

本文介绍 OpenCPN 中 ENC 到 SENC 的转换过程。

ENC(Electric Nautical /Navigational Chart)是由国家官方机构 (HO) 发布的、专供 ECDIS 使用的、符合国际标准的数据库。ENC 除包含为了安全航行所必需的海图信息外, 还可能包含航路指南、港口概况等其他有用的信息。其数据格式主要有矢量方式和栅格方式两种。经 IHO 承认的矢量数据格式标准为 S57/3.0, 栅格数据格式标准为 ARCS。

在海图系统中, 一般将 ENC 海图转换为系统内部格式 SENC (系统 ENC)。OpenCPN 也不例外。OpenCPN 代码中对应功能函数为 BuildSENCFile。函数定义如下:

```
int s57chart::BuildSENCFile(const wxString& FullPath000, const wxString& SENCFileName)
```

其中参数, FullPath000 为 000 文件(即 ENC 文件)全路径; SENCFileName 为转换结果生成的 SENC 文件名。

OpenCPN 的 SENC 文件格式总体分为两部分: 文件头+内容。文件头中包含了关于海图的总体信息, 包括版本号、更新时间、比例尺等; 文件内容为具体海图元素。

1. GetBaseFileAttr -- 获取海图文件基本信息, 用于写 SENC 头。

函数中使用 DDFModule 打开 ENC 文件, 读取比例尺等基本信息所在的 Record, 从而获取需要的数据。

读取海图比例尺:

```
m_native_scale = pr->GetIntSubfield("DSPM",0,"CSCL",0);
```

DDFModule, DDFRecord 是 gdal 库中的类, 用于读取 ENC 文件。关于 S57 标准, 请参考论坛中的帖子: <http://www.landho.cn/thread-49543-1-1.html>

对于 gdal 库, 我接触也不深, 想要深入学习的人可以看此牛人的博客:

<http://blog.csdn.net/liminlu0314?viewmode=contents> 也欢迎讨论交流。

以下是写 SENC 头的部分代码, 简单易懂不用解释:

```
fprintf(fps57, "SENC Version= %d\n", CURRENT_SENC_FORMAT_VERSION);

strncpy(temp, nice_name.mb_str(), 200);
temp[200] = '\0';
fprintf(fps57, "NAME=%s\n", temp);

wxString date000 = m_date000.Format(_T("%Y%m%d"));
strncpy(temp, date000.mb_str(), 200);
fprintf(fps57, "DATE000=%s\n", temp);

strncpy(temp, m_edtn000.mb_str(), 200);
fprintf(fps57, "EDTN000=%s\n", temp);

// Record .000 file date and size for primitive detection of updates to .000 file
wxDateTime ModTime000;
wxString mt = _T("20000101");
if(file000.GetTimes(NULL, &ModTime000, NULL))
    mt = ModTime000.Format(_T("%Y%m%d"));
strncpy(temp, mt.mb_str(), 200);
fprintf(fps57, "FILEMOD000=%s\n", temp);

int size000 = file000.GetSize().GetHi();
int size000l = file000.GetSize().GetLo();
fprintf(fps57, "FILESIZE000=%d%d\n", size000, size000l);
```

2. 类 OGRS57DataSource 封装了 gdal 库，实现对 ENC 文件的读写。过程如下：

1). 设置对于 s57 读取的控制选项

```
char ** papszReaderOptions = NULL;
papszReaderOptions = CSLSetNameValue(papszReaderOptions, S57O_LNAM_REFS, "ON" );
papszReaderOptions = CSLSetNameValue( papszReaderOptions, S57O_UPDATES, "ON");
papszReaderOptions = CSLSetNameValue( papszReaderOptions, S57O_RETURN_LINKAGES, "ON");
papszReaderOptions = CSLSetNameValue( papszReaderOptions, S57O_RETURN_PRIMITIVES, "ON");
poS57DS->SetOptionList(papszReaderOptions);
```

gdal /ogr 中关于 optionlist 的说明：

S57 Control Options

There are several control options which can be used to alter the behavior of the S-57 reader. Users can set these by appending them in the OGR_S57_OPTIONS environment variable.

- **UPDATES=APPLY/IGNORE:** Should update files be incorporated into the base data on the fly. Default is APPLY.
- **SPLIT_MULTIPPOINT=ON/OFF:** Should multipoint soundings be split into many single point sounding features. Multipoint geometries are not well handle by many formats, so it can be convenient to split single sounding features with many points into many single point features. Default is OFF.
- **ADD_SOUNDG_DEPTH=ON/OFF:** Should a DEPTH attribute be added on SOUNDG features and assign the depth of the sounding. This should only be enabled with SPLIT_MULTIPPOINT is also enabled. Default is OFF.
- **RETURN_PRIMITIVES=ON/OFF:** Should all the low level geometry primitives be returned as special IsolatedNode, ConnectedNode, Edge and Face layers. Default is OFF.
- **PRESERVE_EMPTY_NUMBERS=ON/OFF:** If enabled, numeric attributes assigned an empty string as a value will be preserved as a special numeric value. This option should not generally be needed, but may be useful when translated S-57 to S-57 losslessly. Default is OFF.
- **LNAM_REFS=ON/OFF:** Should LNAM and LNAM_REFS fields be attached to features capturing the feature to feature relationships in the FFPT group of the S-57 file. Default is OFF.
- **RETURN_LINKAGES=ON/OFF:** Should additional attributes relating features to their underlying geometric primitives be attached. These are the values of the FSPT group, and are primarily needed when doing S-57 to S-57 translations. Default is OFF.
- **RECODE_BY_DSSI=ON/OFF:** (OGR >= 1.10) Should attribute values be recoded to UTF-8 from the character encoding specified in the S57 DSSI record. Default is OFF.

引自：http://www.gdal.org/ogr/drv_s57.html

2). 调用 OGRS57DataSource::Open 完成读取 Feature 前的准备

```
int open_return = poS57DS->Open(m_tmpup_array->Item(0).mb_str(), TRUE, &s_ProgressCallBack);
if(open_return == BAD_UPDATE) //172
    bbad_update = true;
```

3). 获取 S57Reader

```
poReader = poS57DS->GetModule(0);
```

4). 调用 S57Reader::ReadNextFeature 读取 Feature

5). 循环处理读取到的每一个 Feature，在 CreateSENCRecord 中写入 SENC 文件

函数原型如下：

```
void s57chart::CreateSENCRecord( OGRFeature *pFeature, FILE * fpOut, int mode, S57Reader *poReader )
```

A. 循环写入 Feature 所有 Field：

```
//      In the interests of output file size, DO NOT report fields that are not set.
for( int iField = 0; iField < pFeature->GetFieldCount(); iField++ )
{
    if( pFeature->IsFieldSet( iField ) )
    {
        if( (iField == 1) || (iField > 7))
        {
            OGRFieldDefn *poFDefn = pFeature->GetDefnRef()->GetFieldDefn(iField);

            const char *pType = OGRFieldDefn::GetFieldTypeName(poFDefn->GetType() );

            snprintf( line, MAX_HDR_LINE - 2, " %s (%c) = %s",
                    poFDefn->GetNameRef(),
                    *pType,
                    pFeature->GetFieldAsString( iField ) );

            sheader += wxString(line, wxConvUTF8);
            sheader += '\n';
        }
    }
}
```

B. 根据几何类型写入其他信息。其中内容微复杂。

6). 写 VE Table 和 VC Table

```
if(bcont)
{
    //      Create and write the Vector Edge Table
    CreateSENCVectorEdgeTable(fps57, poReader);

    //      Create and write the Connected NodeTable
    CreateSENCConnNodeTable(fps57, poReader);
}
```

7). 成功写 SENC 文件，把文件内容从零时文件拷贝到指定文件，删除零时文件。
完成。