

Jukka Salminen

SISÄISTEN VIITTAUSTEN TOTEUT- TAMINEN PDF-DOKUMENTTIIN CADS- OHJELMASSA

Opinnäytetyö

Tietotekniikka / Peliohjelmointi

Toukokuu 2015



KYAMK
University of Applied Sciences

Tekijä/Tekijät	Tutkinto	Aika
Jukka Salminen	Tietotekniikan insinööri	Toukokuu 2015
Opinnäytetyön nimi		
Sisäisten viittausten toteuttaminen pdf-dokumenttiin CADS-ohjelmassa.		26 sivua 6 liitesivua
Toimeksiantaja		
Kyndata Oy		
Ohjaaja		
Yliopettaja Paula Posio		
Tiivistelmä		
<p>Opinnäytetyön tarkoituksena on sisäisten viittausten toteuttaminen pdf-dokumenttiin Kyndata Oy:n kehittämässä CADS-ohjelmassa. Kun CAD-suunnitelma tulostetaan, niin syntyy pdf-dokumentti. CADS-ohjelmaan on tarkoitus lisätä ominaisuus, jossa CAD-suunnitelmassa olevat sisäiset linkit tulostuvat pdf-dokumenttiin. Tällöin syntyneessä pdf-dokumentissa näkyvät linkit, joita klikkaamalla tulee näkyviin linkin viittama paikka samassa dokumentissa.</p> <p>Viittausten toteuttaminen pdf-dokumenttiin tehdään Debenu Quick PDF Libraryn avulla C++ -pohjaisesti. Microsoft Visual Studiolla luodaan MFC (Microsoft Foundation Class Library) ohjelma, johon liitetään Debenu Quick PDF Libraryn työkalut DLL-tiedostoina. Kun sisäisiä viittauksia sisältävä CAD-suunnitelma tulostetaan pdf-dokumenttiin, kutsuu CADS tämän opinnäytetyön tuloksena syntynyttä DLL-ohjelmaa, joka muodostaa viittaukset pdf-dokumenttiin.</p>		
Asiasanat		
MS Visual Studio, C++, CADS Planner, Debenu Quick PDF Library		

Author (authors)	Degree	Time
Jukka Salminen	Bachelor of Information Technology	May 2015
Thesis Title		
Implementation of the Internal References of the pdf File in CADs Program.		26 pages 6 pages of appendices
Commissioned by		
Kymdata Oy		
Supervisor		
Paula Posio, Principal Lecturer		
Abstract		
<p>The purpose of this thesis is the implementation of the internal references to the pdf file in the CADs program developed by Kymdata Ltd. When the CAD document is printed, it generates a pdf file. In this research a feature where the internal links of the CAD plan is printed to the pdf document will be added to the CADs program. As the result pdf document links are displayed and by clicking on them the position the link is referring to is displayed in the same document.</p> <p>Implementation of the references to the pdf document is made by Debenu Quick PDF Library based on C++. Microsoft Visual Studio creates the MFC (Microsoft Foundation Class Library) program in which tools of Debenu Quick PDF Library as DLL file form will be attached. When CAD plan containing internal references is printed to the pdf document, the CADs calls generated DLL program, which forms references to the pdf document – which is the result of this thesis.</p>		
Keywords		
MS Visual Studio, C++, CADs Planner, Debenu Quick PDF Library		

SISÄLLYS

1	JOHDANTO.....	5
2	CAD JA PDF.....	6
2.1	CAD.....	6
2.2	Pdf.....	6
3	CADS PLANNER.....	7
3.1	CADS Plannerin käyttö.....	7
4	DEBENU QUICK PDF LIBRARY.....	9
4.1	Debenun asennus ja käyttöönotto.....	10
4.2	Debenun käyttö.....	15
4.3	Visual Basic -pohjaisesti.....	16
4.4	C++ -pohjaisesti.....	16
5	LINKKIEN TOTEUTTAMINEN DEBENU QUICK PDF LIBRARYLLÄ.....	17
5.1	Linkin toiminta CADS Plannerissa.....	17
5.2	Menetelmän kuvaus, perustelu ja käyttö.....	18
5.3	Linkit toteuttava ohjelma luokittain.....	20
5.4	Ohjelman toteuttamat linkit pdf-dokumentissa.....	24
6	ANALYYSI, JOHTOPÄÄTÖKSET JA JATKOKEHITYSMAHDOLLISUUDET.....	25
	LÄHTEET.....	26

LIITTEET

ESIMERKKIOHJELMIA DEBENULLA VISUAL BASIC -POHJAISESTI
DEBENU QUICK PDF LIBRARYN FUNKTIORYHMÄT

LYHENTEIDEN SELITYKSET

Pdf (portable document format) on suositeltava tiedostomuoto, kun jokin valmis esitys halutaan tulostaa paperille. Se on myös hyvin esityskelpoinen tietokoneen näytöllä.

CAD (computer-aided design) tarkoittaa tietokoneavusteista suunnittelua. CAD:llä tehdään useimmiten kaksi- tai kolmiulotteista piirtämistä ja mallinnusta, mutta se sisältää myös numeerista laskentaa ja simulointia.

C++ on yksi yleisimmin käytetyistä ohjelmointikielistä. Se on oliopohjainen ja sisältää luokat, periytymisen, mallit (templates) ja poikkeukset.

MFC (Microsoft Foundation Class Library) on kirjasto, joka liittää osia Windows API:sta (application programming interfaces) C++ -luokkiin (Microsoft Corporation 2010.).

DLL (Dynamic-link library) on Microsoftin toteutus jaetuille kirjastoille.

1 JOHDANTO

Opinnäytetyön tavoitteena on selvittää ja toteuttaa pdf:n sisäiset viittaukset CADS-tulosteessa. Työkaluna tähän käytetään Debenu PDF Librarya C++ -pohjaisesti.

Työssä on saatava yhdistettyä CADS-tulostuksessa syntyvät erilliset pdf-sivut niin, että CADS-ohjelman tekemät linkit toimivat myös yhdistetyssä pdf-dokumentissa.

Työ rajataan siten, että CADS-ohjelmaan liitettävässä pdf-tulostusosiossa on oltava seuraavat ominaisuudet:

- sisältää tietorakenteen, joka kerää CADS-tiedoston tulostuksessa kaikkien siinä esiintyvien linkkien tiedot
- tulostuksen viimeisenä osana on luotava kyseisen tietorakenteen mukaisesti tulostuksessa syntyvään pdf-dokumenttiin toimivat linkit

Työn vaatimusten toteuttaminen on vaatinut sekä CADS Planner -ohjelmiston että Debenu Quick PDF Libraryn alkeisiin tutustumista.

Työn toimeksiantaja Kyndata Oy on vuonna 1979 perustettu suomalainen CAD-ohjelmistotalo. Yrityksen internet kotisivut löytyvät osoitteesta <http://www.cads.fi>.

2 CAD JA PDF

2.1 CAD

CAD (computer-aided design) tarkoittaa tietokoneavusteista suunnittelua. CAD:llä tehdään useimmiten kaksi- tai kolmiulotteista piirtämistä ja mallinnusta, mutta se sisältää myös numeerista laskentaa ja simulointia.

CAD-ohjelmat asettuvat työkaluvalikoimaltaan suunnilleen Microsoft Paint -ohjelman ja CAE-ohjelman (Computer-aided engineering) väliin.

Microsoft Paintilla voidaan piirtää kaksi- ja kolmiulotteisesti, mutta käytettyjen pisteiden koordinaatit ovat aina (x,y)-tasossa. Ohjelma ei toisin sanoen laske kolmiulotteisen pisteen (x,y,z) projektiota halutun kuvakulman määräämään tasoon, vaan se pitää ohjelman käyttäjän tehdä itse.

CAE-ohjelmassa numeerinen laskenta, simulointi tai analysointi on kehittyneempää CAD:iin verrattuna.

CAD-ohjelmissa tyypillisesti käytettäviä objekteja ovat esimerkiksi kuutio, kartio, pallo, pyramidi ja sylinteri.

2.2 Pdf

Pdf (portable document format) on Adoben kehittämä ohjelmistoriippumaton siirrettävä tiedostomuoto. Sitä käytetään pääasiallisesti sähköiseen julkaisuun, tulostamiseen ja painamiseen. Pdf on suositeltava tiedostomuoto, kun jokin valmis esitys halutaan tulostaa paperille; se on myös hyvin esityskelpoinen tietokoneen näytöllä. Tiedostoja voidaan tallentaa pdf-muotoon useimmista tekstinkäsittely ja kuvankäsittelyohjelmista.

3 CADS PLANNER

CADS Planner on ohjelma, jolla voi mallintaa kaksi- tai kolmiulotteisia piirustuksia ja joka sisältää lukuisia laskennallisia toimintoja.

CADS Planneria käytetään pääasiassa talonrakennuksen suunnittelussa ja sillä on markkinajohtajan asema LVI- ja sähkösuunnittelussa Suomessa.

CADS Planneria on kehitelty yli 20 vuotta vuonna 1979 perustetun Kyndata Oy:n toimesta.

CADS Planner on yhteensopiva IFC (Industry Foundation Classes) kanssa, mikä on kansainvälinen rakennusalan standardi oliopohjaisen tiedon siirtoon tietokonejärjestelmästä toiseen.

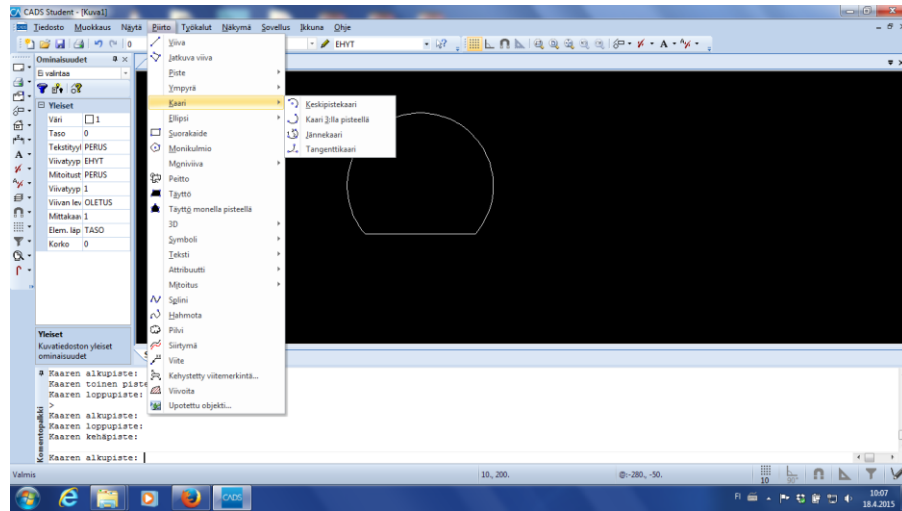
Ohjelma sisältää järjestelmien mitoituksessa ja 3D-mallintamisessa tarvittavia laitevalmistajien tuotekirjastoja.

Ohjelmasta on markkinoilla erityyppisiä, -tasoisia ja -hintaisia versioita. (Kyndata Oy 2015.)

3.1 CADS Plannerin käyttö

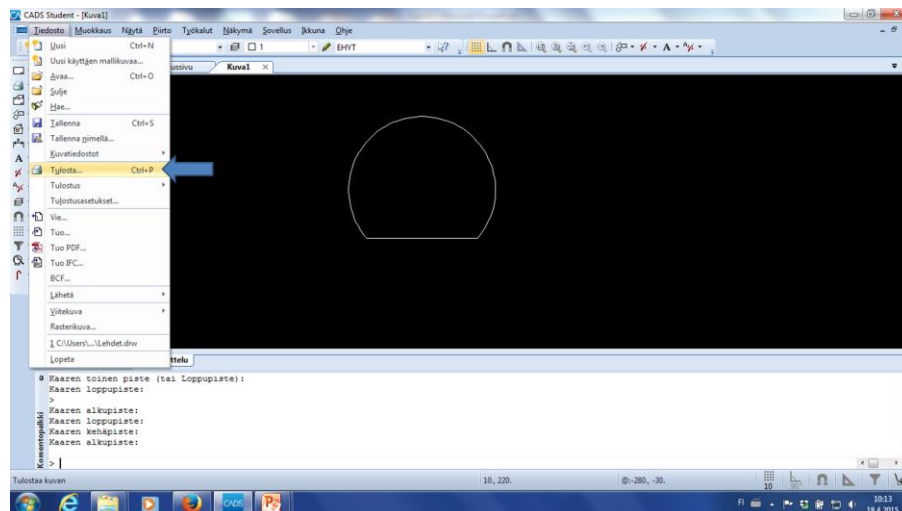
Seuraavassa kuvataan kuvan piirtämistä ja tulostamista CADS Planner -ohjelmalla:

1. Viivan ja kaaren piirtäminen CADS Planner -ohjelmalla tapahtuu niin, että Piirto-valikosta valitaan ensin "Viiva". Sen jälkeen klikataan hiirellä viivan alkupistettä ja sitten viivan loppupistettä. Edelleen valitaan Piirto-valikosta "Kaari" ja tämän alavalikosta "Kaari 3 pisteellä". Sitten klikataan kaaren alkupisteen paikkaa, kaaren välipisteen paikkaa ja lopuksi kaaren loppupisteen paikkaa. Näin piirtyy kaari, jonka kaikki edellä mainitut pisteet ovat tietyn ympyrän kaaren pisteitä. Vain yksi ympyrän kaaren osa voi toteuttaa nämä ehdot (kuva 1).



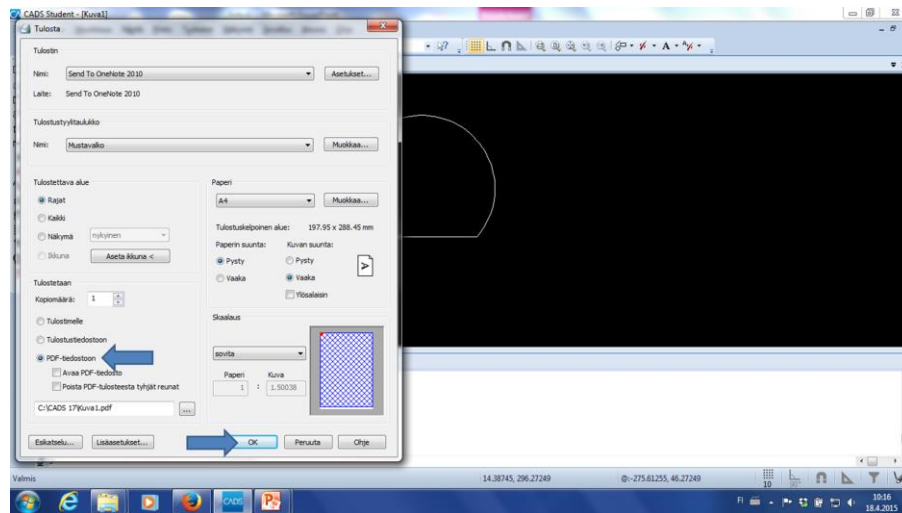
Kuva 1 Viivan ja kaaren piirtäminen

2. Valitaan Tiedosto-valikosta "Tulosta..", jolloin siirrytään tulostuksen asetuksiin (kuva 2).



Kuva 2 Tiedoston tulostus

3. Valitaan tulostus pdf-dokumenttiin (kuva 3).



Kuva 3 Tulostus pdf-dokumenttiin

Kuvatussa esimerkissä sen viimeisen vaiheen jälkeen eli kun on valittu tulostus pdf-dokumenttiin ja klikattu ”OK”, luo CADs-ohjelma pdf-dokumentin, jossa on CADs-ohjelmassa tehty suunnitelma.

4 DEBENU QUICK PDF LIBRARY

Debenu Quick PDF Libraryn avulla voidaan ohjelmallisesti luoda ja muokata pdf-dokumentteja. Sitä voidaan käyttää Windows-, Mac- ja iOS-käyttöjärjestelmissä.

Windows-käyttöjärjestelmässä on käytössä ActiveX-, DLL-, Delphi- tai LIB-tekniikat, Mac-käyttöjärjestelmässä Dylib- ja Delphi- sekä iOS-käyttöjärjestelmässä iOS Static LIB -tekniikat. Kaikissa näissä tekniikat soveltuvat 32-bittiseen ja suurin osa myös 64-bittiseen käyttöjärjestelmään.

Tässä työssä käytetään Windows-käyttöjärjestelmää ja DLL-tekniikkaa.

Tiivistelmä Debenu PDF Libraryn ominaisuuksista:

Debenu PDF Library

- luo ja tulostaa pdf-dokumentteja
- suojaa ja signeeraa pdf-dokumentteja
- luo, täyttää ja muokkaa pdf-lomakkeita
- pilkkoo, yhdistää ja liittää pdf-dokumentteja

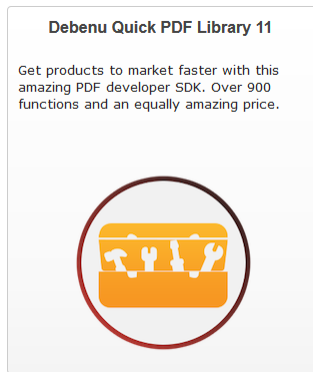
- muuntaa EMF-tiedoston pdf-dokumentiksi (vain Windowsissa)
- liittää tekstiä ja kuvia yhdestä pdf-dokumentista toiseen
- mahdollistaa pdf-tasojen kanssa työskentelyn
- mahdollistaa pdf-dokumentin alkuperäisen näkymän ja asiakirjan ominaisuuksien muokkauksen
- lisää tekstiä, kuvia ja viivakoodeja pdf-dokumenttiin
- sisältää edistyneen tuen JavaScriptille, kirjanmerkeille, vektorigrafialle ja geospaatialiselle pdf-dokumentille
- sisältää "direct access"-toiminnot (tiedostojen lataamiseen levyiltä, ei muistista)
- sisältää huomautukset, kirjanmerkit ja linkit helpolle navigoinnille
- mahdollistaa Windows- ja Mac-sovellusten luonnin (Debenu Pty Ltd 2015.)

4.1 Debenun asennus ja käyttöönotto

Debenu Quick PDF Libraryn asennus ja käyttöönotto vaatii useita vaiheita. Jotta pdf-sivuja luovia ohjelmia voi käyttää, on tehtävä ensin rekisteröinti, Debenun asennus, esimerkikikoodeja sisältävän pakatun kansion purkaminen, skriptien avaaminen Notepadillä, lisenssiavaimen kopioiminen tietystä tekstitiedostosta ja liittäminen ajettavaan ohjelmointikoodiin ja rekisteröitävä Command Promptissa Debenun kirjaston sisältävä dll-tiedosto. Lisäksi käytettäessä 64-bittistä käyttöjärjestelmää, on käytettävä erityistä dll-tiedostoa. Tätä dll-tiedostoa pitää käyttää sekä Command Promptissa rekisteröinnissä että ajettavassa ohjelmointikoodissa.

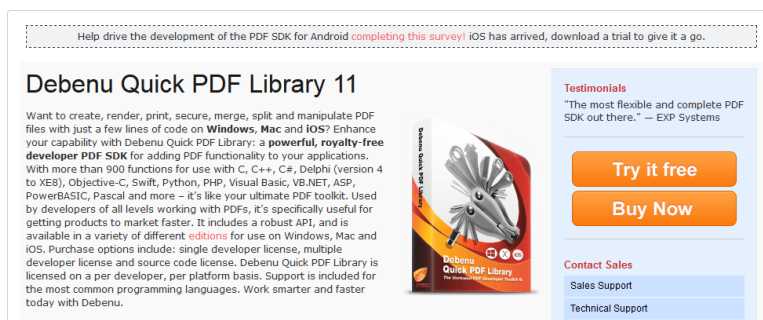
Debenun asennus ja käyttöönotto vaihe vaiheelta:

1. Valitaan Debenun kotisivulta "www.debenu.com" Debenu Quick PDF Libraryn uusin versio (kuva 4).



Kuva 4 Klikataan ”Debenu Quick PDF Library” :n uusinta versiota.

2. Valitaan Debenun ilmainen koekäyttöversio (kuva 5).




Kuva 5 Klikataan ”Try it free”.

3. Rekisteröidytään Debenuun (kuva 6).

Download Free Debenu Quick PDF Library 11 Trial

Debenu Quick PDF Library is a powerful royalty-free toolkit for adding PDF functionality to your applications. It lets you create, render, print, secure, merge, split and manipulate PDF files with only a few lines of code. Support is provided for the most popular programming languages through ActiveX, DLL, LIB, Delphi, iOS Static Lib and Mac Dyllib editions of the library. Download buttons for both the **Windows, Mac** and **iOS** versions will be shown after you fill out the form below. Try the free 30 day trial now.

First Name: *
Last Name: *
Email Address: *

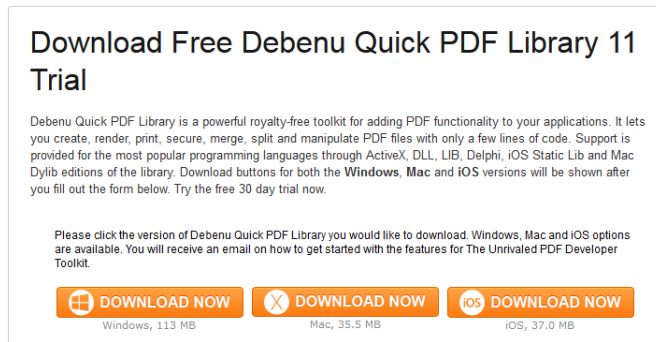


Useful Links

- [Purchase now >](#)
- [Contact support >](#)
- [Visit the forum >](#)

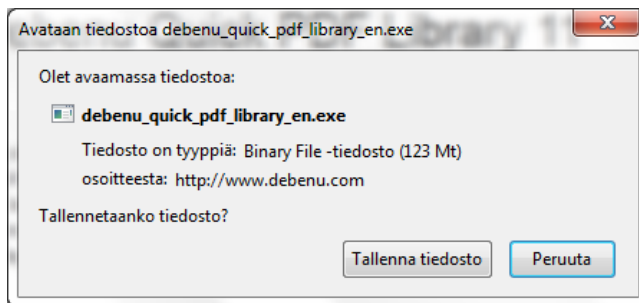
Kuva 6 Täytetään henkilökohtaiset tiedot ja klikataan ”Submit”.

4. Valitaan Debenun asennus Windows-järjestelmään (kuva 7).



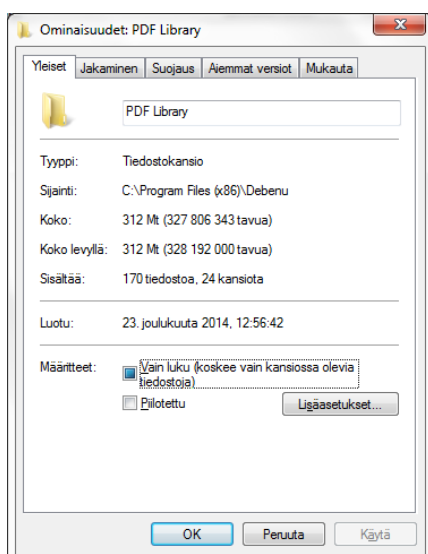
Kuva 7 Klikataan "DOWNLOAD NOW, Windows".

5. Tallennetaan Debenun asentava tiedosto (kuva 8).



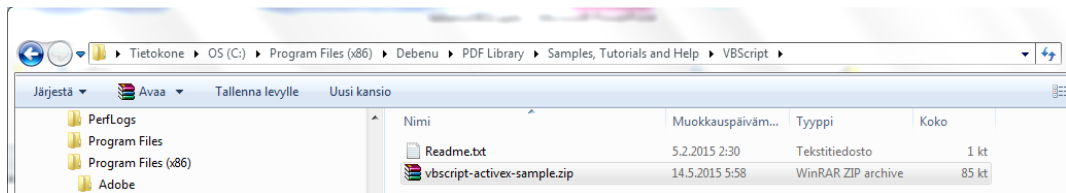
Kuva 8 Klikataan "Tallenna tiedosto"

6. Käynnistetään kyseinen tallennettu tiedosto "debenu_quick_pdf_library_en.exe" ja suoritetaan asennus oletusasetuksin "C:\Program Files (x86)"-kansioon (kuva 9).



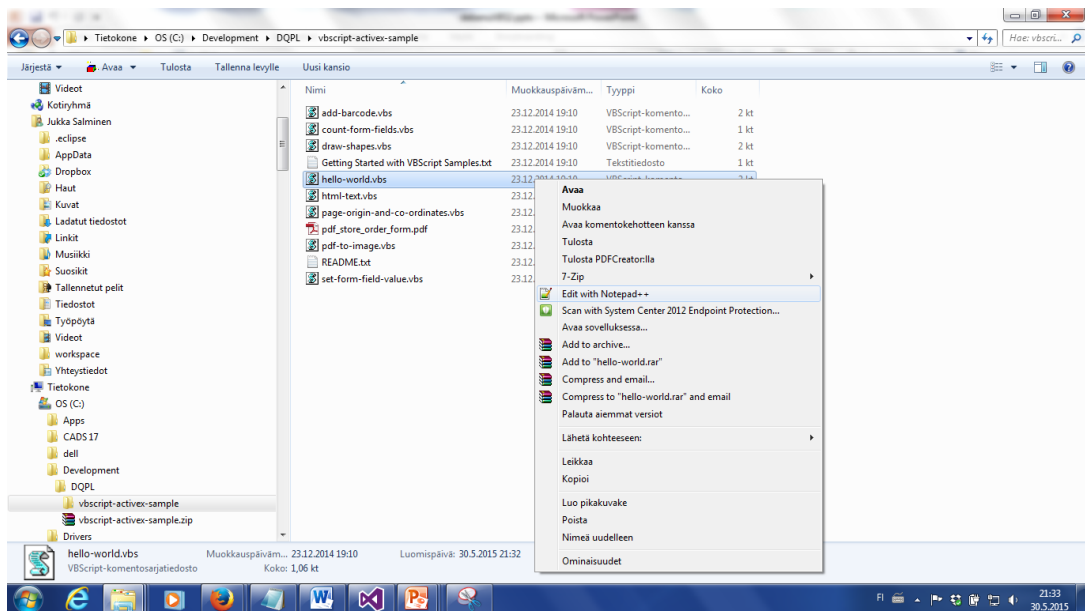
Kuva 9 Asennuksen jälkeen "PDF Library"-kansio sijaitsee oletusasetuksin polussa "C:\Program Files (x86)\Debenun".

7. Polusta "C:\Program Files (x86)\Debenu\PDF Library\Samples, Tutorials and Help\VBScript" löytyy pakattu tiedosto "vbscript-activex-sample.zip"; puretaan se vapaavalintaiseen kansioon (kuva 10).



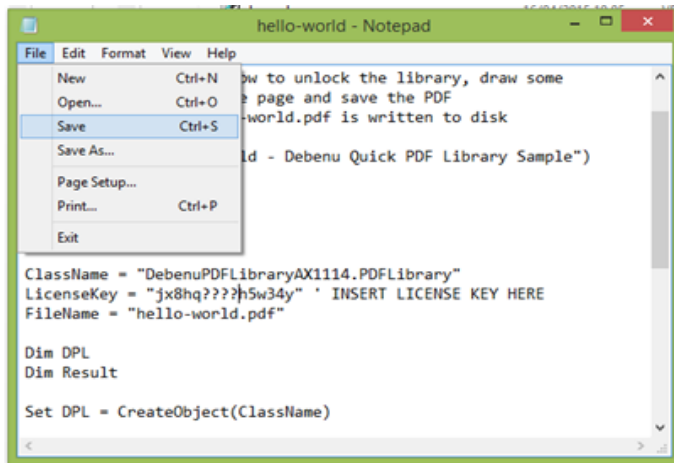
Kuva 10 Puretaan "vbscript-activex-sample.zip"-tiedosto.

8. Zip-tiedoston purkamisen jälkeen kansioista "vbscript-activex-sample" löytyy tiedosto "hello-world.vbs"; muokataan sitä Notepad++:lla tai Notepad:llä (kuva 11).



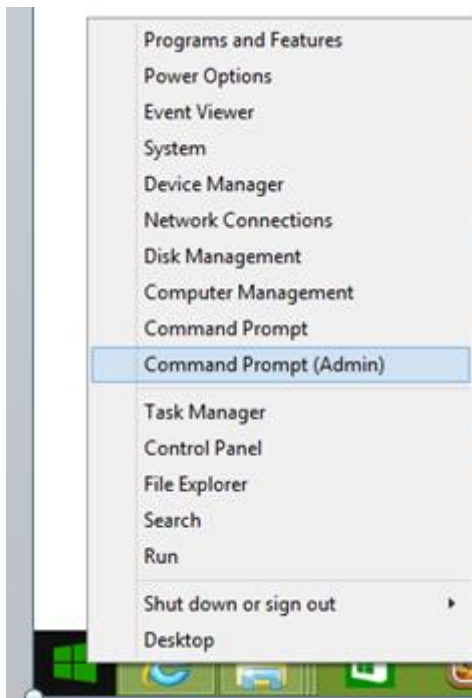
Kuva 11 Klikataan "hello-world.vbs"-tiedostoa hiirioikealla ja klikataan "Edit with Notepad++".

9. Lisätään lisenssiavain ja käytettäessä 64-bittistä käyttöjärjestelmää muutetaan "ClassName"-muuttujan arvoksi "DebenuPDFLibrary64AX1114.PDFLibrary" ja talletetaan tiedosto (kuva 12).



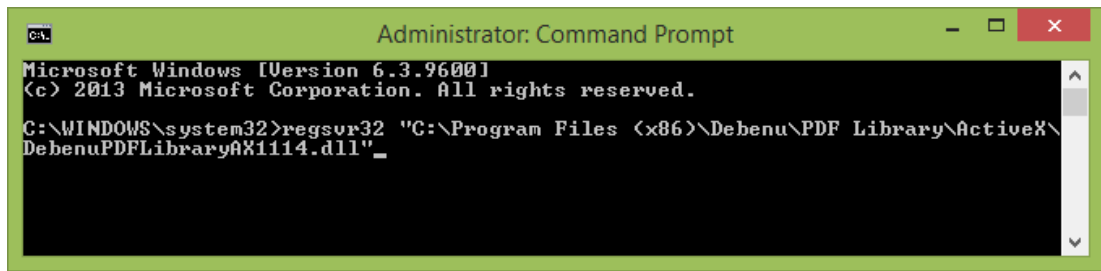
Kuva 12 LicenseKey-muuttujan arvoksi on muutettu "TRIAL_LICENSE_KEY.txt"-tiedostossa oleva lisenssiavain.

10. Avataan Command Prompt järjestelmän valvojana (kuva 13).



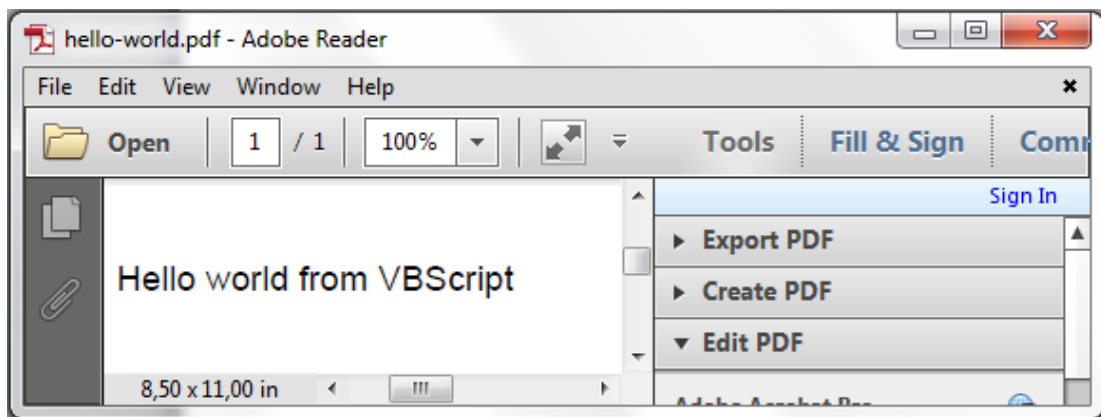
Kuva 13 Windows 8:ssa on klikattu hiirioikealla käynnistä-valikon symbolia ja valittu "Command Prompt (Admin)".

11. Annetaan Command Promptiin käsky `regsvr32 "C:\Program Files (x86)\Debenu\PDF Library\ActiveX\DebenuPDFLibraryAX1114.dll"`. Käytettäessä 64-bittistä käyttöjärjestelmää "DebenuPDFLibraryAX1114.dll" sijaan käytetään "DebenuPDFLibrary64AX1114.dll":ä (kuva 14).



Kuva 14 Command Promptissa rekisteröidään tarvittava dll-tiedosto.

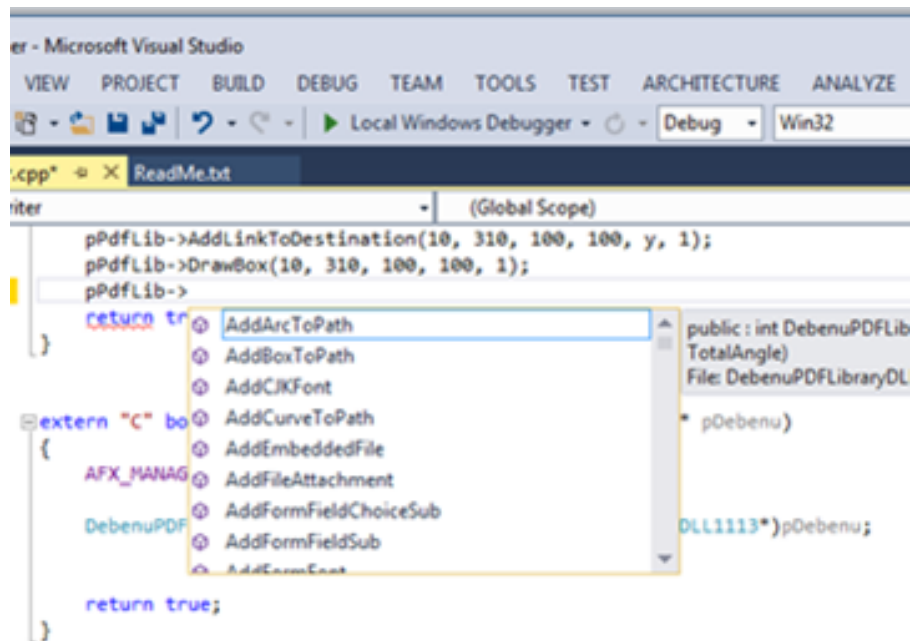
12. Käynnistetään kansiossa "vbscript-activex-sample" sijaitseva "hello-world.vbs" tuplaklikkaamalla, jolloin samaan kansioon syntyy pdf-dokumentti "hello-world.pdf" (kuva 15).



Kuva 15 "hello-world.vbs"-skriptin aikaansaama pdf-tiedosto.

4.2 Debenun käyttö

Kun Visual Studiossa käynnistetään projekti, johon on asennettu Debenu, projektissa on käytettävissä kaikki Debenun luokat ja oliot sekä muut ominaisuudet. Kuvassa on esimerkki Debenu Quick PDF Libraryn käskyvalikoimasta (kuva 16).



Kuva 16 Debenulla on lukuisia funktioita käsitellä pdf-dokumenttia.

4.3 Visual Basic -pohjaisesti

Vbscript-activex käyttöönoton vaiheet:

- Debenu PDF Libraryn lataaminen ja asentaminen Debenun kotisivuilta
- vbscript-activex-sample käyttöönotto ja testaus
 - ✓ lisenssiavaimen kopioiminen lähdekoodiin
 - ✓ rekisteröityminen Command Promptin kautta järjestelmän valvojana käyttäen "regsvr32.exe"-sovellusta
 - ✓ Debenu Quick PDF Libraryn binäärien tuominen projektiin kopioidulla ja liittämällä " DebenuPDFLibraryAX1113.dll" tiedosto samaan kansioon kuin ohjelman suorittavat ".vbs"-tiedostot

4.4 C++ -pohjaisesti

Allaolevassa esimerkissä ohjelma piirtää yksinkertaisen kolmiulotteisen funktion kuvaajan. Debenu Quick PDF Libraryn pohjakoodiin on lisättävä lisenssiavain, tiedostonimi sekä seuraava ohjelmointikoodi (kuva 17):

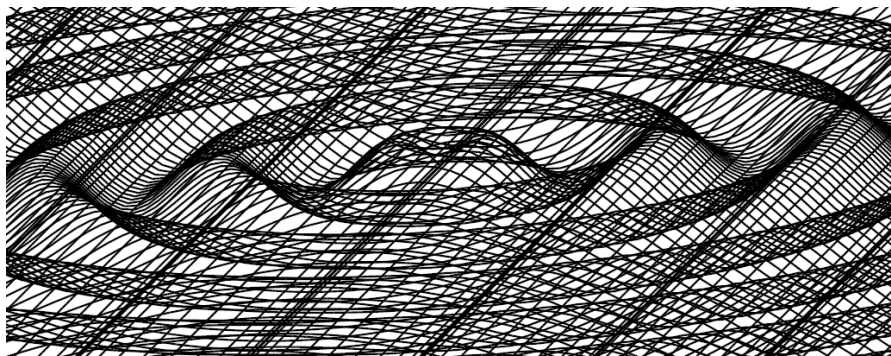

```

1 float XscreenOrigin = 300,screenOrigin = 200, scale = 15, XpreviousScreen, YpreviousScreen, dx = 1.0f, dy = 1.0f;
2 float x0 = -3.0f, y0=-3.0f, x1 = 3.0f, y1=3.0f, y0screen, y1screen, x0screen, x1screen;
3 for (float x = x0; x <= x1 + dx / 2; x = x + dx)
4 {
5     for (float y = y0; y <= y1 + dy / 2; y = y + dy)
6     {
7         float z = sin(pow(pow(x,2)+pow(y,2),0.5f));
8         float y1screen = YscreenOrigin - (z - x / 4.0f)*scale;
9         float x1screen = XscreenOrigin + (y - x / 4.0f)*scale;
10        if (y!=y0)
11        {
12            float y0screen = YpreviousScreen, x0screen = XpreviousScreen;
13            QP.DrawLine(x0screen, y0screen, x1screen, y1screen);
14        }
15        XpreviousScreen = x1screen;
16        YpreviousScreen = y1screen;
17    }
18 }

```

Kuva 17 Eräs C++ -pohjaisen Debenu ohjelmointikoodi

Ohjelman aikaansaama pdf-dokumentti (kuva 18):



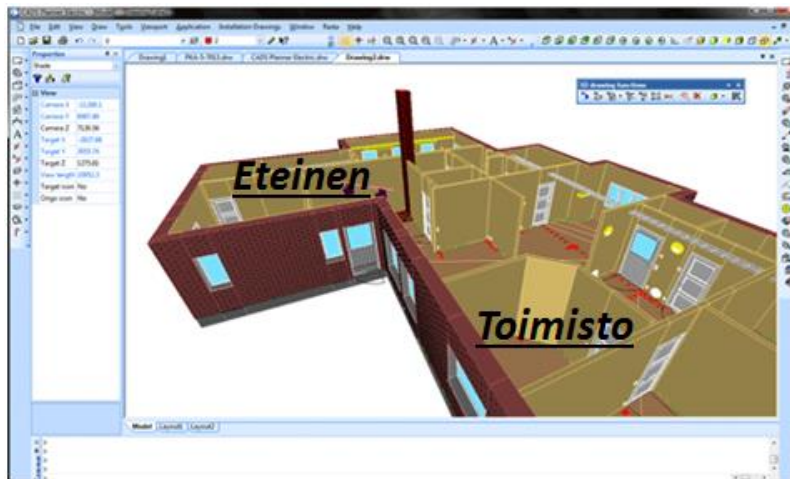
Kuva 18 Pdf-dokumentti kuvaa kolmiulotteista funktiota

5 LINKKIEN TOTEUTTAMINEN DEBENU QUICK PDF LIBRARYLLÄ

CADS Planner -ohjelmaan ollaan lisäämässä ominaisuus, jossa CADs Planner -tiedostossa olevat sisäiset linkit näkyvät myös tulostuksessa syntyvässä pdf-dokumentissa. Sisäisellä linkillä tarkoitetaan CADs Planner -tiedoston yhteen paikkaan rajattua suorakulmion muotoista aluetta, jota klikkaamalla haluttu kohta samasta tiedostosta siirtyy näytön vasempaan yläreunaan.

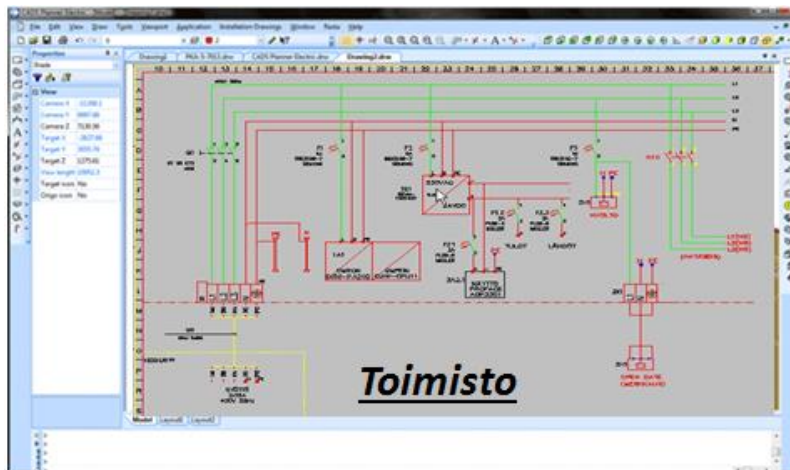
5.1 Linkin toiminta CADs Plannerissa

1. Aluksi CADs Planner näyttää alla olevan osan koko tiedostosta (kuva 19).



Kuva 19 CADS Plannerin suunnitelma rakennuksesta.

2. Kun "Toimisto"-linkkiä klikataan, tulee näyttöön toimiston sähköpiirros saman CADS-tiedoston eri osasta (kuva 20).



Kuva 20 CADS Plannerin suunnitelma toimistohuoneen sähköpiirroksesta

Edellisessä esimerkissä alemman kuvan "Toimisto"-linkkiä klikkaamalla pääsee takaisin ylemmän kuvan näyttöön. Linkin ominaisuuteen kuuluu myös mahdollisuus suurentaa tai pienentää näyttöä; linkin kohteena olevasta tiedoston paikasta pitää tietää pdf-sivun numero (n), tämän sivun piste (x,y) joka laitetaan näytettävän kuva-alueen keskelle ja kuvan suurennos (Z).

5.2 Menetelmän kuvaus, perustelu ja käyttö

CADS-ohjelman tulostuksessa syntyy pdf-dokumentti. Microsoft Visual Studiolla luodaan C++ MFC (Microsoft Foundation Class Library) ohjelma, jonka funktiot suorittavat linkkien toteuttamisen em. pdf-dokumenttiin. Tämä MFC-ohjelma liitetään CADS-ohjelmaan DLL-tiedostona.

Linkkien toteuttavaa MFC (Microsoft Foundation Class Library) ohjelmaa testataan siten, että siihen liitetään Debenu Quick PDF Libraryn työkalut DLL-tiedostoina. Kun tämä C++ -ohjelma käynnistetään, käynnistyy automaattisesti CADs-ohjelma. Tällöin CADs-ohjelmalla voi luoda suunnitelman, joka voi sisältää sisäisiä linkkejä. Kun CADs-ohjelmasta otetaan tulostus, niin ohjelma kutsuu C++ -ohjelman metodeita, jotka suorittavat linkkien luomisen pdf-dokumenttiin.

CADs Planner -ohjelman suunnitelman tulostus pdf-dokumentiksi on jaettu kolmeen osaan:

1. ensimmäisen sivun tulostus, jolloin luodaan lista linkeille
2. välisivujen tulostus, jolloin linkkejä lisätään listaan
3. viimeisen sivun tulostus, jolloin linkit toteutetaan linkkilistan mukaisesti

Microsoft Visual Studio C++ -ohjelmassa tarvitaan seuraavia työkaluja:

1. Luodaan "Link"-luokka, jonka parametreina ovat linkin paikan id, linkin kohdepaikan id, sivunumero (n) ja paikka (x, y).
2. Ensimmäisen sivun tulostuksessa CADs-ohjelma kutsuu Microsoft Visual Studio C++ ohjelman "pdfwriterStartPDF"-funktiota, jossa luodaan "Link"-tyyppinen lista "linkList".
3. Kun minkä tahansa sivun tulostuksessa esiintyy linkki, niin luodaan "Link"-tyyppinen olio, johon linkin tiedot talletetaan.
4. Lisätään tämä olio em. "linkList"-listaan.
5. Viimeisen sivun tulostuksen jälkeen erilliset pdf-dokumentit yhdistetään yhdeksi pdf-dokumentiksi.
6. Luodaan ohjelmallisesti linkit pdf-dokumenttiin "linkList" -listan mukaisesti.

Linkkien alku- ja loppupaikan id:n luomisessa käytetään "GUID"-tunnusta (Globally unique identifier).

GUID-tunnuksen ideana on se, että on erittäin epätodennäköistä olla olemassa kaksi samaa satunnaisesti valittua GUID-tunnusta; näin kutakin GUID-tunnusta voidaan pitää ainutkertaisena.

"GUID (Globally unique identifier) on tietotekniikassa hyödynnetty ainutkertaiseksi tarkoitettu tunnusnumero. GUID on 128-bittinen satunnaisluku, joka esi-

tetään yleensä 32:na heksalukuna yhdysmerkein ryhmiteltynä, esim. {21EC2020-3AEA-4069-A2DD-08002B30309D}.” (Microsoft Corporation 2015.)

5.3 Linkit toteuttava ohjelma luokittain

PDFWriter.h (kuva 21):

```

1 // PDFWriter.h : main header file for the PDFWriter DLL
2 //
3 #pragma once
4 #ifndef __AFXWIN_H__
5 #error "include 'stdafx.h' before including this file for PCH"
6 #endif
7 #include "resource.h"
8 // main symbols
9 // CPDFWriterApp
10 // See PDFWriter.cpp for the implementation of this class
11 //
12 #include "LinkMaker.h"
13 class CPDFWriterApp : public CWinApp
14 {
15 public:
16     CPDFWriterApp();
17     // Overrides
18 public:
19     virtual BOOL InitInstance();
20     DECLARE_MESSAGE_MAP()
21 };

```

Kuva 21 PDFWriter.h-luokan ohjelmointikoodi

PDFWriter.cpp (kuva 22):

```

1 #include "stdafx.h"
2 #include "PDFWriter.h"
3 #include <DebenuPDFLibraryDLL1114.cpp>
4 #include <vector>
5 #include "PdfLink.h"
6 using namespace std;
7 #define PI 3.14159265358979323846
8 #ifdef _DEBUG
9 #define new DEBUG_NEW
10 #endif
11 LinkMaker* linkMaker = new LinkMaker();
12 extern "C" bool PASCAL EXPORT pdfwriterStartNewPage(void* pDebenu)
13 {
14     AFX_MANAGE_STATE(AfxGetStaticModuleState());
15     linkMaker->increasePageNumber();
16     DebenuPDFLibraryDLL1114* pPdfLib = (DebenuPDFLibraryDLL1114*)pDebenu;
17     return true;
18 }
19 extern "C" bool PASCAL EXPORT pdfwriterLinkRectangle
20 (LPCTSTR lpszGuid, LPCTSTR lpszDestinationGuid, POINT* pPoints, int nCount, void* pDebenu)
21 {
22     AFX_MANAGE_STATE(AfxGetStaticModuleState());
23     DebenuPDFLibraryDLL1114* pPdfLib = (DebenuPDFLibraryDLL1114*)pDebenu;
24     int recLeft = pPoints[0].x;
25     int recTop = pPoints[0].y;
26     int recRight = pPoints[2].x;
27     int recBottom = pPoints[2].y;
28     int recWidth = recRight - recLeft;
29     int recHeight = recTop - recBottom;
30     linkMaker->addLinkToLinkList(PdfLink
31     (lpszGuid, lpszDestinationGuid, recLeft, recBottom, recWidth, recHeight, linkMaker->get_pageNumber()));
32     return true;
33 }

```

Kuva 22 PDFWriter.cpp-luokan ohjelmointikoodi

LinkMaker.h (kuva 23):

```

1  #pragma once
2  #include "stdafx.h"
3  #include "PdfLink.h"
4  #include "DebenuPDFLibraryDLL1114.h"
5  #include <vector>
6  using namespace std;
7  class LinkMaker
8  {
9      int pageNumber;
10     vector<PdfLink> linkList;
11     vector<PdfLink> *linkListPointer;
12 public:
13     LinkMaker();
14     ~LinkMaker();
15     int get_pageNumber();
16     vector<PdfLink> get_linkList();
17     void increasePageNumber();
18     void addLinkToLinkList(PdfLink link);
19     void makeLinks(void* pDebenu, DebenuPDFLibraryDLL1114* pPdfLib);
20 };

```

Kuva 23 LinkMaker.h-luokan ohjelmointikoodi

LinkMaker.cpp 1/3 (kuva 24):

```

1  #pragma once
2  #include "stdafx.h"
3  #include "LinkMaker.h"
4  #include "DebenuPDFLibraryDLL1114.h"
5  using namespace std;
6  LinkMaker::LinkMaker()
7  {
8      vector<PdfLink> *linkList = new vector<PdfLink>();
9      this->linkListPointer = linkList;
10     this->pageNumber = 1;
11 }
12 LinkMaker::~LinkMaker()
13 {
14 }
15 void LinkMaker::increasePageNumber()
16 {
17     this->pageNumber++;
18 }
19 void LinkMaker::addLinkToLinkList(PdfLink link)
20 {
21     linkList.push_back(link);
22 }
23 int LinkMaker::get_pageNumber()
24 {
25     return this->pageNumber;
26 }
27 vector<PdfLink> LinkMaker::get_linkList()
28 {
29     return this->linkList;
30 }

```

Kuva 24 LinkMaker.cpp-luokan ohjelmointikoodia

LinkMaker.cpp 2/3 (kuva 25):

```

31 void LinkMaker::makeLinks(void* pDebenu, DebenuPDFLibraryDLL1114* pPdfLib)
32 {
33     AFX_MANAGE_STATE(AfxGetStaticModuleState());
34     std::vector<int> destIdList;
35     for (int i = 0; i < linkList.size(); i++)
36     {
37         LPCTSTR lpszGuid = linkList[i].get_lpszGuid();
38         LPCTSTR lpszDestinationGuid = linkList[i].get_lpszDestinationGuid();
39         int recLeft = linkList[i].get_recLeft();
40         int recBottom = linkList[i].get_recBottom();
41         int recWidth = linkList[i].get_recWidth();
42         int recHeight = linkList[i].get_recHeight();
43         int recRight = recLeft + recWidth;
44         int recTop = recBottom + recHeight;
45         int pageNumber = linkList[i].get_pageNumber();
46         pPdfLib->SelectPage(pageNumber);
47         pPdfLib->SetFillColor(1, 0, 0);
48         int zoomPercentage = 100;
49         int destId = pPdfLib->
50         NewDestination(pageNumber, zoomPercentage, 1, recLeft, recBottom, recRight, recTop);
51         destIdList.push_back(destId);
52     }

```

Kuva 25 LinkMaker.cpp-luokan ohjelmointikoodia

LinkMaker.cpp 3/3 (kuva 26):

```

53     for (int i = 0; i < linkList.size(); i++)
54     {
55         for (int j = 0; j < linkList.size(); j++)
56         {
57             if (i != j && linkList[i].get_lpszGuid() == linkList[j].get_lpszGuid())
58             {
59                 pPdfLib->SelectPage(linkList[i].get_pageNumber());
60                 pPdfLib->SetFillColor(1, 0, 0);
61                 pPdfLib->SetTransparency(50);
62                 pPdfLib->DrawBox(linkList[i].get_recLeft(), linkList[i].get_recBottom(),
63                 linkList[i].get_recWidth(), linkList[i].get_recHeight(), 1);
64                 pPdfLib->AddLinkToDestination(linkList[i].get_recLeft(), linkList[i].get_recBottom(),
65                 linkList[i].get_recWidth(), linkList[i].get_recHeight(), destIdList[j], 1);
66             }
67         }
68     }
69 }

```

Kuva 26 LinkMaker.cpp-luokan ohjelmointikoodia

PDFLink.h (kuva 27):

```

1  #pragma once
2  #include <string>
3  using namespace std;
4  class PdfLink
5  {
6      LPCTSTR lpszGuid;
7      LPCTSTR lpszDestinationGuid;
8      int recLeft;
9      int recBottom;
10     int recWidth;
11     int recHeight;
12     int pageNumber;
13 public:
14     PdfLink(LPCTSTR lpszGuid, LPCTSTR lpszDestinationGuid, int recLeft,
15            int recBottom, int recWidth, int recHeight, int pageNumber);
16     ~PdfLink();
17     LPCTSTR get_lpszGuid();
18     LPCTSTR get_lpszDestinationGuid();
19     int get_recLeft();
20     int get_recBottom();
21     int get_recWidth();
22     int get_recHeight();
23     int get_pageNumber();
24 };

```

Kuva 27 PDFLink.h-luokan ohjelmointikoodi

PDFLink.cpp (kuva 28):

```

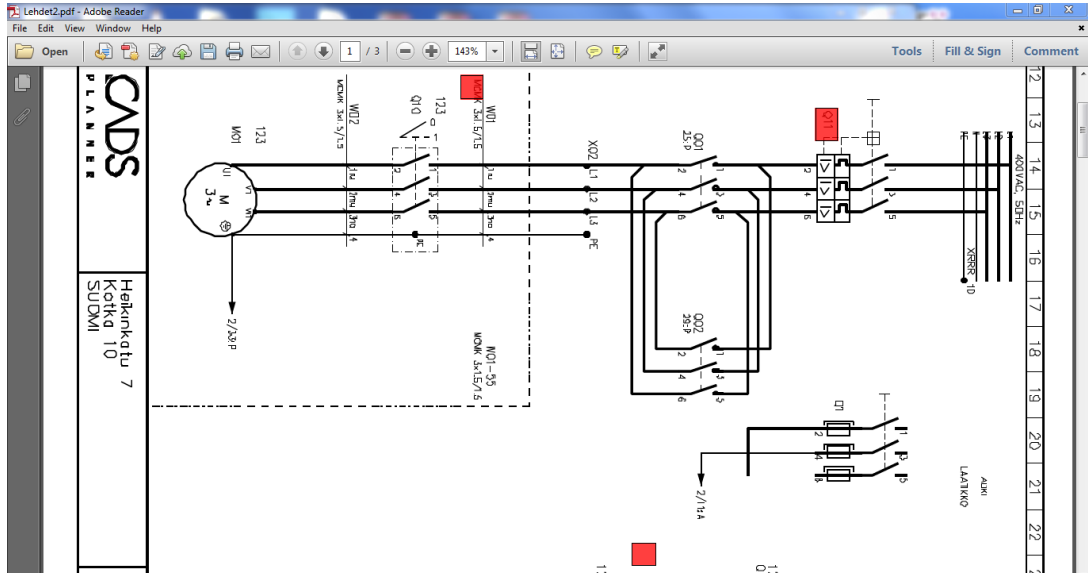
1  #include "stdafx.h"
2  #include "PdfLink.h"
3
4  PdfLink::PdfLink(LPCTSTR lpszGuid, LPCTSTR lpszDestinationGuid, int recLeft,
5                  int recBottom, int recWidth, int recHeight, int pageNumber)
6  {
7      this->lpszGuid = lpszGuid;
8      this->lpszDestinationGuid = lpszDestinationGuid;
9      this->recLeft = recLeft;
10     this->recBottom = recBottom;
11     this->recWidth = recWidth;
12     this->recHeight = recHeight;
13     this->pageNumber = pageNumber;
14 }
15 PdfLink::~PdfLink()
16 {
17 }
18 LPCTSTR PdfLink::get_lpszGuid()
19 {
20     return this->lpszGuid;
21 }

```

Kuva 28 PDFLink.cpp-luokan ohjelmointikoodi

5.4 Ohjelman toteuttamat linkit pdf-dokumentissa

Kun CADs Planner -ohjelmassa tehdään tulostus, linkit toteuttava ohjelma näyttää linkkien paikat läpinäkyvinä, punaisiksi väritettyinä suorakulmioina pdf-dokumentissa (kuva 29).



Kuva 29 Pdf-dokumentissa suorakulmion muotoisia linkkejä klikkaamalla siirtyä näyttö niiden osoittamaan paikkaan samassa dokumentissa.

6 ANALYYSI, JOHTOPÄÄTÖKSET JA JATKOKEHITYSMAHDOLLISUUDET

Opinnäytetyön alussa ei ollut tiedossa, että olisiko Debenu sopiva työkalu CADS Planner -ohjelman käytettäväksi. Työn lopputuloksen perusteella näyttää, että Debenulla pystyy ongelmitta tekemään tavoitteena olleen linkkien toteuttamiseen pdf-tiedostoon.

Jatkokehitysmahdollisuutena on kehittää CADS Planner –ohjelmaa niin, että se lisää CAD-suunnitelmassa olevia komponentteja varten ominaisuustaulukoita. Esimerkiksi jonkin sähkökomponentin jännite ja teho ym. ominaisuudet tulostuisivat pdf-sivulle, jotka tulisivat näkyviin kyseistä komponenttia klikattaessa.

LÄHTEET

Debenu PDF Technology. 2015. Debenu Pty Ltd. Saatavissa:
www.debenu.com [viitattu 3.6.2015]

Debenu Quick PDF Library, function groups. 2015. Debenu Pty Ltd. Saatavissa:
http://www.debenu.com/docs/pdf_library_reference/FunctionGroups.php
[viitattu 30.5.2015]

Tietoa yrityksestä Kyndata Oy. 2015. Kyndata Oy. Saatavissa:
<http://www.cads.fi/fi/Yhteys/Tietoa%20yrityksest%C3%A4/> [viitattu 30.5.2015]

Exploring New C++ and MFC Features in Visual Studio 2010. 2010. Microsoft Corporation. Saatavissa: <https://msdn.microsoft.com/en-us/magazine/ee336130.aspx> [viitattu 3.6.2015]

Guid structure. 2015. Microsoft Corporation. Saatavissa:
<https://msdn.microsoft.com/en-us/library/aa373931%28VS.85%29.aspx> [viitattu 3.6.2015]

ESIMERKKIOHJELMIA DEBENULLA VISUAL BASIC -POHJAISESTI

1. Luodaan pdf-dokumentti ja kirjoitetaan siihen "Hello World" -teksti.

Ohjelmointikoodi kokonaisuudessaan (kuva 30):

```

1  ' Debenu Quick PDF Library Sample
2
3  ' * Remember to set your license key below
4  ' * This sample shows how to unlock the library, draw some
5  '   simple text onto the page and save the PDF
6  ' * A file called hello-world.pdf is written to disk
7  WScript.Echo("Hello World - Debenu Quick PDF Library Sample")
8  Dim ClassName
9  Dim LicenseKey
10 Dim FileName
11 ClassName = "DebenuPDFLibraryAX1113.PDFLibrary"
12 LicenseKey = "... ' INSERT LICENSE KEY HERE"
13 FileName = "hello-world.pdf"
14 Dim DPL
15 Dim Result
16 Set DPL = CreateObject(ClassName)
17 WScript.Echo("Library version: " + DPL.LibraryVersion)
18 Result = DPL.UnlockKey(LicenseKey)
19 If Result = 1 Then
20     WScript.Echo("Valid license key: " + DPL.LicenseInfo)
21     Call DPL.DrawText(100, 500, "Hello world from VBScript")
22     If DPL.SaveToFile(FileName) = 1 Then
23         WScript.Echo("File " + FileName + " written successfully")
24     Else
25         WScript.Echo("Error, file could not be written")
26     End If
27 Else
28     WScript.Echo("- Invalid license key -")
29     WScript.Echo("Please set your license key by editing this file")
30 End If
31 Set DPL = Nothing

```

Kuva 30 Eräs Visual Basic -pohjainen Debenun ohjelmointikoodi

Yllä olevan Debenu Quick PDF Libraryn pohjakoodi toistuu erilaisia pdf-sivuja tehdessä. "Hello-World" -skriptille ominaisia koodirivejä niistä ovat vain seuraavat (kuva 31):

```

1  LicenseKey = "j95sg8g4...59bb9ga6kb7y" ' INSERT LICENSE KEY HERE
2  FileName = "hello-world.pdf"
3  Call DPL.DrawText(100, 500, "Hello world from VBScript")

```

Kuva 31 Edellisen ohjelmointikoodin Debenun pohjakoodista poikkeavat rivit

Ohjelman aikaansaama pdf-dokumentti (kuva 32):

Hello world from VBScript

Kuva 32 Pdf-dokumentin kuva

2. Kirjoitetaan viivakoodi.

Debenu Quick PDF Libraryn pohjakoodiin on lisättävä lisenssiavain ja tiedostonimi - vertaa hello-world.vbs skriptin ohjelmakoodiin – sekä seuraava ohjelmointikoodi (kuva 33):

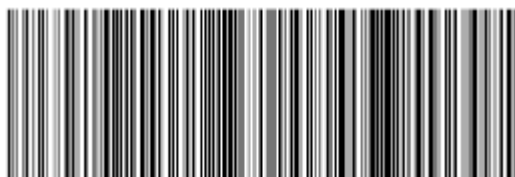
```

1 If DPL.DrawBarcode(300, 600, 120, 40, "Debenu Quick PDF Library Rocks", 3, 0) = 1 Then
2     WScript.Echo("The barcode was successfully added to the selected page.")
3 Else
4     WScript.Echo("The barcode could not be added to the selected page.")
5 End If

```

Kuva 33 Erään ohjelmointikoodin Debenun pohjakoodista poikkeavat rivit

Ohjelman aikaansaama pdf-dokumentti (kuva 34):



Kuva 34 Pdf-dokumentin kuva

3. Lasketaan pdf-sivun sarkaimien määrä.

Debenu Quick PDF Libraryn pohjakoodiin on lisättävä lisenssiavain, tiedostonimi sekä seuraava ohjelmointikoodi (kuva 35):

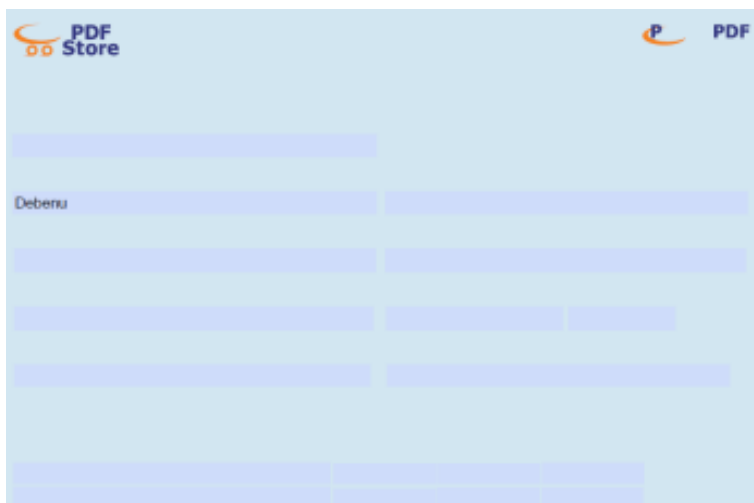
```

1 Dim FormFieldCount
2 Call DPL.LoadFromFile(FileName, "")
3 FormFieldCount = DPL.FormFieldCount()
4 WScript.Echo(FileName & " contains " & FormFieldCount & " form fields")

```

Kuva 35 Erään ohjelmointikoodin Debenun pohjakoodista poikkeavat rivit.

Ohjelman laskee alla olevan pdf-dokumentin tekstikenttien määrän ja ilmoittaa sen ohjelmaa ajettaessa (kuva 36):



Kuva 36 Pdf-dokumentin kuva

4. Piirretään erilaisia muotoja.

Viivoja, suorakulmioita ja ympyröitä piirretty ja väritetty for-käskyllä halutun kokoisia muotoja haluttuun määrään rivejä ja sarakkeita eri RGB-värimallilla. Debenun Quick PDF Libraryn pohjakoodiin on lisättävä lisenssiavain, tiedostonimi sekä seuraava ohjelmointikoodi (kuva 37):

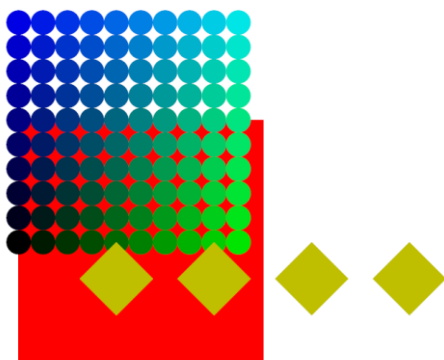
```

1 ' 100% red rectangle
2 Call DPL.SetFillColor(1, 0, 0)
3 Call DPL.DrawBox(100, 300, 200, 200, 1)
4
5 ' Circles become more green towards right and more blue towards down (RGB -color system)
6 Dim X
7 DIM Y
8 For X=0 To 9
9   For Y=0 To 9
10    Call DPL.SetFillColor(0, X/10, Y/10) '(red, green, blue)
11    Call DPL.DrawCircle(100+X*20, 200+Y*20, 10, 1)
12   Next
13 Next
14
15 ' Yellow triangles
16 Call DPL.SetFillColor(0.75, 0.75, 0)
17 For X = 1 To 4
18   Call DPL.StartPath(100 + X * 80, 200)
19   Call DPL.AddLineToPath(70 + X * 80, 170)
20   Call DPL.AddLineToPath(100 + X * 80, 140)
21   Call DPL.AddLineToPath(130 + X * 80, 170)
22   Call DPL.ClosePath
23   Call DPL.DrawPath(1)
24 Next

```

Kuva 37 Erään ohjelmointikoodin Debenun pohjakoodista poikkeavat rivit.

Ohjelman aikaansaama pdf-dokumentti (kuva 38):



Kuva 38 Pdf-dokumentin kuva

5. Kirjoitetaan erilaisia tekstityylejä.

Debenun Quick PDF Libraryn pohjakoodiin on lisättävä lisenssiavain, tiedostonimi sekä seuraava ohjelmointikoodi (kuva 39):

```

1 Call DPL.DrawHTMLText(100, 500, 200, _
2 "Here is some <b>bold</b>, <i>italic</i> and <u>underlined</u> text")
3 Call DPL.DrawHTMLText(100, 400, 200, _
4 "<u><i><b>underlined italic bold</b></i></u>, <i><b>italic bold</b></i> and <b>bold</b> text")

```

Kuva 39 Erään ohjelmointikoodin Debenun pohjakoodista poikkeavat rivit.

Ohjelman aikaansaama pdf-dokumentti (kuva 40):

Here is some **bold**, *italic* and underlined text

Kuva 40 Pdf-dokumentin kuva

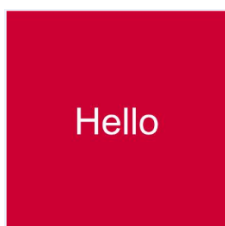
6. Muutetaan pdf-dokumentti jpg-dokumentiksi.

Debenu Quick PDF Libraryn pohjakoodiin on lisättävä lisenssiavain, tiedostonimi sekä seuraava ohjelmointikoodi (kuva 41):

```
1  ' * Draw some objects onto a new PDF
2  Call DPL.SetOrigin(1)
3  Call DPL.SetMeasurementUnits(1)
4  Call DPL.SetPageDimensions(100, 100)
5  Call DPL.SetFillColor(0.8, 0, 0.2)
6  Call DPL.DrawBox(0, 0, 100, 100, 1)
7  Call DPL.SetTextSize(48)
8  Call DPL.SetTextColor(1, 1, 1)
9  Call DPL.SetTextAlign(1)
10 Call DPL.DrawTextBox(0, 0, 100, 100, "Hello", 0)
11
12 ' * Save the new PDF to disk
13 if DPL.SaveToFile(FileName1) = 1 Then
14     WScript.Echo("PDF file written successfully to disk")
15 Else
16     WScript.Echo("Error, PDF file could not be written to disk")
17 End If
18
19 ' * Load the new PDF back into memory
20 Call DPL.LoadFromFile(FileName1, "")
21
22 ' * Render the PDF as an JPG image and save it to disk
23 if DPL.RenderDocumentToFile(96, 1, 1, 1, FileName2) = 1 Then
24     WScript.Echo("JPG image written successfully to disk")
25 Else
26     WScript.Echo("Error, JPG image could not be written to disk")
27 End If
```

Kuva 41 Erään ohjelmointikoodin Debenun pohjakoodista poikkeavat rivit.

Ohjelman tekee ensin pdf-dokumentin ja muuttaa sen jpg-muotoon. Ohjelman aikaansaama jpg-dokumentti (kuva 42):



Kuva 42 Jpg-dokumentin kuva

DEBENU QUICK PDF LIBRARYN FUNKTIORYHMÄT

“Annotations and hotspot links

Create, modify and extract annotations, internal and external links.

Barcodes

Create and place 1D or 2D barcodes that comply with major standards (e.g., Code128, QR Code).

Color

Examine or manage (CMYK and RGB) color usage through the document.

Content Streams and Optional Content Groups

Manipulate Quick PDF Library content streams (layers) as well as Optional Content Groups.

Direct access functionality

Work with files directly on the disk to allow larger documents to be processed quickly.

Document management

Load/save your documents from/to disk or memory.

Document manipulation

Extract pages, split PDFs into smaller documents or merge them together.

Document properties

Modify global document properties including security, fonts, information, and JavaScript.

Extraction

Extract PDF content, including text, full pages, or content-related information (e.g., font properties).

Fonts

Manipulate fonts or find out font properties such as name, size, height, width, kerning.

Form fields

Work with PDF forms - both the original AcroForms technology and static XFA.

HTML text

Basic HTML functionality allows the addition of reflowing text boxes.

Image handling

Add, edit, extract, modify a variety of image types using conventional and Direct Access (DA) functions.

JavaScript

Access the collection of JavaScript functions which turn your PDF into a living dynamic document.

Measurement and coordinate units

Set the units (e.g., mm or inches) to be used by the drawing functions.

Miscellaneous functions

A variety of functions centered around configuring the library such as specifying the unlock key.

Outlines

Add and edit outlines (bookmarks), navigational elements used within a PDF.

Page layout

Add content (including multimedia), configure color, page, and transparency/overprint settings.

Page manipulation

Move, copy, add, rotate, and replace pages using memory or Direct Access functions.

Page properties

Work with page properties like size, orientation, page boxes, page labels, and JavaScript.

Path definition and drawing

Produce vector graphical content using lines and curves.

Rendering and printing

Render PDF content to images for viewing, saving or printing.

Security and Signatures

Encrypt, decrypt documents, apply security properties and work with Digital Signatures.

Text

Add text to a page using a rich set of font characteristics and colors.

Vector graphics

Draw lines, curves or shapes, which can be assigned to drawn directly or assigned to paths.

Deprecated and deleted functions

This is a set of functions that are no longer valid.

Renamed functions

Functions which have been renamed for consistency with the rest of the library." (Deben Pty Ltd 2015.)