



i-net
Clear Reports
2016

.NET Guide

Viewer

1 Overview	2
1.1 Where do I find the example files?	2
1.2 Introduction	2
1.3 Implemented features	2
1.4 URL Parameter	3
1.4.1 Implemented URL Parameter	3
1.4.2 Ignored URL Parameters	3
1.5 Simple Example	3
1.6 Visual Studio Toolbox	4
1.7 Internationalization	4
2 Packages	6
3 Important Classes	7
3.0.1 Entry point	7
3.0.2 Main classes	7

1 Overview

The .NET Viewer is an example of how to use the .NET integrated control to show your reports. The source code is included, which means you can customize the Viewer to your needs - e.g. including them in your .NET application, or changing the User Interface to make it better fit into your Corporate Design. The project was created with Visual Studio 2010 and is optimized for use with .NET 3.5 and higher.

Note: The .NET Viewer can be used with both the Java and the .NET Version of the i-net Clear Reports server.

1.1 Where do I find the example files?

The .NET Viewer example files are in the samples folder of the i-net Clear Reports server installation. Note that during setup you must make sure to include the samples. Then all project files will be included.

You can find the .NET Viewer sample code in the subfolder:

`\documentation\developer\samples\viewer_.net` of the i-net Clear Reports installation folder.

1.2 Introduction

The .NET Viewer enables you to display an i-net Clear Reports report within a .NET environment. The report source is the i-net Clear Reports report server. It is an implemented UserControl based on Windows Forms to be used within the Visual Studio Editor.

1.3 Implemented features

- Drilldown
- Show or Hide Group tree
- Subreports

- Multiple Reports allowed
- Manually set the title of a ReportView (only via code)
- Access reports from an i-net Clear Reports Server
- Page navigation
- Refresh
- Zoom

1.4 URL Parameter

The following URL parameters are implemented for the .NET Viewer. By setting these parameters the default settings from the client and server will be overridden. Some of these properties have an equivalent property in the class ReportView. [URL Parameter documentation](#)

1.4.1 Implemented URL Parameter

1. hasgrouptree
2. hasrefreshbutton
3. hastoolbar
4. haszoomcontrol

1.4.2 Ignored URL Parameters

Note that the following URL parameters are ignored by the viewer as they are used by the server for different report formats or pages.

1. init
2. page

1.5 Simple Example

Example for using the ReportViewer to show a report via server access. First the control `inet.Viewer.WinForms.ReportViewer` has to be added to a Form or Panel in Visual Studio.

```
string url =
```

```
        "http://localhost:9000/?report=file:C:/Reports/Testreport.rpt";  
    // generate the ReportData  
    URLRenderData urlData = new URLRenderData(url);  
  
    // create the view  
    ReportView view = new ReportView(data);  
  
    // add view to the ReportViewer  
    this.reportViewer1.AddReportView(view);
```

For more details have a look at chapter [Entry point](#)

1.6 Visual Studio Toolbox

To use the ReportViewer out of the Toolbox you need to do the following steps:

1. Compile the ReportViewer project
2. Add the compiled dll to your project
3. Follow this link [Add items to toolbox](#) to add the compiled dll

1.7 Internationalization

The locale is depending on the locale of the client computer. The implementation is using the standard .NET localization which is using resource files. The method `ReportViewer.SetLabels()` has to be invoked to change the text and tooltips if the locale was changed. Currently the tooltips of the action buttons and the error messages are localized.

The client locale is transmitted to the server to show the report in a suitable language (if the report supports multiple languages). If the locale is set in the URL parameters, than this will override the system locale. At the moment English and German are supported.

[Translation settings in your reports](#)

Use the following structure to create your own language file if you need to:

- strings.resx (default locale which is English)
- strings.de.resx (German)

2 Packages

This chapter gives a short overview and explanation of the class structure and highlights some of the main classes.

inet.Viewer.Data

Implementation of all the classes that get the data from the server to pass it to the viewer.

inet.Viewer.WinForms

Implementation of the .NET Viewer controls on the basis of Windows Forms.

inet.Viewer.common

Common classes.

inet.Viewer.interfaces

Contains all the interfaces that are used for the .NETViewer

3 Important Classes

3.0.1 Entry point

- `inet.Viewer.WindowsForms.ReportViewerExample.cs`:
An Example on how to use the .NET Viewer. **The entry point** for looking into the source code. Shows how to invoke a report with the server.

3.0.2 Main classes

- `inet.Viewer.WindowsForms.ReportViewer.cs`: The main Control of the ReportViewer. Add ReportViews to this class. A UserControl that also can be used via the Visual Editor.
- `inet.Viewer.Data.UrlRenderData.cs`
The implementation of the IRenderData interface for the access of the data for a report from the server
- `inet.Viewer.WindowsForms.ReportView.cs`
To create an instance of one report to be able to add it to the ReportViewer