PdfViewer Control Reference

View PDF documents in displays.

Reference **Controls** **Viewer** PdfViewer

Show PDF documents, on Windows and Web pages. Maps the URLs to the source of the PDF, facilitating the visualization of online PDF files.

Requirements
Configuration
Customization
    Get Object Control in CodeBehind
    Handling the URL via CodeBehind
    Using Local Files via URL
Runtime Execution

Requirements

This component is Portable. It runs both on Windows (WPF) and on Web Pages hosted in any platform.

Configuration

1. Go to **Displays / Draw**.
2. On the Components Panel, select **Viewer**, then **PdfViewer**.
3. Click or drag-and-drop it on the Drawing area to use it.
4. Double-click the object to open the configuration window.

The only configuration required is to insert the URL in the URL field, which will map to the source of the PDF, facilitating the display of online PDF files.

Customization

Get Object Control in CodeBehind

This is how you get control of this object in CodeBehind:

```
TPdfViewer _pdfViewer = this.CurrentDisplay.GetControl("<objectUid>") as TPdfViewer;
```

where, <objectUid> will be the object Uid defined in object properties.

PDFViewer Customization Properties

```
/// <summary>
/// Defines factor of zoom to be applied, where 1 means 100%
/// </summary>
public double ZoomFactor { get; set; }

/// <summary>
/// Defines the zoom mode for the PDF viewer
/// </summary>
public ePdfViewerZoomMode ZoomMode { get; set; }

/// <summary>
/// Determines whether the horizontal toolbar is displayed
/// </summary>
public bool ShowHorizontalToolbar { get; set; }

/// <summary>
/// Determines whether the vertical toolbar is displayed
/// </summary>

public enum ePdfViewerZoomMode
{
    /// <summary>
    /// Displays with default zoom behavior
    /// </summary>
    Default = 0,

    /// <summary>
    /// Fits the entire page in the viewer
    /// </summary>
    FitPage = 1,

    /// <summary>
    /// Fits the width of the page in the viewer
    /// </summary>
    FitWidth = 2
}
```

Usage example:

```
TPdfViewer _pdfViewer = this.CurrentDisplay.GetControl("ui1") as TPdfViewer;
_pdfViewer.ZoomFactor = 2;
_pdfViewer.ZoomMode = TPdfViewer.ePdfViewerZoomMode.FitPage;
_pdfViewer.ShowHorizontalToolbar = false;
_pdfViewer.ShowVerticalToolbar = false;
```

Handling the URL via CodeBehind

To access the URL for PDF file on CodeBehind, use the `LinkedValue` property. This property provides the URL associated with the control.

If the URL originates from a Tag, it will be tokenized. To retrieve the actual URL, use the following methods:

Handling Tokenized URLs

```
string CustomUrl = TConvert.ToString(TK.GetObjectValue(T.Wpf.RunServices.ConfigHelper.UntokenizeObjectName
(LinkedValue)));
```

Using Local Files via URL

Local files can be referenced using an HTTP URL served by TServer, which works on both Windows and Web clients.

Files placed in the default served folders are accessible without extra configuration:

Target	Default folder
Windows-only (.NET 4.8 / WPF)	`fx-10`
Multiplatform (.NET 10)	`net10`

For files in custom directories, add a `MapPaths` entry in **TServer.json** (under MachineSettings):

Updated URL

```
"MapPaths": {
        "MapPath1": "alias1;C:\\DirTest1"
}
```

The alias then becomes a URL path segment, like the following example:

http://192.168.1.168:3101/alias1/pdf_sample.pdf

Multiple entries are supported (`MapPath1`, `MapPath2`, …).

Method Replacement

The `Validate()` method has been replaced by `StartRuntime` for all controls, including TPdfViewer.

Validate (Outdated)

```
TPdfViewer pf = this.CurrentDisplay.GetControl("pd") as TPdfViewer;
pf.Validate();
```

When initializing or updating a control, use `StartRuntime()` instead of Validate:

StartRuntime

```
TPdfViewer pf = this.CurrentDisplay.GetControl("pd") as TPdfViewer;
pf.StartRuntime();
```

Runtime Execution

When running the application, the PdfViewer Control fetches and displays PDF documents from specified URLs. Upon execution, the PdfViewer Control accesses the provided URL and renders the PDF content within the application.

Running on Windows (Rich Clients)

The PdfViewer is able to access local files and files from the Web, according the URL.

Running on the browser (Web Clients)

The PdfViewer can only files from the Web, or from the Local WebServer (typically 127.0.0.1:3101 or 127.0.0.1:3103, according running the solution in Production or Development modes)

Example of URL acceptable for HTML5 Web Clients

http://localhost:65535/Resources/Document.pdf

In this section...