

自由软件中文化工作指南(L10N)

Aron Xu <happyaron.xu@gmail.com>

Revision 1.1 2009-11-07 更新格式信息

版本历史

Revision 1.0 2009-08-26 初始版本，以网页形式发布于 i18n-zh

内容简介

本文是自由软件中文化过程中的工作指南，是对原工作指南进行了进一步修订，删除了其中陈旧的部分并增加了一些新的内容，主要针对翻译过程中的流程、格式等事项进行了说明。

欢迎对本文内容进行讨论，相关的历史背景和联系方式请参阅文末。

版权信息

Copyright © 2009 by Aron Xu. 本文在 [Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported](https://creativecommons.org/licenses/by-nc-sa/3.0/) 许可下发布，转载时请注明版权信息和获得本文的方法，同时以类似许可再次发布，禁止商业使用。

一、工作流程

不同的翻译项目有不同的工作流程，大致分为准备、协调、翻译和提交。

1. 准备

- 我们的原则是质量优先，强烈不赞成翻译自己不熟悉的软件或文档；
- 仔细阅读本文档和全部与工作相关和要翻译内容相关的文档；
- 最好准备一本英语词典或者使用在线词典以方便查阅，术语、缩写等请先搜索 Google、查阅 Wikipedia 文章以及 Answer.com 上的内容；
- 使用在线翻译工具的(如 Launchpad)可以开始进行翻译；
- 不使用在线翻译工具的，请依照各项目所指定的方法取出源代码、获得 PO 或 POT 文件，然后安装指定的工具并开始翻译(如翻译 PO 文件强烈建议安装 gettext 工具集)。

2. 协调

协调工作的主要目的是防止多个人在互不知情的情况下同时翻译相同的内容，一方面导致了重复劳动，另一方面也造成了合并的困难。

通常情况下的协调工作是通过向小组的邮件列表发送电子邮件说明你要负责的工作，若要翻译的软件或文档不存在则自动成为其协调人开始负责其翻译。若已经有人在 进行则此后需要与之联系以商定分工，请在发送邮件到邮件列表时抄送给原负责人一份，一周内无回复则视为同意你的请求。开始翻译前若需要相关的帐号应提出申请。GNOME 由于使用了 Damned-Lies 站点，请直接使用站点功能；对于在线翻译，如 Launchpad，暂时还没有进行协调工作的有效方式。

3. 翻译

(1) 关于 GNU Gettext

GNU Gettext 是 GNU 本地化和国际化函数库，即 GNU Localization(l10n) & Internationalization (i18n) Library，是自由软件国际化支持的常用工具。

(2) 开始翻译

翻译是我们中文化工作中的重点，在开始翻译前请仔细阅读本文以下章节中的各项要求，保证质量。

PO 文件的翻译过程中不要对 msgid 行做任何修改，即使其中有明显错误，否则合并翻译到主程序的程序将会遇到问题。您在英文字符串上做的更改将会丢失，您的翻译也不会被合并入内。在遇到源文件有问题时最好的办法是向软件开发者提交 Bug 报告。

本文中关于 PO 文件的各项命令都假定读者已经安装了 GNU/Linux 基本系统、gettext 工具集和 intltool 工具集，或者在其他平台上安装了相应的程序。

(3) 检查翻译格式

PO 文件翻译后请使用以下命令进行格式检查：

```
msgfmt --statistics -cv zh_CN.po
```

运行以上命令后会在当前目录下生成一个名为 messages.mo 的文件，此文件便是编译后的机读文件。

(4) 复审翻译

使用 PO 文件的，直接复审 PO 文件，并建议将 PO 文件编译为 MO 文件安装在相应程序的目录下进行实测。

对于文档，构建文档后精读结果文档。

如果发现问题，回到“开始翻译”，修正问题。

(5) 格式化翻译

因为不同的翻译工具可能使用了不同的格式，比如标准 `gettext` 格式、`poedit` 格式。强烈建议所有翻译者使用标准 `gettext` 格式，可以使用标准的 POT 文件与 PO 文件进行合并来达到此目的：

```
msgmerge --no-wrap -U zh_CN.po example.pot
```

4. 提交

不同项目提交翻译的方式不完全相同，多数情况下可以通过翻译小组的邮件列表进行提交，如果您有相关项目的 DVCS 帐号则请直接提交。如果没有负责的小组也没有相关的帐号，请把您的翻译发送到对应项目的翻译者邮件列表(一般名称类似 `xxx-translators`、`i18n`、`intl`、`l10n`；若没有翻译者邮件列表则发送到文档邮件列表，一般名称类似为 `xxx-doc`；若还是没有则发送到开发者邮件列表，一般名称为 `xxx-dev`、`devel`、`developer`)，或者通过填写一个 `Bug` 的方式进行提交。

注意：在请别人代为提交到 TP 项目时 `Last-translator` 不会是您，因为 TP 要求这项需要是提交者的名字和邮件地址，您的信息将会出现在文件头注释域的版权行中。

提交后请根据情况再发送一封邮件来取消原来协调翻译时的占用声明。

二、基本要求

1. 准确表述原文的意思；
2. 中文应该意思清晰且符合中文表达习惯；
3. 原文如果表达不清晰，中文应该意译，并且应根据上下文和注释进行推断并填补相应的信息；
4. 情况 3 不能太多；
5. 对同样短语的翻译，前后必须一致。

三、格式要求

1. 文件编码格式

目前的翻译文件一般使用 `UTF-8` 编码。所以请使用支持 `UTF-8` 编码的编辑工具或者文本编辑器进行翻译，同时需要注意文件头的“`charset`”项设置正确。如果在编码格式上遇到问题，请向翻译组相关邮件列表发送邮件以寻求帮助。

可以使用 `iconv` 或者 `msgconv` 命令进行转换，以下例子为转换 `gb2312 --> utf-8`：

```
iconv -f 源文件编码 -t 目标文件编码 源文件 > 目标文件
iconv -f gb2312 -t utf-8 inputfile > outputfile
```

或者：

```
msgconv -t 目标文件编码 源文件 -o 目标文件
msgconv -t utf-8 inputfile -o outputfile
```

2.关于文件头

下面是一个使用 `msginit` 工具由 POT 文件生成的 PO 文件的初始文件头，使用了 `msginit` 的 `--no-translator` 选项：

```
# Chinese translations for PACKAGE package
# PACKAGE 软件包的简体中文翻译.
# Copyright (C) 2009 Free Software Foundation, Inc.
# This file is distributed under the same license as the PACKAGE package.
# Automatically generated, 2009.
#
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: bug-coreutils@gnu.org\n"
"POT-Creation-Date: 2009-08-20 17:27+0200\n"
"PO-Revision-Date: 2009-08-20 17:27+0200\n"
"Last-Translator: Automatically generated\n"
"Language-Team: none\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=UTF-8\n"
"Content-Transfer-Encoding: 8bit\n"
"Plural-Forms: nplurals=INTEGER; plural=EXPRESSION;\n"
```

编辑后的文件头应为这样，绿色为实际文件中需要修改的，红色为注释：

```
# Chinese translations for coreutils package
#                                     <-- 将 PACKAGE 改为要翻译的软件名，本例中为 coreutils
# coreutils 软件包的简体中文翻译.   <-- 此处 PACKAGE 替换方法同上
# Copyright (C) 1998, 2002, 2004, 2005, 2009 Free Software Foundation, Inc.
#                                     <-- 填入翻译年份，若年份有多个，应当以逗号分割多个数字而不是使用某年至某年的方式
# This file is distributed under the same license as the coreutils package.
#                                     <-- 此处 PACKAGE 替换方法同上
# Yip Chi Lap <clyip@cs.hku.hk>, 1998.
# Abel Cheung <maddog@linux.org.hk>, 2002.
# Anthony Fok <anthony@thizlinux.com>, 2002.
# Funda Wang <fundawang@linux.net.cn>, 2004, 2005.
# Aron Xu <happyaron.xu@gmail.com>, 2009.
#                                     <-- 此处填入您自己的信息，有的 po 编辑器会自动更新此项，有多个年份的以逗号分隔列出
#
# fuzzy <-- 使用 msginit 生成的 po 文件中没有这行，如果直接用 POT 文件请删除此行
msgid ""
msgstr ""
"Project-Id-Version: coreutils 7.5\n"
#                                     <-- 此处 PACKAGE 替换方法同上，VERSION 为对应的软件版本
"Report-Msgid-Bugs-To: bug-coreutils@gnu.org\n"
#                                     <-- 此行一般不需修改，即使有的文件在生成后此项为空
"POT-Creation-Date: 2009-08-20 17:27+0200\n"
#                                     <-- 此行一般不需修改，合并新的 POT 文件时会自动更新
"PO-Revision-Date: 2009-08-24 22:25+0800\n"
#                                     <-- 填入翻译完成的时间，po 编辑器一般能自动更新此项，+0800 代表东 8 时区
"Last-Translator: Aron Xu <happyaron.xu@gmail.com>\n"
#                                     <-- 改成你的名字和电子邮件地址
"Language-Team: Chinese (simplified) <i18n-zh@googlegroups.com>\n"
#                                     <-- 以上这行改成你所在小组的地址，没有小组就填写可以联系到你的电子邮件地址
```

```
"MIME-Version: 1.0\n" <-- 不要进行更改
"Content-Type: text/plain; charset=UTF-8\n"
                <-- 应确保编码"charset"和文件编码均为 UTF-8
"Content-Transfer-Encoding: 8bit\n" <-- 不要进行更改
"Plural-Forms: nplurals=1; plural=0;\n" <-- 简体中文这样填即可, 有的文件可能无此行
```

修改完文件头以后接着就是实质性的翻译了, po 文件的格式是一句 `msgid` 跟着一句 `msgstr`, 以“#”开头的行是注释。您需要做的是把 `msgid` 中的英文翻译成中文写到 `msgstr` 中。例如:

```
#: app/floating_sel.c:198
msgid ""
"Cannot create a new layer from the floating\n"
"selection because it belongs to a\n"
"layer mask or channel."
msgstr ""
"无法从浮动选区创建新\n"
"图层, 因为它属于一个\n"
"图层蒙板或通道。"

#: app/gimphelp.c:194
msgid "Could not find GIMP Help Browser"
msgstr "找不到 GIMP 帮助浏览器"
```

3.关于复数形式

由于某些语言的单复数变化极为复杂, 为了应对这种局面, 符合国际化标准, `gettext` 通过读取 PO 文件头部如下一行中的信息来正确显示不同单复数的译文:

```
"Plural-Forms: nplurals=INTEGER; plural=EXPRESSION;\n"
```

对于我们来说, 正确的设置通常为:

```
Plural-Forms: nplurals=1; plural=0;
```

含义是译文的所有单复数形式只有一种, 所有形式使用一种翻译。

`nplurals` 表示译文单复数变化形式的总数量, 它是一个正整数。中文里一般没有单复数区分, 也就是说没有复数变化, 所以一般情况下 `nplurals=1`;

`plural` 表示原文中表示“n 个”的概念的时候应用复数形式, 这里是一个非负整数, 可以是一个 C 语言表达式的值, 该值必须小于 `nplurals` 且非负。对于中文而言, 可以取到的值只有 0, 于是 `plural=0`。

也有一些情况例如“他/她”以及“他们/她们”的时候中文也存在单复数形式的不同, 通常推荐用语言层面上的方法处理, 当然也不是没有对应的 `Plural-Forms` 设置, 这个时候一般为:

```
Plural-Forms: nplurals=2; plural=(n > 1);
```

含义是单复数一共有两种(一种单数和一种复数), 在所指对象数量超过一个的时候使用复数。

又如一些语言中只有在所指对象数量为一的时候使用复数(英语的默认设置):

```
Plural-Forms: nplurals=2; plural=(n != 1);
```

含义是单复数一共有两种(一种单数一种复数), 仅在数量为一个的时候使用单数(即数量不为有的时候使用复数)。

再举一个正文中的例子, 若原文如下:

```
#: src/msgcmp.c:338 src/po-lex.c:701
#, c-format
msgid "found %d fatal error"
msgid_plural "found %d fatal errors"
msgstr[0] ""
msgstr[1] ""
```

在文件头处设置中文最常用的复数形式后, 上段内容的译文应以下面格式写出:

```
#: src/msgcmp.c:338 src/po-lex.c:701
#, c-format
msgid "found %d fatal error"
msgid_plural "found %d fatal errors"
msgstr[0] "找到 %d 个严重错误"
```

也就是说多余的 `msgstr` 要删除。

4.特殊字符

(1) 标准 gettext 格式 PO 文件中的转义字符和取消转义

标准 `gettext` 格式中的转义字符同 C 语言中的基本相同, 常用的有以下几个:

`\n` 换行

`\t` 水平制表符

`\v` 竖直制表符

您不需要将同样数量的格式标记放在翻译中, 但是如果它们之一有在原文开始或者结束位置的时候, 您必须在翻译中使之包含在对应的开始或者结束之处。

如果要显示非转义字符, 则需要使用 `\` 来表明取消转义, 如 `"` (半角双引号)在 PO 文件中表示字符串的开始或者结束, 如果要在内容中使用该符号, 则需输入 `\`, 还可以参考以下例子:

```
\"\t\"
```

在运行时显示的结果是 `"t"`

(2)XML 中的角括号和“&”号

有一些模块中 XML 被频繁地使用, GConf 是最明显的。XML 将 `<` `>` 和 `&` 视为保留字符, 您不能将其直接添加到翻译中。必须将其实体以下面的形式表示:

`<` 代表 `<`; `>` 代表 `>`; `&` 代表 `&`。

(3) TRUE 和 FALSE

“TRUE”和“FALSE”多出现于 `gtk+` 和 `Gconf`, 以及很多 `Glade` 所生成的文件中(这些文件以 `glade` 为文件名结尾), 不要翻译它们。如果程序找不到它们, 将会出现问题。

5. 标点的使用

一般的原则是：除了小括号、省略号和破折号保留不变以外，都应该使用中文(全角)标点符号。英文标点符号后方常常跟随有一个半角空格，请在翻译成中文标点符号时将其去除。

1. 英文中的 `,` 在中文中可能是 `,` 或者 `、`
2. 英文中的 `.` 在中文中应该是 `,` 或者 `。`，视上下文而定，多数是 `。`
3. 英文中的 `\"%s\"` 在 GUI 程序中应该翻译为 `"%s"`，而不是 `\"%s\"` 或者 `\“%s\”`，而且后者是不符合换码序列要求的。即在 GUI 程序中 ``something`` 和 `'something'` 以及 `\`something\`` 都应该翻译为 `“某事”`
4. 英文中的 `\"%s\"` 在 CLI 程序中应保持为 `\"%s\"`，因为全角引号在文本界面下显示不够美观，所以使用半角双引号，即在 CLI 程序中 ``something`` 和 `'something'` 以及 `\`something\`` 都应译作 `\`某事\``
5. 英文中的 `:` 应该翻译为 `:` (全角)而不是 `:` (半角)，而作为分隔符时(例如时间)，`:` 保留为英文(半角)的，因为这个时候不是标点符号
6. 英文中的 `()` 应该保持不变。由于全角小括号 `()` 很难看，也占地方，所以一律使用半角小括号 `()`
7. 英文中的 `...` 应该保持不变。由于翻译的时候常常难以分清哪些条目是菜单项，哪些条目是一般语句，而后者才能使用中文的省略号 `……`，所以现在统一翻译为 `...`
8. 英文中的 `--` 应该保持不变。由于全角破折号 `——` 兼容性不好，有时显示为两个方格，所以不再使用。
9. 遇到 `%q` 标记的时候，代表此标记显示的是一段引用内容，程序运行时将自动在其两端加上双引号，故不需另加引号。

6. 关于空格

为了美观，通常建议在中文与英文、中文与阿拉伯数字、英文与阿拉伯数字之间加入一个半角空格。例如：

```
msgid "Installing driver for %1"
msgstr "正在安装 %1 的驱动程序"

msgid ""
"Parameter start_num specifies the character at which to start the search. "
"The first character is character number 1. If start_num is omitted, it is "
"assumed to be 1."
msgstr ""
"参数 start_num 指定开始搜索的字符位置。第一个字符序号为 1。如果省略 "
"start_num, 默认它为 1。"
```

对于 CLI 程序，因为文本界面字体显示的原因，一般推荐汉字后接英文时不留空格，英文后接汉字时留一个空格，在显示的时候不会因为英文前后都有空格而造成英文前面的空余空间远大于后面造成的不美观，例如：

```
msgid "Set LC_ALL='C' to work around the problem."
msgstr "请设置 LC_ALL='C' 以避免出现问题。"
```

对于小括号和全角双引号，其两侧不加空格：

```
msgid "Original idea and author (KDE1)"
msgstr "原始创意和作者(KDE1)"
```

```
msgid ""
"The APM Management subsystem seems to be disabled.\n"
"Try executing \"apm -e 1\" (FreeBSD) and see if\n"
"that helps.\n"
msgstr ""
"APM 管理子系统似乎被禁用了。\n"
"试试执行“apm -e 1”(FreeBSD)并看看\n"
"是否有用。\n"
```

包含 XML/HTML 标签的条目，如要在标签中的内容两侧添加空格，请把空格置于标签外侧，否则空格可能显示不出来。

这是 `HTML` 的语法手册

7. 菜单项中快捷字符

快捷字符一律使用大写字母，用小括号括起来放到菜单文字的后面(如果有标点符号则放在标点符号的前面)。在 KDE 中，菜单快捷字符的前缀是“&”；在 GNOME 中，菜单快捷字符的前缀是“_”。但是如果翻译保留了原文的英文单词或阿拉伯数字，且该单词或数字正好是快捷键所在的单词或数字时，应保留原文的快捷键方式(如下面的第二、四个例子)。这里举几个例子：

KDE 菜单：

```
msgid "C&lear"
msgstr "清除(&L)"

msgid "&Glimmer Editor"
msgstr "Glimmer 编辑器(&G)"
msgstr "&Glimmer 编辑器"
```

GNOME 菜单：

```
msgid "_Setup..."
msgstr "设置(_S)..."

msgid "Get _CDDDB Now"
msgstr "现在读取 CDDDB(_C)"
msgstr "现在读取 _CDDDB"
```

注意：以下情况的翻译有点特别。由于“复制”和“剪切”均为“编辑”菜单的条目，只有这样翻译才能保证显示正确！

```
msgid "/_Edit"
msgstr "/编辑(_E)"

msgid "/Edit/C_opy"
msgstr "/编辑(E)/复制(_O)"

msgid "/Edit/C_ut"
msgstr "/编辑(E)/剪切(_U)"
```

8.关于编程语言格式和 msgctxt

翻译过程中常会遇到“c-format”一类的标记，-format 标记代表该段字符串需要按照指定的编程语言格式输出，最常见的就是按照 C 语言格式输出。其他比较常见的还有 python-format、scheme-format、perl-format、php-format、qt-format 和 kde-format 等。

msgctxt 标记(可以理解为 message context)是给出字符串所处的不同环境，通常情况下同一个 msgid 只能在一个 PO 文件中出现一次，但是在有不同的 msgctxt 标记存在时，一个文件中可以出现多个相同的 msgid，因为它们各自的上下文不同，所以可能有不同的译文。

9.翻译中参数的位置

有时候原来的参数顺序不符合中文的语法，一方面，翻译可以通过调整副词、语序等手法来符合中文习惯，另外一方面，在必要的情况下，需要改变参数的位置，例如在 KDE 中：

```
msgid "%1 articles match rule %2"
msgstr "匹配规则 %2 的文章有 %1 个"
```

如果是在 GNOME 中则应该这样写：

```
msgid "%d articles match rule %d"
msgstr "匹配规则 %2$d 的文章有 %1$d 个"
```

即用 1\$、2\$、3\$ 等符号标明参数在原文里出现的位置。同时，任何一个参数的顺序进行了调整，则在这一句译文中所有参数都必须注明原文位置，否则无法通过格式检查。

10.注意注释

文件中有时会有给翻译者的注释，以“#. 开头”，多数情况下还会有“TRANSLATORS”字样提示。通常是对所翻译内容的解释和提示，请在翻译过程中留意。例如：

```
#. TRANSLATORS: ls output needs to be aligned for ease of reading,
#. so be wary of using variable width fields from the locale.
#. Note %b is handled specially by ls and aligned correctly.
#. Note also that specifying a width as in %5b is erroneous as strftime
#. will count bytes rather than characters in multibyte locales.
#: src/ls.c:708
msgid "%b %e %Y"
msgstr "%b %e %Y"
```

当然，有一些文件内主要是地名或国家名，开发者会省略“TRANSLATORS”直接给出提示，例如：

```
#. CN - China. (The official ISO 3166 short English name does
#. not include "The People's Republic of".)
#.
msgid "China"
msgstr "中国"
```

11.关于换行

对于很长的译文，就涉及到了换行问题。多数情况下没有限制，因为不影响最终的显示效果，只要阅读起来方便就行。下面几种格式都是正确的：

```
msgid ""
"Preview failed: neither internal KDE PostScript viewer (KghostView) nor "
```

```

"any other external PostScript viewer could be found."
msgstr ""
"预览失败: 找不到 KDE 内建的 PostScript 查看器 (KGhostView) 或其它外部"
"的 PostScript 查看器。"

msgid ""
"Preview failed: neither internal KDE PostScript viewer (KGhostView) nor "
"any other external PostScript viewer could be found."
msgstr "预览失败: 找不到 KDE 内建的 PostScript 查看器 (KGhostView) 或其它外部"
"的 PostScript 查看器。"

msgid ""
"Preview failed: neither internal KDE PostScript viewer (KGhostView) nor "
"any other external PostScript viewer could be found."
msgstr ""
"预览失败: 找不到 KDE 内建的 PostScript 查"
"看器 (KGhostView) 或其它外部"
"的 PostScript 查看器。"

```

但是, 如果 `msgid` 前方有“`#, c-format`”标记, 或者原文中有强迫换行标记“`\n`”, 那就要手工调整译文的换行, 以便能最终正确地显示在程序界面上。原则是译文长度不大于原文长度, 否则可能译文显示超出原有区域, 或者译文后面部分被截去, 很难看。对于含有 **HTML** 标记的长译文还需要在浏览器中预览显示效果(如果您了解 **HTML** 基本语法的话), 酌情调整行宽。例如:

```

#, c-format
msgid ""
"Error opening file '%s':\n"
"%s"
msgstr ""
"打开文件 “%s” 出错: \n"
"%s"

msgid ""
"Parse a theme dir and generate a \n"
"gtkrellmrc_ksim file that KSim will understand \n"
"better and exit."
msgstr ""
"解析一个主题目录生成 KSim 容易理解\n"
"的 gtkrellmrc_ksim 文件, 然后退出。"

```

多数 **CLI** 程序都需要手工设置换行, 这个时候希望翻译者能够对翻译文件进行测试以确定每行到底应当放多少字符, 尤其是翻译命令参数的时候, 如果不对长句进行强制换行处理或者一味依照文本编辑器所显示的视觉上与英文原文长度相同, 则很容易造成在程序实际运行中的格式非常难看。

12.关于对齐

对齐的问题通常出现在 **CLI** 程序的命令参数上, 英文原文一般会使用空格进行对齐, 但是空格并不能使得最终运行的程序在使用中文的情况下将文字正确对齐, 所以应当使用制表符(**Tab**)代替。制表符的对齐也不总是能在文本编辑器上所见即所得, 也就是说在文本编辑器上已经对齐的行可能在实际运行时前后相差一个制表符的距离。

13.关于模糊译文

如果看到 `#, fuzzy` 标记, 则表示本段译文是由工具软件(通常是 `msgmerge` 命令)猜测翻译得到的。

有时比较准确，有时却谬之千里。另一种情况是译者或校对人员主动加上去的，因为他们对该条译文没有把握。因此请对其译文进行修订，然后去除模糊标记，否则本段译文将不能显示在程序界面上。原则上，所有既往给出的译文，在翻译的时候都应该校对一遍。例如您拿到的文件中有如下行时：

```
#: groupdlg.cpp:209 groupdlg.cpp:216 groupdlg.cpp:229
#, fuzzy
msgid "Get Tagged Articles"
msgstr "取得文章"
```

当您修正可能存在的错误后删除整个“`#, fuzzy`”一行就可以了：

```
#: groupdlg.cpp:209 groupdlg.cpp:216 groupdlg.cpp:229
msgid "Get Tagged Articles"
msgstr "取得标记的文章"
```

可能还会遇到下面的情形：

```
#: groupdlg.cpp:623
#, fuzzy, c-format
msgid "Connecting to server %s"
msgstr "正在连接服务器"
```

在这里，修正翻译后在清除模糊标记时，“`c-format`”标签必须保留：

```
#: groupdlg.cpp:623
#, c-format
msgid "Connecting to server %s"
msgstr "正在连接服务器 %s"
```

14.关于已淘汰译文

老版本的程序中总有一些信息会在新版本中淘汰掉，这些译文由 `msgmerge` 程序自动设置为以 `#` ~ 开头，并被置于整个 `po` 文件的后方，直接删除它们不会影响当前版本的翻译内容，但是因为这些字符串还可能在以后的版本中重新出现，届时 `msgmerge` 还可以使用这些译文来提供翻译建议，所以如果它们没有影响到您的工作建议不要删除它们。例如：

```
#~ msgid "Error: no name"
#~ msgstr "错误: 没有姓名"

#~ msgid "Configure"
#~ msgstr "配置"
```

15.关于“`translator-credits`”字符串

“`translator-credits`”是放置程序运行过程中查看鸣谢信息时显示的翻译者条目，根据不同项目的小组可能有不同的填写要求，最通常的填写办法是将名字和电子邮件地址列表如下：

```
msgid "translator-credits"
msgstr
"Telsa Gwynne <hobbit@aloss.ukuug.org.uk>\n"
"Dafydd Harries <daf@muse.19inch.net>"
```

注意在多个人名时最好每个人占用一行，除最后一行外每一行的结尾都要使用 `\n` 来进行换行。

16.关于时间的译法

鉴于关于日期和时间的译法十分复杂，现在将国家标准中有关规定简要介绍一下，并给出推荐译法。
[国标 GB/T 7408-94 《数据元和交换格式 信息交换 日期和时间表示法》](#)

以下是最需要注意的几个要点：

- (1) 在日期格式中不能出现空格，即“2003 年”这样的格式是不符合国标的。
- (2) 表示日期时，必须按照年月日的顺序；年份一般用四位数字，而月、日必须使用带前导零的两位数字；必须使用“-”作为分隔符，或完全不使用分隔符。即“2004-01-03”或“20040103”来表示 2004 年 1 月 3 日。
- (3) 表示时间时，时、分、秒都必须使用两位数字，中间用“:”(半角括号)分隔，或完全不使用分隔符。小时计法采用 24 小时制，不区分上下午。
- (4) 日期和时间写在一起的时候，应先写日期再写时间。在日期和时间都不使用分隔符的情况下，在中间添加字母“T”进行分隔，即“19850412T101530”。

时间的表示方法同 `date` 命令的表示方法，现将常见的几个对应其含义列表如下：

1. %% 一个 % 字符
2. %A 星期几(如“星期一”)
3. %B 月份(如“八月”)
4. %d 日期(00-31)
5. %D 日期，格式等于 %m/%d/%y (例如：08/25/09)
6. %F 日期，格式等于 %Y-%m-%d (例如：2009-08-25)
7. %g ISO-8601 格式年份的最后两位(例如：09)
8. %G ISO-8601 格式年份(例如：2009)
9. %H 小时(00-23)
10. %i 小时(00-12)
11. %m 月份(01-12)
12. %M 分钟(00-59)
13. %n 换行
14. %p 显示“上午”或者“下午”两个字
15. %S 秒(00-60)
16. %t 制表符
17. %T 时间，等于 %H:%M:%S
18. %x 日期(例如：12/31/99)
19. %X 时间(例如：23:13:48)
20. %y 年份的最后两位(例如：09)
21. %Y 年份(例如：2009)

四、相关工具的使用

1. gettext 工具集

`gettext` 工具集提供了一组使用 PO 文件进行翻译时的实用工具，现在对其中比较常用的几个做简短介绍。

若希望了解关于此工具的详细信息，请浏览 [GNU `gettext' utilities](#)

- (1) `msgfmt` - 检查格式并生成机读的 MO 文件
一般使用的格式为：

```
msgfmt --statistics -cv filename.po
```

这样将会对翻译格式进行检查，如果有格式问题则会显示错误原因并指明行号；如果检查通过则显示翻译的进度情况并在运行目录下生成一个名为 **message.mo** 的文件，该文件便是程序运行时所要读取的二进制翻译文件。

在提交任何 **PO** 文件前都请使用 **msgfmt** 命令对格式进行检查。

(2) msgmerge - 合并文件

一般使用的格式为：

```
msgmerge --no-wrap -o newfile.po fileA.po fileB.po
```

这样进行的结果是将文件 **A** 和文件 **B** 进行合并，若有不相同的地方以文件 **B** 为准，最后输出在 **newfile.po** 中。

可以使用其更新选项来合并原来的 **PO** 文件和最新的 **POT** 文件：

```
msgmerge -U zh_CN.po example.pot
```

这样将会把原来的 **zh_CN.po** 另存为 **zh_CN.po~** 并更新 **zh_CN.po** 为新的内容，**zh_CN.po~** 作为备份文件可以直接删除。

(3) msgconv - 转换文件编码

一般使用的格式如下，作用为把 **zh_CN.po** 从其他编码转换为 **UTF-8** 编码：

```
msgconv -t UTF-8 zh_CN.po
```

msgconv 会自动更新文件头出的 **charset** 设置，但是在开始翻译前不妨再进行一下检查以免出现问题。

(4) msginit - 从 POT 文件创建 PO 文件

一般的使用格式为：

```
msginit -i example.pot -l zh_CN.UTF-8
```

这样将会询问翻译者的电子邮件地址，然后生成一个名为 **zh_CN.po** 的 **UTF-8** 编码文件。

如果运行程序的目录下只有一个 **POT** 文件则 **-i** 选项可以省略；如果运行的 **locale** 为 **zh_CN.UTF-8** 则 **-l** 选项可以省略，也就是说这个命令最简单的时候可以只输入 **msginit**

因为 **PO** 文件和 **POT** 文件的格式存在着一些微小的差异，如果您不是对它们十分了解则请使用此程序以避免可能出现的麻烦。

2. intltool 工具集

intltool 工具集实际上是一组脚本，用以实现一些常规的翻译文件维护，其中部分命令除了支持 **PO** 文件外还支持 **XML**。一共五个命令，对于翻译者来说比较常用的是 **intltool-update** 和 **intltool-merge**。这组工具多数需要在完整的源代码树中的 **po/** 子目录下运行。要了解更多关于 **intltool** 工具的信息，请阅读它的 **man** 手册。

(1) intltool-update - 更新 POT 文件并将 PO 文件与之合并

通常使用它生成 **POT** 文件时使用以下命令：

```
intltool-update -p
```

这样将会在目录中生成一个 POT 文件。

要使用它更新原有的 PO 文件，可以这样运行：

```
intltool-update zh_CN
```

这样将会自动生成新的 POT 文件并更新 zh_CN.po，最后得到的文件是更新后的 zh_CN.po，也可以使用 -o 选项来定义输出到指定文件而非更新原来的 PO 文件。

使用它来查看源代码中的 POTFILES 文件是否被正确维护，通常翻译者不需要做这项工作：

```
intltool-update -m
```

如果输出为空则代表一切正常，否则将有详细提示。

(2) intltool-merge - 合并翻译文件

一般使用方法为：

```
intltool-merge --utf8 po_directory input output
```

其作用为将指定的 po_directory 目录中的所有 PO 文件同 input 文件合并，并将输出写入到 output 文件，output 文件中包含了 input 文件中的原始字符串和其他 PO 文件中的已翻译字符串。在进行合并前会把所有文件都自动转换为 UTF-8 编码。

当 input 为一个 XML 文件时，输出的 output 文件也将是 XML，PO 文件中翻译的字符串将作为 “xml:lang” 属性插入到原 XML 中一并写入 output 文件。

3. xml2po

xml2po 是 gnome-doc-utils 中的一个工具，主要用途是在 XML 和 PO 文件之间进行转换和合并。

要从 XML 文件生成 PO 文件：

```
xml2po -e -o book.pot book.xml
```

反过来要把已翻译的 zh_CN.po 合并回原来的 XML 文件中：

```
/usr/bin/xml2po -e -p zh_CN.po -o book.zh_CN.xml book.xml
```

后记

本文是以 KDE 中国站点上刊载的 [I18N/L10N](#) 工作流程 为蓝本所修订的工作指南，过程中参考了 [i18n-zh](#) 以及 [GNOME Live](#) 上的一些文章。欢迎对此指南的内容进行讨论，请将您的意见发送至 i18n-zh@googlegroups.com。