Foxit

QUICK PDF LIBRARY

LIB Edition
Version 15.11
GETTING STARTED

Installation

The LIB Edition of Foxit Quick PDF Library is generated automatically by converting the DLL Edition into a static link library.

The static link library file is called **DebenuPDFLibraryLIB1511.lib** and a header file called **DebenuPDFLibraryLIB1511.h** is included.

Link dependencies

The following resource files must be linked into the final executable:

```
DebenuPDFLibraryCMap.res
DebenuPDFLibraryDingbats.res
DebenuPDFLibraryGlyphList.res
DebenuPDFLibraryColorProfiles.res
```

The following import libraries must be added to your project:

```
KERNEL32.LIB
GDI32.LIB
ADVAPI32.LIB
OLE32.LIB
USER32.LIB
CRYPT32.LIB
OLEAUT32.LIB
WINSPOOL.LIB
```

In previous versions of the library, these import libraries were also required: MSVCRT.LIB, VERSION.LIB, COMCTL32.LIB but from version 8.15 they are no longer needed.

Initializing/releasing the library

The **DEBENUPDFLIBRARYLIB1511_DllMain** function should be called once before using any of the other functions, using the **DLL_PROCESS_ATTACH** constant for the **fdwReason** parameter.

The **DPLCreateLibrary** function must be called to initialize the library. An InstanceID will be returned which must be passed as the first parameter to all the other functions.

When you are finished with the library, call **DPLReleaseLibrary** to release all allocated memory.

Finally, before the application ends the **DEBENUPDFLIBRARYLIB1511_DllMain** function should be called again. This time the **DLL_PROCESS_DETACH** constant should be used.

Unlocking the library

Once you have an InstanceID, you should call the **DebenuPDFLibraryUnlockKey** function, passing it your license key, to unlock the library.

```
DEBENUPDFLIBRARYLIB1511_DllMain(GetModuleHandle(NULL), DLL_PROCESS_ATTACH, NULL);
int InstanceID = DebenuPDFLibraryCreateLibrary();
if (DebenuPDFLibraryUnlockKey(InstanceID, L"your license key") == 1) {
  DPLDrawText(InstanceID, 100, 500, L"Hello world");
  DPLSaveToFile(InstanceID, L"C:\\Docs\\HelloFromLIB.pdf");
}
DPLReleaseLibrary(InstanceID);
DEBENUPDFLIBRARYLIB1511_DllMain(GetModuleHandle(NULL), DLL_PROCESS_DETACH, NULL);
```

If you are linking the LIB file to a DLL and not an EXE file then you need to change the GetModuleHandle(NULL) reference to GetModuleHandle("mydllname") so that it can correctly load the resources from the DLL.

```
DEBENUPDFLIBRARYLIB1511_DllMain(GetModuleHandle("mydllname"), DLL_PROCESS_ATTACH,
NULL);
  int InstanceID = DebenuPDFLibraryCreateLibrary();
  if (QuickPDFUnlockKey(InstanceID, L"your license key") == 1) {
    DPLDrawText(InstanceID, 100, 500, L"Hello world");
    DPLSaveToFile(InstanceID, L"C:\\Docs\\HelloFromLIB.pdf");
  }
  DPLReleaseLibrary(InstanceID);
  DEBENUPDFLIBRARYLIB1511_DllMain(GetModuleHandle("mydllname"), DLL_PROCESS_DETACH,
NULL);
```

Sending strings to Foxit Quick PDF Library

Most Foxit Quick PDF Library string parameters are defined as PWideChars, which are pointers to 16-bit null-terminated Unicode strings (UTF-16BE).

Some functions accept 8-bit binary data. If the data you need to send to the library may contain null characters, you can ask Foxit Quick PDF Library to create a temporary buffer of a certain size.

Use the **DPLCreateBuffer** and **DPLAddToBuffer** functions to create the buffer and fill it with data. The value returned by the **DPLCreateBuffer** function can then be used for any 8-bit or 16-bit string parameter:

```
char * Buffer;
char * Content = ...; // pointer to the data
Buffer = DPLCreateBuffer(10000);
DPLAddToBuffer(InstanceID, Buffer, Content, 10000);
DPLStoreCustomDataFromString(InstanceID,
  "MyData", Buffer, 1, 0);
DPLReleaseBuffer(InstanceID, Buffer);
```

Receiving strings from Foxit Quick PDF Library

Most functions that return string data will return a PWideChar, a pointer to a Unicode string. The memory for this string is contained within the Foxit Quick PDF Library instance.

The data in the string should be copied out immediately as the same memory will be used for subsequent calls to the library.

To get the length of the returned string, use the **DPLStringResultLength** function. This function returns the number of Unicode characters (multiply by 2 to get the size in bytes).

Some functions may return data that contains 8-bit data, possibly including null characters.

To get the length of the binary data, use the **DPLAnsiStringResultLength** function:

```
char * Content;
int ContentLength;
Content = DPLRetrieveCustomDataToString(InstanceID,
  "MyData", 1);
ContentLength = DPLAnsiStringResultLength(InstanceID);
// copy the data in Content now
```

Product website

Please visit the Foxit Quick PDF Library website for news and information:

http://www.debenu.com/