

Quick PDF Library 8 Upgrade Warnings

This document contains a list of items that developers looking to upgrade to Quick PDF Library 8 should pay attention to. Changes to code may be required in order for your projects to compile using the new version.

1. Changed functions definitions

- The **LoadFromFile**, **LoadFromString**, **LoadFromVariant** and **LoadFromStream** functions now have a **Password** parameter. Note functionality changes below.
- The **ExtractFilePages** and **GetFileMetadata** functions now have a **Password** parameter.
- All functions that previously returned or accepted 8-bit character strings (either UTF-8 encoded, WinAnsi encoding or unspecified encoding) now use UTF-16 Unicode strings.
- Certain functions continue to accept or return 8-bit data, others have been renamed to more easily facilitate working with binary data.
- All the text drawing functions (such as **DrawText**, **DrawHTMLText**) allow the text to be specified using UTF-16 Unicode strings. The fonts automatically convert the text string's Unicode characters to the appropriate encoding for storing in the content stream.

2. Renamed functions

- The following functions have been renamed due to the changes from 8-bit to 16-bit strings. The change results in two new functions to replace each original function:

| | |
|-------------------------------|---|
| GetObjectSource | GetObjectToString (Delphi and DLL) GetObjectToVariant (ActiveX) |
| SetObjectSource | SetObjectFromString (Delphi and DLL) SetObjectFromVariant (ActiveX) |
| GetPageContent | GetPageContentToString (Delphi and DLL) GetPageContentToVariant (ActiveX) * Note functionality changes listed below |
| SetPageContent | SetPageContentFromString (Delphi and DLL) SetPageContentFromVariant (ActiveX) * Note functionality changes listed below |
| DAGetPageContent | DAGetPageContentToString (Delphi and DLL) DAGetPageContentToVariant (ActiveX) |
| DAGetObjectSource | DAGetObjectToString (Delphi and DLL) DAGetObjectToVariant (ActiveX) |
| ExtractFilePageContent | ExtractFilePageContentToString (Delphi and DLL) ExtractFilePageContentToVariant (ActiveX) |

- In QPL v7 the term “layer” was used to describe a part of a page’s content stream. This conflicted with Adobe’s use of the term “layer” for the PDF feature known as optional content groups. To avoid confusion, all the layer functions have been renamed:

| | |
|---------------------------|----------------------------|
| SetLayerOptional | SetContentStreamOptional |
| LayerCount | ContentStreamCount |
| CombineLayers | CombineContentStreams |
| NewLayer | NewContentStream |
| SelectLayer | SelectContentStream |
| EditableLayer | EditableContentStream |
| MoveLayer | MoveContentStream |
| DeleteLayer | DeleteContentStream |
| LayerSafe | ContentStreamSafe |
| UseUnsafeLayers | UseUnsafeContentStreams |
| EncapsulateLayer | EncapsulateContentStream |
| RemoveSharedLayers | RemoveSharedContentStreams |

3. New functions

- **GetContentStreamToString** and **GetContentStreamToVariant** have been added to replace the functionality previously provided by the **GetPageContent** function.
- **SetContentStreamFromString** and **SetContentStreamFromVariant** have been added to replace the functionality previously provided by the **SetPageContent** function.
- **SetTextExtractionOptions** was added to provide greater control over the text extraction functions.
- **CheckFileCompliance** was added to check for PDF/A compliance along with the **GetStringListCount**, **GetStringListItem** functions needed to retrieve the compliance test results.
- **SetPDFAMode** was added to allow the creation of new PDF/A-1b compliant documents.
- Other new functions include **DrawPDF417Symbol**, **AddTrueTypeSubsettedFont**, **SetLineDashEx** and **SetRenderCropType**.

4. Removed functions

- The **SetAdvancePassword** and **SetPassword** functions have been removed. They are no longer necessary because the **LoadFrom*** functions now have an additional parameter for specifying the password.

5. Changed functionality

- When the **LoadFrom*** functions are used to open an encrypted document the objects will be automatically decrypted as necessary. Calls to **SetPassword**, **SetAdvancePassword** and **Decrypt** are no longer necessary to access any parts of an encrypted document. The **SaveTo*** functions can be used to save an encrypted document and the original encryption will remain in place with any new content automatically encrypted to the existing security settings.

- The **LoadFrom*** functions now return 1 on success and 0 on failure. In QPL v7 these functions would return 2 if an encrypted document made use of object streams or cross reference streams. This is no longer necessary as these functions have a new password parameter. If the wrong password is given, the **LoadFrom*** functions will return 0 and the **LastErrorCode** function will return error code 404.
- The **Encrypt** and **Decrypt** functions can still be used to add or remove security and full access to the document is possible even after encryption has been applied.
- In QPL v7, the **GetPageContent** function returned only a portion of the page's content stream. The replacement functions **GetPageContentToString** and **GetPageContentToVariant** now return the entire content stream of the selected page, not just an individual part.
- Similarly, the **SetPageContentFromString** and **SetPageContentFromVariant** functions now set the entire content stream of the selected page, not just an individual part.
- The **SecurityInfo** function returns the active encryption details even though encrypted object in documents are internally decrypted as needed.
- XMP metadata is now added to new documents and maintained in existing documents. Calling functions like **SetInformation** will result in the XMP metadata being updated to keep that particular item in the document information dictionary in sync.