

目录

1. 最新测试版开发情况	1
2. 开发人员指南	6
2.1 编译 - Linux	7
2.1.1 在 Motorola Xoom 上的应用	9
2.2 编译 - Mac OS X	21
2.3 编译 - Windows	22
2.3.1 国际化	26
2.4 代码格式化	27
3. 插件 API	28
3.1 插件指南	29
3.2 不同插件 API 版本	30
4. 问题解决	30
5. 矢量调色板	31
5.1 传统矢量调色板	34
6. 用户界面风格定义	36
7. Bug 和功能追踪	38
8. OpenCPN 发行历史	38
8.1 2012-6-5 发行 3.0	41
8.2 OpenCPN3.2 新功能	44

开发人员手册

1. 最新测试版开发情况

欢迎加入我们的论坛讨论最新开发情况

在[下载](#)页面下载最新测试版

记住:

所有开发情况只放在 **GitHub** 存储库中。

通过以下字符串得到最新测试版

```
"git clone -b beta31_stable https://github.com/OpenCPN/OpenCPN.git"
```

3.1 测试版开发情况

变更日志

3.1.1405 版本

- 183. 更新一些可翻译字符串
- 184. 提高 **Linux** 系统仪表板图形
- 185. 更正航线/航路点上传时序逻辑
- 186. 执行 NMEA0183 APB 语句以便进行**自动导航**输出
- 187. 提高导出所有可视对象 ("Export all visible" objects) 效能
- 188. 停用 RMD 多对象选择 (multiobject selection) 中的一些控制 (全部删除)
- 189. 将 **Garmin** 设备模式 (**Garmin Host Mode**) 状态添加到连接参数自动更新中
- 190. 仪表板**深度**: 将 DPT 而非 DBT 用于**深度**; 如果存在转换器偏差, 将该偏差加到深度值中
- 191. 移除 AIVDO 信息处理过程中的磁航向更新和变差
- 192. 提高 **Garmin** 串行端口可用度.
- 193. 为耗时长的 GPX 和航迹操作添加进度条
- 194. 为 View 仪器更正卫星更新
- 195. **文档**更新, 感谢 Thomas (cagney)

3.1.1328 版本

- 168. 更正航迹开始时间戳的计算
- 169. 提升默认“发送到 GPS” (Send to GPS) 快捷菜单快捷键操作而没有定义端口.
- 170. 增加 Tide 图标可选择点半径提高可选择性
- 171. 使“航线管理对话框” (RouteManagerDialog) 列表可以进行多选操作, 特别是“删除” (Delete)
- 172. 从航线管理对话框移除“导出所有” (Export All) 按钮
- 173. 进一步提高仪表板文本显示
- 174. 更正仪删除表板错误作为参考对话框父框
- 175. 对 Linux 和 Mac 中已经升级的**串**符端口进行检测
- 176. 恢复#166, libtinyxml 依赖项. 现在 2.6.2
- 177. 通过 **linux** 嵌入空间更正 Mark Info html 连接
- 178. 当没有网络接口的时候更正 Windows 中的 partial hang
- 179. 更正 Mac 中重启大型 logfile 文件时出现的系统崩溃
- 180. 在“选项->连接列表” (options->Connections list) 中添加已进行优先等级排序的数据流
- 181. 更新一些可翻译字符串
- 182. 实现**航线**打印功能. 感谢该补丁的“注册” (registry)

3.1.1319 测试版

- 152.更正 NMEA 调试窗口输出数据端口字符串
- 153. 实现 armhf 架构有条件的建置指令
- 154. 移除一些无关的可翻译字符串
- 155. 删除对对矢量海图“暗礁/沉船”(rocks/wrecks)显示有开关功能的热键 "R"
- 156. 如果本船“对地航向”(Cog)或“对地航速”(Sog)没有定义,更正 AIS 目标 CPA 图形
- 157. 将 cr/lf 加入 TCP 输入数据流
- 158. 将“发送到 GPS”(Send to GPS)添加到航线快捷菜单
- 159. 更正航线点删除逻辑, 当该航线部分可视
- 160. 更正/加强 ENC 暗礁/沉船可视逻辑
- 161. 更正上下文菜单对激活航线的选择 (FS#948)
- 162. 对仪表板丢失数据实行超时功能(FS#937) (FS#950)
- 163. 实现“航线管理对话框”(Route Manager)中的“导出所有可视”(Export All Visible) 按钮
- 164. 更正一些 ENC 中外来 OBSTRN 图案 (FS#951)
- 165. 更正每个海图组的海图状态栏显示
- 166. 放松 Debian libtinyxml 对 2.5.3 Squeeze 的依赖
- 167. 提高仪表板文本渲染 (FS#???)

Release Beta 3.1.1309

- 129. 更正连续性快速串行端口写入错误
- 130. 更正读取缓冲区溢位串行端口
- 131. 更正通用 SART 目标 query 报告
- 132. 将“libsdl1.2debian”包作为声音依赖项删除
- 133. 更正以允许对删除的数据流进行一些操作
- 134. 往 NMEA 调试窗口添加优先级更改信息
- 135. 更正航线/航路点上传端口持久性
- 136. 将本船默认最小尺寸(mm)设置为 1
- 137. 从上下文菜单进行航线点 GPS 上传对激活的 GPS 信号的要求
- 138. 允许激活 layer 中的航线和航线点
- 139. 更正 Journeyman 模式下铅笔图标偏移(FS#943)
- 140. 更正真航向超时监视逻辑 (FS#937)
- 141. 将“Garmin 设备模式”(Garmin Host Mode) 和 Furuno 格式上传转换成 global options.
- 142. 更正仪表板可视度逻辑选项
- 143. 更正无效经纬度 AIVDO 处理 (FS#935)(FS#936)
- 144. 为 n/a fields 提高 AIS 目标 query 格式 (FS#934)
- 145. 更正“选项->海图”(Options->Charts)添加/删除/应用逻辑(FS#944)
- 146. 更正创建选项 USE_GARMINHOST=OFF (FS#931)
- 147. 更正可能的 Windows ReadFile() 出错状态. (FS#947)
- 148. 更正应用激活/停用 Mac OSX behavior (FS#938)
- 149. 更正共用航路点可视选择逻辑(FS#942)
- 150. 语言更新
- 151. 内置用户手册重要更新, 特别感谢 Thomas (cagney)

Release Beta 3.1.1224

- 115. 更正以允许 linux /dev/ttyUSB0 访问
- 116. 更正 AIS 端口(共用)"AIS Port (Shared)"自动更新到 3.2 连接架构
- 117. 更正 GPS 上传端口持久性以包含手动输入端口

- 118. 更正 Windows 串行端口写入性能瓶颈
- 119. 更正 AIS 文件扩展名字段更新 (FS#886)
- 120. 实现 portaudio 声音库对 linux.的支持
- 121. 更正仪表板数据流接口
- 122. 提高 Journeyman 模式 AIS 图标和控制台背景.
- 123. 更正 Xport 与 Windows API 不一致
- 124. 更正 UDP 临时输出端口(FS#908)
- 125. 添加 WindHistory 仪器到仪表板, 感谢 Rauch.
- 126. 完成仪表板 AIVDO 处理
- 127. 更新语言文件
- 128. 实现不能修改的 layer 对象的针对性上下文菜单(FS#576)

Release Beta 3.1.1220

- 92. 更正并简化 BOYSPP 符号
- 93. 更正 Rollover 框字体大小逻辑
- 94. 更正 Mac 初始对话框调用顺序
- 95. 更正丢失的“AIS 目标”(AIS Target) 针对性上下文菜单
- 96. 更正 Windows 串行端口输入为非重叠性输入, 以进行可靠性提高的实验
- 97. 通过未指定的端口号更正 UDP 接受故障 (FS#922)
- 98. 更正 Windows 同时安装更新的默认目录选择
- 99. 提高数据流内部线程信息可靠性
- 100. 更正单一海图模式组"sticky" 海图选择逻辑.
- 101. 提高 AIS 工具栏图标显示规律.
- 102. 使最后 GPS 上传端口持久化
- 103. 更正 AIS ATON 扩展名特性(FS#886)
- 104. 就 Class 更正 AIS Msg24 处理 (FS#902)
- 105. 更正一些可翻译字符串, 如 "In, Out"
- 106. 启用航线航路点上下文菜单(FS#926)
- 107. 允许 NMEA 信息回波到接受端口
- 108. 提高仪表板插件仪器标签小字型渲染
- 109. 提高 AIS Area Notice Query 显示, 感谢 Roland
- 110. 更正已拼接的光栅海图下 course-up 模式计算
- 111. 允许航向向上"zero" (Course-Up"zero") 模式平均显示周期
- 112. 添加“选项->连接”(options->Connections) 中单个数据流“激活/禁用”(Enable/Disable) 复选框
- 113. 更正 NMEA0183 Ver 2.3 RMC 信息解析
- 114. 提高数据流过滤器定义和内容语言
- 115. 更正仪表板升级布局持久性

Release Beta 3.1.1206

- 72. 实现应答机 AIVDO (ownership) 信息支持
- 72. 更正一些设备的 gpsd 接口, 即 BU353
- 73. 更正因稀疏区覆盖率导致的 BSB 海图偶然性崩溃
- 74. 确保潮汐/潮汐流选择 (如果存在) 始终出现在上下文菜单中(FS#918)
- 75. 更正 NMEA 调试窗口暂停按钮初始状态
- 76. 提高 Rollover Windows 格式化 (FS#901)
- 77. 允许 UDP 输入数据流有空白地址字段, 意思是 INADDR_ANY (FS#900)

- 78. 提高 AIS 按钮“工具栏工具提示”(Toolbar Tooltips)
- 79. 提高公用航路点“航线管理”(RouteManager)开关可视逻辑
- 80. 更正航线/航迹/航路点“删除所有”(DeleteAll)逻辑,有些项可视
- 81. 提高“航线管理”(route manager)对话框航迹/航线可视性指示器(眼睛),感谢 Happy, Fabbian 和 Bob
- 82. 更正 cm93 部分数据集覆盖逻辑,特别是西半球
- 83. 更正“AIS 目标”(AIS Target)列表排序逻辑 (FS#905)
- 84. 完成 Garmin PVT 模式 GPS 接收器输入逻辑
- 85. 语言更新,2012-11-5 收到的所有内容
- 86. 激活 Windows 下 Garmin USB 设备支持
- 87. 解决 "Port" 国际化模糊问题
- 88. 迫使“B 类 AIS 目标”(Class B AIS targets)在接收 Msg18 时拥有未定义的 NavStatus (FS#905)
- 89. 更正年终潮汐/潮流流计算
- 90. 向端口过滤器添加 AIS 信息类型
- 91. 更正 Mac window 血统关系 (FS#863)

Release Beta 3.1.1105

- 56. 为 ENC 锚详情添加 A 热键. 感谢 nkiesel
- 57. 提高 AIS 船舶 UNC、锚、RAM 等图标,感谢 jesperwe
- 58. 更正 GRIB 插件信息文本.(FS#888)
- 59. 将上下文菜单调整为“集中式上下文菜单”("Focused Context Menus") JesperWe
- 60. 扩展 cm93 覆盖到北纬 80 度
- 61. 启动 S57 ENC 覆盖网格区域所有对象 query 操作
- 62. 完善本船尺寸调整对话框,包括 GPS 接收器偏移
- 63. 完成“仪表板插件图形”(Dashboard PlugIn graphics)的重新设计,感谢 sethdart
- 64. 完成新的智能动态追踪算法
- 65. 更正倾斜光栅海图 AIS 目标渲染
- 66. 实现新的 DaChangelog
- 67. 扩展 PlugIn API Version 1.9 以包括一些制图功能
- 68. 通过有细微差别的颜色渲染 Euro Inland AIS 目标 ("Teal")
- 69. 向“选项->用户界面”(Options->User Interface)添加“确认目标删除”(ConfirmObjectDeletion)复选框
- 70. 将 OpenGL 结构转换成新形式 (wx2.9+)
- 71. 为 Mac Retina 显示 (FS#868)数据源架构提供基本支持

Release Beta 3.1.919

- 54. 更正选项 (Options) 对话框调用故障
- 55. 更正 CMake Mac 建制指令中 typo 以包含 styles in bundle.

Release Beta 3.1.915

- 44. 更正 wxWidgets 2.8.10 光标图像
- 45. 提高状态栏不同屏幕尺寸可读性
- 46. 允许用户独立设置共用 WPTs 可视性
- 47. 扩展鼠标所停位置 S57 照明区
- 48. 更正拼接海图中小比例尺 ENC 未覆盖 (ENC "NoCoverage") 区域.
- 49. 重做 Canvas 弹出菜单以便更好管理.
- 50. 为“仪表板插件”(Dashboard PlugIn)添加新指令
- 51. 实现“选项”(Options)对话框新的布局.
- 52. 提高 Furuno route 上传逻辑.

53. 更正 cm93 旋转过的单一海图的渲染

Release Beta 3.1.814

- 28. 更正以进行多个潮汐/潮汐流数据集加载
- 29. 实现自动/动态大圆航行航线创建
- 30. 更正已旋转的世界海图渲染
- 31. 实现多个部分 cm93 数据集支持
- 32. 完善 KML 对象加强版复制/粘贴操作.
- 33. 实现与航线线路优化程序相关的特定界面
- 34. 改变 GPX 航路点导出原则, 是航线中的航路点作为单独项存在
- 35. 添加 UI 对话框 允许用户对 MOB 图标可视性进行选择
- 36. 提高单一海图模式动态平移/缩放
- 37. 各种 UI 增强功能
- 38. 实现“确认”(“confirm”)对话框删除共用航路点
- 39. 实现航路点创建/移动/删除的撤销/重做功能(ctrl-Z, ctrl-Y)
- 40. 更正不可视航迹片段上下文菜单可选性(FS#866)
- 41. 实现 OpenGL 海图打印 (FS#846)
- 42. 为海图选择栏可视性添加热键 (ctrl-B)
- 42. 修改海图右键上下文菜单以添加子菜单
- 43. 提高“AIS 目标 Query”(AIS Target Query)对话框显示模式

Release Beta 3.1.802

- 9. 提高潮汐/潮汐流管理器和工具栏界面以允许多个并发 HARMONIC 数据集
- 10. 实现工具栏 (Toolbar) 图标可视性选择对话框. jesperwe
- 11. 更正一些插件工具条图标故障, 特别是 WMM 插件
- 12. 当缩小时, 显示本船 COG 和 HDT 向量(FS#842)
- 13. 设置“传统显示风格”(Traditional) 为默认的更新和重新安装模式。
- 14. 扩展插件信息方案
- 15. 如果没有其他方式, 允许 wmm_pi PlugIn 支持系统的磁差
- 16. 更未知 BSB 海图投影处理
- 17. 为 gentoo linux 添加建构原则(FS#853)
- 18. 实现航路点属性窗口及 JumptoPosition 对话框经纬度复制粘贴. JesperWe
- 19. 实现航路点、航线和航迹双击 query. JesperWe
- 20. 实现 AIS 目标真实尺寸显示选择
- 21. 实现二进制 (.tcd) 潮汐/潮汐流数据集支持
- 22. 提高极小型视窗 UI
- 23. 实现新世界海图最为小比例尺背景
- 24. 为 GPX 航线导出设置默认文件名
- 25. 更正依据附近航路点创建航线中出现的错误
- 26. 通过内置 ISO8559 字符更正一些 S57 ENC 处理
- 27. 完善 tidRelease Beta 3.1.802e/潮汐流扩展对话框显示

Release Beta 3.1.714

- 1. 人工合并 3.0.2
- 2. 实现 UI Theme Traditional 和两个新主题的支持, 感谢 Jesperwe!
- 3. 允许用户对状态栏字型进行选择. jesperwe.
- 4. 在航线旋转 (route rollover) 弹出窗口中显示航线总长度. jesperwe

5. 实现航线属性 (Route Properties) 和航迹 (Tracks) 对话框中 "copy-to-clipboard". jesperwe
6. 允许在航线属性 (Route Properties) 对话框中编辑航路点. jesperwe
7. 为 AIS 目标显示动态菜单以显示/隐藏 CPA I 线. 感谢 tom-r
8. 将 CPA/TCPA 添加到 AIS 目标列表中, 感谢 tom-r

2. 开发人员指南

先决条件

以下内容以假设你可以通过源代码构建 OpenCPN, 并且拥有必备工具为前提。访问

http://opencpn.org/compiling_source_linux 和 http://opencpn.org/compiling_source_windows 找到详细指南。

Git magic

如果你按照上述指令操作, 现在你电脑上会克隆有本地存储库及所有必须工具。打开一个终端 (运行 cmd.exe), 进入你克隆源代码的目录。

个人建议: 为解决许多后期问题, 要敢于使用小型单任务分支。听上去可能怪怪的, 特别是那些习惯使用传统控制系统的用户, 但这就是 git 及现代控制系统版本的用途。整合分支比挑选单个任务进行提交要简单的多。

以下列出一些必要的 Git 命令:

更新到最新上行代码 (upstream code)

`git pull origin`

查看本地存储库分支

`git branch`

创建新分支

`git branch [NAME]`

进入另一个分支 (在进入另一个分支前别忘了提交或重置更改, 否则 Git 不会丢掉你的工作及用另一个分支覆盖你的工作)

`git checkout [NAME]`

删除分支

`git branch -D [NAME]`

将某个文件添加到当前分支

`git add [filename.ext]`

将某个文件从当前分支删掉

`git rm [filename.ext]`

将你的本地修改提交到当前分支

`git commit -a -m "My commit message text"`

丢掉最新代码更改, 回到上次提交的状态

`git reset --hard`

注意: 这不是 Git 手册。想要知道如何使用 Git, 打开 <http://book.git-scm.com/> 访问 Git book 或其他网络教程。

代码格式化

在编写 OpenCPN 代码前请阅读[代码格式化](#)页面中的信息

提交代码

为确保代码能正确地提交到 OpenCPN, 首先要创建一个 OpenCPN GitHub 主存储库, 然后把所有提交放到克隆存储库内。当 Github 中一切准备就绪, 你可以发送邮件给开发人员或者在 <http://opencpn.org/flyspray/> 上对你的工作进行描述并参考 Github 提交 URL。

官方发布的开发情况参考: <https://github.com/OpenCPN/OpenCPN>

创建补丁

假设你已经签出分支并在自己的工作目录里面, 运行:

`git diff master > mystuff.patch`

在新创建的为 `mystuff.patch` 的文件中，你可以预览准备通过文本编辑器进行提交的代码。

提价补丁

创建 Flyspray 账户: <http://opencpn.org/flyspray/>

Bugfix:

将补丁及其评论提交到 <http://opencpn.org/flyspray/index.php?project=3> 中 2.2.x 软件缺陷追踪系统 (bugtracker) 对应问题区

新功能:

在 <http://opencpn.org/flyspray/index.php?project=6> “功能请求追踪系统” (Feature requests tracker) 内提交补丁及其评论。

2.1 编译 – Linux

本节内容主要针对 Debian/Ubuntu & Fedora/CentOS 发布的指令，但进行简单的调整后可以用于其他 Linux 系统、BSD 及

Solaris。

1. 必要程序

Debian / Ubuntu

```
sudo apt-get install libgtk2.0-dev gettext git-core cmake gpsd gpsd-clients libgps-dev build-essential wx-common  
libwxgtk2.8-dev libglu1-mesa-dev libgtk2.0-dev wx2.8-headers libbz2-dev libtinyxml-dev libportaudio2 portaudio19-dev
```

Fedora / CentOS

```
yum install git cmake rpm-build gcc-c++ libstdc++-devel gtk2-devel wxGTK-devel mesa-libGLU-devel mesa-libGL-devel  
gettext bzip2-devel portaudio-devel
```

OpenSUSE

```
zypper install git cmake gcc-c++ libstdc++-devel gtk2-devel wxWidgets-wxcontainer-devel gettext-runtime gettext-tools  
libbz2-devel gpsd-devel portaudio-devel tinyxml-devel rpm-build
```

注意: late betas and 3. 必须安装 libportaudio2 和 portaudio19-dev (Ubuntu), 可能叫做 libportaudio 和 libportaudio-dev, 或在其他 distros 上有类似叫法。

警告: “谷歌地球” (Google Earth) 在没有做任何说明的情况下就移走了 portaudio19-dev。OpenCPN 会进行编译但是“声音” (sound) 不能发挥作用。

Cmake 版本必须高于 2.8, 也就是说 Ubuntu 10.4 或后续版本 (启用 backport 存储库的 9.10), 或者 Fedora 13 或后续版本。

Cmake 重点注意事项: 因为 Cmake 中的软件缺陷 "pixbuf-bug", 建议将升级到等于或高于 Cmake 2.8.3.

Fedora 14 已经进行升级, 在 Ubuntu 10.10 上需要启动补丁库。选择 “Synaptic -> 设置->库->不支持更新” (Synaptic->Settings->Repositories->Updates>Unsupported updates) (maverick-backports)

注意这个安装包会安装大量其他依赖项

选择性操作: 删除旧安装

只有在 opencpn .deb / .rpm 已事先安装好的情况才要求

Debian / Ubuntu

```
sudo apt-get remove opencpn
```

Fedora / CentOS

```
yum remove opencpn
```

OpenSUSE

```
zypper remove opencpn
```

2. 下载源代码

运行以下命令对源代码进行本地复制:


```
git clone git://github.com/OpenCPN/OpenCPN.git
```

运行以下命令更新现有本地复制:

```
cd opencpn
```

```
git pull
```

3. 构建 OpenCPN

重要通知, 从 2.6.1718 测试版开始有效

开发人员注意: libgps19 和 ..20.之间 libgps API 变化很大。我们要求使用 libgps20 API 及包含文件进行 OpenCPN 的构建。如果无法找到这些内容会出现“版本偏斜”(version skew)警告。如果是通过 libgps19 包含文件构建的 OpenCPN, libgps 界面无法正常工作。

我们在子目录里创建一切所需内容以保证代码库整洁(易于捕捉变化)

```
cd opencpn # unless already in this directory.
```

```
mkdir build
```

```
cd build
```

```
cmake ../
```

```
make
```

注意:

默认安装目录为/usr/local, 你可以通过提供合适的 cmake 选项(Cmake appropriate option)进行修改:

```
cmake -DPREFIX=/usr ../
```

只有首次构建源代码时才会用到 Cmake, 然后运行 make 就足够了, 即使你更新了源代码。

****Ubuntu 10.10 fails to build...**

Change the path in this line in CMakeCache.txt: `GTK2_GTK_INCLUDE_DIR:PATH=/usr/include/gtk-2.0`

4. 安装或创建一个包

```
sudo make install
```

或者, 你可以创建自己的 .deb 和 .rpm 包(要求 Cmake 版本高于 2.8.2):

在构建目录中运行

```
sudo make package
```

5. Linux 系统 OpenCPN IDEs

你可以使用各种 IDEs 对 Linux 上 OpenCPN 代码进行编辑, 选择一种 **Code::Blocks**, **KDevelop** 及 **Eclipse CDT** 用作 Cmake 支持以生成项目文件。为此, 选择以下任一种与你所选的 IDE 相应的方式替换现有 cmake 配置步骤。不管采用哪种方式你还是可以通过上述方式进行命令行构建。

```
cmake -G "CodeBlocks - Unix Makefiles" ../
```

```
cmake -G "KDevelop3 - Unix Makefiles" ../
```

```
cmake -G "Eclipse CDT4 - Unix Makefiles" ../
```

关于该应用程序的 GUI 部分原型, 请访问: wxFormBuilder.

6. 其他

简化 Git/Cmake 进程脚本

```
#!/bin/sh
```

```
#Change this line to where you want the OpenCPN source on your computer.
```

```
cd /home/thomas/Testing/GitOpenCPN
```

```
GIT=0
```

```
test -d opencpn
```

```
if test $? -eq 1
```

```
then
    git clone git://opencpn.git.sourceforge.net/gitroot/opencpn/opencpn
    GIT=1
fi
cd opencpn
if test $GIT -eq 0
then
    git pull
fi
test -d build
if test $? -eq 1
then
    mkdir build
fi
cd build
rm -f CMakeCache.txt
cmake ../
make
echo "Cmake OK!"
sudo make install
exit
```

编译老版本

CVS 上的老式操作方式不再保留

```
cvs -z3 -d:pserver:anonymous@opencpn.cvs.sourceforge.net:/cvsroot/opencpn co -P opencpn
```

早期版本通过以下基本命令运用 gnu 安装工具链 (automake toolchain):

```
aclocal
automake --add-missing
autoconf
./configure
make
```

进行安装

2.1.1 在 Motorola Xoom 上的应用

入门提醒: 尽管这篇文章主要介绍的是 Motorola Xoom, 但是很有可能只要你的设备使用的是 armel 架构, 你就可以通过该指南在该设备上安装 Linux 和 OpenCPN。这其中的区别主要还是引导装载程序的解锁方式 (root 该设备)。如果你无法找到 root 设备的方法: Android & Windows Phone: Tablets, Apps, & ROMs @ xda-developers, 你可以通过在浏览器中输入 root 或 rooting 及设备模型找到解决办法。

一些设备生产商比如 Motorola 就用使用不会产生 NMEA 语句的 GPS 芯片。一种可以从 Google Play 免费下载的叫做 GPS Share WIFI 的.apk 允许用户通过这类内置 GPS 与 OpenCPN 进行“交流”。除了以下要介绍的 Turbo GPS 外, 拥有这些可产生 NMEA 语句的内置 GPS 的用户大多可以使用该.apk。

自 2013 年 5 月 12 号开始, 我们可以通过以下方法将可产生 NMEA 语句及无法产生 NMEA 语句的内置 GPS 与 OpenCPN 相

连接。

1. 只产生 NMEA 语句的内置 GPS，可以从 Google Play 免费下载'Turbo GPS'，
<http://www.turboirc.com/michael/>
2. 可以从 Google Play 免费下载 'GPS Share WIFI'。不能产生 NMEA 数据的 GPS 芯片必须要用 'GPS Share WIFI'而能产生 NMEA 数据的可能需要使用。你唯一需要的就是一个可用作热点的 Android 设备。不需要外部设备或者网络供应商。因此，该.apk 使得航行过程黄总海员可以在 OpenCPN 上使用内置 GPS。

外部蓝牙 GPS:

1. 除 Linux 发烧友外，对于很多想了解在 OpenCPN 上使用蓝牙 GPS，想利用蓝牙 GPS 进行 kernel 编译、创建和编辑配置文件但是又不知道如何通过 Linux 完成这些操作的用户来说，
<http://www.cruisersforum.com/forums/f134/opencpn-install-on-android-tabl...> 有具体操作步骤（帖子 #162~#167）。
2. USB 相关问题还有待解决。

Rooting 你的设备

访问 XDA 网站了解相关信，网址会在下文给出。注意这里很容易出错！

如果你拥有任何版本的摩托罗拉 Xoom，你可以从这里开始：[\[Root\] Universal Xoom Root - ANY XOOM ANY UPDATE - xda-developers](#)

注意：从现在开始需要确保好的网络连接，网速要快。

你需要一台台式电脑和一根 USB 数据线以便将台式电脑与该设备进行连接。

将 XDA instructions 中提到的文件下载、安装到平板电脑和台式电脑上。点击平板电脑上的谷歌市场（Google Play）图标，使用搜索引擎找到文件。台式电脑上安装的文件是从 XDA 上下载的。

按照说明一步步操作。

一旦成功 rooted 笔记本，我们要开始第一步操作：

第一步：安装 Linux, Android 终端仿真程序及 akRDCFree VNC 查看器。

现在我们还无法直接用 Linux 替换 Android。安装工作已经取得一半成功，但是由于一些问题长期困扰专有驱动，有些硬件无法使用。如果你不需要声音，无线等，你可能想试一试。尽管如此，风险依然很大。

建议你从“谷歌市场”（Google Play）下载 Linux Deploy（稀疏指令 <https://github.com/meefik/linuxdeploy>）。鉴于我的成功经验，我建议使用该安装程序。我并未用过 Complete Linux Installer" 或 Galoula's "Linux Installer"。除此之外还有其他解决办法，但要求熟悉 Linux。

注意：在使用 Linux Deploy 过程中我遇到一个小问题，其 GUI 更新功能并没有做好本职工作反而覆盖我之前对 apt sources.list 文件 (/etc/apt/sources.list) 的任何修改，从而限制我访问各种储存库。解决的办法就是在使用终端执行 apt 和 aptitude 更新、升级和程序/文件安装前，更改 sources.list 文件。最简单的方法是通过 Google Play 的 Jota+ 记录你所选的 Debian 存储库，通过 source.list 文件中前面有 # 的各行注释“ru”存储库，从 Jota+ 复制然后粘贴到你所选的存储库中的 sources.list 文件中 (/etc/apt/sources.list)。

第二步：通过 Google Play 安装 Android 终端仿真程序及 akRDCFree VNC 查看器

该终端用于下载和安装文件依赖项和 OpenCPN。

而查看器则是访问 Linux 桌面必不可少的。

注意不同查看器功能差别很大。之所以在使用 OpenCPN 时选择 VCN 查看器主要是出于鼠标的考虑。其次就是是否包括桌面缩放功能。akRDCFree 和 free bVNC 有这一功能，但是只有 free bVNC 有“捏变焦功能”（pinch and zoom）。我试用过谷歌市场上所有查看器，akRDCFree 鼠标功能最好。如果其放大/缩小功能为捏变焦，它应该是最佳选择，但其+和-也很好用。我首选拥有捏变焦功能的 free bVNC 及模仿触摸板，但总是运行不畅。也许你会有不同的体验，所以不妨一试。

第三步：使用谷歌市场上的 Linux Deploy，安装发行版和桌面环境

在我试图安装组合而非 Debian XFCE 时遇到了问题，所以这次我建议安装 Debian Squeeze 或 Wheezy 及 XFCE 窗口管理程

序。这些 Debian 版本在安装 OpenCPN 和 zyGrib 时都非常成功。通过 Debian Squeeze 使用存储库中的 libqtcore4, qwt-6.0.1 和 zyGrib-6.1.0 源代码。Wheezy 则通过 apt-get 存储库的 qwt 和 qt (per Cruisers Forum member sevanopula) 进行安装。编译和安装 zyGrib version 5.1.4.。我建议创建一个文件循环系统，大小为 **8 gigabytes!** 因为你在下载很多开发文件或实用工具时很可能要用到多个窗口管理程序并且会选择在 Linux 系统上安装 nav.海图。否则，你可以将它们安装在微型 SD 卡上以释放更多内存。

按照 Linux Deploy GUI 上面的指南，创建文件循环系统，安装发行版及窗口管理器。

一旦完成安装，按照以下步骤挂载文件循环系统

打开安卓终端

在提示符后面键入“su”

按下 Enter 键

复制/粘贴或者键入: /data/local/linux/bin/linuxdeploy shell

按下 Enter 键

创建 root 密码:

在终端提示符键入

passwd root

按下 Enter 键

输入不会忘记的密码! 按要求再次输入密码。从 linux vnc 桌面登录该终端时需要输入该密码。

这样，就完成 Linux 文件系统的挂载工作，现在就可以登录了。

想要查看文件系统架构，使用谷歌市场中任何一个文件管理器。然后进入/data/local/linux/mnt。浏览文件系统和在内部终端执行命令时的任何操作都要非常小心。这时你一直在 root 环境下工作，所以很容易破坏安装! 注意你可以通过文件管理器创建目录，复制、移动、重命名和删除文件。剩下的工作需要在内部终端完成。

第四步：安装依赖项及开发文件

有一系列 OpenCPN 依赖项需要下载和安装到你的 Linux 系统中。如果是 DEBIAN SQUEEZE，把下面的一系列文件复制粘贴到终端命令提示符后。

```
apt-get install libglu1-mesa libgl1-mesa-glx zlib1g bzip2 gpsd gpsd-clients xcalib libgtk2.0-dev gettext git-core cmake  
libgps-dev build-essential wx-common libglu1-mesa-dev libgtk2.0-dev libbz2-dev libtinyxml-dev libwxgtk2.8-0  
libwxgtk2.8-dev libwxbase2.8-0 libwxbase2.8-dev libsdl1.2debian-all debhelper devscripts dh-make dpkg-dev fakeroot  
lintian diff patch libtinyxml* libportaudio2 portaudio19-dev
```

按下 Enter 键

注意：如果你选择安装其他 Linux 系统而非 DEBIAN SQUEEZE，请看以下内容：
http://opencpn.org/ocpn/compiling_source_linux。确保你安装了依赖项!! 如果你在编译 OpenCPN 时出现错误，不要尝试其他 Linux 版本，到 OpenCPN 论坛寻求帮助!

这些可能会有它们自己的依赖项，所有下载和安装的文档比上面列出的文档要多。

通过在内部终端键入以下内容来添加 '/home'目录:

```
cd /home
```

按下 Enter

键入:

```
mkdir Downloads
```

按下 Enter

键入:

```
cd Downloads
```

按下 Enter

确保你的计算机仍然与网络保持连接。

自 2013 年 5 月 8 日起,最新版的 OpenCPN, 3.2.2 已经完成编译并在 Debian Squeeze 的 Xoom 和 Ubuntu Precise 的 Nexus 10 上成功安装。访问 Cruisers Forum 的 ‘OpenCPN install on android tablets easier now’ 查看其在其他设备上的安装。

复制并粘贴到终端:

```
git clone -b release32_stable https://github.com/OpenCPN/OpenCPN.git
```

按下 Enter

直到停止下载, 然后进行第四步。 (第五步?)

第四步: 配置和安装 OpenCPN

假定你已经通过以下命令加载好 Linux 系统, 在内部 Android 终端使用: "/data/local/linux/bin/linuxdeploy shell", 在提示符后键入以下命令:

```
cd /home/Downloads/OpenCPN
```

按下 Enter

在提示符处进行复制和粘贴:

```
cmake -DPREFIX=/usr -D CMAKE_VERBOSE_MAKEFILE:BOOL=TRUE /home/Downloads/OpenCPN
```

按下 Enter

注意: 如果在本部分编译和安装过程中遇到错误, 一般是 OpenCPN 安装说明书(installation instructions)寻找 /usr/lib 目录下的文件库——校正 i386 架构的电脑。然而, 使用 ARM 架构的安卓平板电脑中的库可能存在于 /usr/lib/arm-linux-gnueabi 目录下。在本文档的结尾处我给出了 CmakeCache.txt 样本文件以便帮助你 (如果有必要的话) 在文本编辑器中编辑 CmakeCache.txt 文件。或者, 你可以将 -i 添加到 cmake -DPREFIX=/usr -D CMAKE_VERBOSE_MAKEFILE:BOOL=TRUE 命令中, 就像 cmake -i -DPREFIX=/usr -D CMAKE_VERBOSE_MAKEFILE:BOOL=TRUE, 当问及是否想要查看高级项时回答 Yes。

提供最新没有出错且可以执行到下一命令的命令,

```
make
```

按下 Enter

如果 make 继续到 100%

```
make install
```

按下 Enter

现在就完成了 OpenCPN 的安装工作, 我们可以进行使用了。

使用 Linux Deploy GUI 挂载循环文件, 选择开始图标 (start icon), 启动 SSH 和 VNC 服务器。想要解除挂载该系统, 选择暂停按钮。开始 akRDCFree 或其他 vnc 查看器。

开始 akRDCFree

确保按如下信息设置:

Name: "new Server" or any other name you wish to use,

IP Address: "127.0.0.1 or localhost",

Port: "5900",

注意: Linux Deploy 下面的有关指令有可以更改的预设 SSH 和 VNC 密码: "changeme"

Authentication: vnc password. "changeme"

VNC server type: "Autodetect".

Operating system type: "Other".

Linux Deploy 只是要求用户在安装程序内开始（点击播放图标）和停止（点击方形图标）该进程。当然，VNC 查看器的设置参考上面截图。

连接到窗口管理器，启动 OpenCPN

在 VNC 查看器上点击“连接”（connect）。你应该可以看到 XFCE 桌面。这样你很有可能会在 Education 下的主菜单上找到 OpenCPN。按下 Enter 启动该程序。

如果启动 OpenCPN 失败，以 root 用户身份运行此程序。

打开终端，以 root 身份登录，输入 opencpn. Press enter. Enjoy

更多信息请访问 OpenCPN 官网：<http://opencpn.org/ocpn/>

包含建议更改内容的 **CmakeCache.txt** 样本文件。本文件中有很多行因为是不可修改的而被隐藏起来了。它不会影响编译和安装，也不涉及 path 语句：

```
# This is the CMakeCache file.
# For build in directory: /home/android/Downloads/OpenCPN
# It was generated by CMake: /usr/bin/cmake
# You can edit this file to change values found and used by cmake.
# If you do not want to change any of the values, simply exit the editor.
# If you do want to change a value, simply edit, save, and exit the editor.
# The syntax for the file is as follows:
# KEY:TYPE=VALUE
# KEY is the name of a variable in the cache.
# TYPE is a hint to GUI's for the type of VALUE, DO NOT EDIT TYPE!.
# VALUE is the current value for the KEY.
```

```
#####
```

```
# EXTERNAL cache entries
```

```
#####
```

```
//Path to a file.
```

```
BZIP2_INCLUDE_DIR:PATH=/usr/include
```

```
//Path to a library
```

```
BZIP2_LIBRARIES:FILEPATH=/usr/lib/arm-linux-gnueabi
```

```
//Path to a program.
```

```
CMAKE_AR:FILEPATH=/usr/bin/ar
```

```
//CXX compiler.
```

```
CMAKE_CXX_COMPILER:FILEPATH=/usr/bin/c++
```

```
//C compiler.
```

```
CMAKE_C_COMPILER:FILEPATH=/usr/bin/gcc
```

```
//Install path prefix, prepended onto install directories.
```

CMAKE_INSTALL_PREFIX:PATH=/usr

//Path to a program.

CMAKE_LINKER:FILEPATH=/usr/bin/ld

//Path to a program.

CMAKE_MAKE_PROGRAM:FILEPATH=/usr/bin/make

//Path to a program.

CMAKE_NM:FILEPATH=/usr/bin/nm

//Path to a program.

CMAKE_OBJCOPY:FILEPATH=/usr/bin/objcopy

//Path to a program.

CMAKE_OBJDUMP:FILEPATH=/usr/bin/objdump

//Value Computed by CMake

CMAKE_PROJECT_NAME:STATIC=OpenCPN

//Path to a program.

CMAKE_RANLIB:FILEPATH=/usr/bin/ranlib

//Path to a program.

CMAKE_STRIP:FILEPATH=/usr/bin/strip

//Path to a file.

FREETYPE_INCLUDE_DIR_freetype2:PATH=/usr/include/freetype2

//Path to a file.

FREETYPE_INCLUDE_DIR_ft2build:PATH=/usr/include

//Path to a library.

FREETYPE_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi

//Path to a program.

GETTEXT_MSGFMT_EXECUTABLE:FILEPATH=/usr/bin/msgfmt

//Path to a program.

GETTEXT_MSGMERGE_EXECUTABLE:FILEPATH=/usr/bin/msgmerge

//Path to a program.

GETTEXT_XGETTEXT_EXECUTABLE:FILEPATH=/usr/bin/xgettext

//Path to a file.

GTK2_ATK_INCLUDE_DIR:PATH=/usr/include/atk-1.0/atk

//Path to a library.

GTK2_ATK_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi/libatk-1.0.so

//Path to a file.

GTK2_CAIRO_INCLUDE_DIR:PATH=/usr/include/cairo

//Path to a library.

GTK2_CAIRO_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi/libcairo.so

//Path to a file.

GTK2_FONTCONFIG_INCLUDE_DIR:PATH=/usr/include

//Path to a file.

GTK2_GDKCONFIG_INCLUDE_DIR:PATH=/usr/lib/arm-linux-gnueabi/gtk-2.0/include

//Path to a file.

GTK2_GDK_INCLUDE_DIR:PATH=/usr/include/gtk-2.0

//Path to a library.

GTK2_GDK_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi/libgdk-x11-2.0.so

//Path to a file.

GTK2_GDK_PIXBUF_INCLUDE_DIR:PATH=/usr/include/gdk-pixbuf-2.0

//Path to a library.

GTK2_GDK_PIXBUF_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi

//Path to a file.

GTK2_GLIBCONFIG_INCLUDE_DIR:PATH=/usr/lib/arm-linux-gnueabi/glib-2.0/include

//Path to a file.

GTK2_GLIB_INCLUDE_DIR:PATH=/usr/lib/arm-linux-gnueabi/glib-2.0/include

//Path to a library.

GTK2_GLIB_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi

//Path to a file.

GTK2_GOBJECT_INCLUDE_DIR:PATH=/usr/include/glib-2.0

//Path to a library.

GTK2_GOBJECT_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi

//Path to a file.

GTK2_GTK_INCLUDE_DIR:PATH=/usr/include/gtk-2.0

//Path to a library.

GTK2_GTK_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi/libgtk-x11-2.0.so

//Path to a file.

GTK2_PANGO_INCLUDE_DIR:PATH=/usr/include/pango-1.0/pango

//Path to a library.

GTK2_PANGO_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi

//Path to a file.

OPENGL_INCLUDE_DIR:PATH=/usr/include

//Path to a library.

OPENGL_gl_LIBRARY:FILEPATH=/usr/lib/libGL.so

//Path to a library.

OPENGL_glu_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi/libGLU.so.1.3.08005

//Path to a file.

OPENGL_xmesa_INCLUDE_DIR:PATH=OPENGL_xmesa_INCLUDE_DIR-NOTFOUND

//Value Computed by CMake

OpenCPN_BINARY_DIR:STATIC=/home/android/Downloads/OpenCPN

//Value Computed by CMake

OpenCPN_SOURCE_DIR:STATIC=/home/android/Downloads/OpenCPN

注意：即便 **portaudio** 已经安装到你的平板电脑上，以下内容以外的任何其他内容都有可能产生错误并阻止进一步的编译和安装！

//Path to a file.

PORTAUDIO_INCLUDE_DIR:PATH=PORTAUDIO_INCLUDE_DIR-NOTFOUND

//Path to a library.

PORTAUDIO_LIBRARY:FILEPATH=PORTAUDIO_LIBRARY-NOTFOUND

//No help, variable specified on the command line.

PREFIX:UNINITIALIZED=/usr

//Path to a file.

TINYXML_INCLUDE_DIR:PATH=/usr/include

//Path to a library.

TINYXML_LIBRARIES:FILEPATH=/usr/lib/libtinyxml.so

//Enable Garmin Host Mode support

USE_GARMINHOST:BOOL=ON

//Enable S57 ENC support

USE_S57:BOOL=ON

//Path to a file.

X11_ICE_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_ICE_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libICE.so

//Path to a file.

X11_SM_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_SM_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libSM.so

//Path to a file.

X11_X11_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_X11_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libX11.so

//Path to a file.

X11_XShm_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

X11_XSync_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

X11_Xaccessstr_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

X11_Xau_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xau_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXau.so

//Path to a file.

X11_Xcomposite_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xcomposite_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXcomposite.so

//Path to a file.

X11_Xcursor_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xcursor_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXcursor.so

//Path to a file.

X11_Xdamage_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xdamage_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXdamage.so

//Path to a file.

X11_XdmcP_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_XdmcP_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXdmcP.so

//Path to a library.

X11_Xext_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXext.so

//Path to a file.

X11_Xfixes_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xfixes_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXfixes.so

//Path to a file.

X11_Xft_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xft_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXft.so

//Path to a file.

X11_Xi_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xi_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXi.so

//Path to a file.

X11_Xinerama_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xinerama_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXinerama.so

//Path to a file.

X11_Xinput_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xinput_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXi.so

//Path to a file.

X11_Xkb_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

X11_Xkblib_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

X11_Xlib_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

X11_Xrandr_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xrandr_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXrandr.so

//Path to a file.

X11_Xrender_INCLUDE_PATH:PATH=/usr/include

//Path to a library.

X11_Xrender_LIB:FILEPATH=/usr/lib/arm-linux-gnueabi/libXrender.so

//Path to a file.

X11_Xshape_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

X11_Xutil_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

X11_dpms_INCLUDE_PATH:PATH=/usr/include

//Path to a file.

ZLIB_INCLUDE_DIR:PATH=/usr/include

//Path to a library.

ZLIB_LIBRARY:FILEPATH=/usr/lib/arm-linux-gnueabi/libz.so

//Value Computed by CMake

dashboard_pi_BINARY_DIR:STATIC=/home/android/Downloads/OpenCPN/plugins/dashboard_pi

//Dependencies for the target

dashboard_pi_LIB_DEPENDS:STATIC=general;-pthread;general;-lw_x_baseu-2.8;general;-lw_x_gtk2u_core-2.8;general;-lw_x_b
aseu_net-2.8;general;-lw_x_baseu_xml-2.8;general;-lw_x_gtk2u_html-2.8;general;-lw_x_gtk2u_adv-2.8;general;-lw_x_gtk2u_aui
-2.8;general;-lw_x_gtk2u_gl-2.8;

//Value Computed by CMake

dashboard_pi_SOURCE_DIR:STATIC=/home/android/Downloads/OpenCPN/plugins/dashboard_pi

//Value Computed by CMake

grib_pi_BINARY_DIR:STATIC=/home/android/Downloads/OpenCPN/plugins/grib_pi

//Dependencies for the target

grib_pi_LIB_DEPENDS:STATIC=general;-pthread;general;-lw_x_baseu-2.8;general;-lw_x_gtk2u_core-2.8;general;-lw_x_baseu_n
et-2.8;general;-lw_x_baseu_xml-2.8;general;-lw_x_gtk2u_html-2.8;general;-lw_x_gtk2u_adv-2.8;general;-lw_x_gtk2u_aui-2.8;ge
neral;-lw_x_gtk2u_gl-2.8;general;/usr/lib/arm-linux-gnueabi;general;/usr/lib/arm-linux-gnueabi/libz.so;

//Value Computed by CMake

grib_pi_SOURCE_DIR:STATIC=/home/android/Downloads/OpenCPN/plugins/grib_pi

//Path to a program.

wxWidgets_CONFIG_EXECUTABLE:FILEPATH=/usr/bin/wx-config

//Use debug build?

wxWidgets_USE_DEBUG:BOOL=OFF

//Link libraries statically?

wxWidgets_USE_STATIC:BOOL=OFF

//Use unicode build?

wxWidgets_USE_UNICODE:BOOL=ON

//Use universal build?

wxWidgets_USE_UNIVERSAL:BOOL=OFF

//Path to a program.

wxWidgets_wxrc_EXECUTABLE:FILEPATH=/usr/bin/wxrc

2.2 编译 - Mac OS X

1. 安装 Mac 开发工具 (Mac Developer Tools——Xcode)
2. 安装 Cmake. (如果使用 Mac 程序包安装程序, 点击安装链接 ("Install Link") 启用命令行访问.
3. 安装 Glib2 及其依赖项
4. 下载并解压 OpenCPN 源代码
5. cd opencpn (一级) 目录
6. 编辑 CMakeLists.txt 并注释掉这一行 "ADD_DEFINITIONS("`pkg-config --cflags --libs gtk+-2.0`") 因为 clang 不喜欢这一选项
7. mkdir 构建
8. cd 构建
9. cmake -G Xcode ./
10. 打开 Xcode。通过文件->打开 (File->Open), 打开应该已经建在 build 目录里的 OpenCPN.xcodeproj
11. 选择 "opencpn" 激活目标及 "i386" 激活架构
12. 编辑源文件 tri.c 并将内联声明从 function 中移走
13. 点击 "Build & Run" wxWidgets 会发出一系列的警告, 这些警告可以通过点击 "Cancel" 关闭

OSX - Lion (Work in progress)

1. 从应用商店 (App Store) 安装 XCode
2. 下载 wxWidgets-2.9.3.tar.bz2
3. 打开终端创建数据源目录:
4. mkdir ~/src
5. cd ~/src
6. tar jxf ~/Downloads/wxWidgets-2.9.3.tar.bz2
7. cd wxWidgets-2.9.3
8. mkdir macbuild
9. cd macbuild/
10. ../configure --enable-unicode --enable-debug --disable-shared --with-osx_cocoa
--with-macosx-sdk=/Developer/SDKs/MacOSX10.7.sdk --with-macosx-version-min=10.7
11. make
12. sudo make install
13. 下载 gettext-0.18.1.1.tar.gz
14. cd ~/src
15. tar zxf ~/Downloads/gettext-0.18.1.1.tar
16. cd gettext-0.18.1.1/
17. 你需要为 Lion gettext 打补丁。编辑 gettext-tools/gnulib-lib/stpncpy.c 文件, 将 "#undef stpncpy" 添加到 "#ifndef weak_alias" 前面。vim ../gettext-tools/gnulib-lib/stpncpy.c
18. mkdir build_macos
19. cd build_macos/
20. ../configure
21. make
22. sudo make install
23. 克隆 OpenCPN 源

24. 修复 2.9.4... 的一些代码问题 `git remote add balp git@github.com:balp/OpenCPN.git ; git fetch --all ; git checkout balp/mac_osx_lion`
25. `./build_mac.sh`

2.3 编译 – Windows

OpenCPN 2.6 及其后续版本 Windows 官方支持平台有 Microsoft Visual C++ 2010 Express。我们还是有可能通过 Visual Studio 2008 进行构建，但如果你想创建安装包，可能就会遇到版本控制问题。

在开始编译之前，更新 Windows。XP 系统需要 Service Pack 3。

开始编译前的重要注意事项

下列步骤的先后顺序很重要。不要跳过任何步骤，除了标明的选择性操作外，除非你自己非常清楚你的做法，否则不要改变它们的顺序。你最好把整个页面都阅读一遍以确保自己明白里面的内容，软件开发的新手尤其要注意。如果遇到任何问题，请通过我们的论坛联系我们，告诉我们你的问题所在，这样我们就可以帮助你，也能进一步完善指南，服务他人。

准备工具链

注意 VisualStudio 2012

简短的建议：不要通过它构建 OpenCPN，特别是在发布模式下，除非你真的了解下一个语句的意思。

长建议：在通过 VS 2012 进行 OpenCPN 的开发及进行调试构建时你不会遇到任何问题。如果你想在发布模式下构建和运行，情况会变得糟糕。为什么？代码库包含 wxWidgets DLLs 二进制版本，VC 2010 运行时及为简化打包和构建而预构建好的 CrashRpt 库。所有这些说明针对的是我们建议使用的 VS 2010 工具链（toolchain）。要使用 2012 版，为了能够创建安装软件包，或至少可以使 CrashRpt DLL 能够运行你的 Release 二进制文件，你需要使用针对 2012 工具链（toolchain）的版本。即便你按照前述说明进行了所有操作，你的二进制文件可能还是与官方发布的插件不兼容。所以，除了安装 2012 版本还建议安装 free VC++ 2010 Express 或至少不要将这些二进制文件给不知道 Microsoft 2010 和 2012 运行时（runtime）不兼容的普通用户。同时请不要发布通过 VS 2012 开发的插件，因为它可能不适用于使用官方 OpenCPN 安装包的用户。最后，安装 VC++ 2010 所需花费的时间比安装 VS 2012 并使其达到同等效果的时间要少很多。

VC++ 2010 Express

访问 <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-studio-express> 下载

运行安装程序，连续单击完成安装向导，OpenCPN 不需要 Silverlight 和 SQL Server，所以无需安装这两项，这样可以节省安装时间和内存。

安装完成后进行注册（Help->Register Product），否则你只有 30 天的使用时间。

VS 2010 在 XP 或 Vista 上应该运行更快：

<http://news.softpedia.com/news/Speed-Up-Visual-Studio-2010-on-XP-SP3-and-Vista-SP2-152520.shtml>

Service Pack 1

我们有必要进行一切更新。好像如果你没有安装 Visual Studio 2010 Service Pack 1，你就无法将 WX projects 转换成 VS 2010 格式。打开 Help 菜单，选择 Check for Updates。我们有必要重复进行这项操作。

选择性操作: VC++ 2008 Express Edition

如果你没有打算调试 WX 源代码，不要进行此项操作——整个这个步骤都可以跳过。如果你打算进行调试你仍然可以跳过这一步，等到将 WX projects 转换成 VS2010 格式遇到问题后再返回来进行此项操作。

访问 <http://freewareupdate.com/developer-tools/download-microsoft-visual-studio-2008-express-edition> 下载——微软不再支持 VS 2008...

运行安装程序，连续单击完成安装向导。OpenCPN 不需要 Silverlight 和 SQL Server，所以无需安装这两项，这样可以节省安装时间和内存。

Git for Windows

访问 <http://msysgit.github.com/> 下载（下载适用于 Windows 的 Git 而不是 MSysGit）

所有安装的默认设置都没有问题，以下内容除外：

* 在“Adjusting your PATH environment”处选择“Run Git from the Windows Command Prompt”

* 在“Choosing CR/LF behavior”处选择“Commit line endings as they are”。这一点非常重要，代码库使用 Unix Lf 行结

束, 提交 Cr/Lf 会使得提交内容变得很大并隐藏真实内容。如果你的 git 已经完成安装, 请编辑本地工作副本中的 .git/config 并对顶部[core]部分的 autocrlf = input 进行设置。

http://www.lostechies.com/blogs/jason_meridth/archive/2009/06/01/git-for...有很多文章就是教程。

Cmake

访问 <http://cmake.org/cmake/resources/software.html> 下载

在安装的同时, 我们建议安装程序将 Cmake 添加到系统 path 变量中。

Gettext

通过 <http://www.poedit.net/> 安装 Poedit——其 gettext 是可用版本中最好的。

将 Gettext 添加到你的 path 中——如果你安装了 Poedit, 右键我的电脑 (My Computer), 选择属性 (Properties...) 在高级页 (Advanced) 上点击环境变量 (Environment Variables) 按钮, 将 C:\Program Files\Poedit\bin 添加到 PATH 系统变量中。

NSIS (Unicode)

只有当你打算创建 OpenCPN 安装包时需要 NSIS。普通开发不需要用。

访问 <http://code.google.com/p/unsis/downloads/list> 下载。NSIS Unicode 2.46 安装文件还包含在 GIT repository -> ..\buildwin\NSIS_Unicode\nsis-2.46-Unicode-setup.exe 内。直接下载链接:

<http://code.google.com/p/unsis/downloads/detail?name=nsis-2.46.3-Unicode...>

下载和安装 NSIS。因为 Cmake 中存在一个 Bug, NSIS 只有一个安装位置 "HKEY_LOCAL_MACHINE\SOFTWARE\NSIS", Unicode 版本将其注册码 (registry key) 添加到 "HKEY_LOCAL_MACHINE\SOFTWARE\NSIS\Unicode" 中, 我们需要做一些注册调整。

将 "HKEY_LOCAL_MACHINE\SOFTWARE\NSIS\Unicode" 注册码的值 (安装路径) 复制到 "HKEY_LOCAL_MACHINE\SOFTWARE\NSIS" 中。

或者你可以直接运行批处理文件 /CopyNSISUnicodeRegKey.bat', 该文件也存在于 GIT repository -> ..\buildwin\NSIS_Unicode\COPYNSISUnicodeRegKey.bat 中。

也就是说如果你还想使用 ANSI NSIS 版本, 你首先需要按照不同版本的安装路径对 "HKEY_LOCAL_MACHINE\SOFTWARE\NSIS" 的注册码进行更改。

为了使安装包在不同语言中都有适当的名称, 我们有必要修改 X:\Program Files\NSIS\Unicode\Contrib\Language files\Norwegian.nsh 文件, 并将以下命令行

```
!insertmacro LANGFILE "Norwegian" "Norwegian"
```

更改为

```
!insertmacro LANGFILE "Norwegian" "Norsk"
```

先决条件

WxWidgets

http://www.wxwidgets.org/downloads/#latest_stable 下载 wxMSW-2.8.12-Setup.exe 并进行安装。

必要修改

打开你安装和解压 wxWidgets 的文件夹, 编辑文件 include\wx\msw\setup.h, 进行如下设置:

```
#define wxUSE_GRAPHICS_CONTEXT 1
#define wxUSE_GLCANVAS 1
```

编译 wxWidgets 方案 1: 通过命令行 (速度更快, 强烈建议采用)

启动 Visual Studio 2010 Command Prompt, 打开: cd C:\\${WXDIR}\build\msw

通过以下命令使用 makefile 进行编译:

```
nmake -f makefile.vc BUILD=release MONOLITHIC=0 SHARED=1 UNICODE=1 USE_GDIPLUS=1 USE_OPENGL=1
```


如果你想完全调试 OpenCPN, 创建 wxWidgets debug dlls

`nmake -f makefile.vc BUILD=debug MONOLITHIC=0 SHARED=1 UNICODE=1 USE_GDIPLUS=1 USE_OPENGL=1`

编译 **wxWidgets 方案 2: 通过 Visual Studio IDE (选择性操作, 不建议)**

这是一项附加操作, **99%的用户不需要做这项操作**。正如之前所说的, 如果你不熟悉 VisualStudio 和 C++ development 不要这样操作——如果只是想解决 OpenCPN 崩溃的问题, 你不必做这项麻烦的工作。如果你想构建和调试 wxWidgets, 使用 Visual Studio IDE, 打开: C:\\${WXDIR}\build\msw\config.vc, 将"UNICODE = 0"改成 **UNICODE = 1**, 将"USE_OPENGL = 0"改成 **USE_OPENGL = 1**, 将 USE_GDIPLUS = 0"改成 **USE_GDIPLUS = 1**。

将解决方案转换成 **VC++ 2010 格式**

打开: C:\\${WXDIR}\build\msw\wx_dll.dsw, 对于出现的所有项目转化的问题, 回答 Yes。

保存解决方案 (Ctrl-S), 你可以忽视警告。

如果你一直碰到"Project files cannot be loaded. error is: X:\XXX\XXX\XXX.dsp: error: Project upgrade failed."错误, 你需要按照之前的描述安装 VC++ 2008, 将该项目转换成 VS2008 格式。

将解决方案转化成 **VC++ 2008 格式**

打开: C:\\${WXDIR}\build\msw\wx_dll.dsw, 对于出现的所有项目转化的问题, 回答 Yes。

保存解决方案 (Ctrl-S)

关闭 VC++ 2008

运行 VC++ 2010

打开解决方案: (C:\\${WXDIR}\build\msw\wx_dll.sln), 对出现的所有想到转换问题回答 Yes

你可以忽视警告

保存解决方案

在 project **core** 的属性中, 为 **DLL Unicode Debug** 和 **DLL Unicode Release** 配置, 将 **GDIPlus.lib** 添加到 **Configuration Properties -> Linker -> Input -> Additional Dependencies**。

Compile Debug

从配置列表中选择 **DLL Unicode Debug**, 创建解决方案。

Odbc 无法完成编译——可以忽略, 因为 OpenCPN 不会使用

Compile Release

从配置列表中选择 **DLL Unicode Debug**, 创建解决方案。

Odbc 无法完成编译——可以忽略, 因为 OpenCPN 不会使用

使 wxWidgets 操作更简单 (建议使用)

你应该把 C:\\${WXDIR}\lib\vc_dll 添加到 path 环境变量中。完成添加后, 你应该重启和运行该程序 (cmd.exe, cmake-gui, VisualStudio 等等) 确保它们“看到”已经变化的环境变量。如果你无法确定, 重启 Windows, 一切都会完成设置。如果你没有这样做, 日后运行调试版本会遇到问题。

如果 Cmake 没有找到 wxWidgets 安装, 试着创建另一个叫做 **WXWIN** 的环境变量, 值为 C:\\${WXDIR} (比如 C:\wxWidgets-2.8.12)——它会帮助 Cmake 找到各种库。

最后, 再次提醒你不要忘了重启并运行参与构建的程序。

得到 OpenCPN 源代码

运行 cmd.exe, 或者有更好的操作方法: 如果你希望命令行中有工具链, 从开始按钮 (Start menu) → 程序 (Programs) → Microsoft Visual C++ 2008 Express Edition → Visual Studio Tools 运行 „Visual Studio 2008 Command Prompt“

第一次得到源代码, 执行:

`git clone git://github.com/OpenCPN/OpenCPN.git`

更新你之前克隆的代码, cd 源代码目录, 执行:

```
git fetch --all
```

构建 OpenCPN 源代码

在顶层代码目录下创建一个叫做 **build** 的目录

安装方案 1: 通过命令行 (建议):

更改 Build 目录, 执行:

```
cmake -G „Visual Studio 10“ ..
```

安装方案 2: 使用 Cmake-gui

从开始→程序→Cmake 2.8 (Start menu → Programs → Cmake 2.8) 运行 „CMake (cmake-gui)“

填写源代码, 创建目录

点击 Configure 按钮

如果让你选择生成器, 选择 „Visual Studio 10“

如果提示信息为红色并且生成 (Generate) 按钮无法使用, 请在此点击 Configure 按钮

忽略 GTK2_GTK_INCLUDE_DIR-NOTFOUND 和 wxwidgets_wxrc_EXECUTABLE_NOTFOUND

点击生成 (Generate) 按钮

解决方案和项目文件需要建在 build 目录下

重要建议:

使用 CMAKE GUI 工具来安装 OpenCPN 以验证 wxWidgets_LIB_DIR points 指向 {root}/lib/vc_dll 目录。这项验证非常有必要, 因为 cmake FindWxWidgets 模块有时候会找到错误源代码和/或构建配置。

编译方案 1: 通过命令行

从开始→程序→Microsoft Visual C++ 2010 Express (Start menu → Programs → Microsoft Visual C++ 2010 Express)

运行 „Visual Studio Command Prompt (2010)“

修改 build 目录

执行

```
cmake --build
```

等待完成该命令

你在 build 插件时可能会遇到问题, 因为它们在 OpenCPN 之前构建 (build) 的 (在安装过程进行修复)。只要再次执行构建 (build) 就可以修复该问题。

安装方案 2: From Visual Studio

打开 Cmake 产生的解决方案 (build/OpenCPN.sln)

编译这个解决方案或单个项目

在编译任何插件之前要先编译 OpenCPN 项目 (在配置安装中进行修复)

如果你想进行调试, 别忘了将 OpenCPN 选为开始项目 (start-up project)。如果你没有提前将 WX DLL path 添加到 path 环境变量中, 你需要将所需的 WX DLLs 复制粘贴到构建 (build) 目录中 (按照你创建的版本进行调试或发布)。可以在 C:\\${WXDIR}\lib\vc_dll 找到 DLLs。你需要:

Debug: wxbase28ud_net_vc_custom.dll, wxbase28ud_vc_custom.dll, wxbase28ud_xml_vc_custom.dll,
wxmsw28ud_adv_vc_custom.dll, wxmsw28ud_core_vc_custom.dll

Release: wxbase28u_net_vc_custom.dll, wxbase28u_vc_custom.dll, wxbase28u_xml_vc_custom.dll,
wxmsw28u_adv_vc_custom.dll, wxmsw28u_core_vc_custom.dll

(上面是调试, 下面是发行)

通过 VisualStudio 运行调试版

为了能够通过 Inside VisualStudio 运行 OpenCPN, 你需要将以下文件夹从源代码树的 data 子文件夹中复制到构建 (build) 文件夹中: **gshhs, s57data, tcddata**。你需要在构建 (build) 文件夹中创建一个 **uidata** 子文件夹, 然后将以下文件从 src\bitmaps 复制到该文件夹中: **styles.xml, toolicons_journeyman_flat.png, toolicons_journeyman.png, toolicons_traditional.png**

不安装直接运行发行版或通过 VisualStudio 运行

这是反应 2013-05-18 主分支的初步文本。注意, data 目录的复制地址与上述调试版本不同。你需要将以下文件夹从源代码树的 data 子文件夹复制到 build\Release 文件夹: **gshhs, s57data, tcddata**。你还需要创建一个 **uidata** 子文件夹, 然后把以下文件从 src\bitmaps 复制到该文件夹中: **styles.xml, toolicons_journeyman_flat.png, toolicons_journeyman.png, toolicons_traditional.png**, 同时把以下内容从 builwin\crashrpt 复制到 build\Release 中: **CrashRpt1401.dll, CrashSender1401.exe, crashrpt_lang.ini, dbghelp.dll**。

选择性操作: 创建安装包

创建 PACKAGE 项目, **opencpn_X.Y.Z_setup.exe** 在构建 (build) 目录中创建 (X 为发行数值, Y 为版本数值)

目前安装程序将 git 库中的 DLLs 加到包里。如果你想体验不同版本及 wxWidgets 库构建 (build) 设置, 在完成安装后你需要用自定义 DLL 替换这些内容。

运行调试版本

为了能够运行调试版本, 你需要在构建 (build) 目录里创建一个名为 **uidata** 的文件夹, 然后将 **src\bitmaps** 中的 **styles.xml, toolicons_traditional.png, toolicons_journeyman.png, toolicons_journeyman_flat.png** 文件复制到该文件夹内。

为了能够使用矢量海图, 你需要将源代码树上的 **s57data** 文件夹 (data/s57data) 复制到构建 (build) 目录下。否则运行的时候矢量海图支持会被关闭。

2.3.1 国际化

添加新语言

1. 想要将 OpenCPN 翻译成其他任何语言, 在这里下载 **opencpn.pot** 文件。
2. 通过 ISO 语言代码重命名该文件, 并将其扩展名 .pot 改成 .po, 如 **opencpn_it_IT.po** 代表意大利语。
3. 使用 Poedit, 或将已翻译的条目值输入到 **Msgstr** " " 中手动编辑该文件。
4. 选择适当的语言测试 OpenCPN 中的文件。在 Linux 中, 你可能还需要安装语言库 (locales) 以便可以选择所有语言。以下在基于 Ubuntu 的系统中操作的指导:

```
#To list already installed locales
```

```
locale -a
```

```
#To see a list of supported langs
```

```
less /usr/share/i18n/SUPPORTED
```

```
#To install a new locale
```

```
sudo locale-gen de_DE.UTF-8 #This is an example for German, Unicode...
```

```
#To update the list in the system
```

```
sudo update-locale
```

提高或进一步扩展现有翻译

1. 想要完善或进一步扩展现有翻译, 下载相关 .po 文件, 如 **opencpn_it_IT.po** 代表意大利语
2. 使用 Poedit, 或将已翻译的条目值输入到 **Msgstr** " " 中手动编辑该文件。
3. 选择适当的语言测试 OpenCPN 中的文件。在 Linux 中, 你可能还需要安装语言库 (locales) 以便可以选择所有语言。

语言支持（从 2.1.0 Build 415 版开始）:

捷克语
丹麦语
荷兰语（到 openCPN version 2.4.324 测试版为止）
英语
法语
德语
意大利语
波兰语
葡萄牙语
俄语
西班牙语
瑞典语

2.4 代码格式化

从发布后的 3.0 代码库开始，我们一直试图为源代码统一代码风格。所有新的补丁和其他应用贡献都要采用同一代码风格（这一做法的重要性）

首行缩进: 每个 level 要求使用 4 个字符，不允许使用制表符，设置编辑器在制表符的位置插入空格。

注释: 对方法和变量具有功能性描述的自解释代码在注释中可优先使用。只有在一般性阅读时代码及算法使用意图不明显的情况下才使用注释。

首选风格如下。按照这个模板处理大括号、小括号、操作符合控制语句边上的空格。

对于 Eclipse 用户，代码规范化模板参考: <http://opencpn.org/ocpn/downloads/doc/codestyle.xml>

对于 **Uncrustify** 用户，<http://opencpn.org/ocpn/sites/default/files/users/registry/beautify.png> 中有一个 uncrustify.cfg 文件。（另存该文件，这是一个文本文件但是看上去有些像图片，应该被命名为"beautify.cfg"）

```
#include <math.h>
```

```
class Point {
```

```
public:
```

```
    Point( double x, double y ) :
```

```
        x( x ), y( y )
```

```
    {
```

```
    }
```

```
    double distance( const Point& other ) const;
```

```
    int compareX( const Point& other ) const;
```

```
    double x;
```

```
    double y;
```

```
};
```

```
double Point::distance( const Point& other ) const
```

```
{
```

```
    double dx = x - other.x;
```

```
    double dy = y - other.y;
```

```

    return sqrt( dx * dx + dy * dy );
}

int Point::compareX( const Point& other ) const
{
    if( x < other.x ) {
        return -1;
    } else
        if( x > other.x ) {
            return 1;
        } else {
            return 0;
        }
}

namespace FOO {
int foo( int bar ) const
{
    switch( bar ){
        case 0:
            ++bar;
            break;
        case 1:
            --bar;
        default: {
            bar += bar;
            break;
        }
    }
}
} // end namespace FOO

```

3. 插件 API

以下信息源自插件发行时，信息有些过时。

为测试版讨论及测试，OpenCPN 插件架构，**API Version 1.0** 已发布。

设计和发行说明：

1. 插件是动态加载的库

{name}_pi.dll for Windows

lib{name}_pi.so for linux.

2. 当我们将插件放到正确的目录下时，系统会自动检测

Windows: {Where OpenCPN is}\plugins ex: C:\Program Files\OpenCPN\plugins

linux: {prefix}/share/opencpn/plugins ex: /usr/local/share/opencpn/plugins

3. 所有检测到的兼容插件在应用程序启动时被加载并初始化。为进行选择性加载，我们希望能够开发出一个

"PlugInManagerDialog".

4. 插件都是用 C++ 编写的, 使用的是 wxWidgets 工具集。我们建议有兴趣的开发人员可以使用其他更加通用的数据集并建议使用相关有益 API 扩展。同时你还可以结合 wxPython Indenton , 如...
5. 源代码树中的 include/ocpn_plugin.h 对 API 作了简单定义,

我创建了两个插件:

1. grib_pi

通过本插件可以实现当前 GRIB 叠加功能。我的 alpha 测试显示 2.1.0 中 GRIB 的功能几乎是重复的, 还需要全面测试。一些开发人员可能想考虑此插件中已提高的性能。有些地方的代码非常粗糙。缓存和渲染效果不错。

我计划将此插件作为 opencpn Version 2.2 的一部分进行发行。

该插件在 API 级别做了很多有意思的事情:

- a. 工具栏工具添加和管理.
- b. 工具栏选项卡页面添加.
- c. 配置文件读写.
- d. 海图覆盖渲染.
- e. 私人对话框管理.
- f. 鼠标追踪.

细心的读者可能会注意, 以上列出的功能是很多海图覆盖功能所要求的。Google KML 文件支持只是其中一种, 某种 POI 插件是另一个选择。Satallite radio weather...BlackBox radar...或其他的? 如果愿意, 我们可以将 AIS 移到插件中。

2. demo_pi

demo_pi 是一个演示。它可以被用作更高级插件开发所需的构架。我不打算在下一个官方 2.2 发行版中发行 demo_pi。仅供开发人员讨论和使用。事实上它做不了什么。

demo_pi 至少演示了以下内容:

- a. 私人窗口创建.
- b. NMEA 数据流拦截.
- c. 快捷菜单条目添加和管理.

我觉得 demo_pi 是万众期待的仪表盘 (Dashboard) 功能 (常规需求) 的基本包装器。

所以, 这是一个开始。

我还无法预测一个插件的所有 API 要求。因为开发人员已经开始考虑编写插件的时候, 我希望我们可以在 API 规范中添加更多新方法、函数及数据类型。这些我们可以做到。我希望我们可以确保 Version 2.2 的 API 稳定性。我们会尽可能地保持向后兼容性。

插件构建过程是一个半成品。我们鼓励开发人员对 Cmake 脚本所做出评论和改进。

我特别希望能在这里热烈讨论插件发展。

因为每个插件都已准备好进行测试, 我们希望对这些插件进行分页讨论。

祝好

Dave

3.1 插件指南

以下是在 OpenCPN.org 上发布 OpenCPN 插件的指南:

1. OpenCPN.org 上发布的插件需要符合 GPL V2 及其后续版本中的许可条款。
 - 1.1 也就是说 OpenCPN 不会存放插件源代码, 插件作者必须独立发行它们。可以通过 Sourceforge 或 Github 发表, 以便授权其他人使用。
2. OpenCPN.org 上发布的插件必须可移植到以下任一平台:
 - A. Debian style Linux
 - B. Microsoft Windows

- C. Macintosh OS-X
 - D. 开发人员需谨记：插件的可用性及普及度与平台可移植性直接相关，所以应努力使插件适用于所有 OpenCPN 平台。
3. 发布在 OpenCPN.org 上的插件必须可以自由下载，而不收取任何费用。
- 3.1 插件二进制文件可能存在于 OpenCPN 服务器上或其他地方。

如何在 OpenCPN.org 上发布插件：

一旦在 OpenCPN.org 上创建账户和/或登录，填写[插件提交表格](#)。插件正是发布前会有相关审查。发布后，你可以通过编辑插件下载页面对你发布的内容进行实时更新。

3.2 不同插件 API 版本

OpenCPN 3.1.x 与所有预先创建和/或发布的插件向下兼容。换言之：所有较老的插件与新版 OpenCPN 向上兼容。这是一个固定的设计目标，并且我们已经取得了一些成就。

但反过来就不对了。按照 3.1.x API 规范构建的插件与 OpenCPN 3.0.x 不兼容。同理：OpenCPN 3.0.x 与按 3.1.x API 规范构建的插件不兼容。

理论上来说第二种情况非常清晰。OCN 3.0.x 如何预期 3.1.x 插件 API 扩展呢？

所以，我们要这样做：

如果你希望能在 3.0.x 上运行的可靠插件可以在其后续版本上运行，按照 3.0.x API 进行编码。

除非

如果你想要 3.1.x 插件中 API 的扩展性功能，你需要使用 3.1.x API 来构建升级版插件，这样就可以与 3.1.x 及其后版本相兼容。

4. 问题解决

命令行、配置文件及调试器可以帮你解决问题并提供其他低端细节。

OpenCPN 有一个最小的命令行接口 (CLI)。

Usage: opencpn [-unit_test_1] [-p] [-no_opengl]

-unit_test_1

-p

-no_opengl

[-unit_test_1]

该命令在目前已加载的海图数据库中所有可能的海图里循环测试 (test)，调整视窗以自动显示已加载的海图。该测试 (test) 的价值非常明显：它提供一个“无人坚守式” OpenCPN 压力测试，这在拼接模式下非常有趣...

测试 (test) 一直进行直到数据库中所有海图都被测试一次。

我们可以把这看作是一种试用版测试工具。

该转换还有一个副作用。我们可以通过它把一系列 ENC 海图变成 SENC 文件。首先，耗时的 ENC 海图处理过程因此可以通过这种“无人坚守式”方式下完成。

[-p] opencpn.org/ocpn/portable 对可移植选项进行了解释

[-no_opengl] (只适用于 3.2.2) 彻底关闭 OpenGL，比 Options 更彻底。

config 或 ini 文件包含许多会话期间保存下来的设置。

当某些地方出现问题后可以添加调试语句以便输出很多信息。

在一个会话开始前，以下语句可以被识别并添加到配置文件中。

[Settings]

DebugOpenGL=1 为 OpenGL 添加日志文件调试输出

DebugNMEA=500 将 500 个 NMEA 语句添加到日志文件中

DebugCM93=1 将与 CM93 相关的调试语句量写入日志文件

DebugGPSD=1 这样会产生更多与 GPSD 日志输出。
DebugS57 为 S57 添加日志文件调试输出
DebugBSBImg 为 BSBImg 添加日志文件调试输出
DebugGDAL 为 GDAL 添加日志文件调试输出

使用 FIFO 进行 AIS 调试

在 ais.pp 中查看代码

Code:

```
/*  
A special test for a user defined FIFO  
To use this method, do the following:  
a. Create a fifo            $mkfifo /tmp/aisfifo  
b. netcat into the fifo    $nc {ip} {port} > /tmp/aisfifo  
    sample {ip} {port} could be   nc 82.182.117.51 6401 > /tmp/aisfifo  
c. hand edit opencpn.conf and make AIS data source like this:  
    [Settings/AISPort]  
    Port=Serial:/tmp/aisfifo
```

This also works if you have an ascii ais log

for which you can simply \$cat ais.log > /tmp/aisfifo

```
*/
```

```
if(m_pPortName->Contains(_T("fifo")))
```

```
    goto port_ready;
```

你可能已经看出来，如果端口名称里含有"fifo"字符串，端口变成 TTY 的要求就会放松。

5. 矢量调色板

OpenCPN 3.0.

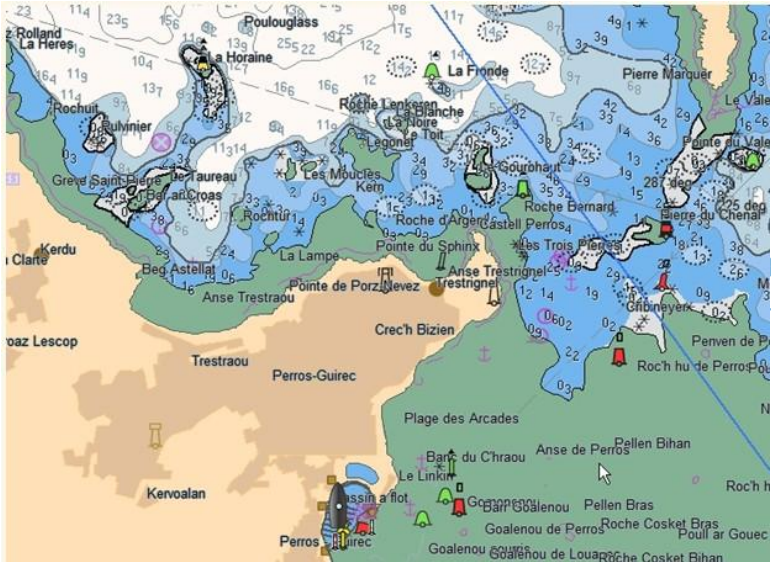
3.0 中海图显示有很大改变。以下会进行详细介绍，想知道如何改变默认风格，可以尝试[下载文件](#)使海图有瑞典风格（新的颜色、声音和暗礁）。解压文件，然后将这两份文件移到 s57data 安装文件夹下，你会看到以下内容：



如果喜欢（法国）SHOM 海图的外观和风格，只需将 s57data/chartsymbols.xml 中的以下符号（symbols）改成 RGB

数值:

- LANDA 255 224 183
- NODTA 225 224 222
- DEPDW 255 249 245
- LANDF 255 200 125
- CHBRN 226 192 156



海图符号

OpenCPN Version 3.0 海图图标应用格式更加现代化也更容易使用。目前为止所有关于如何显示海图图标的信息都储存在一个文件内(S52RAZDS.RLE)。该文件的格式设计不是很实用。

新的处理方式有简单可读的 XML 句法来进行配置。实际的 bitmaps 是从 PNG 文件载入的。你可以通过任何现代图像编辑器对其进行编辑（可能是 Gimp 或 Photoshop）。也就是说我们现在可以用透明 bitmaps 和防混淆图形。

比如，下图是有灯和声音的老版和新版关于左舷浮标的放大截图：



为了使符号编辑更简单, <http://journeyman.se/opencpn/rastersymbols.tif> 有一个多层(layer)TIF 文件, 包含以下层(layer):

- 文本 (Text) : 符号的名称。其功能仅仅是查看你所编辑的符号的名称。非常难看但是比没有强
- 枢轴 (Pivot) : 一个现实符号中心点所在位置的像素
- 符号 (Symbols) : 实际图形符号层, 这个层只能被存为特定色表的<graphics-file>
- 背景 (Background) : 每个符号的矩形定义区。 该位置应该符合 <graphics-location> 及下面要介绍的 XML 文件中的 <bitmap height="" width="">
- 框架 (Frames) : 图像定义区外的一个像素的实线黑色轮廓
- 画布 (Canvas) : 一个有色彩的背景, 你可以进行切换以便在一个纯色背景下查看符号

<http://journeyman.se/opencpn/symbol-vector-orig.svg> 中有一些可用于新符号的矢量原件

简单介绍一下 XML 格式:

chartsymbols.xml 文件中有 5 类标签:

First color tables:

Code:

```
<color-table name="">
  <graphics-file name="" />
  <color name="" r="" g="" b="" />
  ....
</color-table>
```

每个色表 (color table) 都有一个名称。OpenCPN 目前使用以下色表: DAY-BRIGHT, DUSK 和 NIGHT。

每个色表还有一个 <graphics-file name="" />。其名字应该和 XML 文件所在文件夹下的 PNG 文件的名称相同。Bitmap 符号就是从这个文件加载的。它应该是一个使用 alpha 通道的 PNG24 文件。

这些颜色本身是 0-255 之间的 RGB 整数数值。

下一个类型的 Lookups。它们规定了额外图形如何放置, 比如文本及二级符号。除非你觉得有必要, 一般来说我们不需要编辑这些标签, 至少现在它们的可读性已经比老式格式强。目前实际数据 (actual data) 没有改变。

<line-style> 标签决定如何绘制各种复杂的线条 (航道、锚区等等)。实际矢量规范还是 HPGL 格式的 (起始于 80 年代的矢量绘图格式)。然而, 我们还没有一种比较好的解析 SVG 图形的方式, 因此 SVG 图形解析方式没变。

<pattern> 标签规定如何绘制疏浚区、沼泽地等等。注意整体特性, 比如土地和深度轮廓没有图案 (pattern) 所以这里不作规定。以下举一个图案 (pattern) 标签的例子。这个特殊的图案 (pattern) 既有矢量 (vector) 又有 bitmap 定义。默认情况下如果两者同时存在, 系统会使用 bitmap。Bitmap 更快。如果你想使用矢量作为默认方式, 指定 <prefer-bitmap>no</prefer-bitmap> 作为 <pattern> 的子标签。

Code:

```
<pattern RCID="3143">
  <name>MARSSES1</name>
  <definition>V</definition>
  <filltype>S</filltype>
  <spacing>C</spacing>
  <bitmap width="40" height="40">
    <distance min="1500" max="15000" />
    <pivot x="30" y="30" />
    <origin x="0" y="0" />
    <graphics-location x="207" y="1007" />
  </bitmap>
  <vector width="400" height="393">
    <distance min="1500" max="15000" />
    <pivot x="750" y="1016" />
    <origin x="550" y="499" />
  </vector>
  <description>pattern of symbols for a marsh</description>
  <color-ref>ACHBRN</color-ref>
</HPGL>SPA;SW2;PU751,765;PD751,499;SPA;SW2;PU626,892;PD876,892;SPA;SW2;PU550,810;PD950,810;SPA;SW2;PU664,799;PD592,634;SPA;SW2;PU830,799;PD901,637;</HPGL>
</pattern>
```

如果你想做一些编辑, 这里有意义的编辑就是矢量 (vector) 版本的颜色参考 (color-ref), 还有距离 (distances) 等可能也有意义。注意 color-ref 只用于矢量 (vector) 版本, 在 bitmap 中没有效果。

在 bitmap 版本中, 你指定 bitmap 的 size, 显示的时候会图案区域 (pattern area) 进行平铺。

<graphics-location x="" y="" />指向当前已加载的<color-table>的 PNG 文件的位置 (location)。位置 (location) 是从左上角 (0,0) 开始计算的, y 轴朝下为正。我们从上面例子可以看出, 该图案由(207,1007) 到 (207+40,1007+40)的 PNG 文件区的所有内容构成并进行了倾斜。

最后一个标签是<symbol>, 它和<pattern>有些类似。<symbol>同样也可以有矢量 (vector) 和 bitmap 定义, 而 bitmap 也是默认定义。注意出现已旋转的符号不要用 bitmap, 比如 TSS 方向箭头 (TSS direction arrows) 等等。这些符号必须有<prefer-bitmap>no</prefer-bitmap>。

典型的符号 (symbol) 看上去有些像:

Code:

```
<symbol RCID="45">
  <name>BOYSPH68</name>
  <description>spherical buoy,by,full-chart</description>
  <bitmap width="19" height="15">
    <distance min="0" max="0" />
    <pivot x="9" y="11" />
    <origin x="0" y="0" />
    <graphics-location x="661" y="742" />
  </bitmap>
  <definition>R</definition>
</symbol>
```

该导示图形选自当前已加载的 PNG 文件, 本例从 <graphics-location x="661" y="742" />开始, 该区域为 19 x 15 像素, <pivot x="9" y="11" /> 表示该 bitmap 会以什么样的像素出现在海图中。

5.1 传统矢量调色板

...依然能用于 3.0.2。

删除任何 S57DataLocation=行, 把以下信息添加到 opencpn.ini 文件中。

[Directories]

PresentationLibraryData=C:\Program Files\OpenCPN\s57data\S52RAZDS.RLE

OpenCPN 2.5

矢量海图调色板在 s57data\S52RAZDS.RLE 文件中。

该文件顶部有类似如下所列信息

CCIE 31LANDA0.3600(0x1b)0.4000(0x1b)49.00(0x1b)brown

注意: 换码字符(0x1b)

这里的颜色用的是很难翻译的 CCIE color map 或 Yxy (Y=LRV)。

consider line:

CCIE 31LANDA0.4182 0.4220 62.03 brown

0.4182 is x

0.4220 is yOpenCPN 2.5

62.03 is Y

所以颜色是 Yxy (62.03,0.4182,0.4220)。将这些数值放入 EasyRGB.com,每个可用颜色空间都会出现该颜色。

HTTP = #F6C86E

Web safe = #FFCC66

RGB 0-255 = 245.98 200.02 110.02

RGB 0-FF = F5 C8 6E
RGB 0-0.1 = 0.96462 0.78439 0.43144
CMY 0-0.1 = 0.03538 0.21561 0.56856
CMYK % = 0.000 18.684 55.274 3.538
XYZ = 61.471 62.030 23.489
Yxy = 62.030 0.41820 0.42200
CIE-L*ab = 82.929 5.976 50.618
CIE-L*CH = 82.929 50.970 83.267
CIE-L*uv = 82.929 36.230 61.612
HunterLab = 78.759 1.491 37.449
Illuminant = D65
Observer = 2° (1931)

这些名字都有一些描述性功能（比如，LANDA 指的是陆地面积）。我们可以通过代码将 CCIE 转换成 RGB near s52plib.cpp(387)，但是却不能将 RGB 转换成 CCIE。更多信息可以通过 Google 搜索或者访问 easyrgb.com。

如果你浏览了该文件，你会看到各功能的显示原则是以颜色为参考的。比如：

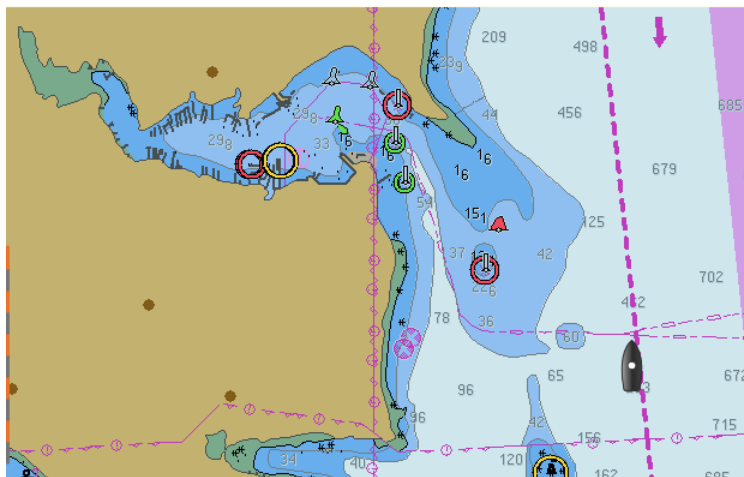
0001 500090
LUPT 40LU00090NILLNDAREA00001SPLAIN_BOUNDARIES
ATTC 1
INST 10AC(LANDA)DISC 12DISPLAYBASE
LUCM 612010
*** 0
Decode: LUPT.....LNDARE.... (a LNDARE feature)
INST.....AC(LANDA) (AreaColor LANDA)

等等

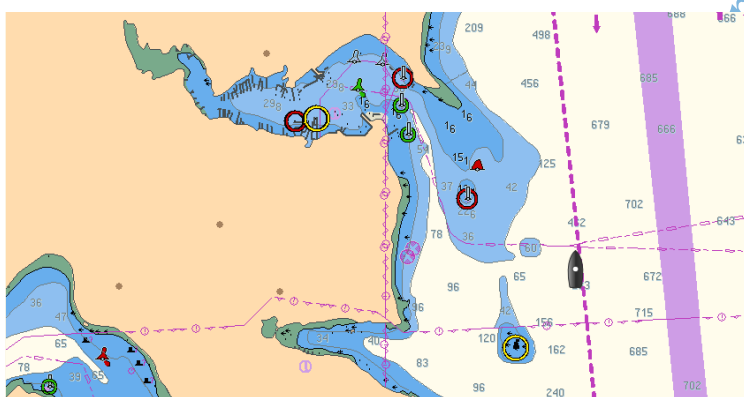
不管怎样你可以随意摆动这些数值且没必要重新进行编译。S52RAZDS.RLE 被作为数据文件读取。你可以在 opencpn.ini 配置文件中指定其位置，如下：PresentationLibraryData=C:\\Users\\Will\\AppData\\Roaming\\opencpn\\CAD.RLE

Openev project libs52 默认调色板:

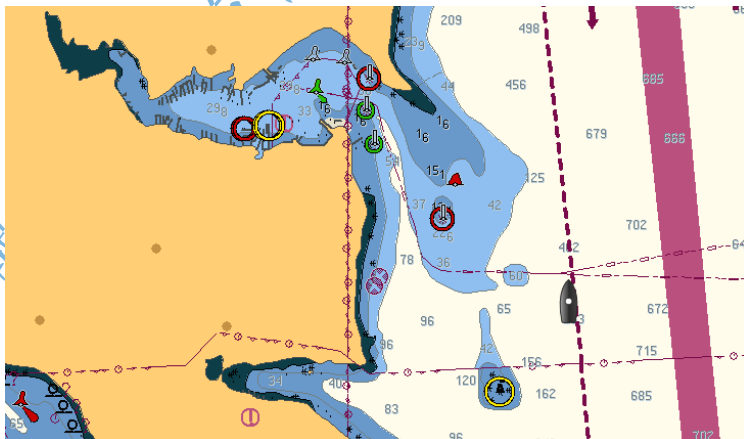




类似 SHOM 海图的 SHOM.RLE 文件修改示例:

下载 [SHOM.RLE](#)

类似加拿大矢量海图的 CAD.RLE 文件修改示例:

下载 [CAD.RLE](#)

6. 用户界面风格定义

风格管理器 (Style Manager) 允许你对 OpenCPN 用户界面的多个方面进行修改:

- 工具栏图标、背景及外形
- 什么工具在工具栏中显示出来
- 海图状态栏大小和背景
- 当前活动航线窗口背景

配置文件

OpenCPN 启动后会寻找安装子目录 `uidata` 下的 `styleXXX.xml` 的文件（这里的 XXX 指任何文本都可以），同时也存在于用户设置目录中（Linux 系统中为 `~/opencpn`）。

每一个 `style.xml` 文件定义一种或多种风格（style）。文件整体结构看上去有些像：

```
<styles>
  <style name="A name here">
    <description>...</description>
    <graphics-file name="iconfile.png" />
    <chart-status-icon width="40"/>
    <chart-status-window transparent="true"/>
    <active-route>
      <font-color r="50" g="50" b="50" />
      <text-background-location x="660" y="260" width="111" height="43" />
    </active-route>
    <tools>
      <horizontal>
        <separation distance="0"/>
        <margin top="4" bottom="4" left="4" right="0" invisible="true"/>
        <background-location x="660" y="60" />
        <toggled-location x="20" y="60" width="38" height="32"/>
        <size x="38" y="32"/>
        <toolbar-start x="100" y="60" width="4" height="32"/>
        <toolbar-end x="140" y="60" width="4" height="32"/>
        <toolbar-corners radius="3"/>
      </horizontal>
      <vertical>
        ...same as for <horizontal />
      </vertical>
      <tool name="...">
        <icon-location x="658" y="20"/>
        <rollover-location x="658" y="220"/>
        <disabled-location x="658" y="320"/>
      </tool>
      ...more tools.
    </tools>
    <icons>
      <icon name="...">
        <icon-location x="700" y="20"/>
      </icon>
      ...more icons.
    </icons>
  </style>
</styles>
```

风格（style）所需的 `bitmaps` 是在 PNG 文件中被定义的，并在 `<graphics-file>` 标签上进行命名。各种 `<xxx-location>` 标签表示 `bitmap` 在这个文件中的位置。位置通过 X/Y 轴坐标表示，左上角为 (0,0)，Y 向下为正值。该文件是 24 bit PNG。支持透明，你可以使用半透明图标及背景来营造不同效果。

工具（`<tools>`）标签包含工具栏水平（`<horizontal>`）和垂直（`<vertical>`）版本的所有参数。这就是的工具栏上有非矩形按钮时还能整齐地横竖显示的原因。它们之间的分隔距离就是两个工具栏按钮间像素间的间隔。当我们使用非矩形按钮时，工具栏按钮在背景的水平垂直方向上位置不同。这种情况下，我们需要将图标正确地放置到水平位置上，然后通过 `<icon-offset x="..." y="..." />` 标签确定其在垂直方位上的偏移量，这里的 X/Y 是指像素，并且可以使负数。

该边缘指的是外部边缘，以像素来衡量的。该边缘同样可设置成透明状。透明工具栏窗口边缘在窗口边缘和按钮间有空间。

背景地点（`<background-location>`）和已切换的地点（`<toggled-location>`）应该提供两种不同的背景，一种用于开关按钮开的状态，另一种用关闭状态。打开状态的按钮背景可以比关闭状态时的背景更大，但是不能比它小。如果你使用了该功能，确保你指定了足够的空间可以很好显示变大了的背景。

尺寸（`<size>`）标签为工具指定默认尺寸，但是可能会被工具自身标签覆盖。

我们可以通过（工具栏开始/结束）`<toolbar-start / end>` 标签使用漂亮的起始和结束图片进一步定义工具栏。这些绘制在其他图形顶部进行。我们可以通过（工具栏-角）`<toolbar-corners>` 标签指定工具栏四角的半径。NB! 圆角只有在使用透明边缘的时候才可以看见。

工具（`<tool>`）是工具栏上的一个条目。它应该提供三种图标：A，普通图标，附带翻转和禁用功能。如果你希望隐藏一个特定工具，你可以通过隐藏于工具栏（`<hide-in-toolbar/>`）标签实现。

图标（`<icon>`）是一个通用图标，工具栏上不使用。光标和航路点标记就是这样一种图标。它只有一种状态。

工具名称和插件

标准的 OpenCPN 工具有自己的名称，你只要查看一下默认的安装 `style` 就知道了。

插件的风格 (style) 表现在 “一般名称” 中。如果你的风格 (style) 为你的插件提供了工具, 其优先等级要高于插件自己提供的图标。但是对于那些将自己的图标用作沟通状态的插件来说, 改插件图标还会获得更高优先等级。(World Magnetic Model 和 Radar Overlay 就是后者的例子)

7. Bug 和功能追踪

试用版和稳定版 Bug 及功能追踪问题可以访问: www.opencpn.org/flyspray/

在你提交报告或者请求之前请查看之前是否有人提过相同的问题。使用该网站不需要注册, 但是如果注册可用功能将更多。

8. OpenCPN 发行历史

1.2.0 2007-04-20

OpenCPN 开发目的如下:

可用作船舶主导航界面, 该导航系统转舵装置全程可见。其他工具更有利于离线航线计划、潮汐和潮流流预报, 在线计程等;

可以快速开启和关闭;

只存在日常航行所必须的工具栏按钮;

可移植性, wxWidgets 核心组件。目前已经在 W98, XP, 和 Linux 上完成测试并进行了推广使用;

支持常见海图格式 (比如, 通用的和现代的)。现实中指的是 BSB 格式光栅海图和 S57ENC 格式矢量海图。

1.2.2 2008-01-02

完成 AIS 符号系统和诊断操作的改进, 包括 RateOfTurn 显示;

完成工具栏图标符号系统的改进, 支持 day/dusk/dark 色彩方案;

优化 Windows 优化主目录选择, 允许程序安装在二进制目录/文件夹内;

提高大圆航行距离/航向计算精确度;

可选择的深度单位 (Optional Depth Units) 可以以凸起的字符形式显示;

海图缩略图中本船位置依据精确的地极 GPS 位置进行显示。

1.2.4 2008-04-20

为 Unicode 系统配置/构建修改, 特别是 Ubuntu。

添加航线 (Route) 和标记 (Mark) 属性表

提高 S57 ENC 符号系统。

1.2.6 2008-08-31

维护和功能升级版;

一些 S57 ENC 符号系统的升级;

更加符合 IHO S52 标准, 及各种增强功能及 Bug 修复。特别有意思的是黄昏 (DUSK0) 和夜晚 (NIGHT0) 显示模式;

特别希望收到用户的反馈意见。

1.2.8 2008-12-25

支持 Mac OS X, 包括可安装的应用程序包 (OpenCPN.dmg), GNU/gcc 兼容系统构建文档及 XCode project 文件;

Windows 平台的 Garmin USB GPS 设备支持;

多项 Bug 修复和用户优化。

1.3.0 2009-05-09

CM93 全球矢量海图支持, 可自动进行海图缩放;

通过可配置 CPA/碰撞警告项提高 AIS 目标追踪;

GPX 兼容航路点和航线导出工具;

以本船为中心的航程范围的定义;

多项 Bug 修复和用户优化。

1.3.2 2009-07-11

AIS 目标描述功能及碰撞监测警告升级;

热键和滚轮鼠标支持;

帮助文档;

各种显示速度和质量的优化,特别是 CM93 海图的显示。

1.3.4 2009-10-01

船舶位置跟踪功能,包括 GPX 导入和导出;

可视性提高,图标和屏幕图像提升;

GPX 接口协议导入和导出航线;

完善 CM93 矢量海图细节控制滑块;

各方面易用性及可靠性的提高。

1.3.6 2010-01-07

添加基本 GRIB 文件解码及海图覆盖显示;

提高和扩大 NMEA 信息类型支持;

文档升级;

添加模块化国际语言支持;

各方面易用性及可靠性的提高。

2.1.0 2010-06-26

- BSBv3 矢量海图和 S57 ENC 海图支持;

- CM93 矢量海图支持;

- S57 矢量海图 IHO S52 兼容显示;

- 单一海图显示和拼接海图显示模式;

- 北点向上 (North-up) 和 航向向上 (Course-up) 显示模式;

- 移动地图 (Moving-map) 显示模式;

- 使用船舶跟踪功能进行航线导航;

- 航路点导航;

- NMEA 0183 GPS 接口波特率可选;

- gpsd 库支持;

- 自动航行 (Autopilot) 输出支持;

- AIS 全目标追踪及碰撞警告导入;

- 锚更/ 警报功能;

- GRIB 气象预报文件输入和显示;

- GPX 航路点和航迹导入导出文件支持;

- 不同地点潮汐和潮汐流预测和显示;

- 多语言支持

- Linux 和 Windows 兼容

2.3.0 2010-12-18

1. 集成式插件体系结构,允许第三方开发及附属功能布局;

2. 两个有效的插件立即可用;

GRIB (气象)文件数据图形覆盖

船舶 NMEA 数据可配置的屏幕显示仪表盘

3 通过可选择对话框强化航线、航迹及航路点管理功能;

4. 基础包加入多语言支持,包括:

葡萄牙语

爱沙尼亚语

希腊语

土耳其语

芬兰语

挪威语

等等

5. 支持 Garmin Host 模式航线和航路点交换及上载到远程 GPS 接收器;
6. 高级国际化 Windows 程序安装程序;
7. 完全支持所有海图类型北点向上 (North-up) 和 航向向上 (Course-up) 显示;
8. 扩展 AIS 目标信息解码, 包括欧洲内陆 AIS 规范.
9. AIS 目标列表对话框 (AIS Target List Dialog) 可以对所有激活的 AIS 目标进行选择 and 跟踪。
10. 基本用户界面的多项改进。

2.3.1 2010-12-31

1. 更正本船 AIS 信息逻辑 (AIVDO)
2. 通过打开目标 query 对话框更正错误关闭 AIS target 列表(FS#310)
3. 通过 BSB 海图航线开关更正偶尔出现的定位问题
4. 提高工具栏刷新逻辑;
5. 将最大海图尺寸从 20 提高到 30 (FS#316);
6. 更正 GoTo 航路点及后续航线删除逻辑 (FS#244)
7. 各种语言文件更新.
8. 更正拼接海图深度 (Depth) 单位显示, 使用非英语语言设置;
9. 在用户模式 (User Mode) 下更正 Win7 Garmin USB 设备界面重设置;
10. 更正 NSIS 脚本中 Windows vc 在发行安装命令。

2.5.0 2011-07-26

1. CM93 海图每个海图单元格的偏移量调整
2. 横切渐长投影海图 (Transverse Mercator chart) 拼接
3. 可选择全屏拼接选项
4. 可配置的经纬度 GID 线
5. 各种屏幕光线调整功能
6. 提高潮汐/潮流信息窗口
7. 配置"Layers"功能, 启用固定目标选择性显示
8. 提高和扩展 AIS 功能, 包括
 - a. Euro Inland 支持
 - b. 基站和 ATONs
9. 可选择性全屏显示模式.
10. 综合性航线计划功能与潮汐表支持
11. 通过浮动透明工具栏提高 GUI 功能
12. 几个新语言选项
13. 提高内存管理
14. 提高自动航行导出支持
15. 新的多航段测量工具
16. 提高矢量海图符号系统
17. 提高综合潮汐/潮流数据准确度
18. 提高航线高清接口, 包括可选择航线颜色
19. 所有平台完全一致, 特别包括 Macintosh OS X
20. 因为广泛的 Beta 测试, 所有平台稳定性大幅度提高

21. 基本用户界面的多想完善, 同时维持 OpenCPN 广为人知的易用优点

3.0.0 2012-06-05

OpenCPN Version 3.0 新功能及功能改进

1. OpenGL 支持, 提高图形性能和功能
2. 提高 ENC 目标 Query 界面
3. 可移植执行选项, 可轻松移入其他系统.
4. 升级到 wxWidgets 2.8.12.
5. 支持 ENC Overlay cells.
6. 升级 gpsd 支持到 2.96 版本
7. 实现插件系统改进, 启用各种插件类型
8. 实现 DSC 和 GPS_Gate Buddy 追踪
9. 实现用户定义的海图组 (Chart Groups)
10. 实现 AIS SART 应答机显示和追踪
11. 提高 ENC 目标渲染和显示
12. 实现用户可定义的潮汐/潮流数据集
13. 所有平台完全一致, 特别包括 Macintosh OS X
14. 因为广泛的 Beta 测试, 所有平台稳定性大幅度提高
15. 基本用户界面的多想完善, 同时维持 OpenCPN 广为人知的易用优点

3.0.2 2012-07-07

本版本更正了当前 3.0.0 field release 中的几个小的 bugs。

没有添加新功能。

3.2.0 2013-02-27

1. 晚上用户选项对话框和上下文菜单, 一体化现代图形和统一格式
2. 显著提高导航仪器输入和输出选项, 包括扩展网络连接功能支持 TCP, UDP 和 GPSD 数据源.
3. 新的用户界面主题支持, OpenCPN 传统及两个新的 low profile themes.
4. 提高剪贴板支持, 包括外部 KML 数据资源支持
5. 提高潮汐/潮流数据集支持, 包括现代完满二进制数据集
6. AIS 数据目标显示和追踪方面的多想提高
7. 自动大圆航线生产选项
8. 仪表板插件 (Dashboard PlugIn) 大幅提高, 包括各种风格的新用户界面选项和仪器
9. 提升 Linux 主机声音系统支持
10. 内置用户手册 (Users Manual) 重要更新.
11. 性能和可靠性完全支持平台明显改善

3.2.2 2013-05-08

本版本更正了当前 3.0.0 field release 中的几个小的 bugs。

没有添加新功能。

8.1 2012-6-5 发行 3.0

OpenCPN 3.0.0 版本

公告:

OpenCPN 3.0 版本已经发布, 你现在可以立即下载。

本版本是 OpenCPN 的一个主要功能升级和维护版本。

OpenCPN 是一款自由、开源导航包, 可以在 Windows、Linux 和 macintosh 计算机上运行。

OpenCPN 的功能包括:

- 拼接光栅(RNC) 和矢量(ENC)数字海图显示
- 行业标准 GPS 接收器接口技术, 有移动地图 ("moving map") 全屏船舶追踪;
- AIS 接收器接口技术, 有可配置的目标安全监测
- 一体化的航线、航迹和航路点数据库, 有行业标准导入/导出功能
- 潮汐和潮汐流预测, 监测到的数据可以在屏幕上重叠显示
- 广泛的国际语言支持

OpenCPN 的设计本身就希望其安装、配置和日常使用简单化。

OpenCPN 的下载和安装是免费且不限时的, 而且无需注册。

OpenCPN Version 3.0 的新功能及强化功能

1. OpenGL 支持, 提高图形显示性能及功能;
2. 对 ENC 目标 Query 界面进行了改进;
3. 可移植的执行项, 确保可以在新系统中使用;
4. 升级到 wxWidgets 2.8.12;
5. 支持 ENC Overlay 网格;
6. 将 gpsd 支持升级到 Version 2.96
7. 实现插件系统升级, 可以接受很多插件类型;
8. 实现 DSC 和 GPS_Gate Buddy 追踪;
9. 实现用户定义海图组;
10. 实现 AIS SART 应答机显示和追踪;
11. 提高 ENC 对象渲染和显示;
12. 实现用户可定义潮汐/潮汐流数据集;
13. 所有平台完全一致, 特别是 Macintosh OS X.
14. 由于大量的测试版测试, 其在所有平台上使用的稳定性大幅提升;
15. 基本用户界面作了很多提升同时延续了 OpenCPN 广为人知的易用性的特点。

访问 opencpn.org/ocpn/node/189, 找到 OpenCPN Version 3.0 详细的发行说明。

我们希望你更喜欢 OpenCPN Version 3.0 上的新功能, 并期盼收到你的反馈。

访问 sourceforge.net/projects/opencpn/ 下载 OpenCPN Version 3.0。

如果需要插件或者想了解其他有用信息, 请访问我们的支持网站 opencpn.org

OpenCPN 开发团队

首席开发人员: Dave Register

OpenCPN version 3.2 新功能

OpenCPN 3.2 发布声明

2013 年 2 月 27 日 即时发布

OpenCPN 团队为能宣布 OpenCPN Version 3.2 发行感到自豪。现在就可以下载 OpenCPN Version 3.2。

本版本是 OpenCPN 功能的重要更新和维护版本。

OpenCPN 跨平台电子导航系统 (ECS) 支持各种制图标准和硬件接口协议。有些版本适用于 Linux, Windows 和 Macintosh 电脑架构。

OpenCPN 的部分功能:

- 拼接光栅(RNC) 和矢量(ENC)数字海图显示
- 行业标准 GPS 接收器接口技术, 有移动地图 ("moving map") 全屏船舶追踪;
- AIS 接收器接口技术, 有可配置的目标安全监测

- 一体化的航线、航迹和航路点数据库，有行业标准导入/导出功能
- 潮汐和潮汐流预测，监测到的数据可以在屏幕上重叠显示
- 广泛的国际语言支持
- 插件架构允许正在进行的第三方功能强化

OpenCPN 是在 GPLV2 许可下发布的。我们可以从 sourceforge.net 直接、自由、无限制地下载可执行程序及其源代码。

作为 OpenCPN3.0 的延续，OpenCPN3.2 包含很多新功能和加强功能，以下列举部分功能：

- 晚上用户选项对话框和上下文菜单，一体化现代图形和统一格式
- 显著提高导航仪器输入和输出选项，包括扩展网络连接功能支持 TCP, UDP 和 GPSD 数据源。
- 新的用户界面主题支持，OpenCPN 传统及两个新的 low profile themes.
- 提高剪贴板支持，包括外部 KML 数据资源支持
- 提高潮汐/潮汐流数据集支持，包括现代完满二级制数据集
- AIS 数据目标显示和追踪方面的多想提高
- 自动大圆航线生产选项
- 仪表板插件 (Dashboard PlugIn) 大幅提高，包括各种风格的新用户界面选项和仪器
- 提升 Linux 主机声音系统支持
- 内置用户手册 (Users Manual) 重要更新.
- 性能和可靠性完全支持平台明显改善

我们希望你喜欢 OpenCPN 所有新功能并期待你的进一步反馈。

通过 [OpenCPN | Free Science & Engineering software downloads at SourceForge.net](http://www.opencpn.org) 下载 OpenCPN3.2

关于插件及其他有用信息访问我们的支持网站: [OpenCPN | Official OpenCPN Homepage](http://www.opencpn.org)

OpenCPN 开发团队

首席开发人员: Dave Register

自从 2012 年 7 月发行 3.0.1 以来，OpenCPN 就保持快速开发的步调。我们介绍了很多新功能及很多小修改、Bug 修复，提高可靠性及其他小细节。

OpenCPN 3.2.2 发布声明

2013 年 5 月 8 日 即时发布

OpenCPN 团队为能宣布 OpenCPN Version 3.2.2 发行感到自豪。现在就可以下载 OpenCPN Version 3.2.2。

本版本是 OpenCPN3.2 功能的重要更新和维护版本。

其更正了一些非关键性的操作缺陷及特殊配置案例。

OpenCPN 跨平台电子导航系统 (ECS) 支持各种制图标准和硬件接口协议。有些版本适用于 Linux, Windows 和 Macintosh 电脑架构。

OpenCPN 的部分功能：

- 拼接光栅(RNC) 和矢量(ENC)数字海图显示
- 行业标准 GPS 接收器接口技术，有移动地图 ("moving map") 全屏船舶追踪；
- AIS 接收器接口技术，有可配置的目标安全监测
- 一体化的航线、航迹和航路点数据库，有行业标准导入/导出功能
- 潮汐和潮汐流预测，监测到的数据可以在屏幕上重叠显示
- 广泛的国际语言支持
- 插件架构允许正在进行的第三方功能强化

OpenCPN 是在 GPLV2 许可下发布的。我们可以从 sourceforge.net 直接、自由、无限制地下载可执行程序及其源代码。

作为 OpenCPN3.0 的延续，OpenCPN3.2 包含很多新功能和加强功能，以下列举部分功能：

- 晚上用户选项对话框和上下文菜单，一体化现代图形和统一格式

- 显著提高导航仪器输入和输出选项，包括扩展网络连接功能支持 TCP, UDP 和 GPSD 数据源.
- 新的用户界面主题支持， OpenCPN 传统及两个新的 low profile themes.
- 提高剪贴板支持，包括外部 KML 数据资源支持
- 提高潮汐/潮流流数据集支持，包括现代完满二进制数据集
- AIS 数据目标显示和追踪方面的多想提高
- 自动大圆航线生产选项
- 仪表板插件（Dashboard PlugIn）大幅提高，包括各种风格的新用户界面选项和仪器
- 提升 Linux 主机声音系统支持
- 内置用户手册（Users Manual）重要更新.
- 性能和可靠性完全支持平台明显改善

我们希望你喜欢 OpenCPN 所有新功能并期待你的进一步反馈。

从 <https://sourceforge.net/projects/opencpn/> 下载 3.2.2 版本

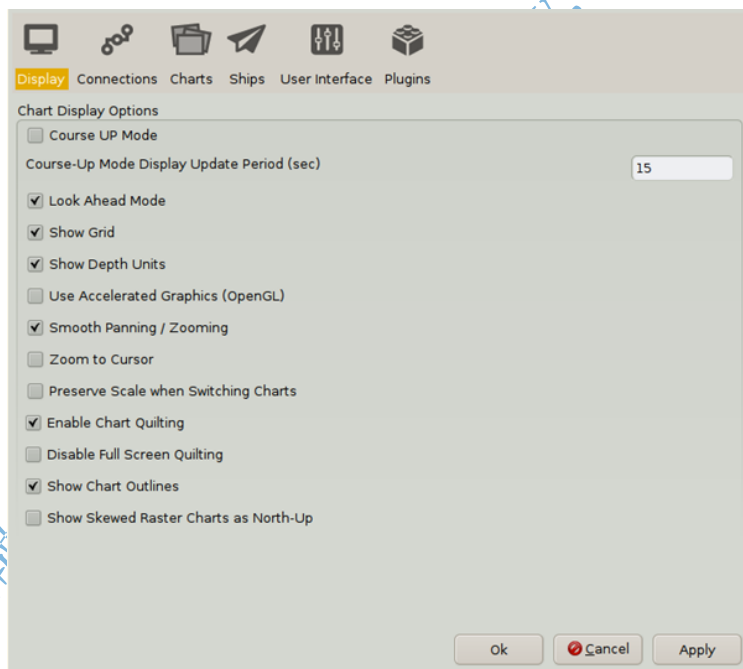
关于插件及其他有用信息访问我们的支持网站: <http://opencpn.org>

OpenCPN 开发团队

首席开发人员: Dave Register

8.2 OpenCPN3.2 新功能

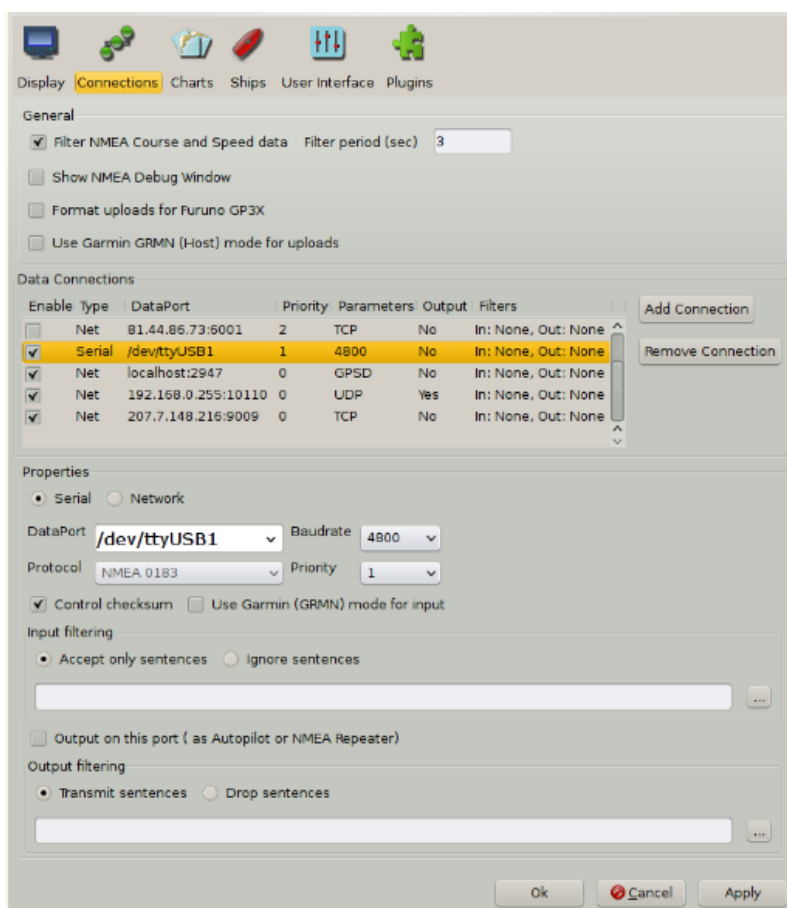
用选项（Options）替换原来的工具栏



实现选项（Options）对话框的新布局以替代原来的工具栏，是条目安排更合理更一致。

更多信息请阅读: http://opencpn.org/ocpn/setting_options

GPS, AIS 和其他 NMEA 数据源新的连接方式



■ **实现新的数据源架构。NMEA 数据源新的连接方式。**这是 3.2 最大的改变。相比较之前的版本有很大的提高。用户会发现大多数之前可以用于 OpenCPN 的第三方应用程序现在都不能使用了。

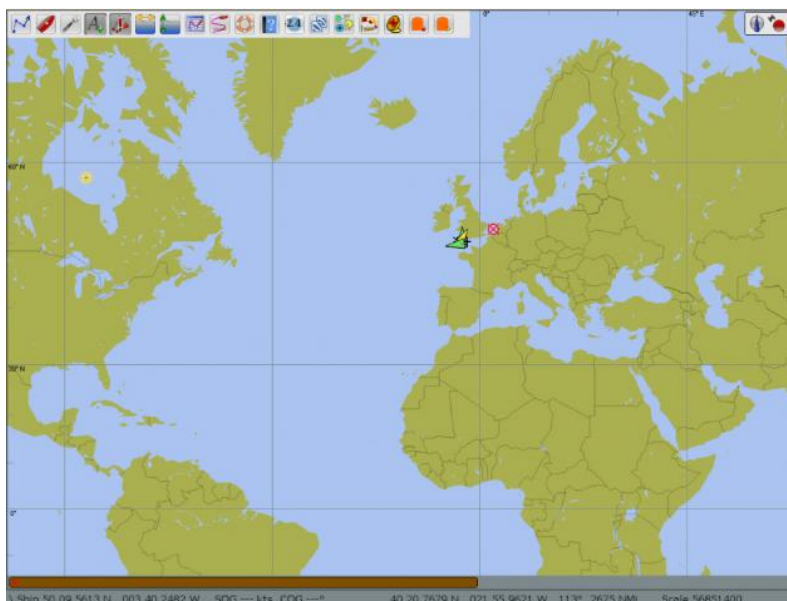
■ **更多信息请阅读：**http://opencpn.org/ocpn/setting_options#connections

进行两项修复

■ **Win7/64 多产 USB 串行适配器。**比如 BU-353 GPS hockey puck，掉线一段时间后不能自动连接的问题得到解决

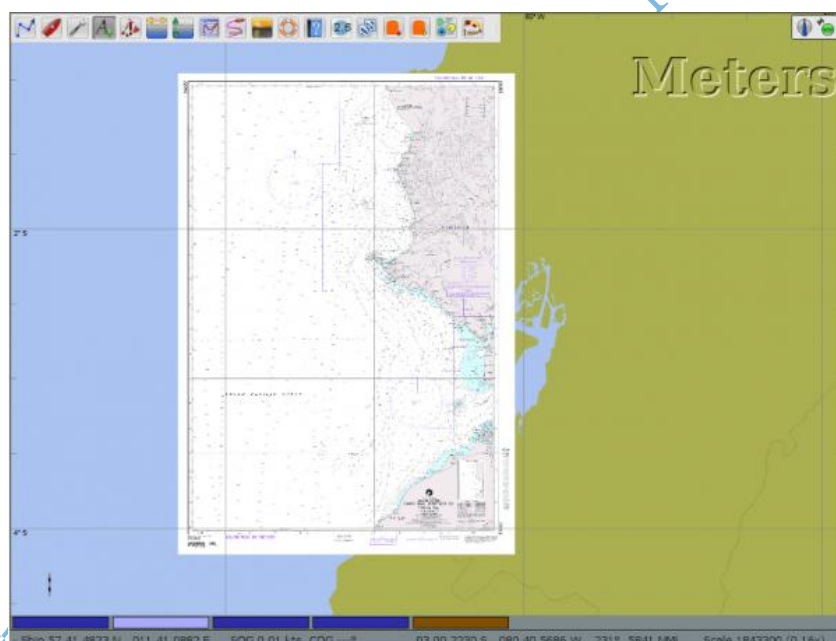
■ **Linux PortAudio 声音二进制文件，解决 Linux 声音问题。**现在 Linux 上的声音功能可以很好地发挥

新的背景世界地图



实现新的世界海图作为小比例尺背景。老式黑色带有蓝色海岸线的地图不用了。

进一步提高单一海图模式显示和缩放功能



■ 单一海图模式显示改变：不再限制单一海图中海图周长。也就是说光栅海图中所有信息都可以简单读取。

■ 不再限制缩小：整张海图都可以在单一海图模式下显示。

■ 实现多个部分 CM93 海图数据集支持。现在可以只加载有用区域而不用加载整张海图。

更多信息请阅读：http://opencpn.org/ocpn/installing_charts

矢量海图中照明区扩展

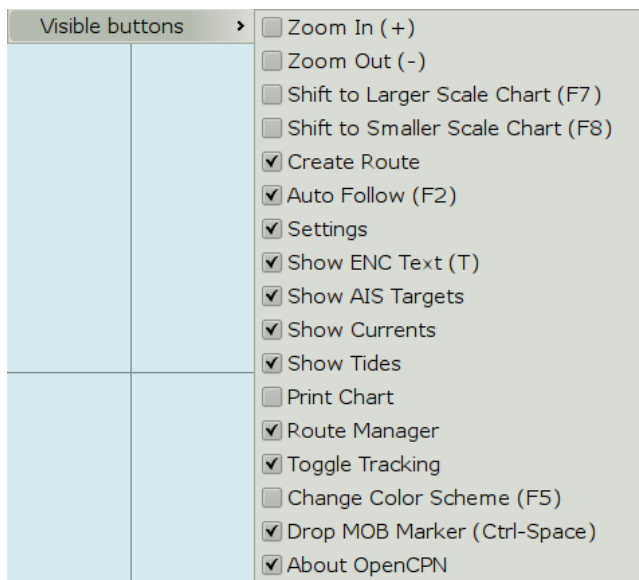


■ 鼠标悬停 S57 照明区：该项功能非常不错。一个特殊灯的功能非常清晰。该照明区对该灯的整体标称范围进行了扩展。

■ 很多矢量海图上的灰色区域没有了。

更多信息请阅读：http://opencpn.org/ocpn/vector_charts

只显示你需要的工具栏！



■ 实现工具栏图标可视化选择对话框

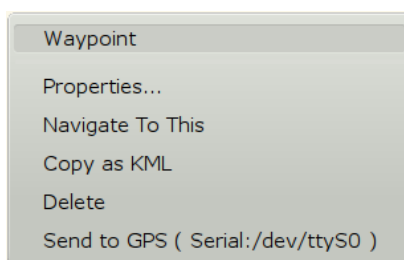
选择你喜欢的显示主题！



■ 主题支持+2 个新主题

更多信息请阅读 http://opencpn.org/ocpn/toolbar_buttons

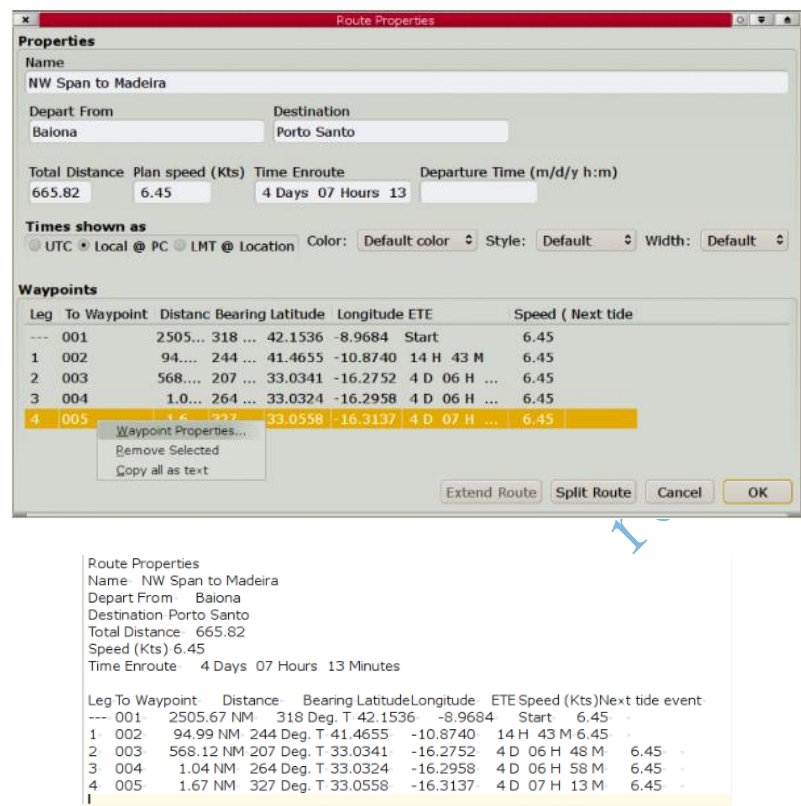
集中式上下文菜单 (Focused Context Menus)



■ 将上下文菜单调整成集中式上下文菜单（Focused Context Menus）。对普通上下文菜单进行进一步发展，以避免屏幕上出现大量大型菜单，而这些内容都是当前任务执行不需要的。

更多信息请阅读：http://opencpn.org/ocpn/right_click_menu

复制和打印航线和航迹



■ 复制到剪贴板航线属性和航迹对话框

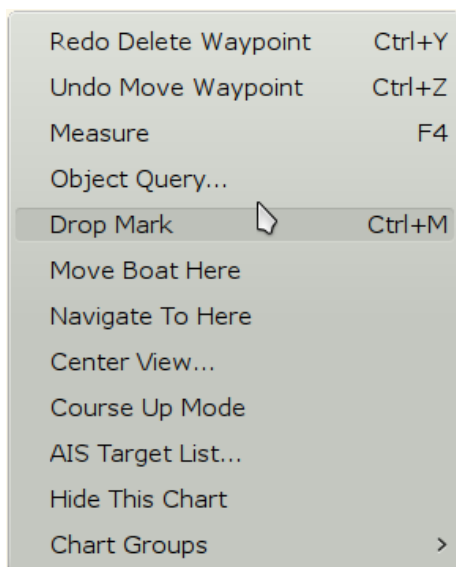
Name	Position	Course	Distance	Description
001	55 55.8 N 012 40.4 E	107 Deg	110.62 NM	
002	55 55.3 N 012 42.9 E	121 Deg	1.44 NM	
003	55 54.9 N 012 44.1 E	186 Deg	0.80 NM	
004	55 54.1 N 012 44.0 E	204 Deg	0.78 NM	
005	55 53.3 N 012 43.3 E	263 Deg	0.88 NM	
006	55 53.2 N 012 41.8 E	322 Deg	0.84 NM	
007	55 53.8 N 012 41.0 E	316 Deg	0.76 NM	
008	55 55.0 N 012 39.0 E	357 Deg	1.65 NM	
009	55 55.6 N 012 38.9 E	346 Deg	0.59 NM	
010	55 57.7 N 012 38.0 E	000 Deg	2.14 NM	

■ 打印航线

双击打开航线等对象的属性窗口

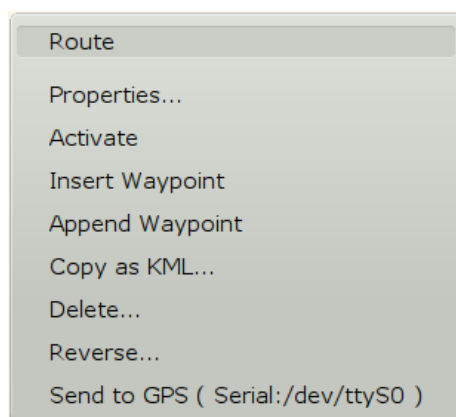
■ 实现航路点，航线和航迹的双击 query

撤销和重做一些操作



■ 实现航路点创建/移动/删除的撤销/重做功能 (ctrl-Z, ctrl-Y)

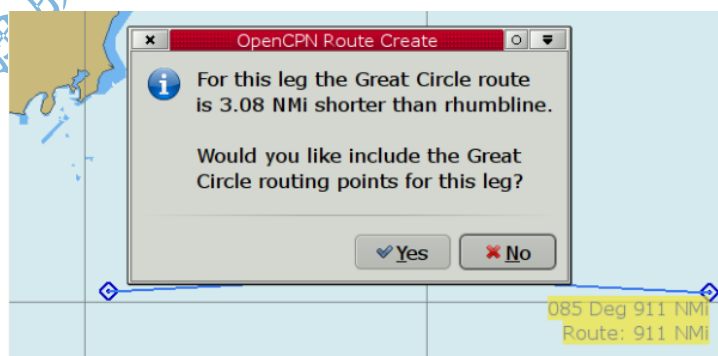
与 qtVlm 和 Google Earth 共同使用



■ 实现 KML 对象复制/粘贴功能的加强。它主要用来将优化过的航线从 QtVLM 复制到 OpenCPN，以及通过 GE 交换航线和标记。

更多信息请阅读: http://opencpn.org/ocpn/the_route_manager

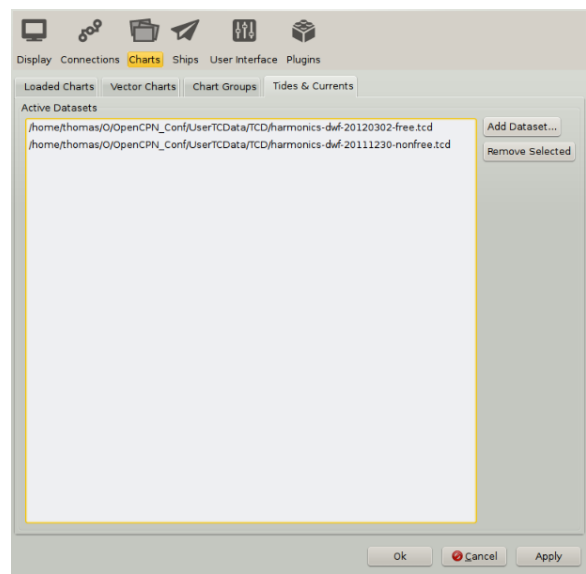
内置大圆航行处理



■ 实现自动/动态单元航行线路创建

更多信息请阅读: http://opencpn.org/ocpn/great_circle_sailing

阅读 tcd 潮汐文件格式

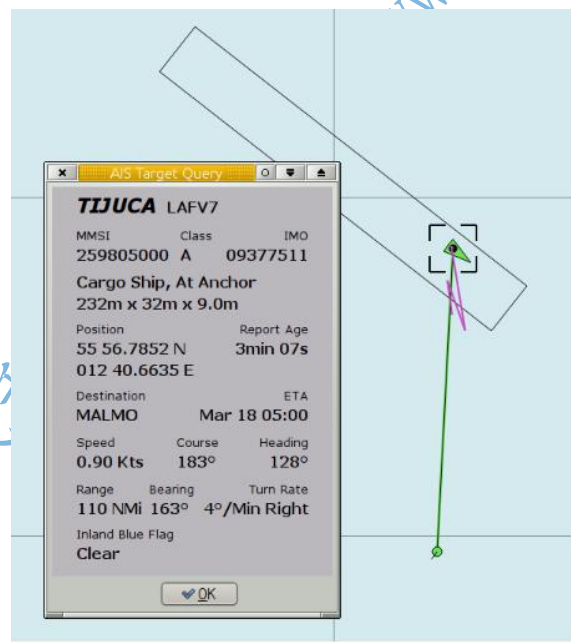


■ 实现二进制（.tcd）潮汐/潮流数据库支持。其优势在于可以直接导入 x-tides 经常更新的文件。已覆盖的区域没有的发到扩展。

■ 提高潮汐/潮流管理器和工具栏 UI 以允许同时存在的 HARMONIC 数据集。

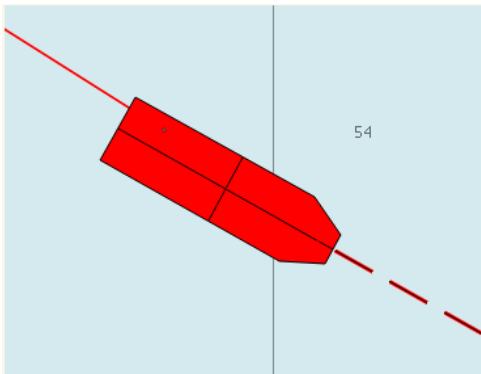
更多信息请阅读：http://opencpn.org/ocpn/tides_and_currents

显示 AIS 目标真实尺寸

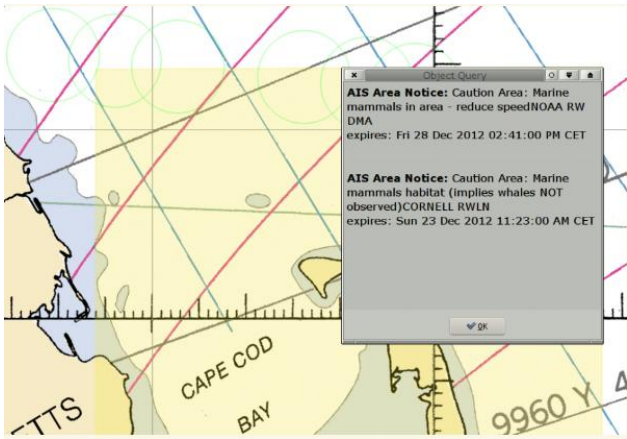


■ 实现 AIS 目标真实尺寸最优显示。放到足够大的时候就可以显示真实尺寸。

显示本船尺寸

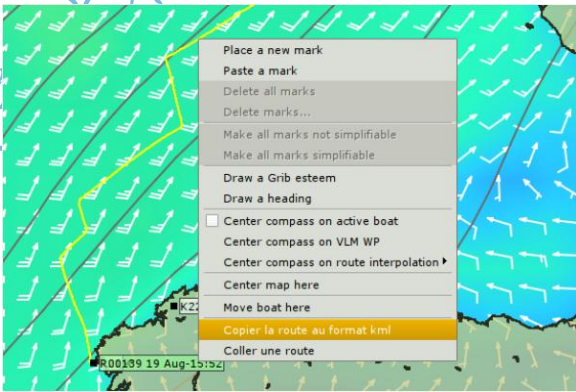


■ 我们现在可以通过不同方式显示本船。GPS 尺寸和位置很容易调整。看到上面的黑点了吗？这就是 GPS 的位置。
提高 AIS 区域通知 (Notices)



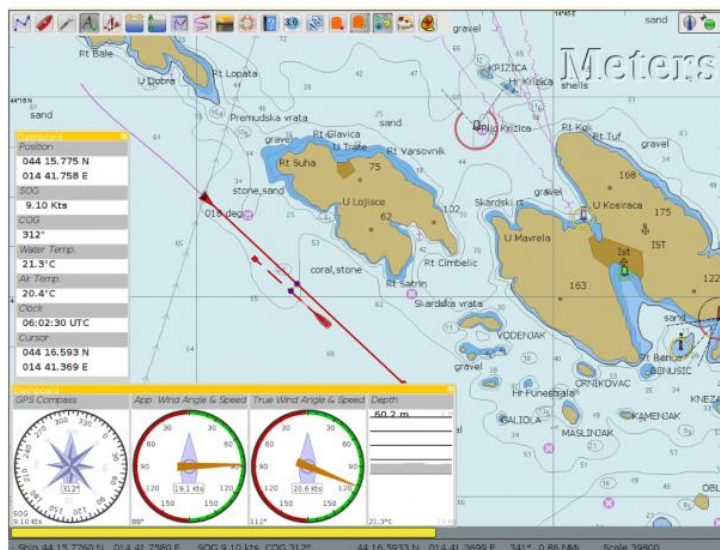
■ 提高 AIS 区域通知 query 显示。AIS 二进制信息显示的提高。
■ 很多 AIS 更新和更正。因为对细节的很多关注，OpenCPN 现在有很多 AIS 功能。
更多信息请阅读：<http://opencpn.org/ocpn/ais>

优化你的航线



■ 实现 QtVlm 航线优化设备相关特定界面。现在可以通过当前 GRB 文件及本船气象导航。优化 QtVlm 中的航线然后导入 OpenCPN。
更多信息请阅读：http://opencpn.org/ocpn/weather_routing

仪表板大幅加强

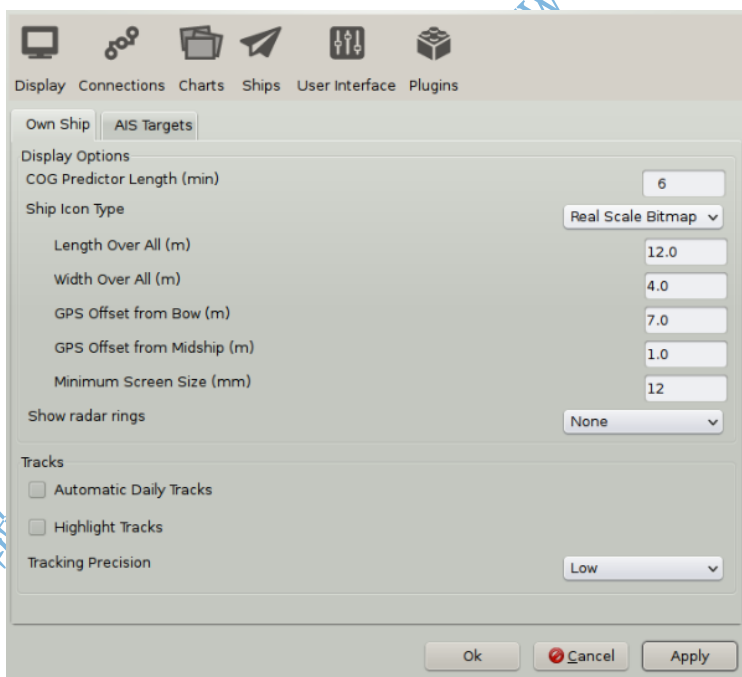


■ 往仪表板插件中添加新仪器

■ 完成仪表板插件图形重设计

更多信息请阅读: http://opencpn.org/ocpn/dashboard_plugin

实现智能追踪及本船尺寸调整



■ 实现新的智能动态追踪计算

■ 实现本船大小调整对话框。包括 GPS 接受器偏移量

更多信息请阅读: http://opencpn.org/ocpn/setting_options#own

注意细节

■ 很多各种 Mac 更新, 使 Mac 能够流畅运行

■ Garmin 和 Furuno: GPS 非常注重细节