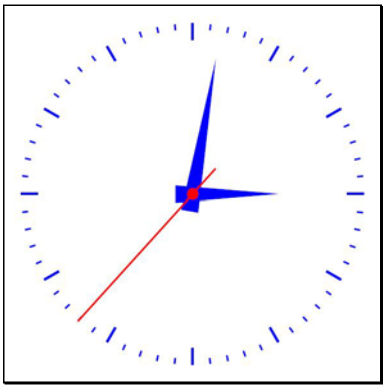
DirectPDF note #01

This document will help you to create well-generated PDF document while you're developing using the COM Component DirectPDF.

1. How can I get a function that adds a graphical clock with the current time on a PDF page?

Adding a clock on a generated PDF document including arrows that show current second, minute and hour is quite easy using DirectPDF.
To complete such goal, you just need to use the drawing function once you get the current time. Here's an example of a generation according to the source code you'll get as implementation from this DirectPDF note.



2. What to code?

2.1. Prerequisite

To run the following code you need to install the COM Component DirectPDF, which is available on website http://DirectPDF.free.fr. Then do not forget to register it (See *DirectPDF Reference Manual*).

2.2. Function DrawClock

DrawClock is a function that draws a clock on the current generated PDF page showing the current generation time.

Modifying the function to fit your need is very easy. The possibility seems endless like:

- Resizing the global drawing to set it as foot page;
- Receiving time to draw using parameter;
- Etc…

Here’s the source code:

```
#include "time.h"

#define RADIUS          200.0f
#define MARGIN          20.0f

void DrawClock(ICOMAccess* pICAccess)
{
HRESULT hr;
float alpha;
time_t timer;
struct tm ltime;

        hr = pICAccess->Pdf_Translate(RADIUS + MARGIN, RADIUS + MARGIN);
        hr = pICAccess->Pdf_SetColor(L"fillstroke", L"rgb", 0, 0, 1, 0);
        hr = pICAccess->Pdf_Save();

        // minute strokes
        hr = pICAccess->Pdf_SetLineWidth(2);
        for (alpha = 0; alpha < 360; alpha += 6)
        {
            hr = pICAccess->Pdf_Rotate(6);
            hr = pICAccess->Pdf_MoveTo(RADIUS, 0);
            hr = pICAccess->Pdf_LineTo((float) (RADIUS-MARGIN/3), 0);
            hr = pICAccess->Pdf_Stroke();
        }

        hr = pICAccess->Pdf_Restore();
        hr = pICAccess->Pdf_Save();

        // 5 minute strokes
        hr = pICAccess->Pdf_SetLineWidth(3);
        for (alpha = 0; alpha < 360; alpha += 30)
        {
            hr = pICAccess->Pdf_Rotate(30);
            hr = pICAccess->Pdf_MoveTo(RADIUS, 0);
            hr = pICAccess->Pdf_LineTo(RADIUS-MARGIN, 0);
            hr = pICAccess->Pdf_Stroke();
        }
        hr = pICAccess->Pdf_Restore();

        time(&timer);
        ltime = *localtime(&timer);

        // draw hour hand
        hr = pICAccess->Pdf_Save();
        hr = pICAccess->Pdf_Rotate((float)(-((ltime.tm_min/60) + ltime.tm_hour - 3.0) * 30));
        hr = pICAccess->Pdf_MoveTo(-RADIUS/10, -RADIUS/20);
        hr = pICAccess->Pdf_LineTo(RADIUS/2, 0);
        hr = pICAccess->Pdf_LineTo(-RADIUS/10, RADIUS/20);
        hr = pICAccess->Pdf_ClosePath();
        hr = pICAccess->Pdf_Fill();
        hr = pICAccess->Pdf_Restore();

        // draw minute hand
        hr = pICAccess->Pdf_Save();
        hr = pICAccess->Pdf_Rotate((float) (-((ltime.tm_sec/60.0) + ltime.tm_min - 15.0) * 6.0));
        hr = pICAccess->Pdf_MoveTo(-RADIUS/10, -RADIUS/20);
        hr = pICAccess->Pdf_LineTo(RADIUS * 0.8, 0);
        hr = pICAccess->Pdf_LineTo(-RADIUS/10, RADIUS/20);
        hr = pICAccess->Pdf_ClosePath();
        hr = pICAccess->Pdf_Fill();
        hr = pICAccess->Pdf_Restore();

        // draw second hand
        hr = pICAccess->Pdf_SetColor(L"fillstroke", L"rgb", 1, 0, 0, 0);
        hr = pICAccess->Pdf_SetLineWidth(2);
        hr = pICAccess->Pdf_Save();
        hr = pICAccess->Pdf_Rotate((float) -((ltime.tm_sec - 15) * 6));
        hr = pICAccess->Pdf_MoveTo(-RADIUS/5, 0);
        hr = pICAccess->Pdf_LineTo(RADIUS, 0);
        hr = pICAccess->Pdf_Stroke();
        hr = pICAccess->Pdf_Restore();

        // draw little circle at center
        hr = pICAccess->Pdf_Circle(0, 0, (float) RADIUS/30);
        hr = pICAccess->Pdf Fill();
}
```

2.3. How to use the function DrawClock

According to the function’s parameters, here’s an example of call to the function DrawClock. It is written as simpler as possible for learning purpose.

Here is the source code:

```
#include "stdafx.h"
#include <atlbase.h>
#include "DirectPDF i.c"
#include "DirectPDF.h"

int main(int argc, char* argv[])
{
ICOMAccess* pICAccess = NULL;
HRESULT hr;
long PDFDocumentIdx = 0;

        // Initialize the COM library (make Windows load the DLLs). Normally you would
        // call this in your InitInstance() or other startup code.  In MFC apps, use
        // AfxOleInit() instead.
        CoInitialize(NULL);

        // Create a COM object, using the COM Access coclass provided by the component.
        // The 4th parameter tells COM what interface we want (ICOMAccess).
        hr = CoCreateInstance(CLSID COMAccess, NULL, CLSCTX INPROC SERVER,
                                 IID ICOMAccess, (void**) &pICAccess);

        if (SUCCEEDED(hr))
        {
                // Open a new PDF Document (Memory)
                hr = pICAccess->Pdf New(&PDFDocumentIdx);
                hr = pICAccess->Pdf SetInfo(L"Title", L"Note 01: PDF clock");
                hr = pICAccess->Pdf SetInfo(L"Creator", L"DirectPDF");
                hr = pICAccess->Pdf SetInfo(L"Author", L"WebSite DirectPDF");
                // Set the saving path of the PDF Document
                hr = pICAccess->Pdf OpenFile(L"C:\\PDFClock.pdf");
                // Create a new page
                hr = pICAccess->Pdf BeginPage((unsigned int) (2 * (RADIUS + MARGIN)),
                                                (unsigned int) (2 * (RADIUS + MARGIN)));

                // Draw a clock
                DrawClock(pICAccess);

                // Close the current page
                hr = pICAccess->Pdf_EndPage();
                // Close and Save the PDF Document according to the saving path
                hr = pICAccess->Pdf Close();

                // Release the ICOMAccess interface.
                pICAccess->Release();
        }

        // Uninit the COM library. In MFC apps, MFC does it for us.
        CoUninitialize();

        return 0;
}
```

3. Support

For the fastest and easiest way to get help, please look for solutions in *DirectPDF Reference manual*.
If you need more help, comments or bug reports, use the web site to send any request:

http://directpdf.free.fr

The Web site contains information, free updates, and all known issues regarding DirectPDF.