

OpenCASCADE DataExchange DWG

eryar@163.com

Abstract. DWG is a file format created in the 70's for the emerging CAD applications. Currently it is the native file format of AutoCAD, a proprietary CAD program developed by Autodesk. Libredwg is a free C library to read and write DWG files. This program is part of GNU project, released under the aegis of GNU. The paper focus on the usage of Libredwg, and use the Libredwg to read a DWG file and output the entities of the DWG to Tcl script for OpenCASCADE Draw Test Harness visualization.

Key Words. OpenCASCADE, DWG, Libredwg, DataExchange, Windows

1. Introduction

DWG 是 CAD 软件 AutoCAD 及基于 AutoCAD 的软件保存设计数据所用的一种专有文件格式，始于 1970 年代的一套 Interact CAD 软件。之后 Autodesk 公司于 1982 年取得版权开始使用这种文件格式。Autodesk 公司拥有、开发并且更新 DWG 文件格式，通常每隔几年 DWG 就会随着在 AutoCAD 中添加新的特性而对 DWG 格式进行更新。

DWG 格式及它的 ASCII 格式变体 DXF 格式，已经成为 CAD 制图数据交换中的事实文件标准，据估计全世界有超过十亿个 DWG 文件。有几家公司正在对 DWG 文件格式进行逆向工程以试图为其它的设计软件提供读写 DWG 文件的能力。Autodesk 公司也提供了一套需要授权的 DWG 读写技术开发包“RealDWG”。

新版的 AutoCAD 可以打开旧版的 DWG 文件，AutoCAD2007 可以打开 2.0 版本的 DWG 文件并且可以保存为 R14 版本。另外 Autodesk 公司提供一个免费的 DWG 查看工具“DWG TrueView”用于查看所有版本的 DWG 文件。另外 Autodesk 公司是 vendor lock-in 策略的强力支持者，尽力保护 DWG 文件格式并且禁止开发支持 DWG 格式的开放源代码库。

2006 年 11 月 12 日，Autodesk 公司对 Open Design Alliance——一款支持 DWG 格式的自由库 OpenDWG 提出了诉讼。

ASCII 格式的 DXF 文件的文档 Autodesk 提供了，但是二进制的 DWG 格式并没有提供相关文档，由上可见对 DWG 文件的读写处理是非常困难的。本文主要介绍如何使用 Libredwg 库来对读取 DWG 中的几何数据，并将几何数据生成为可以在 OpenCASCADE 中显示的 Tcl 脚本，以验证读取数据的正确性。

2. Modify Libredwg for Visual Studio

Libredwg 是一个 Free 的读写 DWG 文件的 C 库，这个程序是 GNU 项目的一部分，授权方式是 GNU GPL3。

Libredwg 是 Libdwg 的一个分支，其目的是创建 OpenDWG 库的一个替代库。也是高优先级的 Free 软件项目：

<http://www.fsf.org/campaigns/priority-projects/priority-projects/highpriorityprojects#ReplaceOpenDWG>，更多信息可访问 <http://www.gnu.org/software/libredwg>。

从网上下载的 Libredwg 源程序是在 Linux 下编译的，并没有配置在 Windows 下编译方法。为了使用 Libredwg 可以在 Windows 上的 Visual Studio 中编译通过，对 Libredwg 做了一些修改，最终编译成功。在 Visual Studio 2008 上成功编译的工程可以文后的链接中下载。

下面给出使用 Libredwg 读取 DWG 文件中直线、圆及文字的例子程序：

```
/*
 * load_dwg.c: load a DWG, get lines, text and circles
 * written by Felipe Castro
 * modified by Felipe Corrêa da Silva Sances
 * modified by Thien-Thi Nguyen
 */

#include <dwg.h>
#include "suffix.c"

void add_line(double x1, double y1, double x2, double y2)
{
    // Make something with that
}

void add_circle(double x, double y, double R)
{
    // Make something with that
}

void add_text(double x, double y, char *txt)
{
    // Make something with that
}

int load_dwg(char *filename)
{
    unsigned int i;
    int success;
    Dwg_Data dwg;

    dwg.num_objects = 0;
    success = dwg_read_file(filename, &dwg);
    for (i = 0; i < dwg.num_objects; i++)
    {
        Dwg_Entity_LINE *line;
        Dwg_Entity_CIRCLE *circle;
```

```

Dwg_Entity_TEXT *text;

switch (dwg.object[i].type)
{
case DWG_TYPE_LINE:
    line = dwg.object[i].tio.entity->tio.LINE;
    add_line(line->start.x, line->end.x, line->start.y, line->end.y);
    break;
case DWG_TYPE_CIRCLE:
    circle = dwg.object[i].tio.entity->tio.CIRCLE;
    add_circle(circle->center.x, circle->center.y, circle->radius);
    break;
case DWG_TYPE_TEXT:
    text = dwg.object[i].tio.entity->tio.TEXT;
    add_text(text->insertion_pt.x, text->insertion_pt.y, text->text_value);
    break;
}
}
dwg_free(&dwg);
return success;
}

int main (int argc, char *argv[])
{
    REQUIRE_INPUT_FILE_ARG (argc);
    load_dwg (argv[1]);
    return 0;
}

```

因为 Libredwg 用的 C 编程风格，没有定义导出定义宏，所以决定将 Libredwg 编译成静态库 libredwg.lib，然后使用其头文件及这个静态库的方式来在程序中使用 Libredwg 库。

经过测试，若 DWG 中只有简单的线和圆等简单实体，Libredwg 还是可以正确读取。但是若用 rewrite 的例子来测试写 DWG 的功能，简单的实例如一个圆的數據都会写出到 DWG 失败，看样子写的功能还没有完全实现好。

3. DWG to OCC

基于上面的例子程序，结合 Libredwg 的读取功能，将 DWG 中的几何数据导出成 Tcl 脚本，这样就可以方便在 OpenCASCADE 的 Draw Test Harness 中来测试结果了。下面给出具体的程序实例及如何在 Draw Test Harness 中来使用生成的 Tcl 脚本。

```
/*
 * Copyright (c) 2014 erylar All Rights Reserved.
 *
 * File      : Main.cpp
 * Author    : erylar@163.com
 * Date      : 2014-10-15 20:46
 * Version   : 1.0v
 *
 * Description : Use libredwg to read data from DWG and
 *               output them to Tcl script for Draw.
 * Key words  : OpenCASCADE, libredwg, Draw Test Harness
 */
#include "dwg.h"
#include <fstream>
#include <iostream>

#pragma comment(lib, "../Debug/libredwg.lib")

// Output the entities to Tcl for OpenCASCADE Draw.
static std::ofstream theTclExporter("d:/dwg2occ.tcl");

void OutputLine(int id, const Dwg_Entity_LINE* theLine)
{
    // Draw Tcl command: vline name xa ya za xb yb zb
    theTclExporter << "vline red" << id << " "
        << theLine->start.x << " " << theLine->start.y << " " << theLine->start.z << " "
        << theLine->end.x << " " << theLine->end.y << " " << theLine->end.z << std::endl;
}

void OutputCircle(int id, const Dwg_Entity_CIRCLE* theCircle)
{
    // Draw Tcl command: circle name x y [z [dx dy dz]] [ux uy [uz]] radius
    // 1. make a curve
    theTclExporter << "circle red" << id << " "
        << theCircle->center.x << " " << theCircle->center.y << " " << theCircle->center.z << " "
        << "0 0 1 " << theCircle->radius << std::endl;

    // 2. make edge from the circle
    theTclExporter << "mkedge e" << id << " "
        << "circle" << id << std::endl;

    // 3. display the circle edge
    theTclExporter << "vdisplay e" << id << std::endl;
}

void OutputText(int id, const Dwg_Entity_TEXT* theText)
{
    // vdrawtext : vdrawtext : vdrawtext name X Y Z R G B hor_align ver_align angle
    // zoomable height Aspect [Font [isMultiByte]]
    theTclExporter << "vdrawtext " << theText->text_value << " "
```

```

        << theText->insertion_pt.x << " " << theText->insertion_pt.y << " 0"
        << " 255 255 000 "
        << theText->horiz_alignment << " " << theText->vert_alignment << " "
        << theText->height << " 1 Times-Roman" << std::endl;
    }

int LoadDwg(char* theDwgFile)
{
    int aResult = 0;

    Dwg_Data aDwgData;

    aResult = dwg_read_file(theDwgFile, &aDwgData);

    for (unsigned int i = 0; i < aDwgData.num_objects; ++i)
    {
        switch (aDwgData.object[i].type)
        {
            case DWG_TYPE_LINE:
                OutputLine(i, aDwgData.object[i].tio.entity->tio.LINE);
                break;

            case DWG_TYPE_CIRCLE:
                OutputCircle(i, aDwgData.object[i].tio.entity->tio.CIRCLE);
                break;

            case DWG_TYPE_TEXT:
                OutputText(i, aDwgData.object[i].tio.entity->tio.TEXT);
                break;
        }
    }

    return aResult;
}

int main(int argc, char* argv[])
{
    theTelExporter.flags(std::ios::fixed);

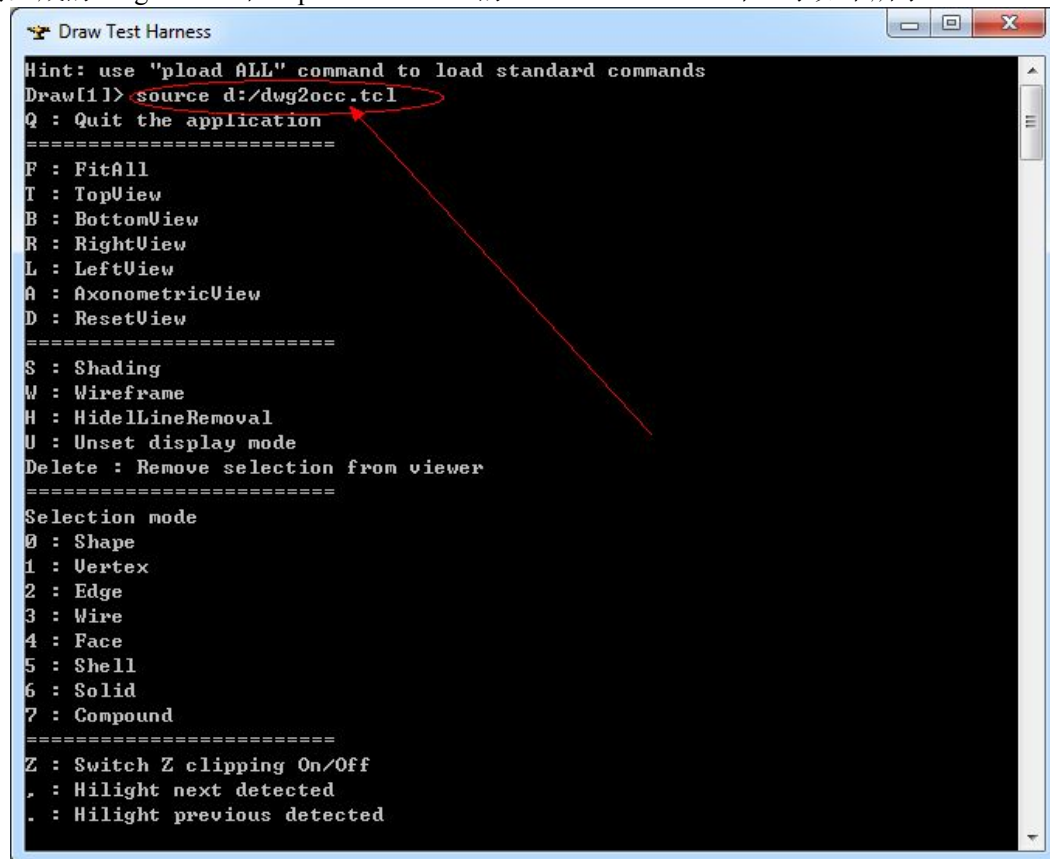
    theTelExporter << "pload ALL" << std::endl;
    theTelExporter << "vinit" << std::endl;
    theTelExporter << "vtrihedron tr" << std::endl;

    if (argc < 1)
    {
        std::cout << "please input the dwg file name!" << std::endl;
    }
    else
    {
        LoadDwg(argv[1]);
    }

    theTelExporter << "vdisplayall" << std::endl;
    theTelExporter << "vfit" << std::endl;
    theTelExporter << "vhelp" << std::endl;
    return 0;
}

```

将生成的 dwg2occ.tcl 在 OpenCASCADE 的 Draw Test Harness 中显示如下所示:



```
Hint: use "pload ALL" command to load standard commands
Draw[1]> source d:/dwg2occ.tcl
Q : Quit the application
=====
F : FitAll
T : TopView
B : BottomView
R : RightView
L : LeftView
A : AxonometricView
D : ResetView
=====
S : Shading
W : Wireframe
H : HideLineRemoval
U : Unset display mode
Delete : Remove selection from viewer
=====
Selection mode
0 : Shape
1 : Vertex
2 : Edge
3 : Wire
4 : Face
5 : Shell
6 : Solid
7 : Compound
=====
Z : Switch Z clipping On/Off
. : Highlight next detected
. : Highlight previous detected
```

Figure 3.1 Import the Tcl script in the Draw Test Hanress of OpenCASCADE



Figure 3.2 Enitites in the DWG file

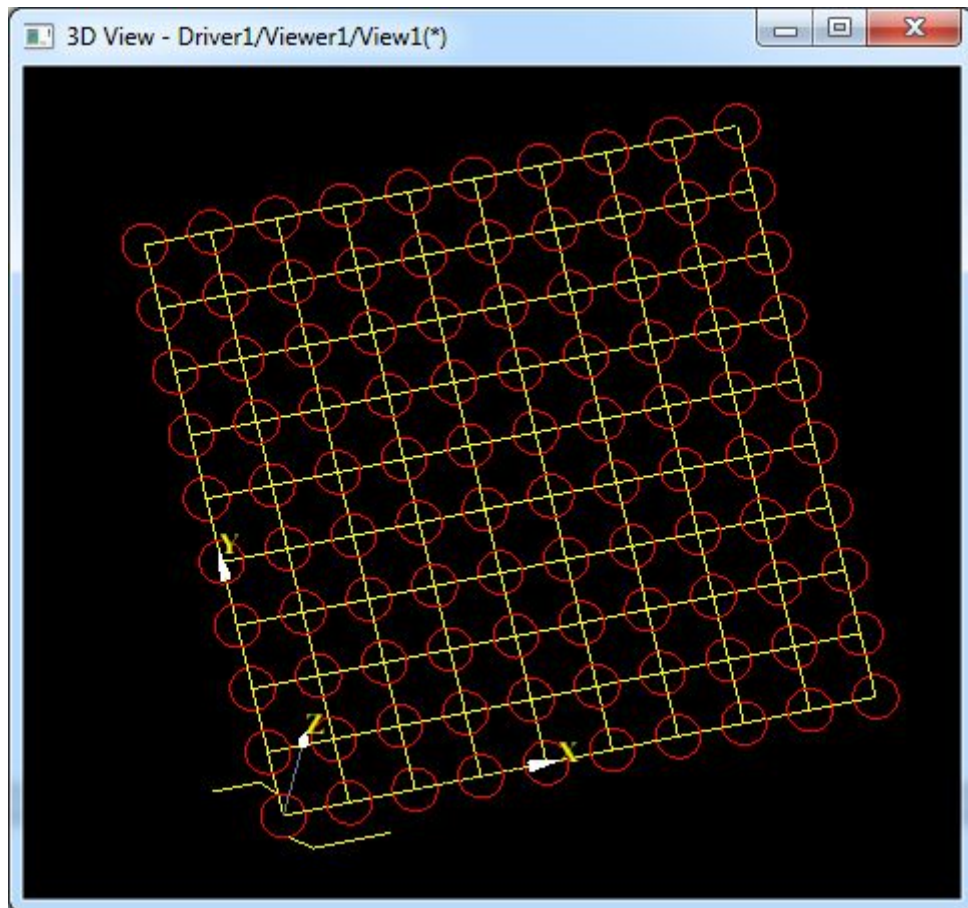


Figure 3.3 Entities in the Draw Test Harness of OpenCASCADE

通过对比发现，直线和圆已经正确读出，但是文字没有读出来。看来 Libredwg 的可靠性还有待提高啊。

4. Conclusion

通过使用 Libredwg 来读取 DWG 格式中几何数据，并将其转换成 OpenCASCADE 的 Draw Test Harness 中能执行的 Tcl 脚本，以方便测试 libredwg 读取数据的正确性。

通过简单测试发现，libredwg 能读取直线和圆，但是文字内容没有正确读出，看来 libredwg 的可靠性还有待提高，但是发现这个开源库的更新很缓慢。

5. References

1. DWG Wiki: <http://en.wikipedia.org/wiki/.dwg>
2. Libredwg: <http://www.gnu.org/software/libredwg>
3. OpenCASCADE Test Harness User's Guide 6.7.1