

使你的 Symbian 操作系统软件本地化的介绍: 第二部分

—Symbian 操作系统 v9

Martin de Jode 和 Mark Shackman

1.1 版

1 介绍

对于 Symbian 操作系统 v9 之前的版本, 在以前的文章中我们概述了需要本地化你软件的措施。在这篇文章中, 我们将通过浏览 Symbian 操作系统 v9 及以后的版本中需要的变动来完成具体的描述。

Symbian 操作系统 v9 的发布是一个显著的分水岭, 系统引入了增强的安全架构。新安全模型强加的各种限制使应用程序结构中一些次要的变化成为必要, 这些限制使程序结构和本地化相关的资源文件位置也产生了一些变化。对于安全平台和软件移植到 v9 的详细介绍, 请读者参考 Craig Heath 的《Symbian 操作系统平台安全》(Wiley 2006)。

在这篇文章中, 我们主要介绍与本地化过程相关的一些变化。幸运的是这些变化非常小, 以前论文概述的那些基本原则仍然有效。

2 本地化你的软件

首先, 让我们看一下 (S60) .MMP 文件。

```
// HelloWorld.mmp
TARGET HelloWorld.exe
TARGETTYPE exe
UID 0x100039CE 0xE0000003
SOURCEPATH .
SOURCE HelloWorld_Main.cpp
SOURCE HelloWorld_Application.cpp
SOURCE HelloWorld_Document.cpp
SOURCE HelloWorld_AppUi.cpp
SOURCE HelloWorld_AppView.cpp
USERINCLUDE .
SYSTEMINCLUDE \epoc32\include
START RESOURCE HelloWorld.rss
HEADER
TARGETPATH \resource\apps
LANG 01 02
END
START RESOURCE HelloWorld_reg.rss
#ifdef WINSCW
TARGETPATH \private\10003a3f\apps
#else
TARGETPATH \private\10003a3f\import\apps
```

```

#endif
END
START RESOURCE HelloWorld_loc.rss
TARGETPATH \resource\apps
LANG 01 02
END
LIBRARY euser.lib, apparc.lib, cone.lib, avkon.lib, eikcore.lib
v9 及以后版本中.MMP 文件的变化主要与 RESOURCE 语法有关。例如：

```

```

START RESOURCE HelloWorld.rss
HEADER
TARGETPATH \resource\apps
LANG 01 02
END

```

对于每一个.RSS文件，都指定一个RESOURCE分组，这个分组以START开始，以END终止。HEADER关键字可选，该关键字使系统包括文件夹...\epoc32\include中产生一个资源头文件(.RSG)。TARGETPATH声明定义资源文件编译后的(.RSC)存储位置。在Symbian操作系统v9及以后的版本中，数据锁定（平台安全架构的一部分）需要资源必须放入正确的路径，要么是公共只读目录\resource，要么是存储程序私有资源的目录\private\<secure_ID>（此处若没有其他定义的话，<secure_ID>是程序的UID3）。关键字LANG与以前意义相同，因此在上面的例子中，资源编译将执行两次，每种语言一次。

3 本地化你的程序资源

对于软件的这一版本，让我们看一下.RSS 文件的内容，对于 S60 用户界面：

```

NAME HEWO
#include <eikon.rh>
#include <eikcore.rsg>
#include <appinfo.rh>
#include <avkon.rh>
#include <avkon.rsg>
#include "helloworld.hrh"
#ifdef LANGUAGE_01
    #include "helloworld.l01"
#elif defined LANGUAGE_02
    #include "helloworld.l02"
#endif
RESOURCE RSS_SIGNATURE { }
RESOURCE TBUF r_default_document_name
{
    buf="HEWO";
}
RESOURCE EIK_APP_INFO
{
    hotkeys=r_example_hotkeys;
    menubar=r_example_menubar;
}

```

```

        cba = R_AVKON_SOFTKEYS_OPTIONS_EXIT;
    }
RESOURCE HOTKEYS r_example_hotkeys
{
    control=
    {
        HOTKEY { command=EAKnSoftkeyExit; key='e'; }
    };
}
RESOURCE MENU_BAR r_example_menubar
{
    titles=
    {
        MENU_TITLE { menu_pane=r_example_first_menu;
        txt=STRING_r_example_first_menu; }
    };
}
RESOURCE MENU_PANE r_example_first_menu
{
    items=
    {
        MENU_ITEM { command=EExampleItem0;
        txt=STRING_r_example_item0; },
        MENU_ITEM { command=EExampleItem1;
        txt=STRING_r_example_item1; },
        MENU_ITEM { command=EExampleItem2;
        txt=STRING_r_example_item2; },
        MENU_ITEM { command=EAKnSoftkeyExit;
        txt=STRING_r_example_close; }
    };
}
RESOURCE TBUF r_example_text_Hello
{ buf=STRING_r_example_text_hello; }
RESOURCE TBUF r_example_text_Item0 { buf=STRING_r_example_item0; }
RESOURCE TBUF r_example_text_Item1 { buf=STRING_r_example_item1; }
RESOURCE TBUF r_example_text_Item2 { buf=STRING_r_example_item2; }

```

这在结构上与 v9 以前的版本像同，使用同样的条件包含我们在第一部分看到的.RLS 文件：

```

#ifdef LANGUAGE_01
#include "01-strings.rls"
#else if defined LANGUAGE_02
#include "02-strings.rls"
#endif

```

3.1 本地化资源字符串

包含本地化字符串的文件，其文件名与程序名字一样，后缀采用_IXX 扩展名，XX 是本地编号。这些文件被 .RSS 文件使用如下语句加载：

```
#ifdef LANGUAGE_01
#include "helloworld.l01"
#elif defined LANGUAGE_02
#include "helloworld.l02"
#endif
```

下面的例子是 helloworld.l02 的内容，它包含语言 02（法语）的资源。

```
// 本地化为法语的字符串
rls_string STRING_r_example_first_menu "Bonjour"
rls_string STRING_r_example_item0 "Élément 0"
rls_string STRING_r_example_item1 "Élément 1"
rls_string STRING_r_example_item2 "Élément 2"
rls_string STRING_r_example_close "Fin"
rls_string STRING_r_example_text_hello "Bonjour Monde!"
rls_string STRING_r_example_short_caption "B M"
rls_string STRING_r_example_caption "Bonjour tout le monde"
```

4 程序注册文件

v9 的另一个主要改变是程序注册文件的引入。由于 GUI 程序现在是 .EXE 文件 (v9 以前是 .APP 文件)，程序注册文件用于告诉应用程序架构，该.exe 是一个 GUI 程序。再看一下 MMP 文件的如下结构（已被简化）：

```
START RESOURCE HelloWorld_reg.rss
TARGETPATH \private\10003a3f\apps
END
```

程序注册文件名与程序名字相同，添加_reg 后缀，替代了早期版本支持的程序信息文件（AIF）文件。这些注册文件必须放进私有的 data caged 目录，由应用程序架构服务器拥有，该服务器负责软件的启动加载过程。将_reg 文件放入私有目录可以避免恶意用户伪造、替换程序的_reg 文件。程序注册文件的内容如下：

```
#include <appinfo.rh>
UID2 KUidAppRegistrationResourceFile
UID3 0xE0000003 // 程序UID
RESOURCE APP_REGISTRATION_INFO
{
    app_file = "HelloWorld";
    localisable_resource_file = "\\resource\\apps\\HelloWorld_loc";
}
```

APP_REGISTRATION_INFO 结构（使用 app_file 声明）提供了二进制文件的文件名（不含扩展名）。如果提供了本地化图标/标题文件，需要提供完整的路径和文件名（象上例一样），除了盘符和扩展名。这个文件定义了程序的标题和图标文件。通常文件名与程序名一样，不过使用 _loc 后缀。

```
#include <AppInfo.rh>
#ifdef LANGUAGE_01
```

```

#include "helloworld.i01"
#elif defined LANGUAGE_02
#include "helloworld.i02"
#endif
RESOURCE LOCALISABLE_APP_INFO
{
short_caption = STRING_r_example_short_caption;
caption_and_icon =
{
CAPTION_AND_ICON_INFO
{
caption = STRING_r_example_caption;
number_of_icons = 1;
icon_file = "\\resource\\apps\\helloworld.mbm";
}
};
}

```

HelloWorld_loc.rss 文件是个标准的 Symbian 本地化资源文件，通过在程序的.MMP 文件中包含类似下列语句，由资源编译器编译：

```

START RESOURCE HelloWorld_loc.rss
TARGETPATH \resource\apps
LANG 01 02
END

```

5 本地化程序的安装

本地化过程的最后一步就是创建安装包 (.SIS) 文件。安装可以被国际化，以至于安装包可以在安装的时候根据电话设备的语言类型安装相应的文件。在这一部分，我们介绍如何取得这一设置。国际化的 Hello World 程序的包文件如下：

```

&01,02
#{ "HelloWorld_v9_s60_EN", "HelloWorld_v9_s60_FR" }, (0xE0000003), 1, 0, 1,
TYPE=SA, NC
:"Symbian Software Ltd"
%{ "Symbian Software", "Symbian Logiciel." }
"E:\Symbian\S60_3rd\EPOC32\RELEASE\ARMV5\UREL\HelloWorld_S60.exe"
- "!\:sys\bin\HelloWorld_S60.exe"
"E:\Symbian\S60_3rd\EPOC32\data\Z\private\10003a3f\apps\HelloWorld_reg.RSC"
C"
- "!\:private\10003a3f\import\apps\HelloWorld_reg.RSC"
{ "E:\Symbian\S60_3rd\EPOC32\data\Z\resource\apps\HelloWorld.R01"
"E:\Symbian\S60_3rd\EPOC32\data\Z\resource\apps\HelloWorld.R02"
} - "!\:resource\apps\HelloWorld.RSC"
{ "E:\Symbian\S60_3rd\EPOC32\data\Z\resource\apps\HelloWorld_loc.R01"
"E:\Symbian\S60_3rd\EPOC32\data\Z\resource\apps\HelloWorld_loc.R02"
} - "!\:resource\apps\HelloWorld_loc.RSC"

```

注意，由于本地化的原因，这里单个字符串或文件名并不相同，在框架范围内，选项的数目必须与.MMP 文件里定义的本地变量的数目匹配（在这种情况下，我们只有两种选择）。

6 总结

在如何本地化 Symbian 操作系统软件的第二部分，我们考虑了移植你的程序到 Symbian 操作系统 v9 及以后版本时必要的改变。基本原则仍然与早期版本相同，但安全平台的需求使 MMP 和有关资源文件的一些语法必须有一些变化。

注意：关于这一文件的当前发行版本，在 CodeWarrior 和 Carbide.c++里有一个缺陷，它使这篇论文里给出的代码示例不能正确构建。关于这一缺陷有一个专门的讨论小组，在最近创建的 Symbian 操作系统 v9.2 和 v9.3 中包括了正在讨论的最新文件。这些最新的文件作为这篇论文附加的下载也被提供，它将要代替在\epoc32\tools 里存在的文件。

[支持开发者库](#)

你想及时了解 Symbian 开发者网络上可获得的新文章吗？

[订阅Symbian社团时事通讯](#)

Symbian 社团时事通讯带给你每月 Symbian 操作系统最新的新闻和资源。