

第四章 软件文档表达

4.5 选择表达方法

4.6 国际化软件开发编档

4.7 软件本地化工程文档处理

4.8 软件编档的风险及防范

4.5 选择表达方法

编制软件文档时，应该根据不同文档的性质、类型以及涉众的类型、专业能力等，在确定文档编制计划、内容和步骤后，还要确定文档编制的表达方法。

需要编入文档的资料以怎样的方式表达，对该文档涉众以及软件产品、文档等的传播有很大的影响。

以软件产品的**用户手册**为例，其表达方式有以下多种：

1. 直叙式
2. 菜谱式
3. 编号指教式
4. 剧本式
5. 四步法

任何一本软件用户手册都可能使用这些方法的其中一种或几种。

用户手册编制时常用表达方式介绍：

1. **直叙式**。按照文档中上下文关系，平铺直叙需要编入文档的内容。这是多数文档常用的表达形式。下图是一个直叙式编档的例：

订购员填写订购输入单 (单据编号 P0123)，这张单据有预先印好的订购号。事务输入月期、供货公司名、货号、数量和价格。

事务员将此单据信息输入到计算机中，并更新订购文件。

注意这种方式会使某些信息在段落中丢失。可以把它和下一种表达方式的例子加以比较

2. 菜谱方式。菜谱方式，顾名思义，就是写成菜肴或点心的烹制法的样子：

搅拌直至均匀，并出现泡沫为止

焙烤 (350°)一小时

语句的第一个词是动词。菜谱方式可以比直叙方式使用更少的字词表达数学内容。

将前面直叙方式表达的内容用菜谱方式再表达一次。同样的内容，字数少了，但实际上说的是同一件事。阅读行越短，意味着阅读和理解越快。

填写订购输入单， P0123

输入日期、供货公司名、货号、数量和价格

将单据数据录入计算机

更新订购文件

提示：每当要给出指示性内容时，使用菜谱表达方式。把动词放在最前面。如果某指示有一个条件，则先说明条件，接着就用一个动词。

3. 编号指教式。以菜谱式为基础，将文档中描述的软件工作步骤或处理过程加以编号，其表达结果会比直叙式更适合阅读理解。

订购员填好订购输入单 (P0123)。

1. 订购号：已印在单据上
2. 日期：输入当前日期。采用 yyyyymmdd 的 8 位表达，如将 2005 年 12 月 2 日表示为 20051202
3. 供货企业：输入接受订货的企业名。最多 25 个汉字
4. 货号：输入要订购的货号。在库存清单中找货号。若为新增，则见新增货物部分
5. 数量：输入订购数量。最多 99999
6. 单价：输入每种货物的价格。最贵 ¥ 99999.99 (系统将计算本次订购的总金额)
7. 制表人：制表人签名
8. 计算机操作员把填写完毕的订购单送计算机录入
9. 系统管理员将录入后数据处理，并更新订购文件

这种表达方式使需要完成的工作十分明确，且数据的长度、格式也有规定，尽管占用的空间稍大，但阅读方便，能使读者快速找到需要的内容。

4. **剧本式**。以菜谱式为基础，将文档以剧本方式表达。也有“演员”，以及动作。下图是这种方法的例：

- 订购员： 1. 完成订购输入单 (P0123)
- a. 订购号：
预先印在单据上的顺序号
 - b. 日期：
输入当前日期。采用 yyyymmdd 的 8 位表达，如将 2005 年 12 月 1 日表示为 20051201
 - c. 供货企业：
输入接受订货的企业名。最多 25 个汉字
 - d. 货号：
输入要订购的货号。在库存清单中找货号。若为新增，则见新增货物部分
 - e. 数量：
输入订购数量。最多 99999
 - f. 单价：
输入每种货物的价格。最贵 ¥ 99999.99 (系统将计算本次订购的总金额)
 - g. 制表人：
制表人签名
- 操作员： 2. 将订购单输入计算机
- 系统员： 3. 数据录入计算机，并更新订购文件

剧本式会比编号指教式占用更多的篇幅。但软件编档的目的并不是为了节省篇幅，而是如何使文档更清晰，更容易阅读理解。

直叙法使文档内容泛而无重点，编号指教法则突出执行者即角色的作用欠缺。而剧本法者使角色的任务十分鲜明，不同用户只需阅读相关部分即可。

5. **四步法**。对有些功能较多，结构复杂，操作较为灵活的系统，用户比以上的例子会有更多的自由。对于使用第三方软件包开发的系统就更是如此。此时，剧本法已不再适用。

即使有几种用户使用软件包，但无法知道谁需要做什么。并且，系统操作过程也不一定是一步接一步，而是需要用户判断接下去做什么，及何时做。

对此种情形，唯有直叙式表达和所谓的四步法适用。因为用户手册需要提供用户何时及为什么执行某项功能的指导。

而四步法将比直叙式能更有效的提供这样的指导：

- (1) 动机或理由。用户需要做什么？为什么要做？
- (2) 效果。当用户操作之后，会发生什么？
- (3) 一般步骤。为实现所要的效果，典型的操作步骤是什么？
- (4) 范例。操作和执行结果的例子。

下图是四步法编档的一个例子。

Wonder-Cal打印命令

利用打印命令，可得到您的表格的纸质拷贝。 Wonder-Cal在一页纸上的打印宽度和纸的宽度有关。通过指定上、下边缘及打印长度和宽度，可修改 Wonder-Cal边缘的设置。

打印表格的一般步骤：

1. 务必连接好打印机，装好打印纸并打开电源
2. 按 F键
3. 如果 Wonder-Cal 预定的边缘可接受 [RETURN] 键，开始打印表格
4. 如果希望改变边缘，按 M 键，可得到边缘修改子命令。屏幕显示 Wonder-Cal 预先设定的边缘参数。使用 [TAB] 键，把光标移到要修改的边缘处，输入新的边缘值，按 [RETURN] 键
5. 完成第 4 步后，仍然处在打印命令状态。按 [RETURN] 键，开始打印表格

例如，当在打印命令状态中按 M 键时，显示 Wonder-Cal 的预设边缘值：

边缘： 左： 6 上： 8 宽： 72

长： 60 纸长： 66

根据实际使用的纸的大小和表格大小，可修改边缘值为：

边缘： 左： 12 上： 3 宽： 60

长： 60 纸长： 66

6. 综合式。

在具体编制用户手册时，可以根据手册内容或手册读者的需要，采用不同的编档表达方法。

例如，对绪言或概述，通常采用直叙式编写。对于读者对象只有一类的那些文档部分，则使用菜谱式或编号指教式比较合适。而有若干类读者的部分，则可以选择采用剧本式编制。

具体在用户手册这类文档的编制过程中，编制者可以根据文档涉众的特点，采用灵活的策略，决定文档的表达方式。

4.6 国际化软件开发编档

在当今的全球经济环境下，使用软件的用户可能说着各种不同的语言、来自不同的国家，有着不同的文化背景，如果我们开发的应用软件能够满足这些用户需要，显而易见是益处多多。

所谓国际化软件开发和编档，就是要理解什么是国际化，并设计全球可用的应用软件，编制相符合的国际化软件文档。

1. 国际化软件开发编档的基本要求

由于国际化软件的用户分布在不同的国家/地区，并且使用不同的语言，因此，国际化软件的开发编档就要充分考虑和满足各种用户的功能和区域特性的需要。与面向单一语言的软件相比，国际化软件开发及编档具有以下基本要求：

(1) 全球可用性是国际化设计的目标

全球可用，可以保证国际化设计的软件顺利实施本地化，使软件能够运行在不同的语言环境中。为了保持较高的开发效率，为各种语言版本创建共同的单一二进制、全球可用的内核，是一种快速有效的方法。全球可用性的目标使得各种不同语言的版本具有一致的外观、风格和功能。

但全球化开发的软件的文档的编制，则必须按照软件去向，符合当地文字规范、文化习俗和阅读习惯。

(2) 支持不同编码字符集的能力

国际化软件及其文档模板应支持不同语言和字符的输入、显示和输出。

对于东亚市场的版本，不但需要支持单字节字符集，还需要正确支持双字节甚至多字节字符集。因此，需要在软件国际化设计时，处理正确的字符集编码。国际化软件编档同样会有类似问题，特别是同一国际化软件的不同语言版本文档采用统一模板编制时。一个较好的解决方案是采用 Unicode 字符编码标准。

(3) 支持区域信息格式

国际化软件需要支持不同区域的信息格式。如，键盘布局、字符串排序、日期、时间、数字、货币、地址、电话号码、度量单位等与特定区域的语言和文化有关的信息。保证不同语言版本的软件能够正确输入、显示、编辑、打印这些信息。相关文档必须完整表达这一类区域信息。

(4) 国际化特征贯穿于软件项目的全过程

软件国际化特征和需求应该从项目最初的市场调研、可行性研究以及项目计划就开始加以考虑。在随后的项目需求分析、系统设计、编码、测试、质量管理，直到软件发布、维护的全过程，应始终贯彻国际化的理念。

尽管国际化软件开发过程的技术文档可以采用所在地文字或事先约定的文字编制，但应为最终完成多语种文档编制做好准备。

(5) 提高软件的本地化能力

提高软件的本地化能力，可以降低本地化的成本，提高软件的质量、可用性和稳定性。因此，要求软件开发人员从软件开发的最初阶段，就应遵循基本的本地化能力准则来开发软件和编档。将需要本地化的内容、文档，与国际化程序代码和文档分开，单独放在资源文件中。

(6) 尽早测试国际化软件

由于必须通过修改程序代码才能解决本地化能力方面的错误，因此，如果到了软件本地化阶段再来修正这些错误，那将会带来很大的困难，并提高开发成本，推迟本地化过程。为此，可以通过伪本地化测试本地化能力，以尽早发现和修改软件的国际化错误。

这项工作可以在编码刚刚完成，甚至在详细设计阶段就开始测试。不仅要测试程序，也要同步测试设计文档。

2. 国际化软件开发编档的基本准则

在进行国际化软件开发和编档的实践中，人们逐步总结出一些通用的准则。遵守这些准则，可以避免国际化项目走弯路甚至失败，可以提高软件质量，降低开发和维护成本。

国际化软件开发和编档需要遵循的通用准则如下：

在国际化软件项目的最初期就融入国际化思想，并且使国际化理念贯串于项目的整个生命周期；

采用单一源文件和文档模板进行多语言版本的本地化，不针对不同的语言编写多套代码和文档；

将需要本地化的文字与软件源代码分离，并存储在单独的资源文件中；

软件代码和文档模板支持处理单字节字符集和多字节字符集文字的输出、输入和显示，并且遵守竖排和折行等规则；

软件代码和文档模板应该支持 Unicode 标准，或者可以在 Unicode 与其他代码页 (Code Pages) 互换；

软件代码和文档不要嵌入字体名，也不要假设使用某种字体；
使用通用的图标和位图，避免不同区域的文化和传统差异，避免在图标和位图中嵌入需要本地化的文字；
菜单、对话框等界面布局及文档模板，能够处理本地化文字的长度扩展；
源语言及文档描述的文字要准确精简，使用一致的术语，避免歧义和拼写错误，容易进行本地化翻译；
保证不同区域的键盘布局都能使用源软件的快捷键及文档的使用的一致方法；
考虑不同区域的法律对软件的要求；
如果软件中采用第三方开发的软件或组件，需要检查和确认是否满足国际化的要求；
保证源语言软件及其中内嵌的电子文档可以在不同的语言文化区域和操作系统上正确运行；
软件代码及文档模板中避免“硬编码”，不使用基于源语言的数字常量、屏幕位置、文件和路径名；
软件代码及文档模板中字符串的缓冲区长度要满足本地化字符扩展的需要；
软件及文档模板能正确支持区域排序和大小写转换。

4.7 软件本地化工程文档处理

软件本地化工程是融合软件工程、翻译技术和桌面出版等专业知识的综合性技术，是软件本地化项目中不可缺少的重要环节。

软件本地化项目包含的本地化内容，经过软件本地化工程处理后，可以提高软件本地化工作（翻译、桌面排版、测试、项目管理）的工作效率，并且有助于保持本地化内容的一致性。

软件本地化工程的目的：

软件本地化工程处理的目的是为软件翻译等人员提供精确的、易于翻译的格式文档，并且对翻译后的文档进行工程后处理，为本地化后的软件编译提供正确的内容。

具体来说，软件本地化的目的至少包含下面三个方面：

1. 通过抽取和重复利用已经翻译的资源文件，提高本地化的效率和一致性；
2. 校正和调整翻译后的资源文件用户界面的大小和位置，保证编译出正确的本地化软件 / 文档；
3. 通过修正软件本地化测试发现和报告的 Bug，提高软件本地化的质量。

软件本地化工程及文档处理的内容：

软件本地化工程处理的内容与具体本地化项目的要求有关。对于一个完整的软件本地化项目，至少包括软件资源文件和联机帮助文档的本地化工程处理。

一般，软件本地化工程及文档处理的内容包括以下多个方面：

1. 制定本地化项目计划，准备本地化工程的专用工具 (Preparing)
2. 检查本地化工程需要处理的文档 (Checking)
3. 抽取并重复利用软件 / 文档的本地化资源 (Leveraging)
4. 生成字数统计表 (Word Count)
5. 生成本地化术语表 (Glossary)
6. 配置本地化编译环境 (Build Environment)
7. 校正翻译的资源文件 (Validating)
8. 调整翻译的用户界面的控件位置和大小 (Resizing)
9. 编译本地化软件和联机帮助文档 (Compiling)
10. 对编译后的本地化软件和联机帮助执行完整性检查 (Sanity Check)
11. 修正测试工程师报告的缺陷 (Fixing bugs)

软件本地化工程及文档处理的流程

1. **软件本地化工程**。软件本地化工程的处理流程与具体处理内容有关，但一般情况下，其流程如下：

准备资源文件本地化工具 检查所需要处理的源内容 抽取软件资源文件 对翻译的资源文件进行工程后处理 检查翻译的文件 调整软件界面 编译本地化软件 检查软件编译的完整性 修正软件测试中发现的缺陷

2. **联机帮助文档本地化**。软件本地化工程一个十分重要的工作是联机帮助文档的本地化，其处理流程如下：

准备联机帮助文档的本地化工具 检查联机帮助工程文件内容 抽取联机帮助工程文件 翻译的联机帮助文档进行工程后处理 检查翻译的文件 编译本地化联机帮助文档 检查编译后联机帮助文档的完整性 修正联机帮助文档测试中发现的缺陷

3. **编制本地化软件文档**。国际软件完成本地化工程后，就基本具备了在本地上市流通的条件了。但是，作为商品软件，在最后上市流通前，还必须完成产品功能性能介绍说明和销售资料的编制，完成必要的用户安装操作说明的编制，以及必要的产品技术、配置、操作、管理等文档的整理翻译。¹⁷

软件本地化工程文档处理的工具

软件本地化工程及文档处理，需要使用很多专业工具软件和自己开发的各种内部工具。根据本地化工程处理的内容不同，可能需要下列多种工具软件：

1. 资源文件本地化专业软件：如，Alchemy Catalyst, PASS Passolo, Microsoft Visual Studio, SDL Insign等；
2. 联机帮助编译工具：Html Help workshop, Microsoft Help Workshop, RoboHelp等；
3. 文档检查工具：Beyond Compare, Microsoft WinDiff, Html QA等；
4. 辅助修改工具：UltraEdit, Orca, Visual C++等；
5. 图像处理工具：Adobe Photoshop, Paint Shop Pro, HyperSnap-DX, Snagit, Camtasia Recorder等；
6. 压缩 /解压缩软件：Winzip, WinRAR等；
7. 文件合并与分割软件：MaxSplitter等；
8. 文件上传和下载工具：CuteFTP, WS_FTP等；
9. 磁盘分区备份软件：Norton Ghost；
10. 通用办公软件：Microsoft Office；

软件本地化工程文档处理的人员

由于软件本地化工程及文档处理需要涉及到多种文件格式，可能牵涉多种操作系统，运用多种工具软件，而且由于本地化工程一般处理时间都比较短。所以，对软件本地化工程人员的要求相对较高：

1. 具有较高的安装和配置通用操作系统及平台的能力；
2. 熟练使用软件本地化常用的联机帮助文档的编译软件；
3. 深入了解并掌握常见的标识语言 (如，Htm和XML等)及其编辑软件；
4. 掌握国际化软件界面布局和基本的桌面排版知识 (如，图形图像、样式表和模板等)；
5. 熟悉宏、脚本语言和图形用户界面开发的相关知识；
6. 熟悉计算机辅助翻译 (CAT)工具的使用；
7. 了解软件国际化和字符集编码的知识；
8. 了解本地化模型、处理流程和项目计划进度等；
9. 了解因特网技术、软件通讯和协议；
10. 专业的工作态度，较强的学习能力，独立快速发现和解决问题的能力；
11. 良好的沟通和表达技巧；
12. 良好的外语水平。

例：联机帮助文档本地化工程的方法：

联机帮助文档本地化工程包括评估与准备、文档格式转换、资源重复利用、检查翻译后的文件、编辑图像、编译本地化联机帮助文档、修复帮助文档的缺陷等。

各种软件的联机帮助文档有多种类型和格式。如，对应于 Windows 操作系统，常用的联机帮助文档类型主要有：hlp、chm、htm和xml。尽管联机帮助文档类型不同，本地化工程处理所用的工具不同，但是工程处理流程和处理内容基本相同，具体讨论如下：

1. 预处理 (Pre-Process)

需要本地化的联机帮助文档的预处理内容包括文档格式转换和数据统计分析。

根据帮助文档的类型，选用适当的工具软件，将原始文档转换成适合文档翻译的文件格式。例如，适合 Trados 翻译的 rt 格式。

预处理时，还要标注需要翻译的文档，以正确区分不需要翻译和需要特殊翻译的内容。

除了提供预处理文档，还要使用翻译记忆工具的统计功能，提供全部需要翻译字符的数量 (包含新增的和更新的)，以便翻译团队估计任务量，分配翻译资源。

2. 后处理 (Post-Process)

在正式编译本地化联机帮助文档之前，需要对翻译完成的文档进行翻译后的处理。主要包括文档格式转换、图像的工程处理和编译环境的设置等。

此处的文档格式转换是指将翻译的文档格式重新转换到适合编译的文档格式。

图像的工程处理主要是对需要本地化的程序位图或图标重新编辑，包括其中的文字、图像外观形状和颜色等。

设置编译环境是编译本地化联机帮助文档的关键步骤，包括定义字符集编码、语言设置、区域名称和编号 ID 等。

3. 编译本地化联机帮助文档 (Compiling)

编译本地化联机帮助文档的具体实现技术与编译的联机帮助文档的类型紧密相关。

通常软件供应商会提供或指定编译工具。本地化工程师设置编译环境，将翻译的联机帮助文档和合适格式的图像复制并替换编译环境中的源语言文档和图像，执行相应的编译操作命令，编译创建本地化联机帮助文档。

4. 检查本地化联机帮助文档 (Validate)

完成本地化软件的编译后，需要确认编译的正确性，包括布局 (Layout) 和功能等方面。通常执行可接受测试 (Acceptance Test)，又叫作冒烟测试 (Smoke Testing)。如果通过了该测试，则可以根据正式测试文档进行正式测试。否则，就需要重新编译，并再次执行可接受测试，直到成功。

本地化联机帮助文档的可接受测试，可以使用自行开发的检查工具，也可以使用专业的检验工具 (例如，HelpQA 或 HtmlQA)，通过对比本地化文档和源语言文档，保证本地化文档的布局或超链接的正确性。另外，也要人工检查是否存在没有翻译的内容，是否存在不能显示的乱码字符等。

5. 修复软件缺陷 (Fixing Bug)

本地化工程师负责修复本地化联机帮助文档测试过程发现和报告的缺陷 (Bugs)。由于本地化工程后，联机帮助文档发生错误或缺陷的可能增多，既可能是本地化工程前就有的，也可能是本地化工程新引入的。而本地化工程师只需要修复由于本地化产生的错误，如翻译错误、用户界面布局错误、文档显示格式错误、翻译错误等引起的本地化软件的错误。对于源语言联机帮助文档中的错误，则由软件供应商负责处理。

由于本地化文档的编译和测试通常在本地化项目的后期才能进行，因此，修复本地化联机帮助文档缺陷的时间通常比较紧张，本地化工程师应该及时、快速、正确地修复缺陷。

通过软件本地化工程及文档处理过程高效和精确的工作，可以减少本地化翻译的工作难度，有效的保证软件本地化及其文档的质量，也为本地化项目管理提供了良好的条件，便于更好的项目管理、成本控制和进度跟踪。

最后，由于软件国际化和本地化开发技术的不断发展，本地化工程中必然将处理更多种类的文件格式，因此，需要本地化工程及文档处理的专业人员能够不断选择和学习更多的新软件和新技能，使得在软件本地化工程及文档处理工作中，能够发挥更重要的作用。

4.8 软件编档的风险及防范

1. 编档风险识别

与软件工程中软件项目风险的识别处理类似，软件编档的风险识别也从以下几个方面着手：

- (1) 编制的文档是否和软件吻合；
- (2) 文档表达是否符合涉众阅读、理解的水平；
- (3) 编档进程是否能够符合软件开发或发行的进度；
- (4) 编档人员是否对开发的软件有足够的了解；
- (5) 编档资源 (人员、资料、时间、环境、成本) 是否有足够保证；
- (6) 编档工作是否得到管理部门的支持。

编档风险的识别同样可以使用“风险项目检查表”，来列出所有可能的与每一个风险因素有关的提问。

如，管理人员或项目组长可以通过回答下列问题，得到有关人员对编档风险的认识：是否确保编档人员的投入或参与编档的相关人员有合理的编档时间吗？编档人员对编档工作重视吗？人们对手头上的编档工作是否有正确的目标？参与编档的人员接受过必要的编档培训吗？

对于这些提问，通过判定分析或假设分析，给出确定的回答，就可以帮助管理人员或计划人员估算编档风险的影响。

2. 编档风险评价

在风险分析过程中进行风险评价的时候，应该同时建立一个风险三元组：

$$[r_i, l_i, x_i]$$

其中， r_i 是风险， l_i 是风险出现的概率， x_i 是风险对编档及其结果的影响。具体在做风险评价时，应当进一步检验在风险估计时所得到的估计的正确性，尝试对已暴露的风险进行优先排队，并着手考虑控制和(或)消除可能出现风险的方法。

具体进行风险评价的方法可以参阅相关软件工程资料了解。

3. 风险防范和监控

风险防范是指在软件编档过程，针对已经发现或可能发生的风险利用某些技术，如编档自动化、软件心理学、阅读心理学、传播心理学、文档评审技术、信息资源管理方法等，设法转移或规避风险。

例如，假如人员的频繁流动是一项风险 r_i ，基于以往的历史和管理经验，频繁流动可能性的估算值 l_i 为 0.70 (即 70%，属相当高)，而影响 x_i 的估计值是，项目的开发时间增加 15%，总成本增加 12%。有了这些数据后，就可以制定防范这个风险的对策：

- (1) 了解人员流动的原因；
- (2) 在项目开始前，把缓解这些原因 (规避风险) 的工作列入已拟定的风险防范计划；
- (3) 当项目启动时，做好人员流动可能出现的准备。采用一些措施以确保一旦人员离开时，项目仍能继续 (削弱风险)；
- (4) 建立项目组，以使各成员都了解有关项目开发活动的信息；
- (5) 制定文档标准，并确保编档工作的顺利进行；
- (6) 对所有工作都组织细致的评审 (以使大家都能按计划进度完成自己的工作)；
- (7) 对每一个关键性的技术岗位，都要物色和培养后备人员。

当然，这些风险防范步骤会带来额外的成本开支。因此，当这样的成本超出因为防范风险而带来的收益时，就有必要对风险防范工作进行评估。编档风险防范的评估通常都会和项目风险防范的评估一起进行。

通常，只有当编档软件的规模较大，软件文档结构较为复杂时，才有必要单独对编档工作制订风险防范和监控计划 (RMP)，启动风险识别和防范工作。否则，只要将编档风险的识别和防范工作纳入到项目风险的识别和防范工作中即可，其计划也可一并制订。