



图形化翻译操作指南

November 18, 2021

Contents

1 所需的文件和工具	2
1.1 下载 PoEdit	2
1.2 下载 KiCad 源	2
1.3 下载现有的翻译和文档	2
2 查找要翻译的句子	2
3 用于翻译的 KiCad 树	3
3.1 词典树	3
3.2 搜索路径	3
3.3 文件	4
4 使用 poedit	4
4.1 安装	4
4.2 KiCad 准备	4
4.3 PoEdit 配置	5
4.4 项目配置	6
4.5 路径和文件配置	7
4.6 关键字配置	8
4.7 保存项目	8
5 创建或编辑词典	8
6 在 KiCad 源代码中添加新的语言条目（仅限开发人员）	9
6.1 步骤	11
6.1.1 在 include/id.h 中添加新的 id。	11
6.1.2 添加新图标（仅限审美目的）	11
6.1.3 Editing bitmaps_png/CMakeLists.txt	11
6.1.4 Editing include/bitmaps.h	12
6.1.5 Editing common/edaappl.cpp	12
6.1.6 重新编译	13

参考手册

Copyright

本文件版权所有 (c) 2010-2015, 作者如下。您可以根据 GNU 通用公共许可证 (<https://www.gnu.org/licenses/gpl.html>)、第 3 版或更高版本或知识共享署名许可证 (<https://creativecommons.org/licenses/by/3.0/>) 的条款分发和/或修改它。

本指南中的所有商标均属于其合法所有者。

Contributors

Jean-Pierre Charras, Fabrizio Tappero, Wayne Stambaugh.

翻译

taotieren <admin@taotieren.com>, 2019

Telegram 简体中文交流群: https://t.me/KiCad_zh_CN

反馈

请将任何错误报告、建议或新版本引导到此处:

- 关于 KiCad 文档: <https://gitlab.com/kicad/services/kicad-doc/issues>
- 关于 KiCad 软件: <https://gitlab.com/kicad/code/kicad/issues>
- 关于 KiCad 软件 i18n: <https://gitlab.com/kicad/code/kicad-i18n/issues>

出版日期和软件版本

2015 年 5 月 30 日发布。

1 所需的文件和工具

创建和/或维护翻译不需要任何 C++ 编程技能: 在 KiCad 文件中没有任何更改。

翻译很容易使用工具 **PoEdit** 定位 (在 KiCad 源中) 要翻译的句子, 并且能够使用此工具创建的翻译为 KiCad 创建字典。因此, 您需要安装 PoEdit, 并获取最新的 KiCad 源, 并且对于现有翻译, 请获取最新的翻译。翻译可以在 Linux, Window 或 MacOSX 下进行。

1.1 下载 PoEdit

请参见: <https://www.poedit.net/>

1.2 下载 KiCad 源

KiCad 源当前托管在快速启动板上:

<https://launchpad.net/kicad>

可以使用名为 “bazaar” (命令中的 bzt) 的工具从启动板下载文件。所以:

- 安装名为 bazaar 的工具 (易于在所有平台下安装), 如果尚未完成): 请参阅 <https://bazaar.canonical.com/>
- 下载 KiCad 源使用命令 bzt 分支 lp: kicad <directory where sources files are copied>
- 您可以在 http://docs.kicad.org/zh/gui_translation_howto.html 上的文档中找到有关翻译和 PoEdit 配置的此文档

1.3 下载现有的翻译和文档

KiCad 翻译和文档也托管在 github 上: <https://gitlab.com/kicad/code/kicad-i18n/>

使用命令下载翻译:

```
git clone https://gitlab.com/kicad/code/kicad-i18n.git
```

2 查找要翻译的句子

KiCad 中的不同菜单和工具提示是国际化的, 可以很容易地翻译成本地语言 而无需修改源代码。

规则是:

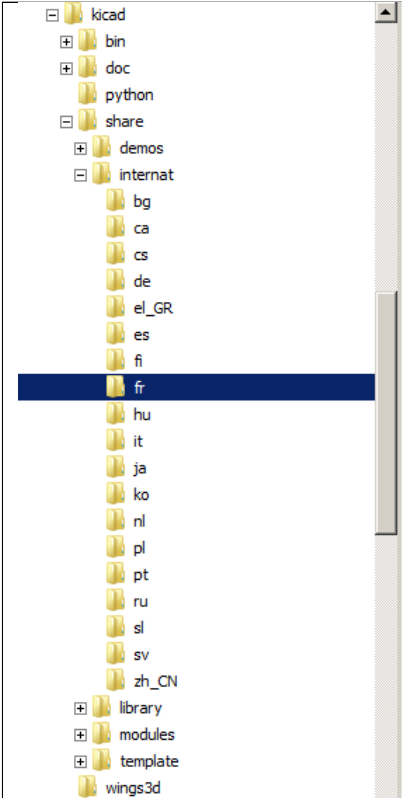
- 它们是用英语写的。
- 必须翻译的所有字符串都写成: _(“hello world”), 并显示 “hello world” 但是如果在显示之前发现字典翻译成语言环境语言。
- 词典英语 > 区域设置句柄翻译 (一种词典按语言)。

创建和维护词典英语 > 区域设置的更简单方法是使用 **poedit**。PoEdit 扫描 KiCad 源, 并允许您输入翻译。您必须下载 KiCad 源并设置 PoEdit 才能创建翻译。

3 用于翻译的 KiCad 树

3.1 词典树

KiCad 只有在合适的路径中才会找到该词典：



一个

合适的路径是 `kicad/internat/xx`,
或 `kicad/internat/xx_yy`
使用: `xx` = 标准化区域设置指示符 (缩写), 如:
* `fr` = 法语 * `en` = 英语 * `es` = 西班牙语 * `pt` = 葡萄牙语
或: `xx_yy` = 规范化的区域设置指示符 (长格式), 如:
* `fr_FR` * `en_GB` * `en_US`

3.2 搜索路径

按此顺序搜索词典和联机帮助文件：

- 在规范化区域设置指示器中的路径（长格式）（`kicad/internat/xx_yy`）
- 在规范化区域设置指示器（简短形式）中的路径（`icad/internat/xx`）

对于在线帮助文件，搜索范围包括：

- 在规范化区域设置指示器中的路径（长格式）（`kicad/help/xx_yy`）

- 在规范化区域设置指示器（简短形式）中的路径（kicad/help/xx）
- kicad/help/en
- kicad/help/fr

Note

从二进制路径检索的主要 KiCad 路径，或（如果未找到）：

在 WINDOWS 下：

- c:\kicad
- d:\kicad
- c:\Program Files\kicad

在 LINUX 下：

- /usr/share/kicad
 - /usr/local/share/kicad
 - /usr/local/kicad/share/kicad
 - /usr/local/kicad
-

3.3 文件

在每个目录中有 2 个文件 **kicad/internat/xx**：

- internat.po（字典文件）
- internat.mo（PoEdit 工作文件）

4 使用 poedit

4.1 安装

下载并安装 PoEdit (<https://www.poedit.net>)。PoEdit 存在于 Windows, Linux 和 Mac OS X 中。

下载并解压缩 KiCad 源代码。

4.2 KiCad 准备

KiCad 源：在这个示例中，文件位于 f:/kicad/。要翻译的所有字符串都标记为 （“要翻译的字符串”）。

poedit 必须搜索 （下划线）符号才能找到这些字符串。

必须在 KiCad 中添加适合字典的目录（**kicad/share/internat/xx**）。在此示例中，目录为 **kicad/share/internat/fr**。

4.3 PoEdit 配置

运行 PoEdit。

运行文件/新目录。

你应该看到类似的东西：

编目属性

翻译属性

源路径

源关键字

项目名称和版本:

翻译的项目名称

语言团队:

团队名称和电子邮件地址或 URL

语言:

中文

复数形式:

☒ 使用默认语言

☐ 使用自定义表达式

nplurals=1; plural=0;

[了解复数形式](#)

字符集:

UTF-8 (推荐)

源代码字符集:

高级提取设置...

确定

取消

4.4 项目配置

编目属性

翻译属性 | 源路径 | 源关键字

项目名称和版本:

语言团队:

语言:

复数形式: ☒ 使用默认语言
☐ 使用自定义表达式

[了解复数形式](#)

字符集:

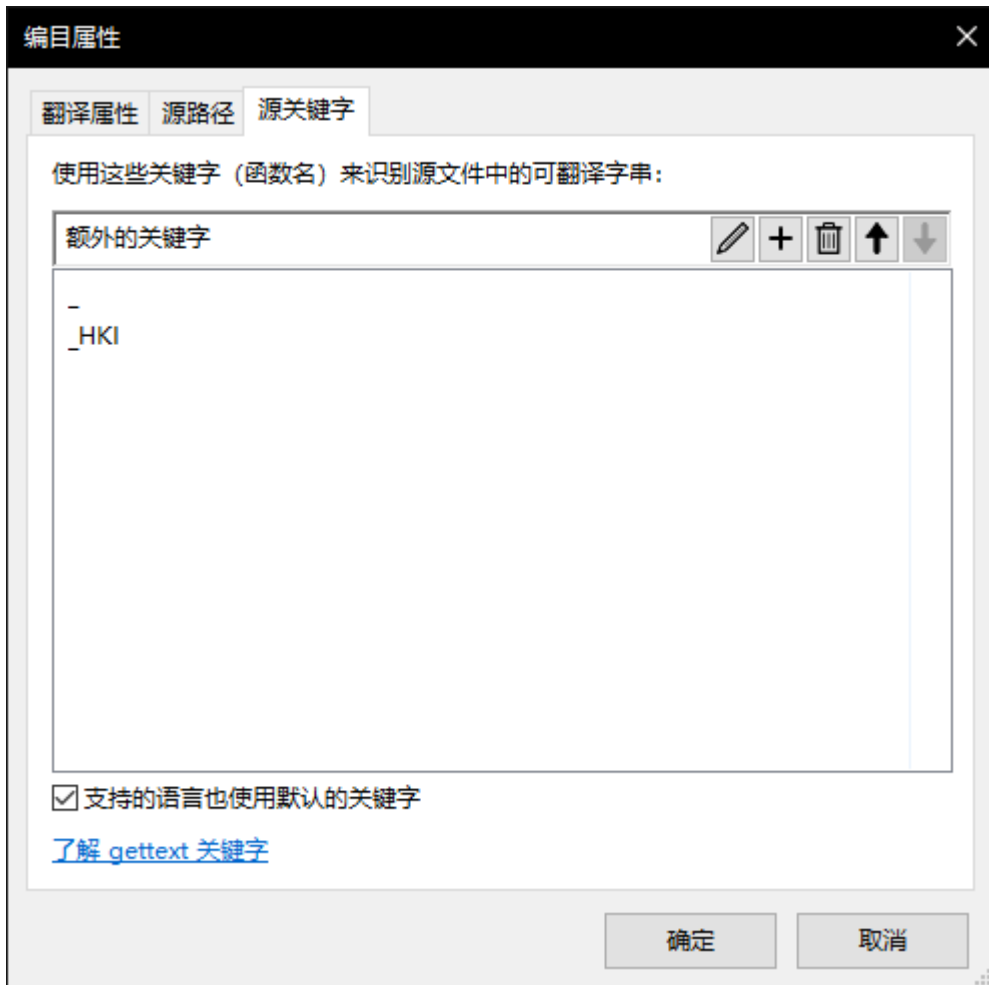
源代码字符集:

源文件是英文的，因此无需为源代码选择一些内容。

4.5 路径和文件配置



4.6 关键字配置



要在此处输入几个关键字：

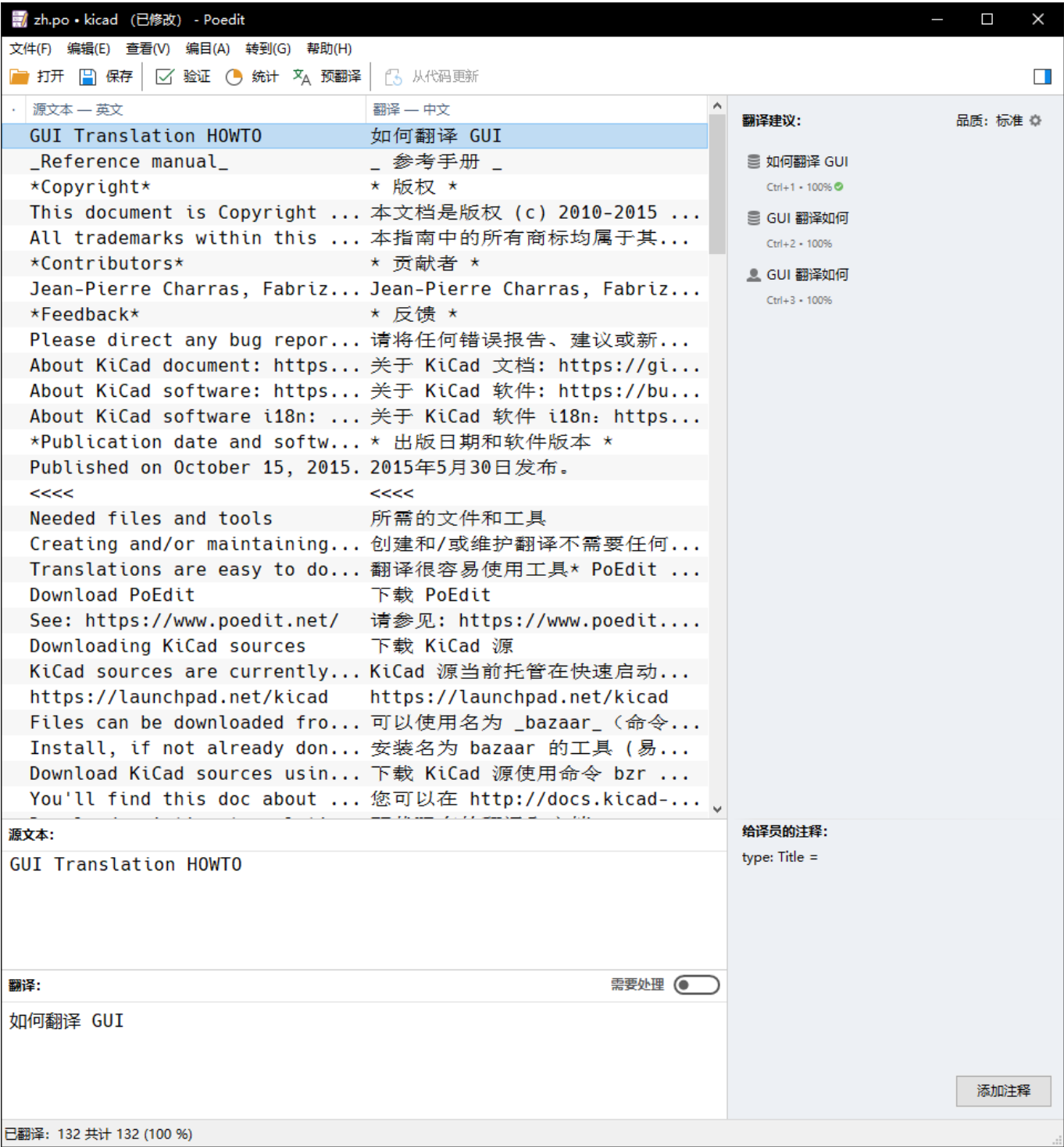
- _（下划线）用作通用源文件中的标记
- _HKI 用作热键描述翻译的标签。

4.7 保存项目

将新项目保存在 `kicad/share/internat/xx` 中，名称为 `kicad.po`。

5 创建或编辑词典

运行 PoEdit 并加载项目（此处：`kicad.po`）。



运行命令 目录/从源更新。

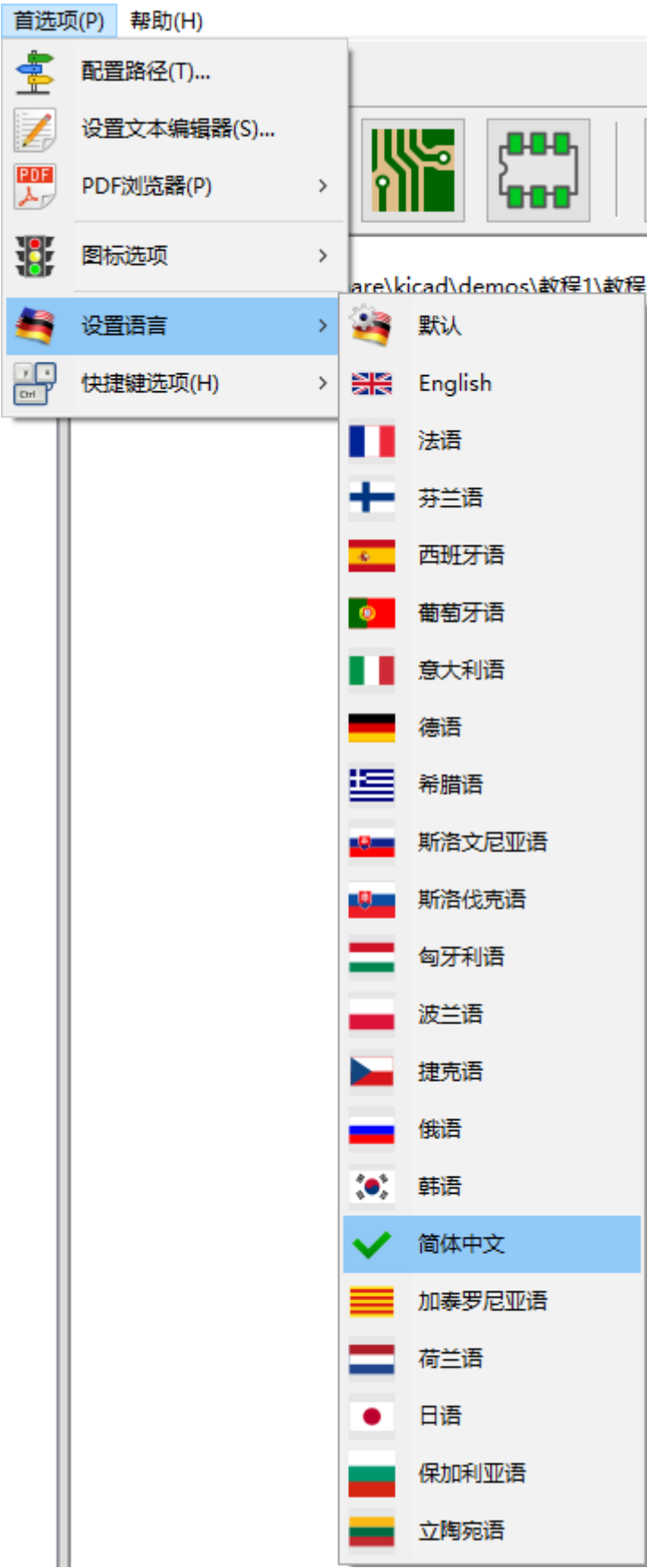
新的字符串（尚未翻译）将显示在窗口的顶部。

6 在 KiCad 源代码中添加新的语言条目（仅限开发人员）

此步骤不需要。它仅对开发人员有用，仅用于测试目的。

在 KiCad 中，我们可以强制使用所使用的语言。

强烈建议使用默认语言。



但由于开发人员必须测试翻译，因此语言列表中的新条目可用于测试目的。

6.1 步骤

6.1.1 在 `include/id.h` 中添加新的 `id`。

→ 在 `include/id.h` 中，找到如下序列：

```
ID_LANGUAGE_CHOICE,
ID_LANGUAGE_DEFAULT,
ID_LANGUAGE_ENGLISH,
ID_LANGUAGE_FRENCH,
ID_LANGUAGE_SPANISH,
ID_LANGUAGE_GERMAN,
ID_LANGUAGE_RUSSIAN,
ID_LANGUAGE_PORTUGUESE,
```

并在列表中添加一个新条目（稍后将在菜单中使用），如：

在 `ID_LANGUAGE_CHOICE_END` 之前 `ID_LANGUAGE_MY_LANGUAGE`。

6.1.2 添加新图标 (仅限审美目的)

→ 在 SVG 中创建一个新图标（例如使用 Inkscape）格式：通常是国家标志。例如 `lang_new.svg`

其他语言图标位于 `common/bitmaps_png/source` 中

6.1.3 Editing `bitmaps_png/CMakeLists.txt`

→ 找到文本：

```
lang_catalan
lang_chinese
lang_bg
lang_cs
lang_def
lang_de
lang_en
lang_es
lang_fr
lang_fi
lang_gr
lang_hu
lang_it
lang_jp
lang_ko
lang_nl
lang_pl
```

```
lang_pt
lang_ru
lang_sl
```

并添加新文件名（不带扩展名）: lang_new

6.1.4 Editing include/bitmaps.h

→ 找到文本:

```
EXTERN_BITMAP( lang_bg_xpm )
EXTERN_BITMAP( lang_catalan_xpm )
EXTERN_BITMAP( lang_chinese_xpm )
EXTERN_BITMAP( lang_cs_xpm )
EXTERN_BITMAP( lang_def_xpm )
EXTERN_BITMAP( lang_de_xpm )
EXTERN_BITMAP( lang_en_xpm )
EXTERN_BITMAP( lang_es_xpm )
EXTERN_BITMAP( lang_fr_xpm )
EXTERN_BITMAP( lang_fi_xpm )
EXTERN_BITMAP( lang_gr_xpm )
EXTERN_BITMAP( lang_hu_xpm )
EXTERN_BITMAP( lang_it_xpm )
EXTERN_BITMAP( lang_jp_xpm )
EXTERN_BITMAP( lang_ko_xpm )
EXTERN_BITMAP( lang_nl_xpm )
EXTERN_BITMAP( lang_pl_xpm )
EXTERN_BITMAP( lang_pt_xpm )
EXTERN_BITMAP( lang_ru_xpm )
EXTERN_BITMAP( lang_sl_xpm )
```

并添加一行以包含名为 lang_new_xpm 的新图标名称（_xpm 已添加到文件名中）。

6.1.5 Editing common/edaappl.cpp

→ 找到:

```
struct LANGUAGE_DESCR
{
    int            m_WX_Lang_Identifier;           // wxWidget locale identifier (see ↩
    wxWidget doc)
    int            m_KI_Lang_Identifier;           // kicad identifier used in menu ↩
    selection (see id.h)
    const char**   m_Lang_Icon;                   // the icon used in menus
    const wxChar*  m_Lang_Label;                   // Label used in menus
    bool           m_DoNotTranslate;              // set to true if the m_Lang_Label ↩
    must not be translated
};
```

```
#define LANGUAGE_DESCR_COUNT 14
static struct LANGUAGE_DESCR s_Language_List [LANGUAGE_DESCR_COUNT] =
{
    {
        wxLANGUAGE_DEFAULT,
        ID_LANGUAGE_DEFAULT,
        lang_def_xpm,
        _( "Default" )
    },
    {
        wxLANGUAGE_ENGLISH,
        ID_LANGUAGE_ENGLISH,
        lang_en_xpm,
        wxT( "English" ),
        true;
    },
    {
        wxLANGUAGE_FRENCH,
        ID_LANGUAGE_FRENCH,
        lang_fr_xpm,
        _( "French" )
    },
    {
```

并添加一个新条目，如：

```
{
    wxLANGUAGE_MY_LANGUAGE,
    ID_LANGUAGE_MY_LANGUAGE,
    lang_new_xpm,
    _( "My_language" )
},
```

wxLANGUAGE_MY_LANGUAGE 是国家/地区的 wxWidgets 语言标识符（请参阅 wxWidget 文档）。

6.1.6 重新编译

你应该是一个 PNG 维护者（参见 `bitmaps_png/CMakeLists.txt` 文件），即用选项 `MAINTAIN_PNGS` 编译 KiCad 显然，这是下一个也是最后一步。