# HCNetSDK代码反向建模工具包

## 背景

HCNetSDK中存在大量的复合数据结构（结构体、联合体、类）、宏定义及用户接口API。不同于ISAPI这种制定初期就建立完善体系的协议，HCNetSDK缺少健全的软件模型和接口管理。长时间积累后众多的数据结构及接口加大管理的难度，大量新增的接口及数据结构很难及时跟新到文档中，为HCNetSDK集成开发者制造了障碍。而目前若企图以人工方式重新建立软件模型，几十万行的代码的分析和数千个数据对象的建模又存在工作量巨大的问题。

以下介绍一种方法，可以通过自动化方式建立HCNetSDK中用户层复合数据结构，宏定义及用户接口的软件模型。软件模型建立以后，可利用软件模型完成其他功能。

且该方法可推广到其他SDK产品中使用，保证相同的有效性。

## 技术路线

### 开发工具

开发语言：python

核心工具：doxygen

必要三方库：PyQt4、ElementTree、numpy、sicpy

存储格式：mat文件

### 开发目标

* HCNetSDK应用层数据结构及用户接口数据库

为HCNetSDK的应用层数据结构及接口建立软件模型，重点关注HCNetSDK.h中struct、union、class、及function对象

* GUI集成工具包界面程序

工具包包含：

* **对象数据库索引工具**

提供界面增、删、改、查语法对象数据库

* **一键代码生成工具**

由于建立了语法对象数据库，只需要建立相应的语法转换规则，即可产生一键产生对应代码，比如转jni、jna、c#等

* **私有协议校验工具**

如果使用同样方法，建立HCNetSDK私有协议的数据库。那么可以通过将从wireshark提取的数据拷贝到私有协议校验工具中，进行协议的验证

### 实施方案

一、使用doxygen工具，输入c++源代码文件，输出c++语法结构xml文件

具体，下载->安装->启动doxygen GUI->配置源代码路径->勾选输入文件格式为xml->配置解析语法元素包含局部变量、宏、全局函数、成员函数、预编译宏等->运行doxygen->产生xml语法描述结构。

*参考：使用doxygen产生c++语法结构xml文件方法（*[*附*](#_附：)*）*

二、使用python解析xml文件建立语法结构数据库

使用python xml解析包解析doxygen生成的xml文件，建立以struct、union、class、function等语法单元为核心的逻辑结构，并建立索引。最后保存在.mat文件数据库中，形成完整的HCNetSDK语法结构数据库。

三、使用PyQt4开发集成工具

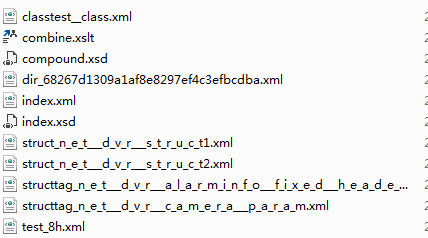
# 附：

## 使用doxygen产生xml数据库

doxygen默认不产生xml文件，需要在expert->xml启用xml才会产生

### doxygen产生的xml数据库分析

产生的xml数据库，目录如下：



#### 索引文件

其中，起到索引作用的文件是index.xml，对全局语法对象的索引。index.xml中将复合类型CompoundKind分为如下类，复合组是doxygen为复合语法元素定义的分类

|  |
| --- |
| class /\*\*  struct \* 复合数据类型  union \*/  interface  protocol  category  exception  file // 除复合数据类型以外的语法元素都归为此类  namespace  group  page  example  dir |

MemberKind为复合组下属的小分类，即原子语法单元，定义如下：

|  |
| --- |
| define  property  event  variable  typedef  enum  enumvalue  function  signal  prototype  friend  dcop  slot |

如果只是分析c/c++的语法元素，则只需要关注其中标红的语法元素复合组即可。下面提供一个索引文件的示例

|  |
| --- |
| <compound refid="structXXX" kind="struct"><name>XXX</name>  <member refid="structXXX\_1a1e1849aa56c12ebd8addf493e9fbfe76" kind="variable">  <name>fCameraHeight</name>  </member>  <member refid="structXXX\_1aa112b83f51ad71a45383575bcd002127" kind="variable">  <name>fCameraAngle</name>  </member>  </compound>  <compound refid="classXXX" kind="class"><name>XXX</name>  <member refid="classXXX\_1a51c17fc1302330b5453c263ed9bfe66c" kind="variable">  <name>byEnableHeight</name>  </member>  <member refid="classtest\_\_class\_1a06608dd4b9a6ec170aa7e402b27db649" kind="function">  <name>test\_func</name>  </member>  </compound>  <compound refid="union\_union\_test" kind="union">  <name>UnionTest</name>  </compound>  <compound refid="test\_8h" kind="file"><name>test.h</name>  <member refid="test\_8h\_1aaa0a105941a262d3efc09f89db90ebf0" kind="define">  <name>MAX\_NAMELEN</name>  </member>  <member refid="test\_8h\_1acf657caf678d125698170633d9d821f1" kind="enum">  <name>tagLINE\_MODE</name>  </member>  <member refid="test\_8h\_1acf657caf678d125698170633d9d82f774b8f45" kind="enumvalue">  <name>HEIGHT\_LINE</name>  </member>  <member refid="test\_8h\_1a798af1e30bc65f319c1a246cecf59e39" kind="typedef">  <name>DWORD</name>  </member>  <member refid="test\_8h\_1a53ab5c0efac09b056eef3305d501c1ae" kind="function">  <name>NET\_DVR\_STDXMLConfig\_Conv</name>  </member>  </compound> |

可以从index.xml文件中检索所有复合语法单元，然后我们可以从<compound refid=...></compound>获取复合语法单元的文件名，进而取获取细节信息。注：index文件中只包含语法单元基本，细节信息需查看语法单元的文件。

#### 复合语法单元文件

复合语法单元的类型[参考](#复合语法元素定义的分类)

从c++语法的角度，只需关心如下四类复合元素：

|  |
| --- |
| class /\*\*  struct \* 复合数据类型  union \*/  file // 除复合数据类型以外的语法元素都归为此类 |

doxygen为每一个复合语法单元生成一个详细信息文件，注：file类型的复合语法文件只有一个，所以可能超大！

##### file复合语法单元文件

file复合语法单元包含原子语法单元较多，基本分为

|  |
| --- |
| innerclass  sectiondef |

其中每一个复合数据类型struct, union, class占用一个<innerclass></innerclass>元素描述，示例：

|  |
| --- |
| <innerclass refid="struct\_n\_e\_t\_\_\_d\_v\_r\_\_\_s\_t\_r\_u\_c\_t2" prot="public">NET\_DVR\_STRUCT2</innerclass>  <innerclass refid="union\_union\_test" prot="public">UnionTest</innerclass>  <innerclass refid="classtest\_\_class" prot="public">test\_class</innerclass> |

这一类复合元素的详细信息单独有文件说明，故在file文件中描述扼要。

file文件的关注点在于其他非复合数据类型的语法元素

|  |
| --- |
| define  enum  typedef  func |

示例：

|  |
| --- |
| <sectiondef kind="define">  <memberdef kind="define" id="test\_8h\_1aaa0a105941a262d3efc09f89db90ebf0" prot="public" static="no">  <name>MAX\_NAMELEN</name> <!-- “宏名”-->  <initializer>16</initializer><!-- “值”-->  <briefdescription>  </briefdescription>  <detaileddescription>  </detaileddescription>  <inbodydescription>  </inbodydescription>  <location file=".../test.h" line="1" column="9" bodyfile=".../test.h" bodystart="1" bodyend="-1"/>  </memberdef>  ...  </sectiondef> |
| <sectiondef kind="enum">  <memberdef kind="enum" id="test\_xxxx8d125698170633d9d821f1" prot="public" static="no" strong="no">  <type></type>  <name>tagLINE\_MODE</name> <!-- “枚举类型”-->  <enumvalue id="test\_8h\_1acf657caf678d446d77872aaef774b8f45" prot="public">  <name>HEIGHT\_LINE</name> <!-- “枚举项”-->  <initializer>= 12</initializer><!-- “枚举项初始值”-->  <briefdescription>  </briefdescription>  <detaileddescription>  </detaileddescription>  </enumvalue>  <enumvalue id="test\_8h\_1acf657ca4f00bf0e6587f681c70c0" prot="public">  <name>LENGTH\_LINE</name>  <briefdescription>  </briefdescription>  <detaileddescription>  </detaileddescription>  </enumvalue>  <briefdescription>  </briefdescription>  <detaileddescription>  </detaileddescription>  <inbodydescription>  </inbodydescription>  <location file=".../test.h" line="19" column="1" bodyfile=".../test.h" bodystart="18" bodyend="22"/>  </memberdef>  ...  </sectiondef> |
| <sectiondef kind="typedef">  <memberdef kind="typedef" id="test\_8h\_1a798af1e30bc65f319c1a246cecf59e39" prot="public" static="no">  <type>unsigned int</type>  <definition>typedef unsigned int DWORD</definition>  <argsstring>  </argsstring>  <name>DWORD</name>  <briefdescription>  </briefdescription>  <detaileddescription>  </detaileddescription>  <inbodydescription>  </inbodydescription>  <location file="D:/3\_test\_program/HCStructModel/XmlModel2/src/test.h" line="7" column="1" bodyfile="D:/3\_test\_program/HCStructModel/XmlModel2/src/test.h" bodystart="7" bodyend="-1"/>  </memberdef>  </sectiondef> |
| <sectiondef kind="func">  <memberdef kind="function" id="test\_8h\_1a5b6b0xxx8b1ad2b313a4" prot="public" static="no" const="no" explicit="no" inline="no" virt="non-virtual">  <type>  <ref refid="test\_8h\_1ad50602a3ef40bd7a6c81f4b245921380" kindref="member">NET\_DVR\_API</ref>  <ref refid="test\_8h\_1a9154c0d0c21af4686624543215b4e5f2" kindref="member">LONG</ref>  \_\_stdcall  </type>  <definition>NET\_DVR\_API LONG \_\_stdcall NET\_DVR\_SDKChannelToISAPI</definition>  <argsstring>(LONG lUserID,LONG lInChannel,BOOL bSDKToISAPI)</argsstring>  <name>NET\_DVR\_SDKChannelToISAPI</name>  <param>  <type>  <ref refid="test\_8h\_1a9154c0d0c21af4686624543215b4e5f2" kindref="member">LONG</ref>  </type>  <declname>lUserID</declname>  </param>  <param>  <type>  <ref refid="test\_8h\_1a9154c0d0c21af4686624543215b4e5f2" kindref="member">LONG</ref>  </type>  <declname>lInChannel</declname>  </param>  <param>  <type>  <ref refid="test\_8h\_1ae4cc35dcc70810fa972cc8a5185a28fa" kindref="member">BOOL</ref>  </type>  <declname>bSDKToISAPI</declname>  </param>  <briefdescription>  </briefdescription>  <detaileddescription>  </detaileddescription>  <inbodydescription>  </inbodydescription>  <location file="D:/3\_test\_program/HCStructModel/XmlModel2/src/test.h" line="96" column="1"/>  </memberdef>  </sectiondef> |