
KAPITOLA 10

Implementace mezinárodní podpory a lokalizace

V této kapitole:

- ◆ Řetězce ve zdrojovém kódu
 - ◆ Internacionalizace nápovědy
 - ◆ Internacionalizace ostatních zdrojů
 - ◆ Administrace a příprava lokalizovaných zdrojů
 - ◆ Shrnutí
-

Přeložte si svoji aplikaci!

Tato kapitola ukazuje, jak do aplikace implementovat mezinárodní podporu (internacionalizovat ji), a upravit ji tak pro různé jazyky. Naučíte se specifika internacionalizace zdrojového kódu a souborů manifestu, stejně jako překladu nápovědy a přizpůsobení dalších zdrojů, jako jsou obrázky.

Profesionální flexibilní aplikace se musí navrhovat co nejjednodušším způsobem tak, aby se mohly snadno přizpůsobit pro koncové země a jazyky. Pro snadnou internacionalizaci aplikace lze využít podporu, kterou nabízí Java API, platforma NetBeans a NetBeans IDE.

Řetězce ve zdrojovém kódu

Řetězce se přemísťují ze zdrojového kódu aplikace do souborů properties. Jazykově závislé řetězce tak lze oddělit od kódu logiky aplikace a snadno přeložit do jiných jazyků. To je dokonce možné i po uvolnění aplikace. Jazykové zdroje se pak uloží jako dvojice klíč/hodnota v souboru properties:

```
CTL_MyTopComponent = My Window
HINT_MyTopComponent = This is My Window
```

Jákykoliv takovýto soubor s jazykovými prostředky používá třída `ResourceBundle` jazyka Java. Třída `ResourceBundle` je odpovědná za jazykové prostředky podle konkrétního místního nastavení národního prostředí, které specifikuje zemi a jazyk. Pro snadnou obsluhu souborů properties a přístupu k instanci `ResourceBundle` poskytuje platforma NetBeans třídu `NbBundle`. Soubor s jazykovými prostředky musí nést jméno `Bundle.properties` a většinou se tento soubor vytváří pro každý balíček. Nejjednodušší způsob, jak vytvořit objekt `ResourceBundle`, demonstruje následující volání:

```
ResourceBundle bundle = NbBundle.getBundle(MyTopComponent.class);
```

Třída `NbBundle` vytváří objekt `ResourceBundle` pro soubor `Bundle.properties` umístěný v balíčku se třídou `MyTopComponent`. Požadovaný řetězec se jednoduše načte metodou `getString()` třídy `ResourceBundle`:

```
String msg = bundle.getString("CTL_MyTopComponent");
```

Pokud vyžadujeme pouze malé množství řetězců uvnitř naší třídy, použijeme metodu `getMessage()` pro načtení řetězce přímo bez vytváření instance `ResourceBundle`:

```
String msg = NbBundle.getMessage(MyTopComponent.class, "CTL_MyTopComponent");
```

V řetězcích lze použít také zástupný znak (tzv. placeholder). Toho lze nejčastěji využít pro konkrétní jména nebo konkrétní cesty k souborům a adresářům. Jako zástupný znak se použije pár složených závorek, který obsahuje číslo parametru:

```
Result = {0} MP3-Files found for {1}
```

Tyto parametry předáme metodě `getMessage()`, která nahradí zástupný znak daným parametrem. Podobně můžeme předat až tři parametry, nebo dokonce pole s neomezeným počtem parametrů:

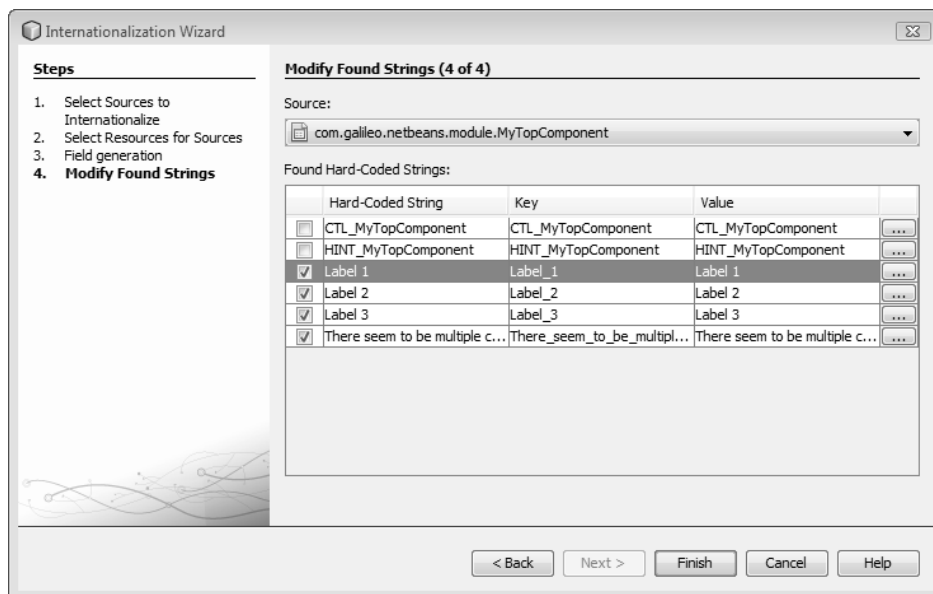
```
String label = NbBundle.getMessage(MyTopComponent.class,
    "Result",
    new Integer(results.size()),
    search.getText());
```

V každém souboru properties se ukládají pouze řetězce jednoho jazyka. Pro přidání dalšího jazyka do aplikace uložíme řetězce – se stejnými klíči – do souboru properties se jménem `Bundle_<jazyk>_<země>.properties` do stejného adresáře. Třída `NbBundle` vrací `ResourceBundle` volá-

ním metody `getBundle()`. Ten odpovídá místnímu nastavení národního prostředí, které vrací `Locale.getDefault()`. Soubor `Bundle.properties`, který neobsahuje identifikaci jazyka a země, je základní balík jazykových prostředků. Tento balík se použije v případě, že není k dispozici balík jazykových prostředků pro aktuálně nastavené místní národní prostředí. Konkrétní balík prostředků pro určité národní prostředí lze také získat předáním objektu `Locale` metodě `getBundle()`. Pro upřesnění, v jakém pořadí se balíky prohledávají, poskytuje metoda `NbBundle.getLocalizingSuffixes()` seznam všech přípon v pořadí prohledávání.

Metoda `Locale.getDefault()` vrací místní nastavení národního prostředí implicitní pro virtuální stroj. Pro spuštění celé aplikace ve specifickém nastavení národního prostředí nastavte argument `locale` příkazové řádky. Tento parametr předá identifikaci jazyka a země aplikaci. Více informací naleznete v kapitole 11.

NetBeans IDE poskytuje průvodce pro internacionalizaci řetězců pro vaše zdrojové soubory. Použití průvodce pro vyhledání řetězců v našich souborech nám pomůže s přesunem těchto řetězců do souborů `properties` (viz obrázek 10.1). Takto můžeme upravit klíč, hodnotu a kód pro vložení na místo řetězce. Průvodce můžeme nalézt v nabídce **Tools** → **Internationalization** → **Internationalization Wizard**.



Obrázek 10.1: Automatický přesun řetězců ze zdrojového kódu do balíku jazykových zdrojů a vložení potřebného zdrojového kódu průvodcem internacionalizace

Řetězce v souboru manifestu

Kromě řetězců ve zdrojových kódech můžeme také internacionalizovat textové informace v souboru manifestu. Jsou dvě možnosti, jak toto udělat. První možností je připojit identifikátor jazyka k atributu v manifestu, a použít tak stejný atribut v souboru vícekrát:

```
Manifest-Version: 1.0
OpenIDE-Module: com.galileo.netbeans.module
```

```
OpenIDE-Module-Name: My Module
OpenIDE-Module-Name_de: Mein Modul
```

Druhou možností (preferovaná autorem) je přesunout atributy, které se mají internacionalizovat, do souboru properties. Jména atributů se pak použijí jako klíče a následně se použijí stejně v balíčcích jazykových prostředků pro každý jazyk. Načtení atributů z balíku zaneseme do souboru manifestu použitím atributu `OpenIDE-Module-Localizing-Bundle`, jak můžeme vidět ve výpisích 10.1 až 10.3 (viz také kapitola 3).

Výpis 10.1: Manifest.mf

```
Manifest-Version: 1.0
OpenIDE-Module: com.galileo.netbeans.module
OpenIDE-Module-Localizing-Bundle: com/galileo/netbeans/module/Bundle.properties
```

Výpis 10.2: Bundle.properties

```
OpenIDE-Module-Name = My Module
```

Výpis 10.3: Bundle_de.properties

```
OpenIDE-Module-Name = Mein Modul
```

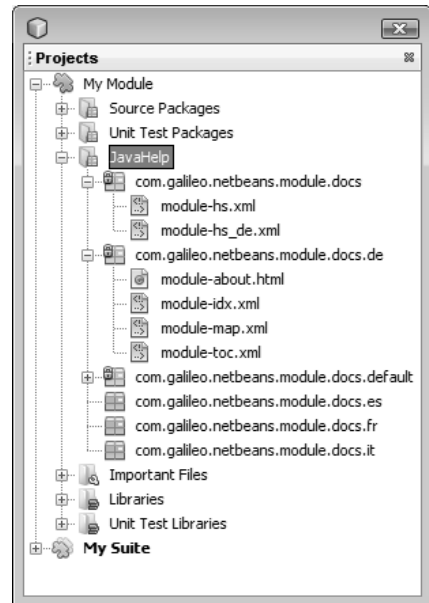
Internacionalizace nápovědy

Stránky nápovědy včetně konfiguračních souborů nápovědy se internacionalizují stejně jako soubory `.properties` (viz část „Řetězce ve zdrojovém kódu“) připojením identifikátorů země a/nebo jazyka. Důsledkem toho, že komplex nápovědy aplikace se typicky skládá z většího množství souborů, je matoucí struktura. Proto je zde možné ukládat soubory určené k internacionalizaci do podadresáře (viz obrázek 10.2). Identifikátor jazyka a země pak už není potřeba, protože jej reprezentuje podadresář.

Pouze soubor `module-hs.xml` kolekce témat nápovědy zůstává v základním adresáři a nemá žádný identifikátor. V tomto souboru pak odkazujeme na odpovídající adresáře (viz výpis 10.4). Soubor kolekce témat nápovědy bez identifikátoru se vždy používá v případě, že aktivní místní nastavení národního prostředí neodpovídá přítomným existujícím souborům. Je běžnou praxí, že základní balíček obsahuje anglickou verzi nápovědy.

Výpis 10.4: Definiční soubor nápovědy odkazující na konkrétní balíček jazyka

```
<maps>
  <homeID>com.galileo.netbeans.module.about</homeID>
  <mapref location="default/module-map.xml" />
</maps>
<view mergetype="javax.help.AppendMerge">
  <name>TOC</name>
  <label>Contents</label>
  <type>javax.help.TOCView</type>
  <data>default/module-toc.xml</data>
```



Obrázek 10.2: Kolekce témat nápovědy pro konkrétní jazyky se ukládají v oddělených adresářích

```
</view>
<view mergetype="javafx.help.AppendMerge">
  <name>Index</name>
  <label>Index</label>
  <type>javafx.help.IndexView</type>
  <data>default/module-idx.xml</data>
</view>
```

Internacionalizace ostatních zdrojů

Výše zmíněné způsoby internacionalizace aplikace patří k nejdůležitějším možnostem implementace mezinárodní podpory v aplikacích. Mimo ně existují však i další doplňující možnosti pro internacionalizaci dalších komponent aplikace. Platforma NetBeans právě tyto možnosti poskytuje.

Obrázky

Nejenom text lze přizpůsobit pro konkrétní jazyk a zemi, ale také grafiku, jako např. obrázky a ikony. Pro tento účel poskytuje třída `ImageUtilities` verzi metody `loadImage()`, která se obvykle používá pro nahrávání grafiky. Nastavte Boolean parametr určující, zda se má pro konkrétní jazyk/zemi nahrát odpovídající dostupná verze ikony s ohledem na aktuální místní nastavení národního prostředí. Metoda `NbBundle.getLocalizingSuffixes()` vrací seznam možných identifikací použitých v daném pořadí při vyhledávání.

```
Image img = ImageUtilities.loadImage("resources/icon.gif", true);
```

Pokud je při tomto volání aktuální místní nastavení národního prostředí např. `de_DE`, pak se nejprve vyhledají `icon_de_DE.gif` a `icon_de.gif`.

Libovolný jiný soubor

Platforma NetBeans definuje speciální protokol pro nahrání ostatních internacionalizovaných zdrojů. Je to protokol `nbresloc`, což je rozšíření protokolu `nbres`, který nahrává zdroje všech dostupných modulů. Pro tento protokol můžeme jednoduše vytvořit objekt URL pro adresovaný zdroj:

```
URL u = new URL("nbresloc:/com/galileo/netbeans/module/icon.png");
ImageIcon icon = new ImageIcon(u);
```

Pokud je místní nastavení `de_DE` a existuje-li soubor se jménem `icon_de_DE.png` nebo `icon_de.png`, pak se tato ikona nahraje namísto `icon.png`.

Adresáře a soubory

Systémový `FileSystem` poskytuje dva speciální atributy pro internacionalizaci jmen a ikon souborů a adresářů. To má smysl např. pro nabídky, jejichž jména se deklarují pouze v souboru `layer` a nelze je načíst třídou `NbBundle`. To se týká například jmen akcí. Jedná se o atributy `SystemFileSystem.localizingBundle` a `SystemFileSystem.icon`. První atribut, což je odkaz na váš balík zdrojů jazykových prostředků, se uvede bez přípony `.properties`. V tomto balíku se vyhledá klíč, který odpovídá kompletní cestě adresáře nebo souboru, který obsahuje atribut `SystemFileSystem.localizingBundle`. Ukázka v následujícím příkladu (viz výpisy 10.5 až 10.7) se týká `Menu/MyMenu` a `Menu/MyMenu/MySubMenu`. S atributem `SystemFileSystem.icon` můžeme případně nastavit i ikonu pro adresář nebo soubor. V případě, že chceme nahrát internacionalizovanou verzi ikony, použijeme protokol `nbresloc`.

Výpis 10.5: layer.xml

```
<folder name="Menu">
  <folder name="MyMenu">
    <attr name="SystemFileSystem.localizingBundle"
      stringvalue="com.galileo.netbeans.module.Bundle"/>
    <folder name="MySubMenu">
      <attr name="SystemFileSystem.localizingBundle"
        stringvalue="com.galileo.netbeans.module.Bundle"/>
      <attr name="SystemFileSystem.icon"
        urlvalue="nbresloc:/com/galileo/netbeans/module/icon.png"/>
    </folder>
  </folder>
</folder>
```

Výpis 10.6: Bundle.properties

```
Menu/MyMenu=Extras
Menu/MyMenu/MySubMenu=My Tools
```

Výpis 10.7: Bundle_de.properties

```
Menu/MyMenu=Extras
Menu/MyMenu/MySubMenu=Meine Tools
```

Navic k těmto dvěma atributům představila verze 6.5 platformy NetBeans všeobecně použitelný přístup k lokalizujícím atributům. Namísto `stringvalue` či `urlvalue` můžeme využít `bundlevalue` pro odkaz na řetězec v balíku jazykových prostředků `properties`. Můžeme použít `bundlevalue` pro libovolný atribut v `XMLFileSystem` jako v systémovém souborovém systému. Už jsme tento přístup použili, když jsme v kapitole 4 registrovali akce, v kapitole 7 registrovali `DataLoader` a když jsme přidávali panely nastavení v kapitole 9.

Podívejme se nyní na registraci akce v souboru `layer`. Chceme umístit záznam se jménem akce do balíku vlastností jazykových prostředků namísto toho, abychom jej zapsali přímo do `layer` souboru. Pro nahrání hodnoty atributu `displayName` z balíku vlastností se záznam definuje tak, jak uvádí výpis 10.8.

Výpis 10.8: Použití atributu `bundlevalue`

```
<file name="com-galileo-netbeans-module-MyFirstAction.instance">
  <attr name="displayName"
    bundlevalue="com.galileo.netbeans.module.Bundle#CTL_MyFirstAction"/>
</file>
```

S tímto záznamem jsme schopni umístit jméno akce do souboru `properties` tak, jak uvádí výpis 10.9.

Výpis 10.9: Hodnoty atributů souboru `layer` v `properties` souboru

```
com/galileo/netbeans/module/Bundle.properties
  CTL_MyFirstAction=My First Action
```

Administrace a příprava lokalizovaných zdrojů

Až doposud se lokalizované zdroje ukládaly ve stejném adresáři modulu, kde se hledají soubory `Bundle.properties` nebo ikony. Ale jak udržet zdroje pro každý jazyk oddělené na jiném místě a později již uvolněný modul snadno rozšířit dalším překladem do jiného jazyka? I pro tento scénář má platforma NetBeans snadnou podporu. Nabízí totiž možnost, jak oddělit zdroje od ostat-

ních (v podstatě hlavně tříd) modulu. Poskytuje adresář locale, který má své umístění v úložišti modulu (viz obrázek 10.3). Zdroje pro jazyk v archivu JAR v tomto adresáři se rozšíří o identifikátor jazyka/země. Archiv musí mít stejné jméno jako JAR archiv modulu. V tomto archivu rozšířeném o lokalizované zdroje se spravují veškeré jazykově/místně specifické zdroje. Mají stejnou balíčkovou strukturu jako vlastní modul. Zdroje se tak oddělí, a lze je dokonce aktualizovat nezávisle. Překlad modulů platformy NetBeans se realizuje stejným způsobem.

Je nutné podotknout, že oddělené lokalizované zdroje však stále potřebují své doplnění o identifikátor jazyka/země. Archiv rozšířený o lokalizované zdroje však nepotřebuje soubor manifestu, neboť archiv identifikuje výhradně jméno lokalizovaného balíku a jeho umístění na cestě zavaděče třídy modulu (viz kapitola 2).

Zde uvedený příklad demonstruje umístění zdrojů pro němčinu v adresáři locale v archivu rozšiřujícím aplikaci o lokalizované zdroje. Zdroje pro angličtinu, které nemají žádný identifikátor, jsou základní zdroje a poskytují je přímo archiv JAR modulu. Je nutno zmínit, že můžeme také vložit základní zdroje do archivu rozšiřujícího lokalizované zdroje, protože ten nemá žádný identifikátor, má stejné jméno jako modul a je v adresáři locale. Pak dojde ke kompletnímu oddělení zdrojů, které plánujeme lokalizovat, od vlastního modulu. Později se další práce usnadní v případě přidání dalšího jazyka například třetí stranou, protože je již zřejmé, které zdroje se musí lokalizovat.



Obrázek 10.3: Umístění jazykově specifických zdrojů v odděleném archivu JAR v adresáři locale

Shrnutí

V této kapitole jsme se naučili, jak svoji aplikaci uzpůsobit pro různé země a jazyky (internacionalizace) a jak poskytnout různé zdroje pro přizpůsobení odlišným zemím a jazykům (lokalizace). Začali jsme řetězci ve zdrojovém kódu. Viděli jsme, že s pomocí třídy `NbBundle` a spolu s průvodcem v NetBeans IDE je velice jednoduché vložit jazykově specifický obsah do odděleného souboru `properties`.

Také jsme se podívali na lokalizaci stránek nápovědy. Nejenom řetězce lze lokalizovat, existuje totiž i podpora pro nahrávání místně specifické grafiky (ikony, obrázky). Naučili jsme se také, jak lokalizovat hodnoty v souboru `layer` a také jak distribuovat svůj lokalizovaný obsah v odděleném modulu.