

## Table of Contents

ReadMe	2
Release Notes	2
Version 2.2.0.310	2-3
Version 2.1.0.260	3-4
Version 2.0.0.200	4-5
Installation	5-8
Product Requirements	8
License Information	8-9
Upgrade to Latest Version	9
Redistribution	9-10
Viewing Help Content	10
Technical Support	10
Contacting Sales	10
End-User License Agreement	10

## ReadMe

This readme contains basic information about **GrapeCity Documents for PDF**, also referred to as **GcPdf**.

- [Release Notes](#)
- [Installation](#)
- [Product Requirements](#)
- [GcPdf License Information](#)
- [Redistribution](#)
- [Viewing Help Content](#)
- [Technical Support](#)
- [Contacting Sales](#)
- [End-User License Agreement](#)

## Release Notes

Refer to the following release notes for the major releases of the product.

- [Release Notes for Version 2.2.0.310](#)
- [Release Notes for Version 2.1.0.260](#)
- [Release Notes for Version 2.0.0.200](#)

For details about latest hot fixes, see the [nuget page](#).

## Version 2.2.0.310

### New features and improvements

The following features have been added with this version of the product.

- Added `GcPdfDocument.ImportFormDataFromFDF()`, `GcPdfDocument.ImportFormDataFromXFDF()` and `GcPdfDocument.ImportFormDataFromXML()` methods to import document's form data from FDF, XFDF and XML formats.
- Added `GcPdfDocument.ExportFormDataToFDF()`, `GcPdfDocument.ExportFormDataToXFDF()` and `GcPdfDocument.ExportFormDataToXML()` methods to allow users to export document's form data to FDF, XFDF and XML formats.
- Added `GcPdfDocument.ImportFormDataToFDF()`, `GcPdfDocument.ImportFormDataToXFDF()` and `GcPdfDocument.ImportFormDataToXML()` methods to allow users to import document's form data to FDF, XFDF and XML formats.
- Added `RedactAnnotation` class to support redact annotations. However, it is important to note that GcPdf doesn't support the actual removal of the content which is marked for redaction.
- Added `Page.Draw()` and `Page.DrawAnnotations()` methods to support PDF rendering on the `GcGraphics` class.
- Added `SaveAsBmp()`, `SaveAsPng()`, `SaveAsGif()`, `SaveAsJpeg()` and `SaveAsTiff()` methods in the `GcPdfDocument` class, which convert the PDF document to various image formats. Also, the methods `Page.SaveAsPng()` and `Page.SaveAsBmp()` have been added.

### Changes From the Previous Version

This version of the product has the following changes.

- Earlier, some PDF generators used to render the same text multiple times at nearly the same position in order to emulate bold the text. But now, GcPdf handles such scenarios more efficiently while building the text map.
- Now, the property `SignatureField.Value` is writable. This means that it will allow users to remove an existing

signature.

## Bug Fixes

The following issues have been resolved since the last release.

- The internal document signature flags are now correctly updated while adding or removing the signature fields.
- The method `GcPdfDocument.GetText()` now works correctly with all the PDF files.
- Now, the `StackOverflow` exception is not thrown while loading certain PDF files.
- Now, the `NotImplementedException` is not thrown while enumerating the `ThreadArticleBeadCollection` and everything works appropriately.

## Version 2.1.0.260

### New features and improvements

The following features have been added with this version of the product.

- Added static property `ICMapProvider GcPdfDocument.CMapProvider`, allows to define `ICMapProvider` which will be used by `GcPdfDocument` to request predefined `CMap`'s not existing in `GrapeCity.Documents.Pdf.dll`.
- Added `GrapeCity.Documents.Pdf.Resources.dll` containing additional resources used by `GrapeCity.Documents.Pdf.dll`, currently it contains additional predefined `CMap`'s.
- `AnnotationBase.DefaultAppearanceString` was removed, `WidgetAnnotation.DefaultAppearanceString` and `FreeTextAnnotation.DefaultAppearanceString` were added, use them instead.
- Added properties `TextField.RichTextValue`, `TextField.DefaultStyleString`, allows to set value of field as rich formatted string, see PDF specification for more details. Note! `GcPdf` does not generate automatically appearance streams for RTF text fields, it is not supported.
- Added `AssociatedFiles` property for `Page`, `GcPdfDocument`, `StructElement`, `AnnotationBase` objects, allows to associate list of embedded files with object. This property can be used to generate PDF/A-3x documents if they contains embedded files.
- Added `PdfAConformanceLevel Metadata.PdfA` and `GcPdfDocument.PdfAConformanceLevel` properties allows to set PDF/A conformance level, `GcPdfDocument.PdfAConformanceLevel` is a wrapper around `Metadata` property, property `GcPdfDocument.PdfACompliant` now obsolete.
- Added `Metadata.CreatorTool` property it has same purpose as `DocumentInfo.Creator`.
- Added support of Sound annotations, see `SoundAnnotation` class.
- Added `GcPdfGraphics.SoftMask` property, allows to set `FormXObject` as a drawing mask.
- Method named `ITextMap.GetSelectionxxx(...)` is renamed to `ITextMap.GetFragmentxxx(...)`.
- Added `Page.TransitionEffect`, `Page.TransitionDuration` properties, allows to define page behavior in presentation mode.
- Added `GcPdfDocument.ArticleThreads` list. It allows to define article threads in the document.
- Added `ITextMap Page.GetTextMap(float dpiX = 72, float dpiY = 72)` method. It allows to get page's text map represented by `ITextMap` interface which can be used for hit testing, retrieve list of text fragments on the page etc.
- Added `GcPdfDocument.PageLabelingRanges` property, allows to define dictionary of `PageLabelingRange` objects which define labels for document's pages.

## Bug Fixes

The following issues have been resolved since the last release.

- Fixed the issue where exception occurred on using `SignatureLockedFields(SignatureLockedFieldsType, IEnumerable<string>)` constructor.

- Fixed the issue where appearance streams are generated incorrectly for Acroform fields in certain scenario.
- Fixed the issue where text renders incorrectly in GcPdfGraphics if FontSizeInGraphicUnits is true and GcPdfGraphics.Resolution is not 72.

## Version 2.0.0.200

### GcPdf

#### Breaking Changes

- GcPdf .NET 2.0.0 version comes with an evaluation license key that allows users to use the product without any limitations for a time period of 30 days.
- GcPdf version 2.x.x.x requires a new license key. Please note this is a free upgrade for the customers who have purchased GcPdf v1 license. Our sales team will send a new license key to the licensed customers.
- Property type of GcPdfDocument.CompressionLevel is now changed from integer to standard System.IO.Compression.CompressionLevel enumeration. CompressionLevel.NoCompression corresponds to old compression level 0, Fastest to 1, Optimal to 6.
- Property type of ViewerPreferences.DisplayDocTitle is now changed to bool. Note that it is written to PDF stream, if it is not null.

#### Bug Fixes

The following issues have been resolved since the last release.

- GcPdfDocument could not process encrypted signed files. (DOC-749)
- Disabled bold font emulation for semi-bold fonts.

#### New Features and Improvements

The following features have been added with this version of the product.

- Added GcPdfDocument.MergeWithDocument(GcPdfDocument sourceDoc, MergeDocumentOptions options) method. Use this method to combine pages from different PDF documents.
- Added constructor FormXObject(GcPdfDocument doc, Page page). It allows to insert pages from a PDF document into another document as images, and allows to duplicate a page reusing page resources.
- Added new method GcPdfGraphics.DrawForm(FormXObject, ...) that allows to draw FormXObject on GcPdfGraphics with full support for positioning, scaling and transformations.
- Added property Metadata.PdfUa which allows to set the supported version of PDF/UA.
- Added IPdfImage.ImageMask property. It indicates whether the image is defined as a stencil mask for painting in the current color.
- Added support of AES-256 encryption as specified in Adobe® Supplement to the ISO 32000. To specify 256 bit encryption use the StandardSecurityHandlerRev5 class. Note that StandardSecurityHandlerRev5 is derived from StandardSecurityHandlerRev4, however, its EncryptionAlgorithm and KeyLength can be AES and 256 respectively; this limitation comes from the PDF specification.
- Added new action type: ActionHide, hides or shows one or more annotations on the screen by setting or clearing their Hidden flags.
- Added ActionBase.Next property that allows to set the next ActionBase or sequence of actions to be performed after this action.
- Added property Page.AnnotationsTabsOrder.
- Added GcPdfDocument.GetImages(...) and Page.GetImages(...) methods, allowing to enumerate and retrieve images from the whole document or from a specific page respectively.
- Added GcPdfDocument.EmbeddedFiles property, which allows to specify document level file attachments.
- Added CreationDate, ModificationDate, CheckSum, Size properties to the EmbeddedFileStream class.

## GcCommon

### Breaking Changes

This version of the product has the following changes.

- Image, GcGraphics, Pen, Brush, TextRenderer classes moved to GrapeCity.Documents.Imaging package, which is now required for GcPdf to work. If you reference the GcPdf NuGet package, after updating to the new GcPdf a reference to GrapeCity.Documents.Imaging package will be added automatically.

### Resolved Issues

The following issues have been resolved since the last release.

- Issue when applying LineSpacingScaleFactor to inline objects in TextLayout.
- Single-character line breaks did not work with TextLayout with specified ObjectRects.
- Several issues in ContentWidthFitsInBounds and ContentHeightFitsInBounds properties of the TextLayout class.
- Unnecessary trimming of trailing spaces in TextLayout.
- Exception occurred when loading Font from a stream that does not support seeking. (DOC-707)
- TextFormat.Tag property associated with GlyphRun had an incorrect value.

### New Features and Improvements

The following features have been added with this version of the product.

- Added TrimmingGranularity and EllipsisCharCode properties to TextLayout for text trimming.
- Added DelimiterCharCode and DelimiterCharCount properties to TextLayout to specify the portion of text to be preserved from trimming.
- Added JustifiedSpaceShrinking property to TextLayout to make justified text more compact.
- Added RecalcFontSizeForFallbackFonts property to TextFormat: indicates whether the size of fallback font should be coerced to the size of original font (true by default).
- Added the GlyphAdvanceOffset property to TextFormat specifying the additional increment to the glyph advance value.
- Added MeasureStringWithTrailingWhitespace() method to GcGraphics.
- Added ContentWithWhitespaceBounds and other related properties to TextLayout.
- Added font fallback support to combining marks.
- The value of GcGraphics.FontCollection is now copied to TextLayout.FontCollection in the GcGraphics.CreateTextLayout() method.
- Added opacity parameter to GcGraphics.DrawImage(...) methods.

## General

The following packages now include japanese localized assemblies.

1. ja/GrapeCity.Documents.Word.resources.dll
2. ja/GrapeCity.Documents.Pdf.resources.dll
3. ja/GrapeCity.Documents.Imaging.resources.dll
4. ja/GrapeCity.Documents.Common.resources.dll
5. ja/GrapeCity.Documents.Common.Windows.resources.dll

## Installation

## Setting up an application

GcPdf references are available through NuGet, a Visual Studio extension that adds the required libraries and references to your project automatically. To work with GcPdf, you need to have following references in your application:

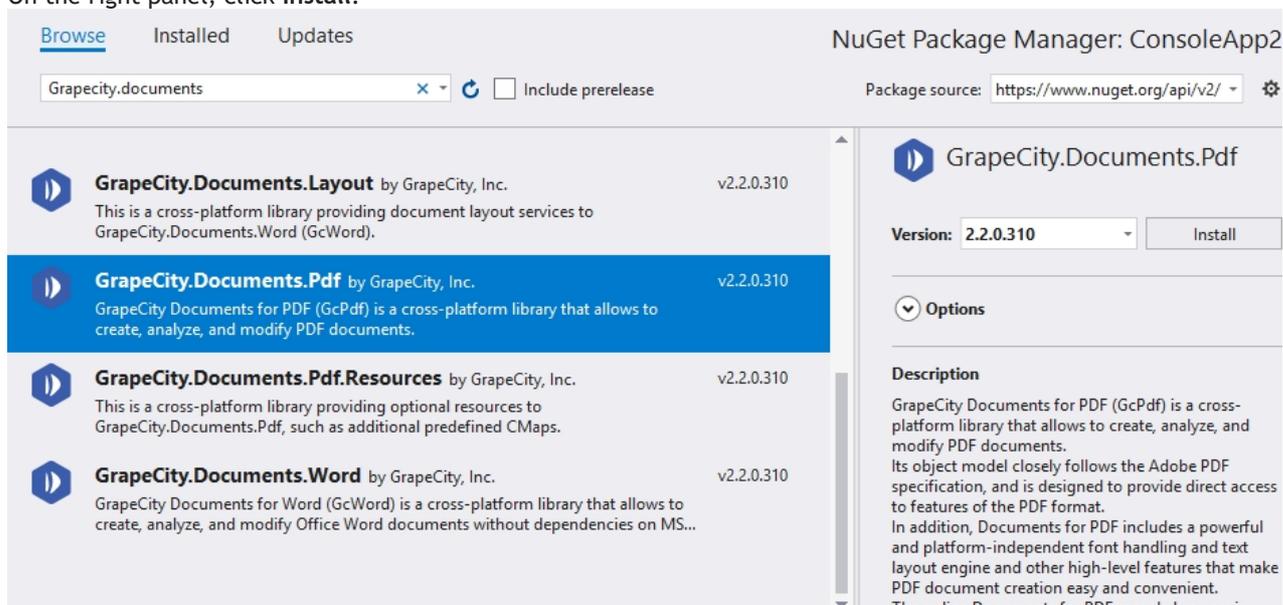
Reference	Purpose
<a href="#">GrapeCity.Documents.Pdf</a>	To use GcPdf in an application, you need to reference (install) just the GrapeCity.Documents.Pdf package. It will pull in the required infrastructure packages.
<a href="#">GrapeCity.Documents.BarCode</a>	To render barcodes, install the GrapeCity.Documents.Barcode (aka GcBarcode) package. It provides extension methods allowing to render barcodes when using GcPdf.
<a href="#">GrapeCity.Documents.Imaging</a>	GrapeCity.Documents.Imaging provides image handling. You do not need to reference it directly.
<a href="#">GrapeCity.Documents.Common</a>	GrapeCity.Documents.Common is an infrastructure package used by GcPdf and GcBarcode. You do not need to reference it directly.
<a href="#">GrapeCity.Documents.Common.Windows</a>	On a Windows system, you can optionally install GrapeCity.Documents.Common.Windows. It provides support for font linking specified in the Windows registry, and access to native Windows imaging APIs, improving performance and adding some features (e.g. TIFF support).
<a href="#">GrapeCity.Documents.DX.Windows</a>	GrapeCity.Documents.DX.Windows is an infrastructure package used by GrapeCity.Documents.Common.Windows. You do not need to reference it directly.

### Add reference to GcPdf in your application from NuGet.org

In order to use GcPdf in a .NET Core, ASP.NET Core, .NET Framework application (any target that supports .NET Standard 2.0), install the NuGet packages in your application using the following steps:

## Visual Studio for Windows

1. Create any application (any target that supports .NET Standard 2.0).
2. Right-click the project in Solution Explorer and choose **Manage NuGet Packages**.
3. In the **Package source** on top right, choose **nuget.org**.
4. Click **Browse** tab on top left and search for "GrapeCity.Documents".
5. On the left panel, choose **GrapeCity.Documents.Pdf**
6. On the right panel, click **Install**.



7. In the **Preview Changes** dialog, click **OK** and choose **I Accept** in the next screen.
8. (Optional) If you want to add barcodes in your PDF file, you need to install the package **GrapeCity.Documents.Barcode** using the steps 5 to 8 above.

This will add all the required references of the package to your application.

## Visual Studio for Mac

1. Create any application (any target that supports .NET Standard 2.0).
2. In tree view on the left, right-click **Dependencies** and choose **Add Packages**.
3. In the Search panel, type "GrapeCity".
4. From the list of packages displayed in the left panel, select **GrapeCity.Documents.Pdf** (and **GrapeCity.Documents.Barcode** if you want to render barcodes in your Pdfs) and click **Add Packages**.
5. Click **Accept**.

This will automatically add references of the package and its dependencies to your application.

## Visual Studio Code for Linux

---

1. Open Visual Studio Code.
2. Install **Nuget Package Manager** from **Extensions**.
3. Create a folder "MyApp" in your **Home** folder.
4. In the Terminal in Visual Studio Code, type `cd MyApp`
5. Type command `dotnet new console`  
*Observe:* This creates a .NETCore application with MyApp.csproj file and Program.cs.
6. Press **Ctrl+P**. A command line opens at the top.
7. Type command: `>`  
*Observe:* "**Nuget Package Manager: Add Package**" option appears.
8. Click the above option.
9. Type "**GrapeCity**" and press Enter.  
*Observe:* GrapeCity packages get displayed in the dropdown.
10. Choose **GrapeCity.Documents.Pdf**.
11. (Optional) Repeat above steps to add **GrapeCity.Documents.Barcode** if you want to add barcodes to your PDF.  
*Observe:* The packages would be added to your .csproj file.
12. Type following command in the Terminal window: `dotnet restore`.

This will add references of the package to your application.

## Product Requirements

GcPdf system requirements, depending upon the framework you are using to create an application, are:

- Our packages include two targets, NET Standard 2.0 and .NET Framework 4.6.1. In order to use them, your application needs to target either of the following:
  - .NET Core 2.0 or later
  - .NET Framework 4.6.1 or later
- Visual Studio 2015+/Visual Studio for MAC/Visual Studio Code for Linux

For OS versions supported in .NET Core 2.0+, see [.NET Core 2.0+ - Supported OS versions](#).

## License Information

GrapeCity Documents for PDF supports the following types of license:

- **Unlicensed**
- **Evaluation License**
- **Licensed**

### Unlicensed

After downloading the product, the product works in unlicensed mode. The following limitations are imposed when the product is used without license:

- Only 5 pages of the PDF file can be loaded for analyzing.
- On saving the PDF file, a watermark is displayed on all the pages in that file.  
*'Unlicensed copy of GrapeCity PDF. Loading is limited to 5 pages. Contact [us.sales@grapecity.com](mailto:us.sales@grapecity.com) to get your 30-day evaluation key.'*

 Note that if you run a sample that uses a signed PDF without a valid license key of GcPdf, the original signature in the generated PDF is invalidated. This happens because a license header is added to the PDF in such cases which changes the original signed document.

## Evaluation License

GcPdf evaluation license is available to users for 30 days to evaluate the product. If you want to evaluate the product, you can ask for evaluation license key by sending an email to [us.sales@grapecity.com](mailto:us.sales@grapecity.com).

The evaluation version has an expiration date that is determined when an evaluation key is generated. After applying the evaluation license key, you can use the complete product until the license expiry date.

After the expiry date, the product works in unlicensed mode with the above mentioned limitations.

In such case, following watermark is displayed in the PDF file:

'Created with expired evaluation copy of GrapeCity PDF. Loading is limited to 5 pages. Contact [us.sales@grapecity.com](mailto:us.sales@grapecity.com) to purchase license.'

 Note that if you run a sample that uses a signed PDF without a valid license key of GcPdf, the original signature in the generated PDF is invalidated. This happens because a license header is added to the PDF in such cases which changes the original signed document.

## Licensed

GcPdf production license is issued at the time of purchase of the product. If you have a production license, you can access all the features of GcPdf without any limitations.

## Apply License

To apply evaluation/production license in GcPdf, the long string key needs to be copied to the code in one of the following two ways.

- Pass it as an argument to the GcPdfDocument's ctor:  

```
var doc = new GcPdfDocument("key")
```

This licenses the instance being created.
- Call a static method on GcPdfDocument:  

```
GcPdfDocument.SetLicenseKey("key");
```

This licenses all the instances while the program is running.

## Upgrade to Latest Version

To upgrade GcPdf license from v1 to v2, you will obtain a new license key from GrapeCity Sales team. This is a **free** upgrade for customers who purchased GcPdf v1 license. Once you have received a new license key through email, follow these steps:

1. Open an existing application created using GcPdf v1 license.
2. Right-click the project in **Solution Explorer** and choose **Manage NuGet Packages**.
3. In the **Package source** on top right, select **nuget.org**.
4. Click **Updates** tab on the top. A list of all the installed NuGet packages is displayed.
5. On the left panel, select the **Select all packages** checkbox and click **Update**.
6. In the **Preview Changes** dialog, click **OK** and choose **I Accept** in the next screen.
7. Switch to the code view and replace the old key with new v2 key received through email.
  - To upgrade the license of a particular instance:  

```
var doc = new GcPdfDocument("new key")
```
  - To upgrade the license of all the instances:  

```
GcPdfDocument.SetLicenseKey("new key");
```

## Redistribution

In order to distribute the application, make sure you meet the installation criteria specified in the [Product Requirements](#) page in this documentation. Further, you also need to have a valid Distribution License to successfully distribute the application.

 GcPdf makes it easy to deploy your application to your local servers or cloud offerings such as Azure.

For more information about Distribution License, contact our Sales department using one of these methods:

World Wide Web site	<a href="https://www.grapecity.com/">https://www.grapecity.com/</a>
E-mail	<a href="mailto:us.sales@grapecity.com">us.sales@grapecity.com</a>
Phone	1.800.858.2739 or 412.681.4343
Fax	(412) 681-4384

## Viewing Help Content

For detailed information on Documents for PDF, please refer [online documentation](#).

## Technical Support

If you have a technical question about this product, consult the following source:

- Product forum: <https://www.grapecity.com/forums>
- Email: [us.sales@grapecity.com](mailto:us.sales@grapecity.com)

## Contacting Sales

If you would like to find out more about our products, contact our Sales department using one of these methods:

World Wide Web site	<a href="https://www.grapecity.com/">https://www.grapecity.com/</a>
E-mail	<a href="mailto:us.sales@grapecity.com">us.sales@grapecity.com</a>
Phone	(800) 858-2739 or (412) 681-4343 outside the U.S.A.
Fax	(412) 681-4384

## End-User License Agreement

The GrapeCity licensing information, including the GrapeCity end-user license agreement, frequently asked licensing questions, and the GrapeCity licensing model, is available online. For detailed information on licensing, refer [GrapeCity Licensing](#). For GrapeCity end-user license agreement, refer [End-User License Agreement For GrapeCity Software](#).