

## PDFDevice Enhancements

**ID :** 1973**Implemented in version :** 5.0.0.0**Short Description:** New static \$file... functions**Full Description:** Version 5 implements a set of file based static functions for merging, encrypting, signing and reading PDF files.

PDF Device.\$filemerge( cInputFile1, cInputFile2, cOutputFile) Returns err  
 Merges two PDF files appending cInputFile2 to cInputFile1.  
 Returns 1 if succesful, 0 or a negative error code on failure. Use

PDF Device.\$fileencrypt( cInputFile, cOutputFile, cUserPassword,  
 cOwnerPassword, iEncryption, iPermissions ) Returns err  
 Encrypts the file cInputFile using the provided passwords, encyption and permissions,  
 and writes the result to cOutputFile.  
 Returns 1 if succesful, 0 or a negative error code on failure. Use

PDF Device.\$filesign( cInputFile, cOutputFile, cCertificateFile, cKeyFile, cKeyFilePassword  
 [, cSignReason, bSigPrintable=kFalse, bSigUseExisting=kTrue,  
 cSigFieldName, wAnnotationAtts] ) Returns err  
 Signs cInputFile using the certificate, private key and various options, and writes the resulting  
 PDF to cOutputFile. When cCertificateFile file is a .p12 file that includes both the certificate  
 and private key, cKeyFile can be left empty but cKeyFilePassword must be specified as it is  
 needed to extract the information from the p12 file.  
 Returns 1 if succesful, 0 or a negative error code on failure. Use

PDF Device.\$filereaddata( cInputFile, cOutputListName ) Returns err  
 Reads the field names and their data from Acroform fields of the given input file.  
 The output list name must be contextually accessible from within the context of  
 the calling method.  
 Returns 1 if succesful, 0 or a negative error code on failure. Use

PDF Device.\$filegetlasterror( cOutErrorText ) Returns err  
 Returns additional error information.

**Comments :****ID :** 1974**Implemented in version :** 5.0.0.0**Short Description:** New form field report object**Full Description:** Version 5 implements a form field report object for embedding acro form fields in PDF files. Supported acro form field types are push-button, checkbox, text, choice and signature fields. Radio button fields are currently not supported, but the choice field can be used instead.

Form Field Properties Summary

\$dataname: PDF Form Object's initial value (if blank, \$defaultvalue is used instead, otherwise data replaces default value). For pushbuttons, the data or default value specifies the action of which there

are currently three types supported:

"URL~" When the value begins with the phrase "URL~" the subsequent data specifies an URL for submitting the form data using a standard html POST message. Example:

"URL~https://demos.brainydata.com/test.php"

"Reset~" Generates a simple reset form action for the button and no additional data follows the phrase "Reset~"

"Script~" The data that follows the phrase "Script~" is Acrobat Javascript. For more details see the Adobe Acrobat Javascript reference at

[https://opensource.adobe.com/dc-acrobat-sdk-docs/acrobatsdk/pdfs/acrobatsdk\\_jsapiref.pdf](https://opensource.adobe.com/dc-acrobat-sdk-docs/acrobatsdk/pdfs/acrobatsdk_jsapiref.pdf)

\$defaultvalue: PDF Form Object's default value (initial and when form is reset). See \$dataname for details relating to pushbuttons.

\$fieldtype: PDF Form Object's type (Pushbutton, Checkbox, Text, Choice).

Note: The Radio type is currently not supported

\$fieldname: The internal name of form field which can be used by javascript to identify and manipulate form fields during actions. This maps to the 'T' entry in the field dictionary in PDF. See the adobe PDF reference for more details.

\$submitname: the name to be used when the data is submitted. This maps to the 'TM' entry in the field dictionary in PDF. See the adobe PDF reference for more details.

\$displayname: The name to be displayed in the user interface. This maps to the 'TU' entry in the field dictionary in PDF. See the adobe PDF reference for more details.

\$fieldflags: Set of flags for field. Which flags are relevant depend on \$fieldtype (one of the kDevPdffff... constants). This maps to the 'Ff' entry in the field dictionary in PDF. See the adobe PDF reference for more details.

\$strokewidth, \$strokestyle, \$strokecolor, \$strokeradius, \$fillcolor: These properties can be used to specify additional adornments that are not an integral part of the control itself. Together, they specify the fields framing and fill options.

\$fontname, \$fontencoding, \$fontsize, \$textcolor: These properties specify the form fields text properties. The \$fontname property can be set to one of the standard font constants (kDevPdffffsf...).

The \$fontencoding can be set to one of the encoding constants (kDevPdffffe...).

Note: the text properties of the first form field will be used as the default text setting for editable text fields.

#### Form Field Constants (kDevPdffff...)

kDevPdfffft... Field types used with \$fieldtype

...Button - Form button field (Pushbutton, Checkbox or Radio button, see kDevPdffff flags)

...Text - Form text field (single/multi-line/password, etc, see kDevPdffff flags)

...Choice - Form choice list (listbox or combo, see kDevPdffff flags)

...Signature - Form signature field

kDevPdffffb... Fieldborder stroke types used with \$strokestyle

...Solid - Solid line

...Dash - Dashed line

...Dot - Dotted line

...DashDot - Dash-Dot line

...DashDotDot - Dash-Dot-Dot line

kDevPdfffff... Field flags used with \$fieldflags

...ReadOnly - If set, the user may not change the value of the field.

...Required - If set, the field must have a value at the time it is exported by a submit-form action.

...NoExport - If set, the field must not be exported by a submit-form action.

...Multiline - If set, the field can contain multiple lines of text

...Password - If set, the field is intended for entering a secure password

...PushButton - If set, the field is a pushbutton that does not retain a permanent value. (button fields)  
...Combo - If set, the field is a combo box; if clear, the field is a list box. (list fields)  
...ComboEdit - If set, the combo box includes an editable text box as well as a drop-down list (list fields)  
...Sort - If set, the field's option items should be sorted alphabetically. (list fields)  
...MultiSelect - If set, more than one of the field's option items may be selected simultaneously. (list fields)  
...FileSelect - If set, the field value represents the pathname of a file to be submitted. (text fields)  
...DoNotSpellCheck - If set, text entered in the field is not spell-checked. (text/combo fields only)  
...DoNotScroll - If set, the field does not scroll horizontally or vertically. (text fields only)  
...RichText - If set, the value of this field contains rich text, i.e. fully-formed XML. (text fields only)  
...CommitOnSelChange - If set, the new value is committed as soon as selection changes. (list fields only)

kDevPdfFFFE... Font encoding used with \$fntencoding

...platform - Either kDevPdfFFFEmacRoman or kDevPdfFFFEwinAnsi

...MacRoman - Mac standard encoding

...WinAnsi - Windows standard encoding

...Symbol - Symbol font encoding

...ZapfDingbats - ZapfDingbats font encoding

kDevPdfFFSF... Standard fonts used with \$fntname (Postscript name supported by Adobe Readers)

...Helvetica (Helvetica)

...HelveticaBold (Helvetica-Bold)

...HelveticaOblique (Helvetica-Oblique)

...HelveticaBoldOblique (Helvetica-BoldOblique)

...TimesRoman (Times-Roman)

...TimesBold (Times-Bold)

...TimesItalic (Times-Italic)

...TimesBoldItalic (Times-BoldItalic)

...Courier (Courier)

...CourierBold (Courier-Bold)

...CourierOblique (Courier-Oblique)

...CourierBoldOblique (Courier-BoldOblique)

...Symbol (Symbol)

...ZapfDingbats (ZapfDingbats)

### Known Problems

- The bSigUseExisting option for \$filesign() is not working.
- Radio buttons will not be supported in version 5. Instead use the choice list fields (i.e. combo or list box). We will re-consider the support of radio buttons for a future intermediate feature release, should it become important.
- Error reporting for the new \$file... functions is working in general, but on occasion can be incoherent due to some errors being stacked.
- Printing form objects to any destination other than PDF Device will not show anything at all. This is a technical limitation related to Omnis report comment objects that we must use to insert the special information required to produce PDF form fields in the final output, while at the same time avoiding the insertion of unwanted visual representations that would be required for other destinations.

### Open Source Commitment

The open source we used to build PDFDevice version 5 is available to anyone with a developer support subscription. To receive the source please use our online support request form at [https://support.brainydata.com/request\\_form.htm](https://support.brainydata.com/request_form.htm).

We will provide the source as is, but cannot offer additional support to help with the building of the open source, albeit we will provide projects for building wherever possible.

**Comments :**

