Forms Data Format

Introduction

Forms Data Format (FDF) is a format for handling forms within PDF documents. You should read the documentation at » http://partners.adobe.com/asn/acrobat/forms.jsp for more information on what FDF is and how it is used in general.

The general idea of FDF is similar to HTML forms. The difference is basically the format how data is transmitted to the server when the submit button is pressed (this is actually the Form Data Format) and the format of the form itself (which is the Portable Document Format, PDF). Processing the FDF data is one of the features provided by the fdf functions. But there is more. One may as well take an existing PDF form and populated the input fields with data without modifying the form itself. In such a case one would create a FDF document (fdf_create()) set the values of each input field (fdf_set_value()) and associate it with a PDF form (fdf_set_file()). Finally it has to be sent to the browser with MimeType *application/vnd.fdf*. The Acrobat reader plugin of your browser recognizes the MimeType, reads the associated PDF form and fills in the data from the FDF document.

If you look at an FDF-document with a text editor you will find a catalogue object with the name *FDF*. Such an object may contain a number of entries like *Fields*, *F*, *Status* etc.. The most commonly used entries are *Fields* which points to a list of input fields, and *F* which contains the filename of the PDF-document this data belongs to. Those entries are referred to in the FDF documentation as /F-Key or /Status-Key. Modifying this entries is done by functions like fdf_set_file() and fdf_set_status(). Fields are modified with fdf_set_value(), fdf_set_opt() etc..

Installing/Configuring

Requirements

You need the FDF toolkit SDK available from
» http://partners.adobe.com/asn/acrobat/forms.jsp. As of PHP 4.3.0 you need at least SDK version 5.0. The FDF toolkit library is available in binary form only, platforms supported by Adobe are Win32, Linux, Solaris and AIX.

Installation

You must compile PHP with *--with-fdftk[=DIR]*.

Note
If you run into problems configuring PHP with fdftk support, check whether the header file *fdftk.h* and the library *libfdftk.so* are at the right place. The configure script supports both the directory structure of the FDF SDK distribution and the usual *DIR/include* / *DIR/lib* layout, so you can point it either directly to the unpacked distribution directory or put the header file and the appropriate library for your platform into e.g. */usr/local/include* and */usr/local/lib* and configure with *--with-fdftk=/usr/local*.

Note
Note to Win32 Users In order for this extension to work, there are DLL files that must be available to the Windows system *PATH*. For information on how to do this, see the FAQ entitled " How do I add my PHP directory to the PATH on Windows ". Although copying DLL files from the PHP folder into the Windows system directory also works (because the system directory is by default in the system's *PATH*), this is not recommended. *This extension requires the following files to be in the PATH: fdftk.dll*

Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

Resource Types

Most fdf functions require a `fdf` resource as their first parameter. A `fdf` resource is a

handle to an opened fdf file. *fdf* resources may be obtained using fdf_create(), fdf_open() and fdf_open_string().

Predefined Constants

The constants below are defined by this extension, and will only be available when the extension has either been compiled into PHP or dynamically loaded at runtime.

FDFValue (integer)

FDFStatus (integer)

FDFFile (integer)

FDFID (integer)

FDFFf (integer)

FDFSetFf (integer)

FDFClearFf (integer)

FDFFlags (integer)

FDFSetF (integer)

FDFClrF (integer)

FDFAP (integer)

FDFAS (integer)

FDFAction (integer)

FDFAA (integer)

FDFAPRef (integer)

FDFIF (integer)

FDFEnter (integer)

FDFExit (integer)

FDFDown (integer)

FDFUp (integer)

FDFFormat (integer)

FDFValidate (integer)

FDFKeystroke (integer)

FDFCalculate (integer)

FDFNormalAP (integer)

FDFRolloverAP (integer)

FDFDownAP (integer)

Examples

The following examples shows just the evaluation of form data.

Example #1 - Evaluating a FDF document

```
<?php
// Open fdf from input string provided by the extension
// The pdf form contained several input text fields with the names
// volume, date, comment, publisher, preparer, and two checkboxes
// show_publisher and show_preparer.
$fdf = fdf_open_string($HTTP_FDF_DATA);
$volume = fdf_get_value($fdf, "volume");
echo "The volume field has the value '<b>$volume</b>'<br />";

$date = fdf_get_value($fdf, "date");
echo "The date field has the value '<b>$date</b>'<br />";

$comment = fdf_get_value($fdf, "comment");
echo "The comment field has the value '<b>$comment</b>'<br />";

if (fdf_get_value($fdf, "show_publisher") == "On") {
 $publisher = fdf_get_value($fdf, "publisher");
 echo "The publisher field has the value '<b>$publisher</b>'<br />";
} else
 echo "Publisher shall not be shown.<br />";

if (fdf_get_value($fdf, "show_preparer") == "On") {
 $preparer = fdf_get_value($fdf, "preparer");
 echo "The preparer field has the value '<b>$preparer</b>'<br />";
} else
 echo "Preparer shall not be shown.<br />";
fdf_close($fdf);
?>
```

FDF Functions

fdf_add_doc_javascript

fdf_add_doc_javascript -- Adds javascript code to the FDF document

Description

bool **fdf_add_doc_javascript** (resource `$fdf_document`, string `$script_name`, string `$script_code`)

Adds a script to the FDF, which Acrobat then adds to the doc-level scripts of a document, once the FDF is imported into it.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`script_name`
The script name.

`script_code`
The script code. It is strongly suggested to use *\r* for linebreaks within the script code.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #2 - Adding JavaScript code to a FDF

```
<?php
$fdf = fdf_create();
fdf_add_doc_javascript($fdf, "PlusOne", "function PlusOne(x)\r{\r  return
x+1;\r}\r");
fdf_save($fdf);
?>
```

will create a FDF like this:

```
%FDF-1.2
%âãÏÓ
1 0 obj
<<
/FDF << /JavaScript << /Doc [ (PlusOne)(function PlusOne\(x\)\r{\r  return
x+1;\r}\r)] >> >>
>>
```

```
endobj
trailer
<<
/Root 1 0 R

>>
%%EOF
```

fdf_add_template

fdf_add_template -- Adds a template into the FDF document

Description

bool **fdf_add_template** (resource `$fdf_document`, int `$newpage`, string `$filename`, string `$template`, int `$rename`)

Warning
This function is currently not documented; only its argument list is available.

fdf_close

fdf_close -- Close an FDF document

Description

void **fdf_close** (resource `$fdf_document`)

Closes the FDF document.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

Return Values

No value is returned.

See Also

- fdf_open()

fdf_create

fdf_create -- Create a new FDF document

Description

resource **fdf_create** (void)

Creates a new FDF document.

This function is needed if one would like to populate input fields in a PDF document with data.

Return Values

Returns a FDF document handle, or **FALSE** on error.

Examples

Example #3 - Populating a PDF document

```
<?php
$outfdf = fdf_create();
fdf_set_value($outfdf, "volume", $volume, 0);

fdf_set_file($outfdf, "http:/testfdf/resultlabel.pdf");
fdf_save($outfdf, "outtest.fdf");
fdf_close($outfdf);
Header("Content-type: application/vnd.fdf");
$fp = fopen("outtest.fdf", "r");
fpassthru($fp);
unlink("outtest.fdf");
?>
```

See Also

- fdf_close()
- fdf_save()
- fdf_open()

fdf_enum_values

fdf_enum_values -- Call a user defined function for each document value

Description

bool **fdf_enum_values** (resource `$fdf_document`, callback `$function` [, mixed `$userdata`])

Warning
This function is currently not documented; only its argument list is available.

fdf_errno

fdf_errno -- Return error code for last fdf operation

Description

int **fdf_errno** (void)

Gets the error code set by the last FDF function call.

A textual description of the error may be obtained using with fdf_error().

Return Values

Returns the error code as an integer, or zero if there was no errors.

See Also

- fdf_error()

fdf_error

fdf_error -- Return error description for FDF error code

Description

string **fdf_error** ([int `$error_code`])

Gets a textual description for the FDF error code given in *`error_code`*.

Parameters

`error_code`
: An error code obtained with fdf_errno(). If not provided, this function uses the internal error code set by the last operation.

Return Values

Returns the error message as a string, or the string *no error* if nothing went wrong.

See Also

- fdf_errno()

fdf_get_ap

fdf_get_ap -- Get the appearance of a field

Description

bool **fdf_get_ap** (resource `$fdf_document`, string `$field`, int `$face`, string `$filename`)

Gets the appearance of a *`field`* (i.e. the value of the /AP key) and stores it in a file.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`field`

`face`
The possible values are **FDFNormalAP**, **FDFRolloverAP** and **FDFDownAP**.

`filename`
The appearance will be stored in this parameter.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

fdf_get_attachment

fdf_get_attachment -- Extracts uploaded file embedded in the FDF

Description

array **fdf_get_attachment** (resource `$fdf_document`, string `$fieldname`, string `$savepath`)

Extracts a file uploaded by means of the "file selection" field *`fieldname`* and stores it under *`savepath`*.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`fieldname`

`savepath`
May be the name of a plain file or an existing directory in which the file is to be created under its original name. Any existing file under the same name will be overwritten.

Note
There seems to be no other way to find out the original filename but to store the file using a directory as *`savepath`* and check for the basename it was stored under.

Return Values

The returned array contains the following fields:

- *`path`* - path were the file got stored
- *`size`* - size of the stored file in bytes
- *`type`* - mimetype if given in the FDF

Examples

Example #4 - Storing an uploaded file

```
<?php
 $fdf = fdf_open_string($HTTP_FDF_DATA);
 $data = fdf_get_attachment($fdf, "filename", "/tmpdir");
 echo "The uploaded file is stored in $data[path]";
?>
```

fdf_get_encoding

fdf_get_encoding -- Get the value of the /Encoding key

Description

string **fdf_get_encoding** (resource `$fdf_document`)

Gets the value of the */Encoding* key.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

Return Values

Returns the encoding as a string. An empty string is returned if the default *PDFDocEncoding/Unicode* scheme is used.

See Also

- fdf_set_encoding()

fdf_get_file

fdf_get_file -- Get the value of the /F key

Description

string **fdf_get_file** (resource `$fdf_document`)

Gets the value of the */F* key.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

Return Values

Returns the key value, as a string.

See Also

- fdf_set_file()

fdf_get_flags

fdf_get_flags -- Gets the flags of a field

Description

int **fdf_get_flags** (resource `$fdf_document`, string `$fieldname`, int `$whichflags`)

Warning
This function is currently not documented; only its argument list is available.

fdf_get_opt

fdf_get_opt -- Gets a value from the opt array of a field

Description

mixed **fdf_get_opt** (resource `$fdf_document`, string `$fieldname` [, int `$element`])

Warning
This function is currently not documented; only its argument list is available.

fdf_get_status

fdf_get_status -- Get the value of the /STATUS key

Description

string **fdf_get_status** (resource `$fdf_document`)

Gets the value of the */STATUS* key.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

Return Values

Returns the key value, as a string.

See Also

- fdf_set_status()

fdf_get_value

fdf_get_value -- Get the value of a field

Description

mixed **fdf_get_value** (resource `$fdf_document`, string `$fieldname` [, int `$which`])

Gets the value for the requested field.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`fieldname`
Name of the FDF field, as a string.

`which`
Elements of an array field can be retrieved by passing this optional parameter, starting at zero. For non-array fields, this parameter will be ignored.

Return Values

Returns the field value.

ChangeLog

Version	Description
4.3.0	Support for arrays and the *`which`* parameter were added.

See Also

- fdf_set_value()

fdf_get_version

fdf_get_version -- Gets version number for FDF API or file

Description

string **fdf_get_version** ([resource `$fdf_document`])

Return the FDF version for the given document, or the toolkit API version number if no parameter is given.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

Return Values

Returns the version as a string. For the current FDF toolkit 5.0 the API version number is *5.0* and the document version number is either *1.2*, *1.3* or *1.4*.

See Also

- fdf_set_version()

fdf_header

fdf_header -- Sets FDF-specific output headers

Description

void **fdf_header** (void)

This is a convenience function to set appropriate HTTP headers for FDF output. It sets the *Content-type:* to *application/vnd.fdf*.

Return Values

No value is returned.

fdf_next_field_name

fdf_next_field_name -- Get the next field name

Description

string **fdf_next_field_name** (resource `$fdf_document` [, string `$fieldname`])

Gets the name of the field after the given field. This name can be used with several functions.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`fieldname`
Name of the FDF field, as a string. If not given, the first field will be assumed.

Return Values

Returns the field name as a string.

Examples

Example #5 - Detecting all fieldnames in a FDF

```
<?php
$fdf = fdf_open($HTTP_FDF_DATA);
for ($field = fdf_next_field_name($fdf);
   $field != "";
   $field = fdf_next_field_name($fdf, $field)) {
   echo "field: $field\n";
}
?>
```

See Also

- fdf_get_value()

fdf_open_string

fdf_open_string -- Read a FDF document from a string

Description

resource **fdf_open_string** (string `$fdf_data`)

Reads form data from a string.

You can use fdf_open_string() together with *$HTTP_FDF_DATA* to process FDF form input from a remote client.

Parameters

`fdf_data`
The data as returned from a PDF form or created using fdf_create() and fdf_save_string().

Return Values

Returns a FDF document handle, or **FALSE** on error.

Examples

Example #6 - Accessing the form data

```
<?php
$fdf = fdf_open_string($HTTP_FDF_DATA);
/* ... */
fdf_close($fdf);
?>
```

See Also

- fdf_open()
- fdf_close()
- fdf_create()
- fdf_save_string()

fdf_open

fdf_open -- Open a FDF document

Description

resource **fdf_open** (string `$filename`)

Opens a file with form data.

You can also use fdf_open_string() to process the results of a PDF form POST request.

Parameters

`filename`
: Path to the FDF file. This file must contain the data as returned from a PDF form or created using fdf_create() and fdf_save().

Return Values

Returns a FDF document handle, or **FALSE** on error.

Examples

Example #7 - Accessing the form data

```
<?php
// Save the FDF data into a temp file
$fdffp = fopen("test.fdf", "w");
fwrite($fdffp, $HTTP_FDF_DATA, strlen($HTTP_FDF_DATA));
fclose($fdffp);

// Open temp file and evaluate data
$fdf = fdf_open("test.fdf");
/* ... */
fdf_close($fdf);
?>
```

See Also

- fdf_open_string()
- fdf_close()

- fdf_create()
- fdf_save()

fdf_remove_item

fdf_remove_item -- Sets target frame for form

Description

bool **fdf_remove_item** (resource `$fdf_document`, string `$fieldname`, int `$item`)

Warning
This function is currently not documented; only its argument list is available.

fdf_save_string

fdf_save_string -- Returns the FDF document as a string

Description

string **fdf_save_string** (resource `$fdf_document`)

Returns the FDF document as a string.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

Return Values

Returns the document as a string, or **FALSE** on error.

Examples

Example #8 - Retrieving FDF as a string

```
<?php
$fdf = fdf_create();
fdf_set_value($fdf, "foo", "bar");
$str = fdf_save_string($fdf);
fdf_close($fdf);
echo $str;
?>
```

The above example will output:

```
%FDF-1.2
%âãÏÓ
1 0 obj
<<
/FDF << /Fields 2 0 R >>
>>
endobj
2 0 obj
[
<< /T (foo)/V (bar)>>
]
endobj
trailer
<<
/Root 1 0 R
```

```
>>
%%EOF
```

See Also

- fdf_open_string()
- fdf_close()
- fdf_create()
- fdf_save()

fdf_save

fdf_save -- Save a FDF document

Description

bool **fdf_save** (resource `$fdf_document` [, string `$filename`])

Saves a FDF document.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`filename`
If provided, the resulting FDF will be written in this parameter. Otherwise, this function will write the FDF to the default PHP output stream.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- fdf_close()
- fdf_create()
- fdf_save_string()

fdf_set_ap

fdf_set_ap -- Set the appearance of a field

Description

bool **fdf_set_ap** (resource `$fdf_document`, string `$field_name`, int `$face`, string `$filename`, int `$page_number`)

Sets the appearance of a field (i.e. the value of the */AP* key).

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`field_name`

`face`
The possible values **FDFNormalAP**, **FDFRolloverAP** and **FDFDownAP**.

`filename`

`page_number`

Return Values

Returns **TRUE** on success or **FALSE** on failure.

fdf_set_encoding

fdf_set_encoding -- Sets FDF character encoding

Description

bool **fdf_set_encoding** (resource `$fdf_document`, string `$encoding`)

Sets the character encoding for the FDF document.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`encoding`
The encoding name. The following values are supported: " *Shift-JIS* ", " *UHC* ", " *GBK* " and " *BigFive* ". An empty string resets the encoding to the default *PDFDocEncoding/Unicode* scheme.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

fdf_set_file

fdf_set_file -- Set PDF document to display FDF data in

Description

bool **fdf_set_file** (resource `$fdf_document`, string `$url` [, string `$target_frame`])

Selects a different PDF document to display the form results in then the form it originated from.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`url`
Should be given as an absolute URL.

`target_frame`
Use this parameter to specify the frame in which the document will be displayed. You can also set the default value for this parameter using fdf_set_target_frame().

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #9 - Passing FDF data to a second form

```
<?php
 /* set content type for Adobe FDF */
 fdf_header();

 /* start new fdf */
 $fdf = fdf_create();

 /* set field "foo" to value "bar" */
 fdf_set_value($fdf, "foo", "bar");

 /* tell client to display FDF data using "fdf_form.pdf" */
 fdf_set_file($fdf, "http://www.example.com/fdf_form.pdf");

 /* output fdf */
 fdf_save($fdf);
```

```
 /* clean up */
 fdf_close($fdf);
?>
```

See Also

- fdf_get_file()
- fdf_set_target_frame()

fdf_set_flags

fdf_set_flags -- Sets a flag of a field

Description

bool **fdf_set_flags** (resource `$fdf_document`, string `$fieldname`, int `$whichFlags`, int `$newFlags`)

Sets certain flags of the given field.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`fieldname`
Name of the FDF field, as a string.

`whichFlags`

`newFlags`

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- fdf_set_opt()

fdf_set_javascript_action

fdf_set_javascript_action -- Sets an javascript action of a field

Description

bool **fdf_set_javascript_action** (resource `$fdf_document`, string `$fieldname`, int `$trigger`, string `$script`)

Sets a javascript action for the given field.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`fieldname`
Name of the FDF field, as a string.

`trigger`

`script`

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- fdf_set_submit_form_action()

fdf_set_on_import_javascript

fdf_set_on_import_javascript -- Adds javascript code to be executed when Acrobat opens the FDF

Description

bool **fdf_set_on_import_javascript** (resource `$fdf_document`, string `$script`, bool `$before_data_import`)

Warning
This function is currently not documented; only its argument list is available.

See Also

- fdf_add_doc_javascript()
- fdf_set_javascript_action()

fdf_set_opt

fdf_set_opt -- Sets an option of a field

Description

bool **fdf_set_opt** (resource `$fdf_document`, string `$fieldname`, int `$element`, string `$str1` , string `$str2`)

Sets options of the given field.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`fieldname`
Name of the FDF field, as a string.

`element`

`str1`

`str2`

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- fdf_set_flags()

fdf_set_status

fdf_set_status -- Set the value of the /STATUS key

Description

bool **fdf_set_status** (resource `$fdf_document`, string `$status`)

Sets the value of the */STATUS* key. When a client receives a FDF with a status set it will present the value in an alert box.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`status`

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- fdf_get_status()

fdf_set_submit_form_action

fdf_set_submit_form_action -- Sets a submit form action of a field

Description

bool **fdf_set_submit_form_action** (resource `$fdf_document`, string `$fieldname`, int `$trigger`, string `$script`, int `$flags`)

Sets a submit form action for the given field.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`fieldname`
Name of the FDF field, as a string.

`trigger`

`script`

`flags`

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- fdf_set_javascript_action()

fdf_set_target_frame

fdf_set_target_frame -- Set target frame for form display

Description

bool **fdf_set_target_frame** (resource `$fdf_document`, string `$frame_name`)

Sets the target frame to display a result PDF defined with **fdf_save_file()** in.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`frame_name`
The frame name, as a string.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- **fdf_save_file()**

fdf_set_value

fdf_set_value -- Set the value of a field

Description

bool **fdf_set_value** (resource `$fdf_document`, string `$fieldname`, mixed `$value` [, int `$isName`])

Sets the *`value`* for the given field.

Parameters

`fdf_document`
The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`fieldname`
Name of the FDF field, as a string.

`value`
This parameter will be stored as a string unless it is an array. In this case all array elements will be stored as a value array.

`isName`

Note
In older versions of the FDF toolkit last parameter determined if the field value was to be converted to a PDF Name (= 1) or set to a PDF String (= 0). The value is no longer used in the current toolkit version 5.0. For compatibility reasons it is still supported as an optional parameter beginning with PHP 4.3, but ignored internally.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

ChangeLog

Version	Description

4.3.0	Support for arrays in the *value* parameter was added.

See Also

- fdf_get_value()
- fdf_remove_item()

fdf_set_version

fdf_set_version -- Sets version number for a FDF file

Description

bool **fdf_set_version** (resource `$fdf_document`, string `$version`)

Sets the FDF *`version`* for the given document.

Some features supported by this extension are only available in newer FDF versions.

Parameters

`fdf_document`
: The FDF document handle, returned by fdf_create(), fdf_open() or fdf_open_string().

`version`
: The version number. For the current FDF toolkit 5.0, this may be either *1.2*, *1.3* or *1.4*.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

See Also

- fdf_get_version()