

Euclidean Vault SDK

使用指南



euclidean
UNLIMITED 3D

Euclidean Vault SDK (简称“VDK”)是面向开发人员的一款产品,是一个定期发布的动态可视化平台。本指南所包含的一些信息和屏幕截图,可能来自较早前版本,但在实际使用中,请使用最新的版本。

简介

通过 VDK,您可以利用优立的无限细节技术,开发属于您的定制化解决方案,本指南将帮助您在特定的计算机环境下进行安装,并提供部分代码。

打开 VDK 并查找:

Directory	Content
\docs\	Documentation for all the functions
\include\	Header files for VDK
\lib\	Library and DLL files
\samples\	Sample program for starting VDK
\testdata\	Sample UDS files you can play around with

VDK 与 Windows、Mac 或 Linux 操作系统兼容。您需要创建自己的封装包,这些功能的详细文档可以在/docs/文档下找到。

Windows

Python

- 创建新的 Python 脚本
- 复制 \include\win_x64\VaultSDK.dll 到文件夹
- 确保在 python 脚本中,您已加载 VDK 并获取所需功能

```
vaultSDK = CDLL("vaultSDK.dll");

vaultContext_Connect = getattr(vaultSDK, "vdkContext_Connect");
vaultContext_Disconnect = getattr(vaultSDK, "vdkContext_Disconnect");

vaultUDModel_Load = getattr(vaultSDK, "vdkPointCloud_Load");
vaultUDModel_Unload = getattr(vaultSDK, "vdkPointCloud_Unload");

vaultUDRenderContext_Render = getattr(vaultSDK, "vdkRenderContext_Render");
vaultUDRenderContext_Create = getattr(vaultSDK, "vdkRenderContext_Create");
vaultUDRenderContext_Destroy = getattr(vaultSDK, "vdkRenderContext_Destroy");

vaultUDRenderView_Create = getattr(vaultSDK, "vdkRenderView_Create");
vaultUDRenderView_Destroy = getattr(vaultSDK, "vdkRenderView_Destroy");
vaultUDRenderView_SetTargets = getattr(vaultSDK, "vdkRenderView_SetTargets");
vaultUDRenderView_SetMatrix = getattr(vaultSDK, "vdkRenderView_SetMatrix");

class vdkRenderInstance(Structure):
    _fields_ = [("pointCloud", c_void_p),
               ("worldMatrix", c_double * 16)]
```

Visual Studio 2015

C/CPP

- 创建一个新的空白的 C++ 项目
- 复制文件目录下的所有头文件
- 复制源文件目录下的 lib\win_x64\VaultSDK.lib
- 在其他依赖项下添加 VaultSDK.lib
- 将\include\win\u64\VaultSDK.dll 复制到生成文件夹中

C#

- 创建一个新的 C# 项目
- 将\include\win_x64\VaultSDK.dll 复制到生成文件夹中

Linux/Mac

即将推出!

如有需求, 请直接联系我们以获得支持: support@euclideon.com.cn

认证

在您设置好所有必须文件之后, 您还需要一个使用 VDK 的授权证书。

Vault 以及扩展的 VDK 需要凭证才能进行渲染和转换。必须创建 vdkContext 并调用函数 vdkContext_Connect 来验证您的认证, 类似 Vault 客户端的登录。有关此功能的详细信息将在本指南的结尾阐述。

到这个步骤, 您应该已经收到优立向您发送了一封电子邮件, 里面详细说明了您的凭据 (Vault 登录用户名和密码) 以及要连接的服务器, 如下面所示。如果你还没有收到这封电子邮件, 请联系 support@euclideon.com.cn 索取。

To access the latest version of **VAULT Client (Desktop)** firstly go to <https://earth.vault.euclideon.com/>, then login using the below:

Server URL: <https://earth.vault.euclideon.com/>
Username: example
Password: example123

建议您查看优立 Vault 客户端并尝试登录以验证您的凭据, 利用这个机会也可以一并测试 VDK。Vault 客户端是用 VDK 构建的, 可以演示使用开发工具包进行渲染和转换的模型。一旦您成功登录到优立 Vault 客户端并验证了您的凭据, 就可以用 VDK 编写您的第一个程序了!

Vault SDK 示例程序

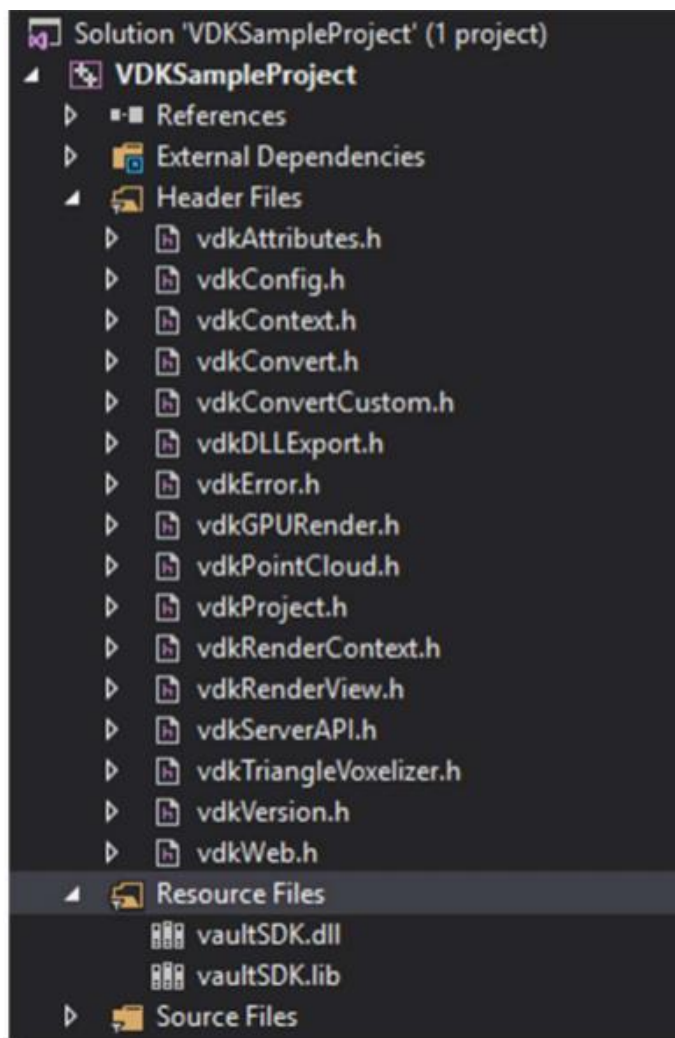
我们捆绑了两个测试程序,您可以构建它们来测试 VDK 并查看用 C、CPP、C 或 Python 编写的示例程序。在本指南,我们将以 CPP 版本和 Visual Studio 2015 为例。

第一个程序是一个 UDS 文件绘制成的 PNG 图像。另外,VDK 可以渲染优立的专有 UDS 文件格式,渲染需要 Vault 许可证。

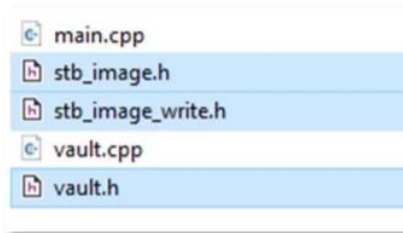
第二个程序将文件类型转换为 UDS 文件。目前,转换支持以下格式: LAS、XYZ、TXT、ASC、CSV、e57、PTS、PTX 和 SLPK。与渲染类似,转换需要 Vault 许可证。

PNG printer

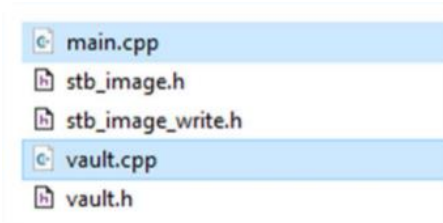
假设您按照步骤在 Visual Studio 2015 设置好了,则应如下所示:



- 把相应文件复制到项目里:
 1. 在 \sample\cpp\, 把所有头文件复制到相应目录下



2. 然后，把源文件复制粘贴到相应目录



- 提供 Vault 凭证

1. 打开 main.cpp

```
main.cpp + X
VDKSampleProject (Global Scope)
1 // A Vault Client Hello world! program in C++.
2
3 #include "vault.h"
4
5 #define STB_IMAGE_IMPLEMENTATION 1
6 #include "stb_image.h"
7 #define STB_IMAGE_WRITE_IMPLEMENTATION 1
8 #include "stb_image_write.h"
9
10 #include <iostream>
11 #include <string>
12
13 int main(int argc, char **ppArgs)
14 {
15     const int width = 1280;
16     const int height = 720;
17
18     std::string serverPath = "ServerURL";
19     std::string username = "Username";
20     std::string password = "Password";
21     std::string modelName = "FilePath";
22     bool pause = false;
23
24     for (int i = 0; i < argc; ++i)
25     {
26         if (strcmp(ppArgs[i], "-u") == 0 && i + 1 < argc)
27             username = ppArgs[++i];
28         else if (strcmp(ppArgs[i], "-p") == 0 && i + 1 < argc)
29             password = ppArgs[++i];
30         else if (strcmp(ppArgs[i], "-s") == 0 && i + 1 < argc)
31             serverPath = ppArgs[++i];
32         else if (strcmp(ppArgs[i], "-m") == 0 && i + 1 < argc)
33             modelName = ppArgs[++i];
34         else if (strcmp(ppArgs[i], "-pause") == 0)
35             pause = true;
36     }
37
38     const double cameraMatrix[] = {
39         1,0,0,0,
40         0,1,0,0,
41         0,0,1,0,
42         5,-75,5,1
43     };
44
45     vdkError error;
46     Vault::Context context;
47     Vault::RenderContext renderer;
48     Vault::RenderView renderView;
49     std::vector<Vault::PointCloud> models;
```

2. 在 main () 下，提供您自己的凭据以及分配给您的服务器 URL (请参阅优立发给您的电子邮件)。

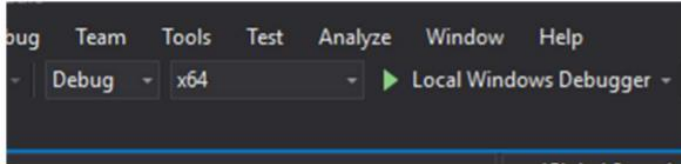
```
std::string serverPath = "ServerURL";
std::string username = "Username";
std::string password = "Password";
```

3. 然后提供要渲染的 UDS 文件的文件路径（示例 UDS 文件在/samples/）

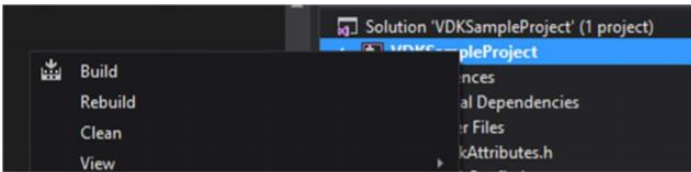
```
std::string modelName = "FilePath";|
```

4. 建立项目并执行

提供必要的参数后，将其设置为在 x64 中运行



右键 project 并选择 “生成”



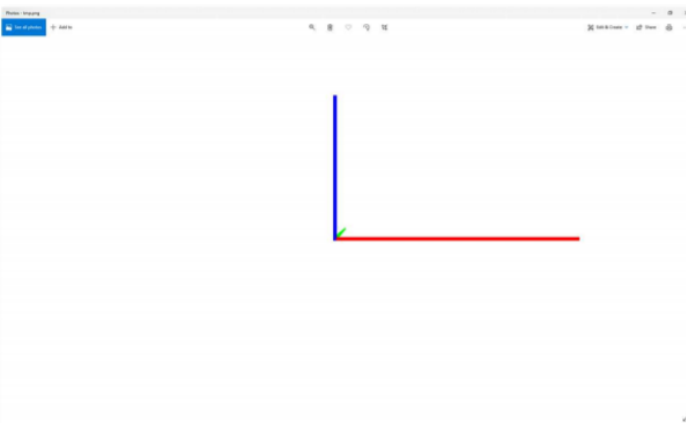
操作正确的话，您就可以成功生成

```
==== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

运行在生成文件夹中找到的可执行文件

File Name	Created	Type	Size
testdata	16/10/2019 4:55 PM	File folder	
tmp.png	16/10/2019 5:13 PM	PNG File	36 KB
vaultSDK.dll	1/08/2019 2:56 PM	Application extens...	1,279 KB
VDKSampleProject.exe	16/10/2019 5:13 PM	Application	281 KB
VDKSampleProject.ilink	16/10/2019 5:13 PM	Incremental Linke...	2,463 KB
VDKSampleProject.pdb	16/10/2019 5:13 PM	Program Debug D...	1,924 KB

执行之后，应该有一个名为 tmp.png 的文件

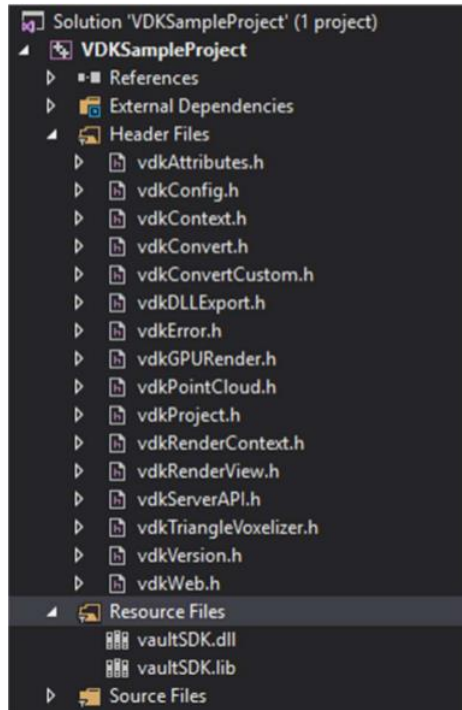


现在您已经使用 VDK 进行了第一次渲染！PNG printer 是一个基本程序，展示如何在 API 内进行渲染。您可以查看主文件和头文件，以了解功能如何实现。优立 Vault 客户端还提供了无限细节渲染的浏览，建议您试一下。此外，我们提供了一个完整的文档，可以在

/docs/下找到。

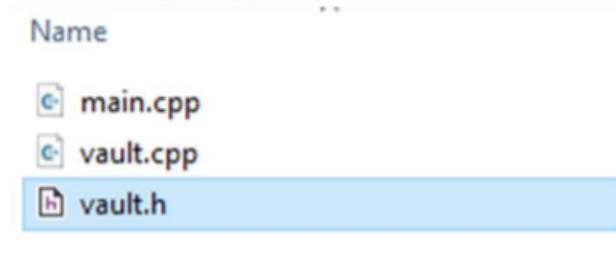
转换 UDS

如果您按照 VDK 步骤设置好了 Visual Studio 2015，您看到的项目列表应该如下：

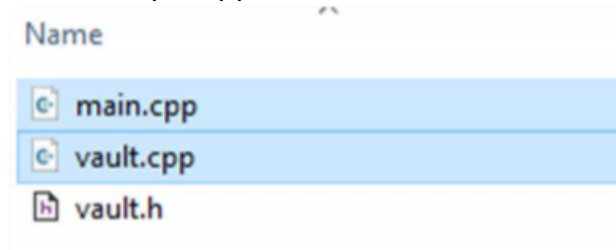


复制文件到项目：

1. 在 \sample\cpp\, 复制所有头文件并粘贴到相应文件夹



2. 在 \sample\cpp\, 复制所有源文件并粘贴到相应文件夹



3. 打开 main.cpp

```
main.cpp * X
% VDKSampleProject (Global Scope)
1 // A Vault Client Hello World! program in C++.
2
3 #include "vault.h"
4
5 #define STB_IMAGE_IMPLEMENTATION 1
6 #include "stb_image.h"
7 #define STB_IMAGE_WRITE_IMPLEMENTATION 1
8 #include "stb_image_write.h"
9
10 #include <iostream>
11 #include <string.h>
12
13 int main(int argc, char **ppArgv)
14 {
15     const int width = 1280;
16     const int height = 720;
17
18     std::string serverPath = "ServerURL";
19     std::string username = "Username";
20     std::string password = "Password";
21     std::string modelName = "FilePath";
22     bool pause = false;
23
24     for (int i = 0; i < argc; ++i)
25     {
26         if (strncmp(ppArgv[i], "-u") == 0 && i + 1 < argc)
27             username = ppArgv[++i];
28         else if (strncmp(ppArgv[i], "-p") == 0 && i + 1 < argc)
29             password = ppArgv[++i];
30         else if (strncmp(ppArgv[i], "-s") == 0 && i + 1 < argc)
31             serverPath = ppArgv[++i];
32         else if (strncmp(ppArgv[i], "-m") == 0 && i + 1 < argc)
33             modelName = ppArgv[++i];
34         else if (strncmp(ppArgv[i], "-pause") == 0)
35             pause = true;
36     }
37
38     const double cameraMatrix[] = {
39         1,0,0,0,
40         0,1,0,0,
41         0,0,1,0,
42         5,-75,5,1
43     };
44
45     vdkError error;
46     Vault::Context context;
47     Vault::RenderContext renderer;
48     Vault::RenderView renderView;
49     std::vector<Vault::PointCloud> models;
```

4. 在 main () 下，提供您自己的凭据以及分配给您的服务器 URL（请参阅优立发给您的电子邮件）

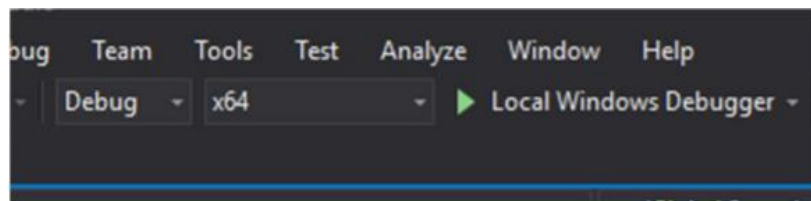
```
std::string serverPath = "ServerURL";
std::string username = "Username";
std::string password = "Password";
```

5. 提供你想要转换文件的路径

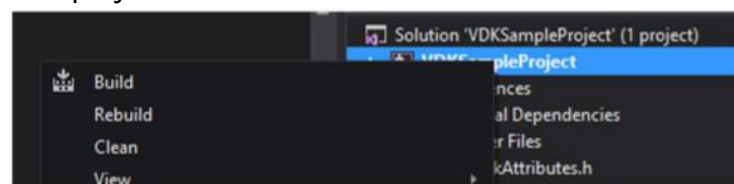
```
std::string modelName = "FilePath";
```

6. 建立项目并执行

提供必要的参数后，将其设置为在 x64 中运行



右键 project 并选择创建 build



操作正确的话，您将成功创建

```
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
```

运行在生成文件夹中找到的可执行文件

testdata	16/10/2019 4:55 PM	File folder	
Tmp.uds	27/05/2019 1:14 PM	UDS File	24,900 KB
vaultSDK.dll	1/08/2019 2:56 PM	Application extens...	1,279 KB
VDKSampleProject.exe	16/10/2019 5:13 PM	Application	281 KB
VDKSampleProject.ilc	16/10/2019 5:13 PM	Incremental Linke...	2,463 KB
VDKSampleProject.pdb	16/10/2019 5:13 PM	Program Debug D...	1,924 KB

作为一个基本的程序，UDS 转换器展示了 VDK 如何完成转换工作，更多高级的功能，例如添加元数据或将地理位置插入到模型中，也可以在 VDK 完成。有关详细信息，请参阅 /docs/下的完整文档。

其他技术支持

希望本操作指南能让您对 VDK 有一个基本的了解。如果您需要更多信息，或者任何有关 VDK 的反馈，请发送电子邮件至 support@euclidean.com.cn。我们的技术支持团队将为您提供一对一的支持。

另外，在 Github 网站 <https://github.com/Euclidean/vaultclient> 也可以下载最新的源代码。

联系我们

获取更多关于优立 Vault 资料，请联系我们 info@euclidean.com.cn 或访问官网 www.euclidean.com.cn