DynaPDF 2.5 Features

This document describes improvements and new features which were added in DynaPDF 2.5. Changes which affect backward compatibility are described in the file compatibility_notes.pdf.

Color spaces

All PDF color spaces can now also be used with images. Most color spaces were already available in DynaPDF 2.0 but the corresponding handling for images was not implemented.

It is also possible to import the image color space as is. This can be very useful if the image is defined in a device independent color space such as Lab or any ICC based color space. See SetGStateFlags() for further information. The section "Color Spaces" in the help file provides detailed information about color spaces and the corresponding image handling.

Content Parser

The function ParseContent() is now fully documented (as well as all other functions) and easier to understand examples were added to the package. It is now also allowed to write contents to the page while the parser is executed.

EMF Conversion

Spool fonts are now automatically loaded when converting spool EMF files. It is no longer required to add the user's temp directory to the list of font search paths. Existing applications should be changed to avoid unnecessary calls of AddFontSearchPath() / ClearHostFonts(). AddFontSearchPath() does no longer consider fonts with the extension tmp.

The EMF converter produces a warning if a spool font cannot be loaded.

Reverse mapping for glyph indexes

DynaPDF applies now a reverse mapping when glyph indexes are used to output text. This enables the extraction of human readable text from the resulting PDF file.

Encryption

The new encryption flag rsEmbFilesOnly can be used to encrypt embedded files only. This flag can be used in combination with 128 or 256 bit AES encryption.

External Signatures

PDF files can now be signed with an external signature handler like the Windows CryptAPI. This enables also the usage of hardware or software certificates of the system's certificate store. The signature handler must only be able to create DER encoded PKCS# 7 objects because CER encoded PKCS# 7 objects are not supported by Adobe's Acrobat or Reader. See CloseAndSignFileExt() in the help file for further information.

Font handling

DynaPDF supports now OpenType fonts with Postscript outlines as well as external CMaps. External CMaps are very important when creating PDF files for Asian locales because this technology enables the usage of non-embedded OpenType or TrueType based CID fonts in combination with arbitrary CJK encodings. Due to the large number of characters which are typically used in Asian scripts it is often not possible or wished to embed all fonts in the PDF file. The usage of non-embedded CID fonts can greatly reduce the size of the resulting PDF file.

Support for external CMaps is also required to extract human readable text from such documents. See SetCIDFont(), SetCMapDir(), GetPageText(), and ParseContent() for further information.

Font Selection

The font selection algorithm has been improved to enable the selection of arbitrary font weights from 100 (Thin) through 1000 (UltraBlack) in combination with the family name. Prior versions supported the font weights regular and bold only and due to this limitation it was impossible to select other weights with the family name if more weights were installed.

It is now possible to select all font weights from 100 through 1000. If a specific font weight is not available, DynaPDF selects the next thinner weight end emulates the missing weight if necessary (the emulation can also be disabled with SetFontWeight()).

In addition, one font can be installed in up to three different formats (OpenType, TrueType, or Type1) and all these formats are selectable. It is also possible to explicitly disable certain font formats from selection (see SetFontSearchOrder() for further information).

The default font search order has been changed to TrueType, OpenType, Type1, StdFonts. The 14 PDF standard fonts had the highest search priority in prior versions. Also if a standard font was installed on the system it was impossible to select it unless standard fonts were explicitly disabled. This handling was not optimal. System fonts should have a higher priority than standard fonts.

Missing Glyphs

All functions to output text produce now a warning if one or more glyphs could not be found. The warning is passed to the error callback function if set. The return value of the used text function is still true in such a case. This is required to preserve backward compatibility. If no error callback function is used the application can call pdfGetMissingGlyphs() or GetErrorMessage() to determine whether certain glyphs are absent.

Interactive Forms

The new function GetFieldEx() offers much easier access to practically all field properties. It is now also possible to access the children of a group field or radio button directly. The creation of field appearances has been revoked to archive identical results in comparison to Acrobat 9.

Template handling

Placing an imported page on a destination page could be rather difficult with DynaPDF 2.0 especially if the original page was rotated and if it contained a non-normalized media or crop box.

The new function PlaceTemplateEx() considers all the painful things like a maybe existing crop box, original page orientation, clipping and so on automatically when placing an imported page on a destination page. This function simplifies the handling of such templates a lot.

List of new functions

Many functions in the following list were already available in DynaPDF 2.0 but not documented.

Font API:

```
fntBuildFamilyNameAndStyle    // Returns the family or postscript name
fntTranslateRawCode           // Converts a byte sequence to Unicode
```

DynaPDF API:

```
pdfAddDeviceNProcessColorants // Defines used process colorants
pdfAddDeviceNSeparations      // Defines used spot colorants
pdfAddImage                   // Helper function to extract images
pdfCloseAndSignFileExt        // External signatures
pdfCloseImage                 // Helper function to extract images
pdfCloseTag                   // Tagged PDF
pdfComputeBBox                // Computes the visible area of a page
pdfConvertColors              // Converts inline color operators
pdfCreateAnnotAP              // Custom appearance for stamps
pdfCreateDeviceNColorSpace    // Creates a DeviceN color space
pdfCreateImage                // Helper function to extract images
pdfCreateStructureTree        // Tagged PDF
pdfDeleteAcroForm             // Deletes an Interactive Form
pdfDeleteEmbeddedFile         // Deletes an embedded file
pdfDeleteJavaScripts          // Deletes all global JavaScripts
pdfDeleteXFAForm              // Deletes an existing XFA form
pdfFinishSignature            // Finishes an external signature
pdfFlattenAnnots              // Flattens annotations
pdfFreeImageBuffer            // Helper function to extract images
pdfGetAnnotEx                 // Extended version of GetAnnot()
pdfGetCheckBoxCharEx          // Returns the character of a check box
pdfGetCMap                    // Properties of an external CMap
pdfGetCMapCount               // Number of available external CMaps
pdfGetColorSpaceCount         // Number of color space objects
pdfGetColorSpaceObj           // Properties of a color space
pdfGetColorSpaceObjEx         // Properties of a color space
pdfGetDeviceNAttributes       // Attributes of a DeviceN color space
pdfGetFieldEx                 // Extended version of GetField()
pdfGetFieldEx2                // Version to access children of a field
pdfGetFontCount               // Number of fonts in the document
pdfGetFontEx                  // Properties of a font object
pdfGetFontSearchOrder         // Returns the font search order
```

```
pdfGetImageBuffer              // Helper function to extract images
pdfGetInIsXFAForm              // Contains the open PDF file a XFA form?
pdfGetInNamedDest              // Properties of a named destination
pdfGetInNamedDestCount         // Number of named destinations
pdfGetInPrintSettings          // Print dialog settings of a PDF file
pdfGetJavaScriptAction2        // Including event that executes it
pdfGetMissingGlyphs            // Number of glyphs not found in a font
pdfGetNeedAppearance           // Interactive Forms: Default is false
pdfGetPageAnnot                // Per page access
pdfGetPageAnnotCount           // Number of annotations used in page
pdfGetPageAnnotEx              // Extended version of GetPageAnnot()
pdfGetPageFieldEx              // Per page access
pdfGetPrintSettings            // Current print dialog settings
pdfGetTextFieldValue           // Value of a text field
pdfHighlightAnnot              // Creates a highlight annotation
pdfLoadCMap                    // Loads an external CMap
pdfLoadFDFData                 // Import FDF data
pdfLoadFDFDataEx               // Accepts a file buffer
pdfOpenTag                     // Tagged PDF
pdfPlaceTemplateEx             // Special version for imported templates
pdfSet3DAnnotProps             // Sets certain 3D annotation properties
pdfSet3DAnnotScript            // Adds a JavaScript to a 3D annotation
pdfSetAnnotString              // Sets or deletes certain strings
pdfSetCIDFont                  // Loads a font with an external CMap
pdfSetCMapDir                  // Loads external CMap directories
pdfSetFillColorEx              // Supports up to 32 color channels
pdfSetFontSearchOrder          // Changes the font search order
pdfSetImportFlags2             // Additional import flags
pdfSetNeedAppearance           // Sometimes useful
pdfSetPrintSettings            // Sets the default print dialog settings
pdfSetStrokeColorEx            // Supports up to 32 color channels
pdfSetTextFieldValueEx         // Extended version of SetTextFieldValue
pdfStampAnnot                  // Creates a stamp annotation
```