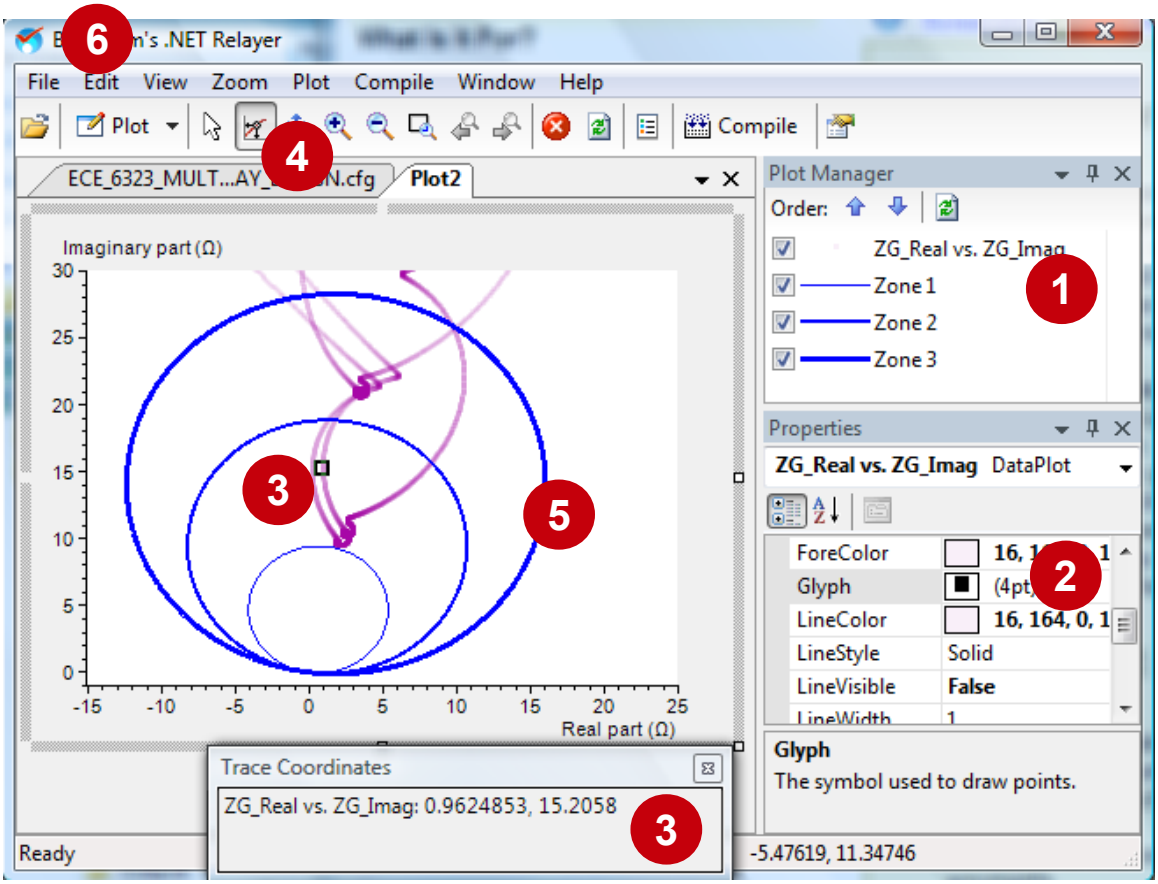
. Optionally, specify the destination editor.
3. To plot a family of waveforms, click Data | Plot | Plot Series Family. Plot families are drawn with shades of the same color.
4. To plot one waveform versus another (complex trajectories), click Data | Plot | X:<name1>; Y:<name2>
5. To plot a subset of certain waveforms against time, select the rows to plot and click Data | Plot | Plot Selected Rows (as shown in the illustration below).



Manipulating Plots

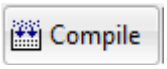
The **.NET Relayer** software includes powerful plotting tools to achieve a wide range of displays:

1. Plot manager to select active displays and view/copy legend. The plot manager automatically appears upon plotting data for the first time. The plot manager can be activated using View | Plot Manager or from the toolbar.
2. Properties Window to modify plot attributes (line, color, limits, scale, offset, axes, title, symbols/glyphs).
3. Trace window that displays the coordinates of select items. To activate Trace for any waveform: Select it; In the Properties Window, set the ShowCursor property to true; Select View | Show Cursors. A small square appears on each plot activated for cursor display, and coordinates are updated in the Trace window as you move the cursor.
4. Zoom tools to pan, focus on details or to return to graph overviews.
5. Easy setup of relay zones for visualizations (Plot | Setup Relay Zones).
6. Copy plots and legend to clipboard (Edit | Copy as Bitmap/Metafile).

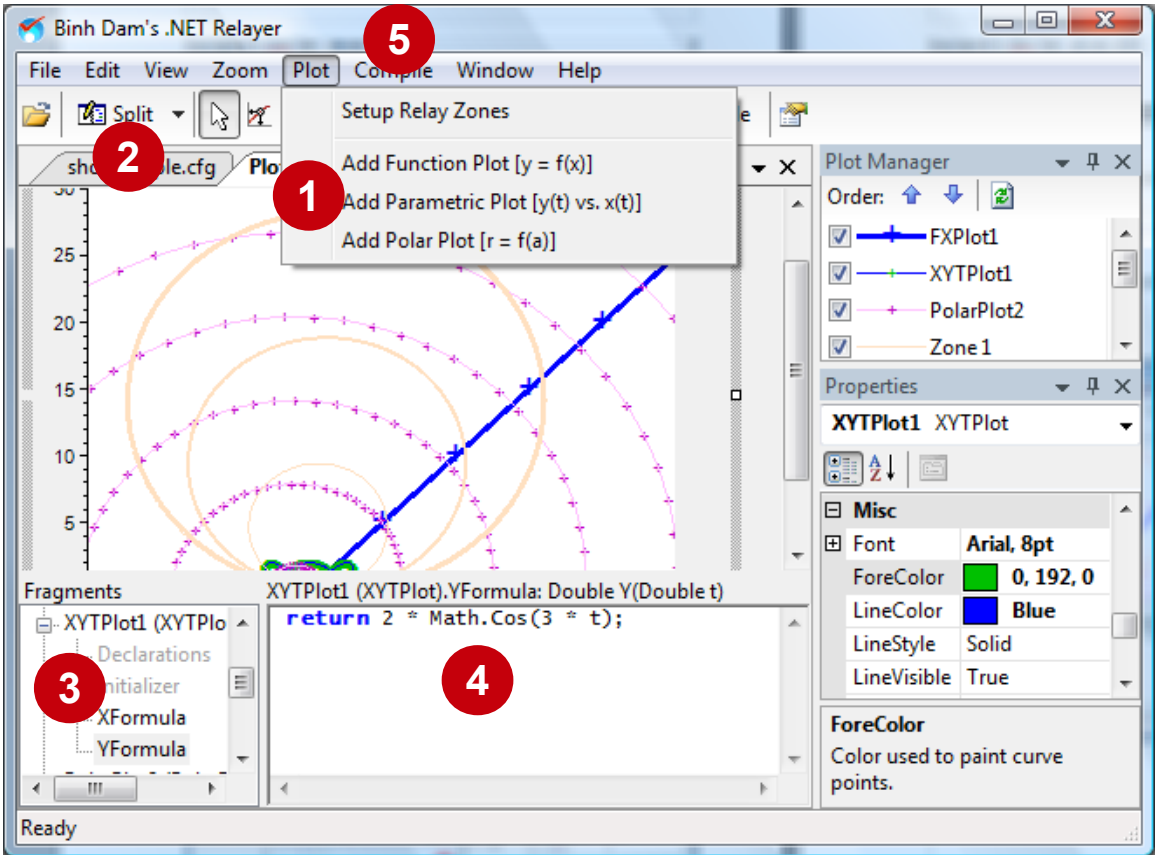


Plotting Formulas

.NET Relayer allows users to plot formulas to check if waveforms obey certain criteria. Mathematical plots are easy to create and configure:

1. Use the Plot menu to select which type of plot to insert: function plots, parametric plots, and polar plots. A plot is inserted and drawn with default settings.
2. Use the Split view (View | Split) to show the code editor side-by-side with the graphical display.
3. In the Code view, under Fragments, select Formula or XFormula or YFormula depending on the type of plot.
4. Type a return expression (C#). If private variables, initialization code, or external references are needed, or to change the language, see Advanced topics in this guide.
5. Click Compile | Compile Plot Group or use the 

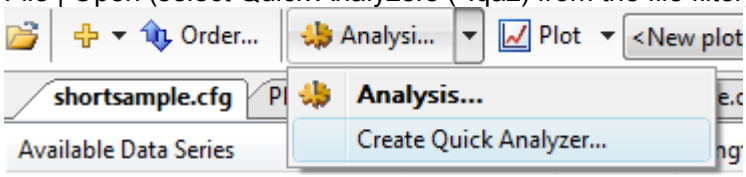
toolbar button to apply changes. The plot editor is automatically updated with the new plots.



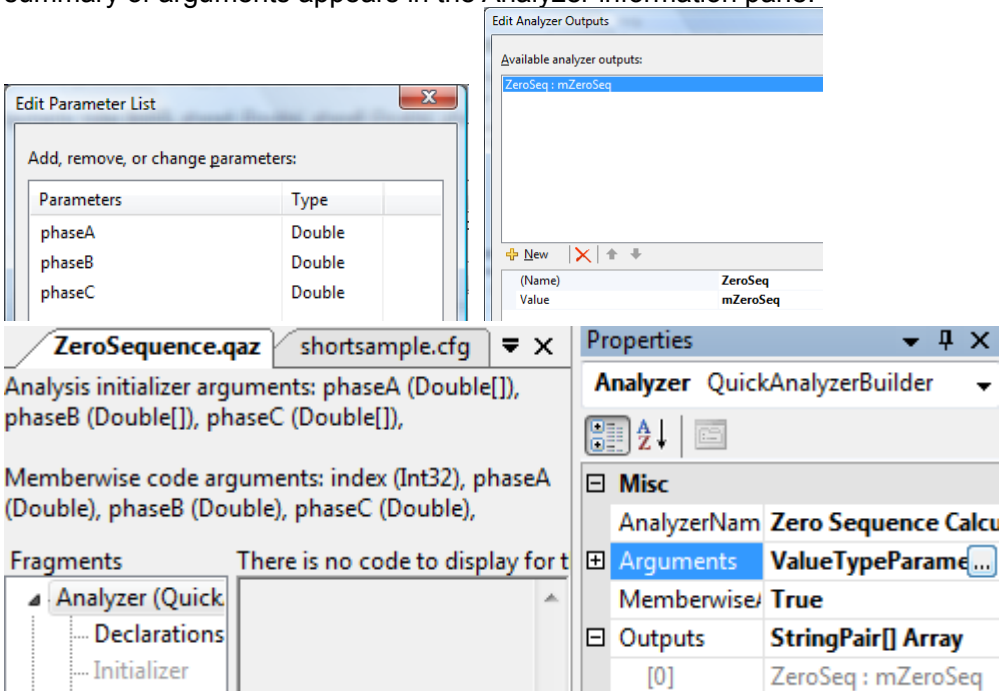
Building Your Own Relay Functions

The QuickAnalyzers feature facilitates the writing and testing of custom relay functions in a few easy steps:

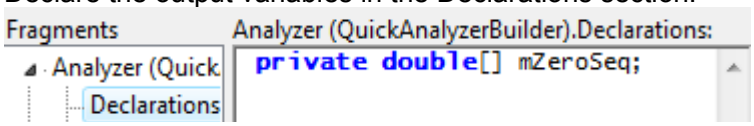
1. From the Data Series editor, select Data | Create Quick Analyzer, or open on from File | Open (select Quick Analyzers (*.qaz) from the file filter).



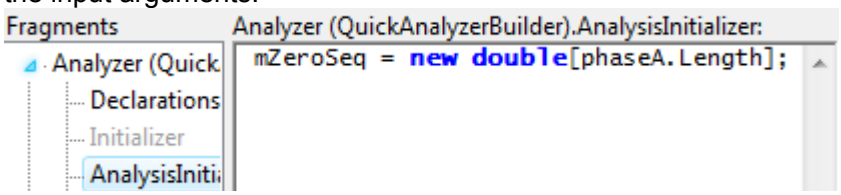
2. From the Properties Window, edit the Arguments and the Output property. A summary of arguments appears in the Analyzer information pane.



3. Declare the output variables in the Declarations section.

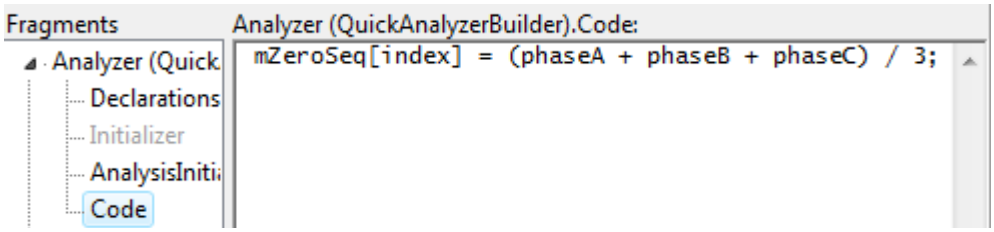


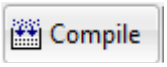
4. Initialize the output variables in the AnalysisInitializer section based on the length of the input arguments.



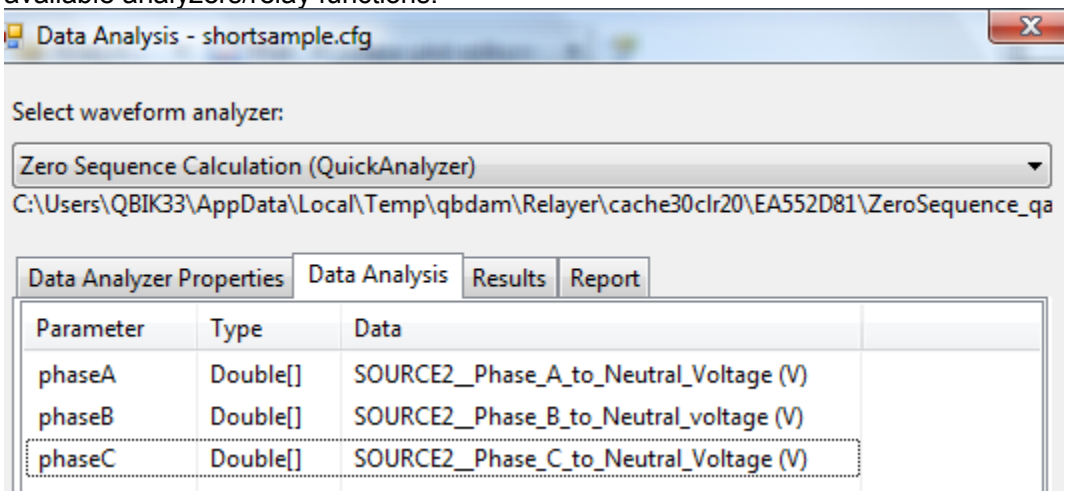
5. In the Code section, write what each member of the output variables should be based on the input arguments. Note that the MemberwiseAnalysis property of the

analyzer determines whether the code section is based on each sample or on the whole data series.



6. Click Compile | Compile QuickAnalyzer or use the 

toolbar button. You will be warned of any errors. If there are no errors, no message will be displayed, and you may proceed to next step.
7. To use the newly created QuickAnalyzer, return to the editor for an open COMTRADE file, and start an analysis. The QuickAnalyzer appears in the list of available analyzers/relay functions:

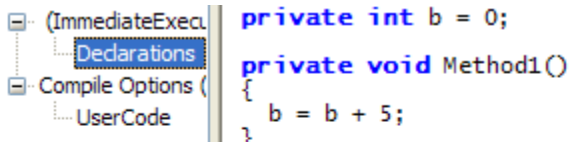


Advanced Configurations

Custom Variables and Methods

To add code for variables and methods, **switch to Code View**. Select **(PlotX or QuickAnalyzer) > Declarations** in the **Fragment List**, and add the code for variables and methods in the text editor. Since these variables and methods are in the same class than the plot or QuickAnalyzer to evaluate, you can use them directly without providing the fully qualified name of the variables and methods.

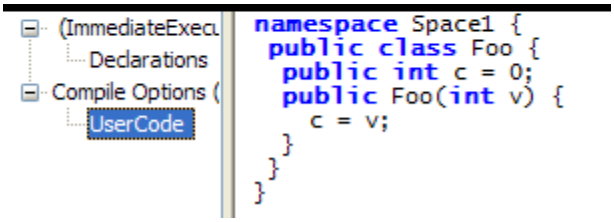
> Note: the Declarations fragment uses the namespace and imports provided in the Build Options properties.



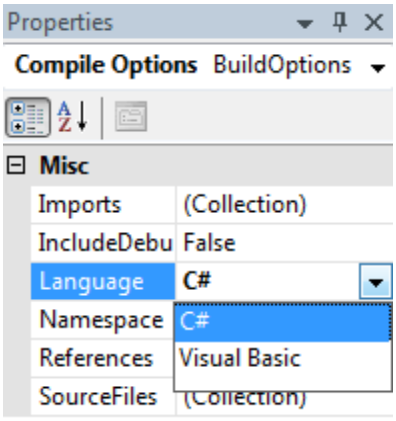
Custom Class Declarations

To provide additional namespace/class declarations, **switch to Code View**. Select **Build Options > UserCode** in the **Fragment List**. Add the code for new namespaces and classes. Because the UserCode is in a separate unit from the input code, you have to use the fully qualified name of static methods, and provide the appropriate namespaces when instantiating a class (Figure 2b).

> Note: the UserCode fragment does not use the imports found in the Build Options, you must provide namespace imports as if it was a separate file.



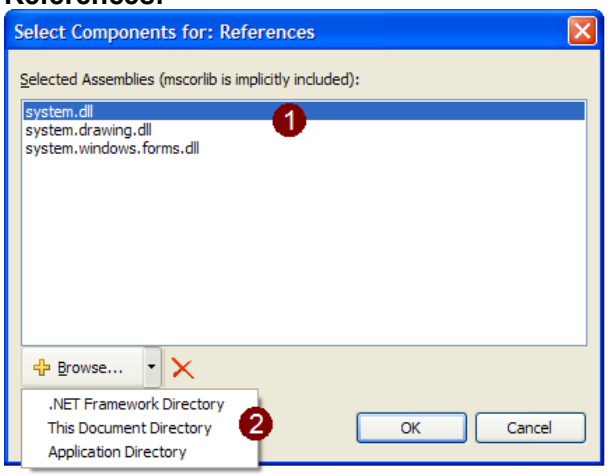
Configuring Compile Options



In most basic cases, you don't have to do anything else to execute code as you write it.

In some cases, it will be necessary to change some settings. To do so, make sure the **Properties Window** is visible (View | Properties Window, or F4 key), and **select Compile Options** in the drop-down list. The **Properties Window** displays several pairs of settings and their values.

- **Imports:** Lists all namespaces imported (using directives, Imports directives). Some namespaces are already listed in the namespace selector, so you can check them in one click instead of full typing them every time.
  Note: This property also applies to the UserCode property, but does NOT apply to referenced source files. Source files must include their own namespace import statements.
- **IncludeDebugInfo:** If true, makes debugging info (symbols and source code) available to use in a .NET debugger. Enable debugging info to spot errors in documents with large amount of code. The default is false.
- **Language:** Choose from C# (default) or Visual Basic .NET.
- **Namespace:** The namespace in which to build the code.
- **References:**



1. The Reference selector lists all assemblies referenced in the active document. mscorlib.dll is always referenced and does not appear in the references list.
2. The **drop-down button** just right of the **Add/Browse button** provides shortcuts to folders where common .NET libraries are located. Shortcuts point to the application directory, the document directory (if available) and the installation folder of the utilized .NET version.

- **SourceFiles:** Lists all files which contain source code used by the input. Useful to maintain some reusable source code without having to compile it in a library. The user interface for the source file selector is very similar to the assembly reference

selector.

- **UserCode:** Contains any additional class declarations you need. To edit the user code, display the **Code Pane** by clicking the **Code view button**, or by activating View | Code in the menu, and look for the BuildOptions | UserCode in the Fragments tree.

> Note: Code in the UserCode section should be ready to build as if it was a separate file, independently from any document code. **Namespaces listed in the Imports property will apply to the user code section, but you must explicitly import the namespaces in the referenced source files.**

About the Provided Samples

- COMTRADE Files
  - shortsample.cfg/.dat: some simulation subtransients waveform.
  - TestE2.cfg/.dat: waveform with step in current magnitude.
- QuickAnalyzers
  - cosine.qaz: a cosine waveform generator.
  - HarmCosine.qaz: a cosine waveform generator with harmonics distortion (up to harmonic 7).
  - ZeroSequence: computes the zero sequence of a three-phase signal.
- Visual Studio Project Files (Visual Studio may be need to be configured before these projects can be built and utilized completely)
  - NativeRelay: a C DLL project with a native function that can be called from a QuickAnalyzer or other .NET project.
  - SampleRelay: a C# project that creates a relay DLL from pure C# code or from C# code that calls the native DLL above.

Contact Information

The author may be contacted through the following means:

- Email: qbdam@yahoo.com
- Telephone: +1 404-384-9106
- Product website: http://www.geocities.com/qbdam/relaying.html

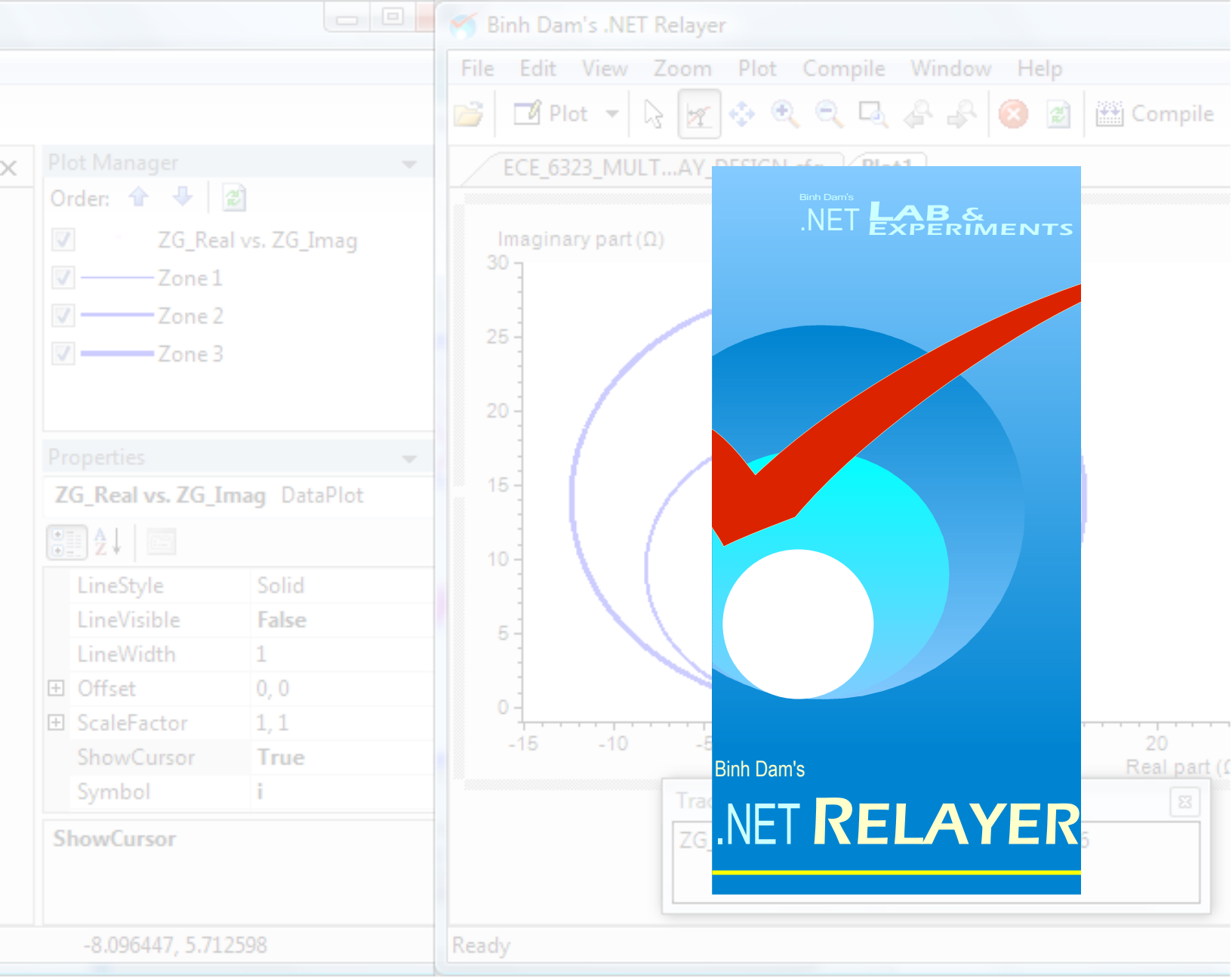
Submit feedback by using the web form located at:

http://www.geocities.com/qbdam/participate.html

You may also access this web form from the software using Help | Send Feedback.

Additional services for the **.NET Relayer** software may be arranged:

- On-site training may be arranged for a nominal fee.
- Extended support and software update notification to be provided with subscription.



.NET Relayer Software

*Thank you for your interest in the **.NET Relayer** software!*

April 2009