
NetIQ® Identity Manager

Localization Toolkit Guide

Februaury 2017

Legal Notice

For information about NetIQ legal notices, disclaimers, warranties, export and other use restrictions, U.S. Government restricted rights, patent policy, and FIPS compliance, see <https://www.netiq.com/company/legal/>.

Copyright (C) 2017 NetIQ Corporation. All rights reserved.

Contents

About This Guide	5
1 Introduction to the Localization Toolkit	7
About the Localization Toolkit	7
Standard Localization Resource Groups	8
Custom Localization Resource Groups	9
Localization Toolkit Contents.	9
Using the Localization Toolkit	10
2 Using the Localization Toolkit	13
Exporting User Application Strings	13
Exporting User Application Driver Strings	13
Exporting Strings from the User Application WAR	16
Importing User Application Strings	19
Importing Strings to the User Application WAR	19
Importing User Application Driver Strings	22
Installing a Language Pack	30
Customer Requirements	30
Files for Customer Distribution	30
Customer Setup Steps	31
Running the Patch Installer	31
3 Working with Custom Localization Resource Groups	37
Creating Custom Localization Resource Groups	37
Required Changes for Custom Localization Resource Groups	38
Portlets	38
Portal Artifacts	38
Custom Localization Resource Group Example	39
Finding Bookmark Portlet Dependencies	39
Creating an Export File	40
Creating a Bookmark Localization Resource Group JAR	40
Adding a Custom Localization Resource Group and New Locale	40
Importing Translated Files	41
Modifying the User Application WAR	41
Packaging, Installing and Testing	41
Updating the Permission Index for a New Language	41
4 Glossary of Identity Manager User Application Terms	43

About This Guide

This guide describes how to use the Identity Manager User Application Localization Toolkit to add a language not typically supported by the Identity Manager User Application. It includes these sections:

- ♦ [Chapter 1, “Introduction to the Localization Toolkit,” on page 7](#)
- ♦ [Chapter 2, “Using the Localization Toolkit,” on page 13](#)
- ♦ [Chapter 3, “Working with Custom Localization Resource Groups,” on page 37](#)
- ♦ [Chapter 4, “Glossary of Identity Manager User Application Terms,” on page 43](#)

Audience

This guide is intended for Novell field offices and consultants who want to translate the Identity Manager User Application into a language that is not part of the standard language support set.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Additional Documentation

For documentation on other Identity Manager features, see the [Identity Manager Documentation Web site](http://www.netiq.com/documentation/idm402) (<http://www.netiq.com/documentation/idm402>).

1 Introduction to the Localization Toolkit

The Localization Toolkit allows you to translate the Roles Based Provisioning Module's strings to a language that is not in the standard set of supported languages listed in [Table 1-1 on page 7](#). Topics include:

- ♦ [“About the Localization Toolkit” on page 7](#)
- ♦ [“Localization Toolkit Contents” on page 9](#)
- ♦ [“Using the Localization Toolkit” on page 10](#)

NOTE: This toolkit is not intended to be used as a tool for rebranding the User Application.

Table 1-1 Supported Languages

Locale Code	Language
da	Danish
de	German
en	English
es	Spanish
fr	French
it	Italian
ja	Japanese
nl	Dutch
pt	Portuguese
ru	Russian
sv	Swedish
zh-CN	Chinese (China)
zh-TW	Chinese (Taiwan)

About the Localization Toolkit

The Localization Toolkit enables you to translate the entire User Application user interface strings or a subset of user interface strings (such as the strings used in the [Identity Self-Service](#) tab) into a language that is not one of the standard set of supported languages. You can choose any of the

supported languages as the source language to begin the translation. For example, if you wanted to translate the user application to Portuguese, you could use the Brazilian Portuguese strings as the base language.

The user interface strings are assigned to categories or **localization resource groups**. Localization resource groups provide a way to logically organize the strings that comprise the user interface. A single localization resource group typically includes all of the strings needed by a particular area of the user interface. You can translate one or more of the localization resource groups, depending on how your customers use the user application.

Standard Localization Resource Groups

The toolkit includes a set of standard localization resource groups that correspond to the main parts of the User Application. They are listed in [Table 1-2](#). If you wanted to localize all of the User Application except the **Admin** tab, you would select all of the localization resource groups except for Administrative.

There are dependencies among localization resource groups. For example, the Common localization resource group contains all of the common framework strings, so it must always be translated regardless of what other localization resource groups you translate.

Table 1-2 Standard Localization Resource Groups

Localization Resource Group Name	Description	Generates a JAR of this name (where xx indicates the language)
Accessory Portlets	Includes the strings for the accessory portlets. See “Creating Custom Localization Resource Groups” on page 37 for more information about translating a subset of accessory portlets.	accessory-resgrp_xx.jar
Administrative	Includes the strings for the Admin tab of the user application.	admin-resgrp_xx.jar
Base	Includes the strings for Juice and directory abstraction layer error messages.	base-resgrp_xx.jar
Common	Includes the common and framework strings. This group of strings must always be translated.	common-resgrp_xx.jar
Identity	Includes the strings for the Identity Self-Service tab.	identity-resgrp_xx.jar
Password Management	Includes the strings for the password self-service module (an external password management WAR).	pwdmgt-resgrp_xx.jar
Requests & Approvals	Includes the strings needed for the Requests & Approvals tab.	apwa-resgrp_xx.jar
Roles	Includes the strings needed for the Roles tab. You should include the Requests & Approvals localization resource group when you translate the Roles group. Roles and Requests & Approvals are related. If you translate only the strings related to the Roles tab, your user interface will not be consistent.	roles-resgrp_xx.jar

Custom Localization Resource Groups

The Localization Toolkit also enables you to create one or more *custom* localization resource groups. Custom localization resource groups allow you to reduce the contents of a localization resource group. For example, you might have a customer that only uses one or two accessory portlets. You can create a custom localization resource group that contains only those portlets. Your customer can then translate just the contents of that localization resource group rather than the entire accessory portlet localization resource group. Creating a custom localization resource group is a complex task. When you use custom localization resource groups, be sure to allow extra testing time in your schedule to ensure that the user interface language is consistent (particularly if you create a custom localization resource group for the any of the main user interface tabs.) For more information, [“Creating Custom Localization Resource Groups” on page 37](#).

Localization Toolkit Contents

The Localization toolkit (L10nTools.zip) contains the utilities described in [Table 1-3, “Localization Toolkit Utilities,” on page 9](#) and other files such as the reports and JARs described in [Table 1-4, “Supporting Localization Toolkit Files,” on page 10](#).

Table 1-3 Localization Toolkit Utilities

Utility Name	Description
Localization Toolkit	Located in fl10n-utility.jar. Use this utility to: ♦ Export strings stored in the Identity Manager User Application WAR in a properties file format for translation. ♦ Import translated strings. ♦ Package the translated strings into an installable patch for distribution to customers. Launch the utility using this command: <pre>java -jar fl10n-utility.jar</pre>
PatchUserApp	A patch installer: Use this to add the JAR containing the translated strings to the User Application WAR. To launch the patch installer, use: ♦ Windows*: PatchUserApp.exe ♦ Linux: PatchUserApp.bin ♦ Solaris: PatchUserApp.sol.bin
TranslationVerifier	Use this utility to verify that: ♦ For every base file sent to translation, there is a translated file. ♦ There is a one-to-one correspondence between the keys in the base files and the keys in the translated files. This utility runs as part of the localization toolkit. You can also run it separately. Launch the utility by using this command: <pre>java -cp fl10n-utility.jar com.novell.soa.fl10ntools.TranslationVerifier</pre>

Utility Name	Description
CreateDummyPropertiesFiles	Use this utility to create properties files for a dummy language. This is useful if you want to do a test run of the localization process. This utility copies the existing properties files and prepends a string that you specify to each of the existing strings. It appends the string to the name of each JAR. Launch the utility using by this command: <pre>java -cp fl10n-utility.jar com.novell.soa.fl10ntools.CreateDummyPropertiesFiles <directory-name> <stringtoinsert></pre>

Table 1-4 Supporting Localization Toolkit Files

File	Description
Localization resource group language string JARs	A set of JARs, one for each localization resource group, in each supported language. For more information about localization resource groups, see Table 1-2, “Standard Localization Resource Groups,” on page 8 .
dependencies.html	Use this file when you create a custom resource bundle. It lists the dependencies among resources.
fl10n-word-count.txt	This file lists the word counts per localization resource group in English. This file is provided to help you estimate the cost of translating each localization resource group. You can run this tool by using this command: <pre>java -cp fl10n-utility.jar com.novell.soa.fl10ntools.CountWordsInPropertiesFiles \$DirForPropertiesFiles \$langcode \$outputfile</pre> Where <i>\$DirForPropertiesFiles</i> is the directory where the properties files (JARs containing the strings) are located, <i>\$langcode</i> is the Java* language code (for example, en for English), and <i>\$outputfile</i> is the name of the report file to create.

Using the Localization Toolkit

The following table provides an overview of the steps for using the localization toolkit along with links to additional information:

Table 1-5 Localization Steps

Step	Description	For more information
1	Obtain and install the Localization Toolkit from Novell, Inc.	See the Readme.html for more information on installation and requirements.

Step	Description	For more information
2	Export string resources for translation	You export: ♦ User Application driver data such as: ♦ Display labels for entities, provisioning request definitions, and teams ♦ E-mail notification templates See “Exporting User Application Driver Strings” on page 13 ♦ User Application WAR strings. See “Exporting Strings from the User Application WAR” on page 16.
3	Send the ZIP containing the property files to translators.	The translator does the following: ♦ Translates the strings. ♦ Packages the translated strings into a JAR that: ♦ Has the same directory structure as the original JAR. ♦ The new JAR can have any name and extension. But the exported files typically include the Java two-character locale code as a naming convention. WARNING: If the file structure and naming conventions are not followed, the import of the translated strings fails.
4	Add the new locale to User Application driver and any custom localization resource groups	See “Adding Locales to the User Application Driver” on page 22 .
5	Import the translated User Application driver property files	See “Importing User Application Driver Strings” on page 22 .
6	Import the JARs containing the new language strings using the Identity Manager User Application Localization utility	Import the translated strings by using the toolkit. This verifies and validates the strings that you are importing, and packages the translations into a ZIP file that you install by using the patch installer. For more information, see “Importing Strings to the User Application WAR” on page 19.
7	Package the translated files for distribution to customers	See “Installing a Language Pack” on page 30 .

NOTE: Locale changes require an application server restart before they appear in the user application.

2 Using the Localization Toolkit

This section describes how to use the Localization Toolkit string export and import utilities. Topics include:

- ♦ [“Exporting User Application Strings” on page 13](#)
- ♦ [“Importing User Application Strings” on page 19](#)
- ♦ [“Installing a Language Pack” on page 30](#)

Exporting User Application Strings

User Application strings are stored in the User Application WAR file and in the User Application driver. To export strings stored in the User Application WAR, you use the User Application Localization utility; to export strings stored in the User Application driver, you use Designer for Identity Manager. This section includes the following topics:

- ♦ [“Exporting User Application Driver Strings” on page 13](#)
- ♦ [“Exporting Strings from the User Application WAR” on page 16](#)

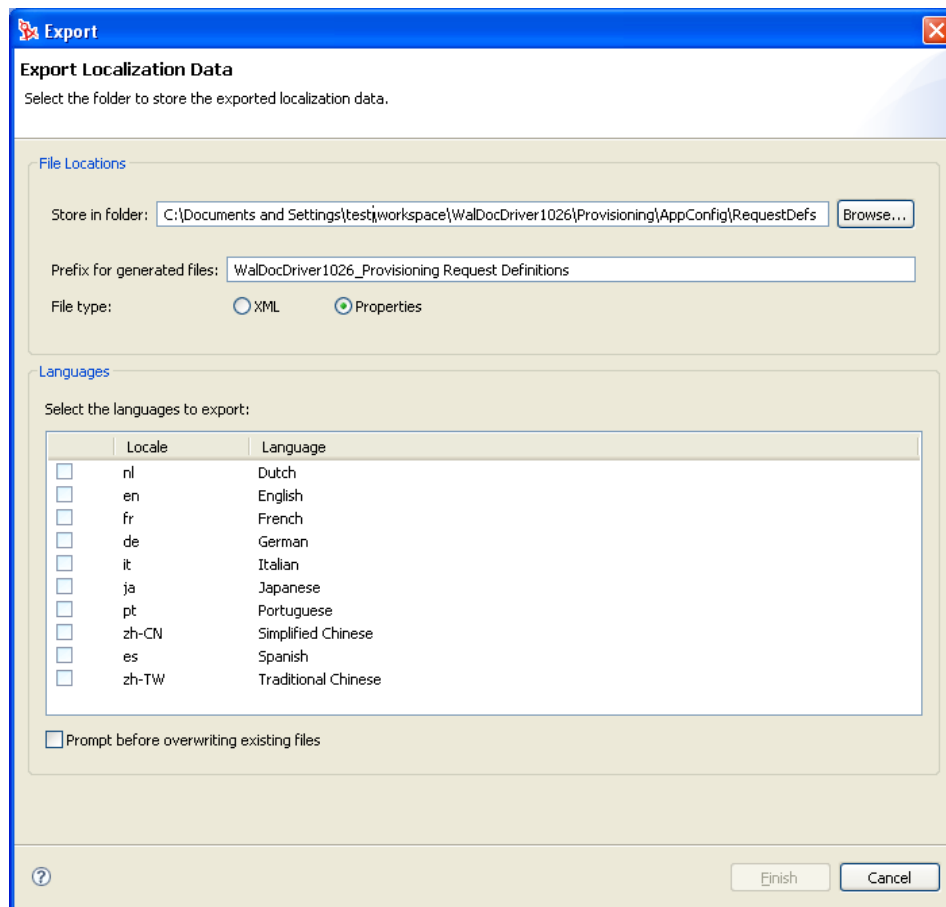
Exporting User Application Driver Strings

User Application driver strings include display labels defined for directory abstraction layer objects, provisioning request definition objects, provisioning team definitions, and role catalog. They also include the contents of e-mail notification templates. Designer provides a wizard that lets you export display labels and e-mail notification templates from a project to a properties file that can be localized then imported. Designer allows you to export an entire driver, all objects of a single type (such as provisioning request definitions), or a single object at a time.

- ♦ [“Exporting Display Labels” on page 13](#)
- ♦ [“Exporting E-Mail Templates” on page 15](#)

Exporting Display Labels

- 1 To launch the Export Localization Data Wizard, right-click a container node or an object in the Provisioning view.
- 2 Select **Localize > Export Localization Data**.



3 Fill in the fields as follows:

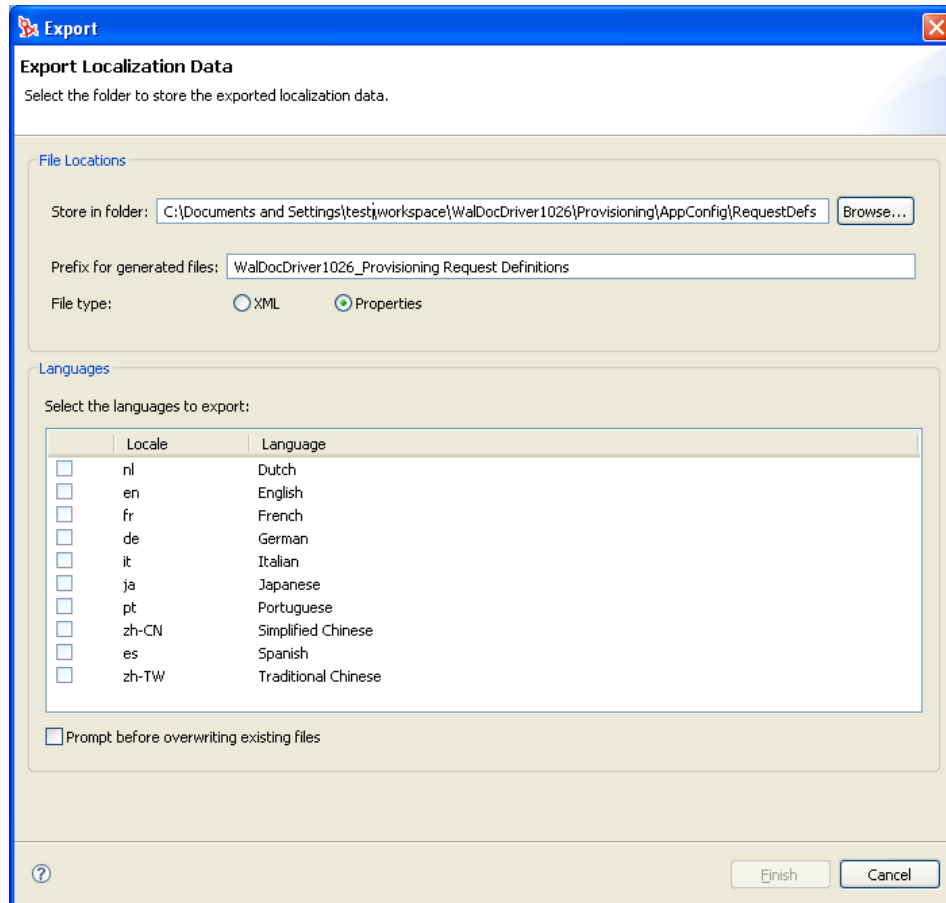
Field	Description
Store in folder	Specify the name of a local folder where the exported files should be written.
Prefix for generated files	Specify a prefix for the generated files. Determine a naming strategy so you are able to identify the files.
File Type	Select Properties . Properties use Unicode* for encoding.
Select the languages to export	Select the base language for the export file. Designer generates a file for the specified language that contains the display labels in the specified language. The translator must provide the same file structure for the new language. The localizations need to be added to this file in the proper format so you can import them to the proper User Application driver objects.
Prompt before overwriting existing files	If this option is selected, Designer prompts you before it overwrites any existing files of the same name in the target directory.

4 Click **Finish**.

Designer displays a message describing the result of the export operation

Exporting E-Mail Templates

- 1 To launch the Export Localization Data Wizard, right-click the Default Notification Collection container or one of the templates.
- 2 Select **Localization > Export Localization Data**.



3 Fill in the fields as follows:

Field	Description
Store in folder	Specify the name of a local folder where the exported files should be written.
Prefix for generated files	Specify a prefix for the generated files. Determine a naming strategy so you are able to identify the files.
File Type	Select Properties . Properties use Unicode for encoding.
Select the languages to export	Select the base language for the export file. Designer generates a file for the specified language that contains the display labels in the specified language. The translator must provide the same file structure for the new language. The localizations need to be added to this file in the proper format so you can import them to the proper User Application driver objects.
Prompt before overwriting existing files	If this option is selected, Designer prompts you before it overwrites any existing files of the same name in the target directory.

- 4 Click **Finish**. Designer displays a message describing the result of the export operation.

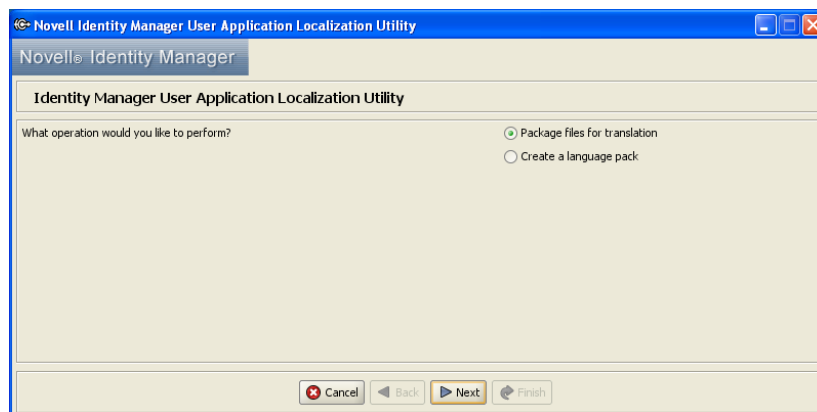
Exporting Strings from the User Application WAR

Use the User Application Localization utility to export the strings stored in the User Application WAR. These strings are organized into localization resource groups. Follow these steps to export one or more localization resource groups.

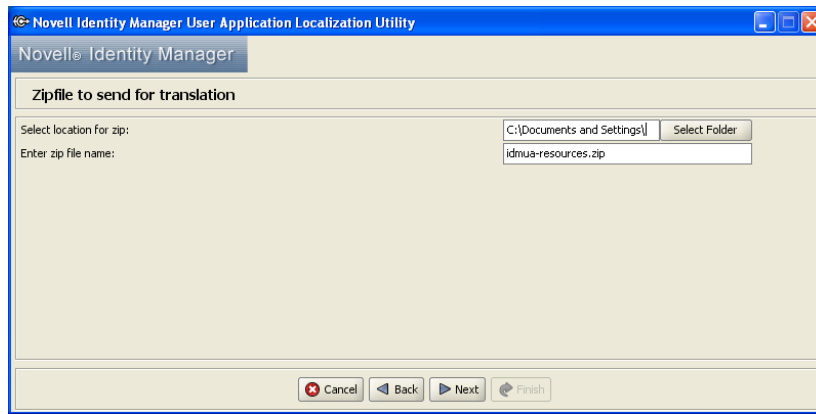
- 1 Launch the User Application Localization utility with this command:

```
java -jar fl10n-utility.jar
```

You are prompted to choose an action.



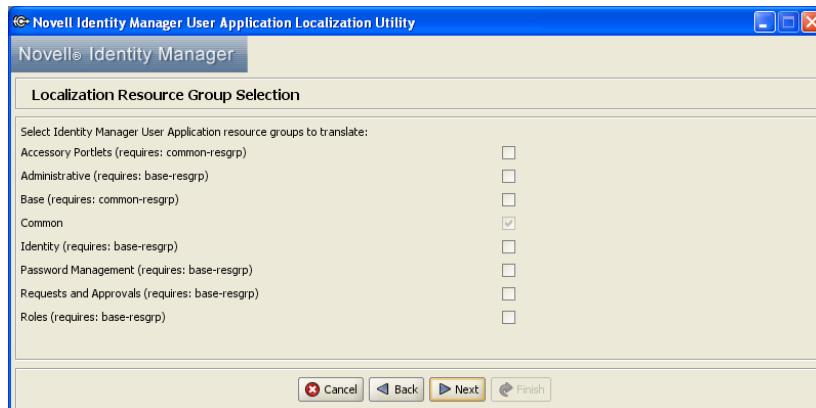
- 2 Choose **Package files for translation**, then click **Next**.



3 Fill in the fields as follows:

Field	Description
Select location for zip	Type the name of the folder where the export file should be created or click Select Folder . This folder must already exist.
Enter Zip file name	Type the name of the file the export should create or accept the default.

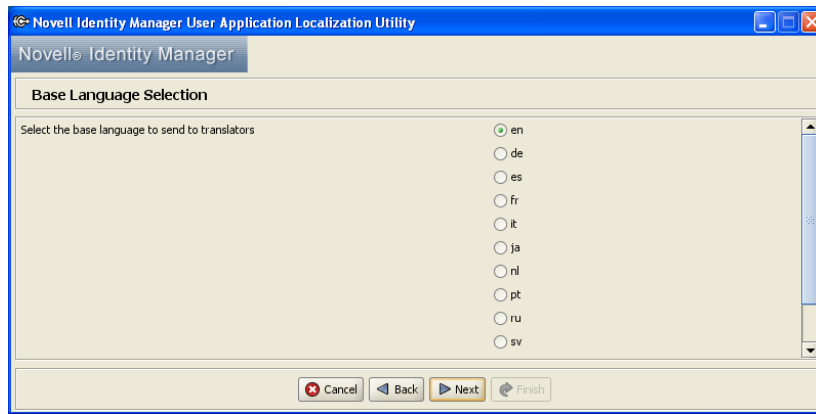
4 Click **Next**. You are prompted for a localization resource group selection.



5 Select one or more localization resource groups to translate.

For more information about what localization resource groups to select, see [Table 1-2, “Standard Localization Resource Groups,” on page 8](#). For information on creating your own custom localization resource groups, see [“Creating Custom Localization Resource Groups” on page 37](#)

6 Click **Next**.

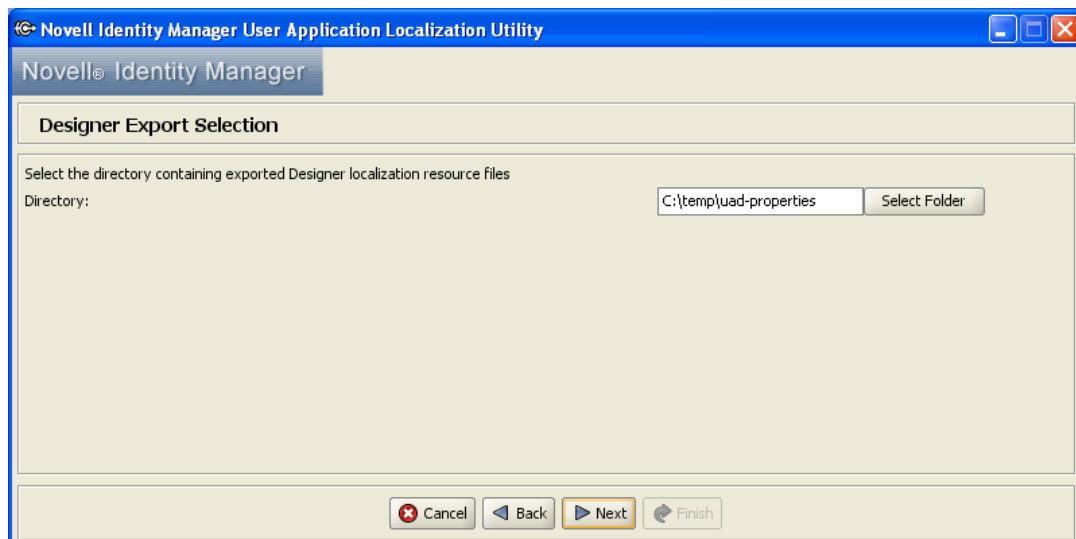


7 Select a language.

This language determines the language of the strings written to the export file that you send to the translator. The translator uses this as the base language to perform the translations.

8 Click **Next.**

You are prompted for the name and location of the User Application driver export files. The utility adds these files to the ZIP file that you send to translation.



9 Click **Next, then click **Finish**.**

The utility generates the ZIP file in the location you specified. The ZIP file contains the strings for the localization resource groups in the base language you specified. When it is complete, you can send this ZIP file for translation. For more information on what to do with the translated strings, see [“Importing User Application Strings” on page 19](#).

Roles Resource Group and Non-Latin-1 Languages

For Roles reports to display correctly in non-latin-1 languages, you must manually modify the NRFRsrc_xx.properties file to include the entries below:

- ♦ REPORT_LABEL_PDF_FONT=STSong-Light
- ♦ REPORT_LABEL_PDF_ENCODING=UniGB-UCS2-H

These entries are automatically included in the ru, ja, zh_CN, and zh_TW default localization files. If you are adding a new non-latin-1 translation, such as Korean, you must set the properties specified in the NRFRsrc_ko.properties file.

The NRFRsrc_xx.properties is located in the roles-resgrp\com\novell\idm\nrf\resources folder of the ZIP file containing the exported strings.

NOTE: For some languages, you may also need to perform some additional steps to ensure that all characters display properly in PDFs generated for role reports. For example, if you generate a role report PDF in Polish, you may notice that some characters are missing. To address this problem, you would need to modify the role detail XML data and configuration XML data in iManager. For details, see the topic “Ensuring that Characters Display Properly in Role Report PDF Files” in [Localizing and Customizing User Application Strings](http://www.netiq.com/documentation/idm402/agpro/?page=/documentation/idm402/agpro/data/bg1knuk.html) (<http://www.netiq.com/documentation/idm402/agpro/?page=/documentation/idm402/agpro/data/bg1knuk.html>).

Importing User Application Strings

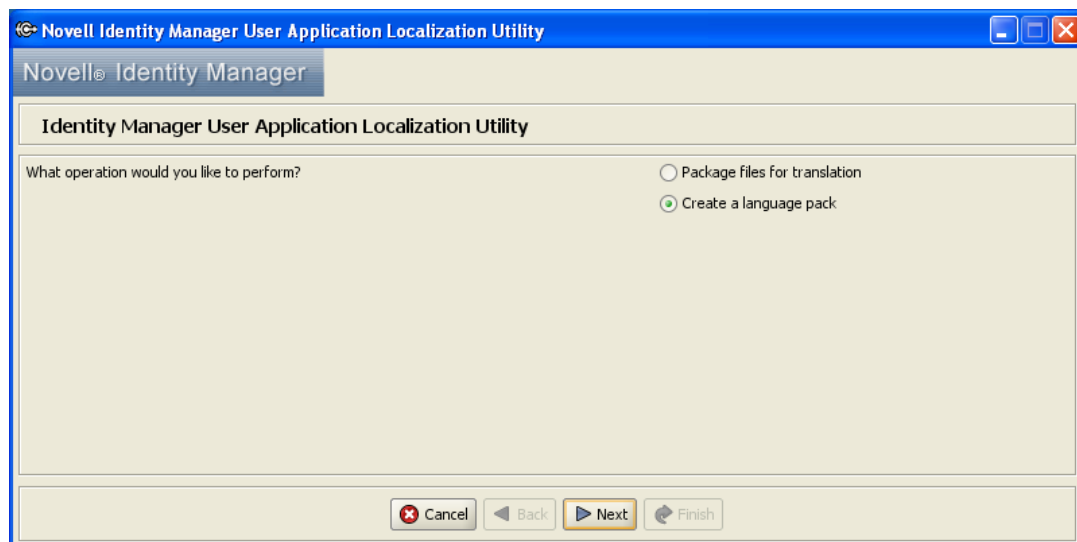
This section describes how to import the newly translated strings. Topics include:

- ♦ “Importing Strings to the User Application WAR” on page 19
- ♦ “Importing User Application Driver Strings” on page 22

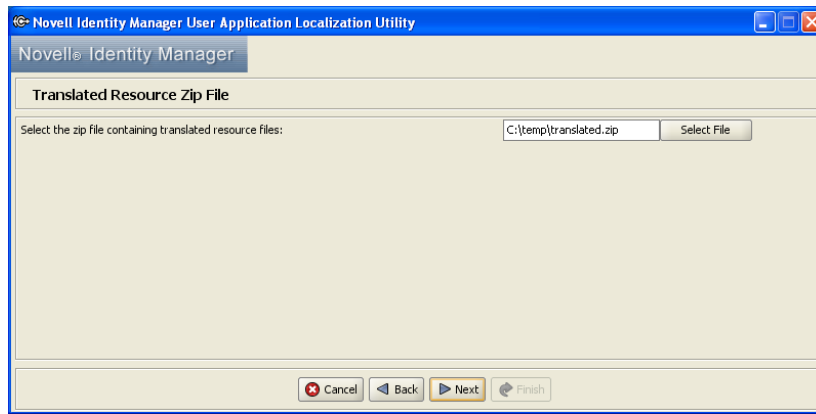
Importing Strings to the User Application WAR

- 1 Use the following command to launch the User Application Localization utility:

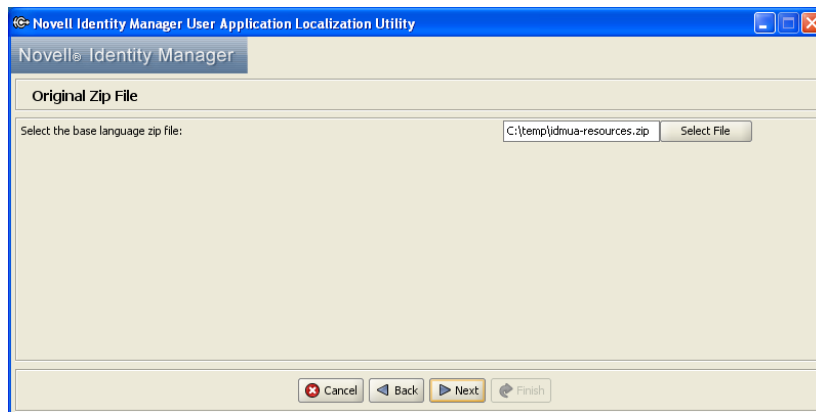
```
java -jar fl10n-utility.jar
```



- 2 Select **Create a language pack**, then click **Next**.



- 3 Select the name and location of the ZIP file that you received from the translator, then click **Next**.



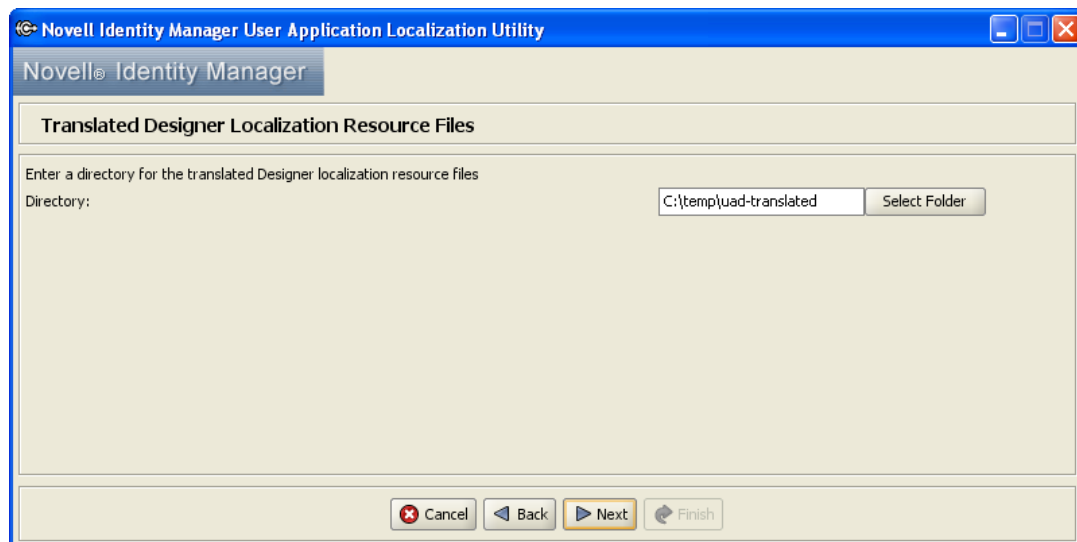
- 4 Select the name and location of the folder containing the original files that you sent for translation.

This is used to compare the newly translated files against the original files

- 5 Click **Next**, then supply the name and location for a new ZIP file.

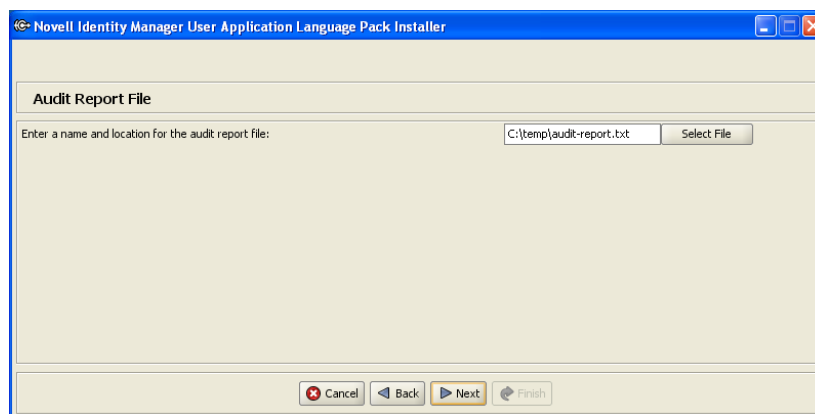
The utility creates this ZIP file. It adds the translated strings and any other resources needed for customers to use with the patch installer.

- 6 Click **Next**, then specify the name and location where the utility can locate the User Application driver's translated strings



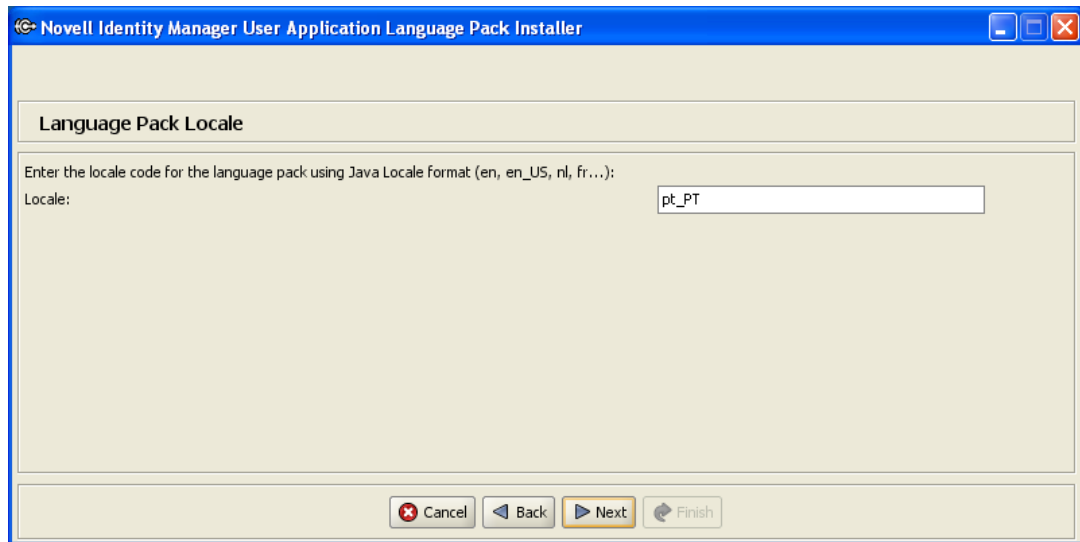
7 Click Next.

The tool verifies multiple things about the translated files (as described in [“About the Localization Toolkit” on page 7](#)). Use this panel to select a location for the tool to write those audit records.

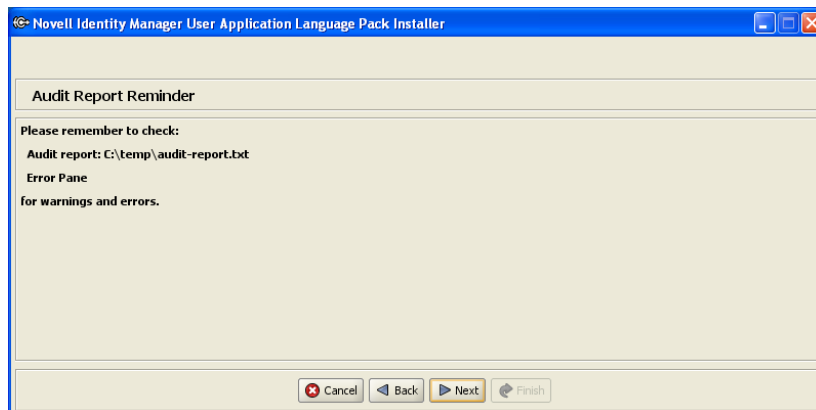


8 Click Next, then type the Java language code for the newly created language pack.

See the Java Web Site for these codes. The language you enter here must correspond to the language that you define for the supported driver locales that you specify in Designer.



9 Click **Next**, then click **Finish**.



Importing User Application Driver Strings

Designer provides a wizard that lets you import a properties file containing the translated display labels in your User Application driver project. You can import an entire driver, all directory abstraction layer or provisioning request definitions, or a single object at a time. Before you can import display labels for a new locale, you must add the new locale to the list of supported locales as described in [“Adding Locales to the User Application Driver” on page 22](#).

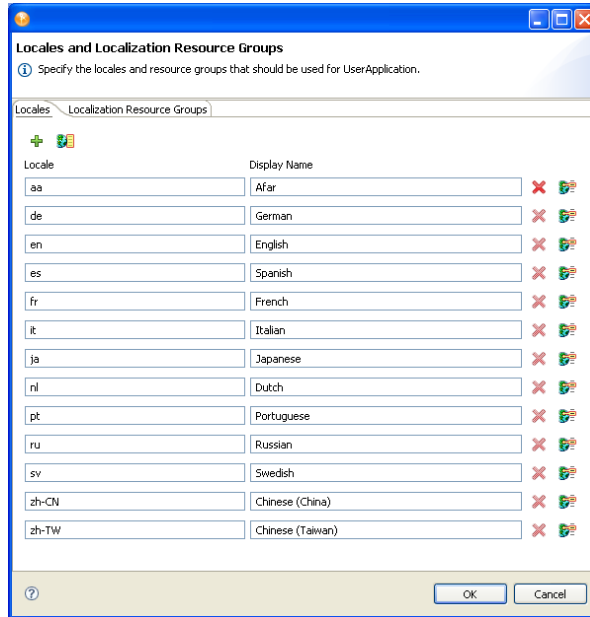
- ♦ [“Adding Locales to the User Application Driver” on page 22](#)
- ♦ [“Importing Localized Display Labels” on page 26](#)
- ♦ [“Importing E-Mail Templates” on page 27](#)
- ♦ [“Ensuring that Dates Display Properly” on page 29](#)
- ♦ [“Ensuring that Dates Display Correctly in Norwegian” on page 29](#)

Adding Locales to the User Application Driver

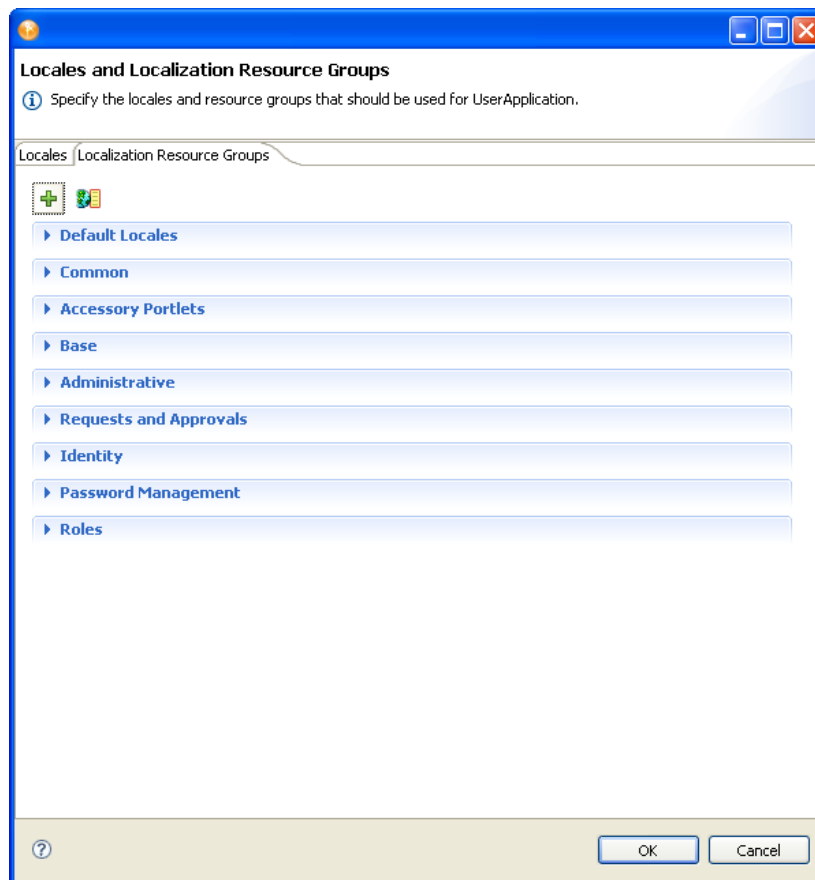
To add a new locale to the User Application driver’s supported locales list:

- 1 Right-click the User Application driver in the Provisioning view, then select **Configure > Locales**.

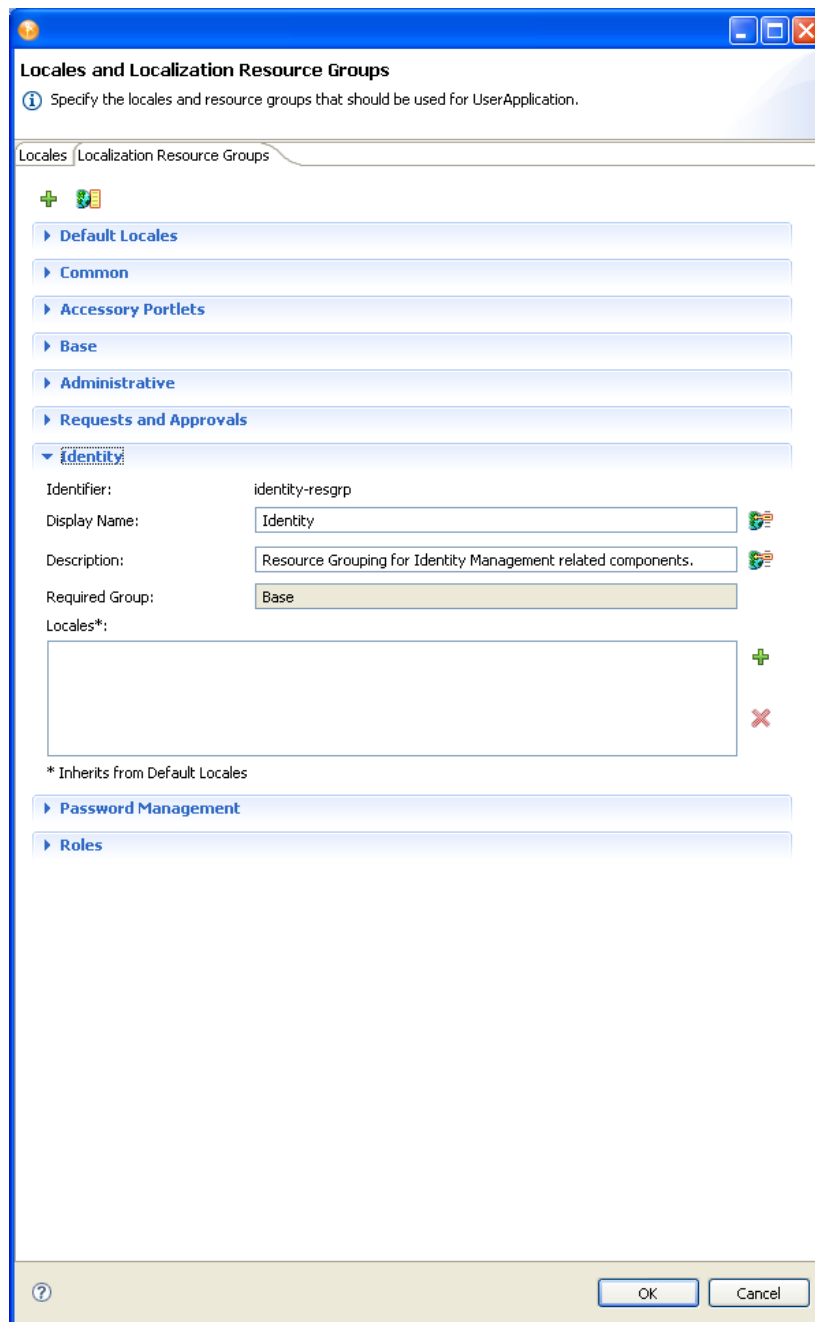
Designer displays the Supported Locales dialog box.



- 2 Click , then select the Language and optionally the Country from the drop-down or type the value. The language is required, but the country is not. The language is displayed as the Locale in the Supported Locales dialog box.
- 3 Click **OK**.
- 4 Click *Localization Resource groups*.

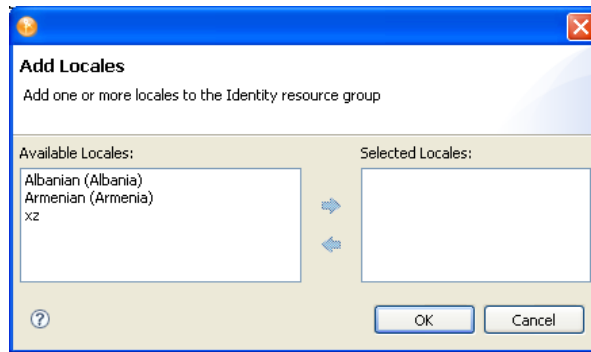


- 5 Select the **Localization Resource Group** where you want the new locale to be supported.
For example, if you want the locale to be used on the **Identity Self-Service** tab, you would select the Identity resource group.



NOTE: If the localization resource group you select includes a **Required Group**, Designer adds the locale to that group also. For example, when you add a locale to the Identity resource group, Designer also adds the locale to the Base resource group. But if you add the locale to the Base resource group first, Designer does not automatically populate the Identity resource group with the new locale.

- 6 Click  in the Locales area, then select the locale from the **Available Locales** and move it to the Selected Locales list.

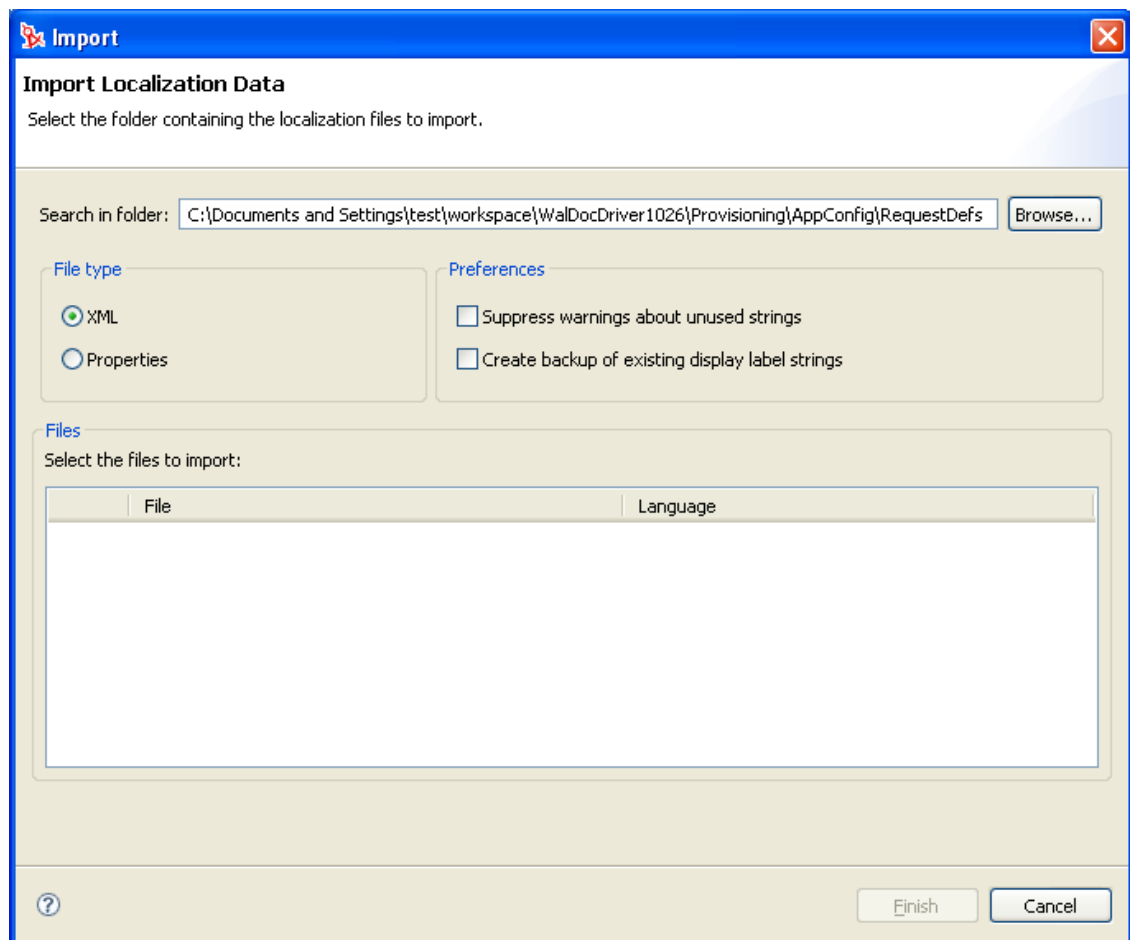


7 Click **OK**.

Importing Localized Display Labels

To import localized files:

- 1 To launch the Import Localization Data Wizard, right-click a container node or an object in the Provisioning view.
- 2 Select **Localize > Import Localization Data**.



3 Fill in the fields as follows:

Field	Description
Search in folder	Specify the folder location where the files to import are located.
File Type	Select Properties .
Preferences	Select Suppress warnings about unused strings if you want the wizard to suppress warning messages. Select Create backup of existing display label strings if you want the wizard to create a backup of the existing strings before the import. Useful if you need to revert.
Files	Select the files to import. This table is populated with the files from the folder location and File Type specified above. If it is blank, no files of the specified type are located in the target folder. The wizard attempts to determine the language by looking at the filename. If the name cannot be determined, it defaults to English. You can change the language using the language selection list. This list is populated with the list of languages from the supported locales list. You can change the Language column if the wizard assumes the wrong language. The wizard changes the file name to reflect the language you specify and imports the display labels to the corresponding language.

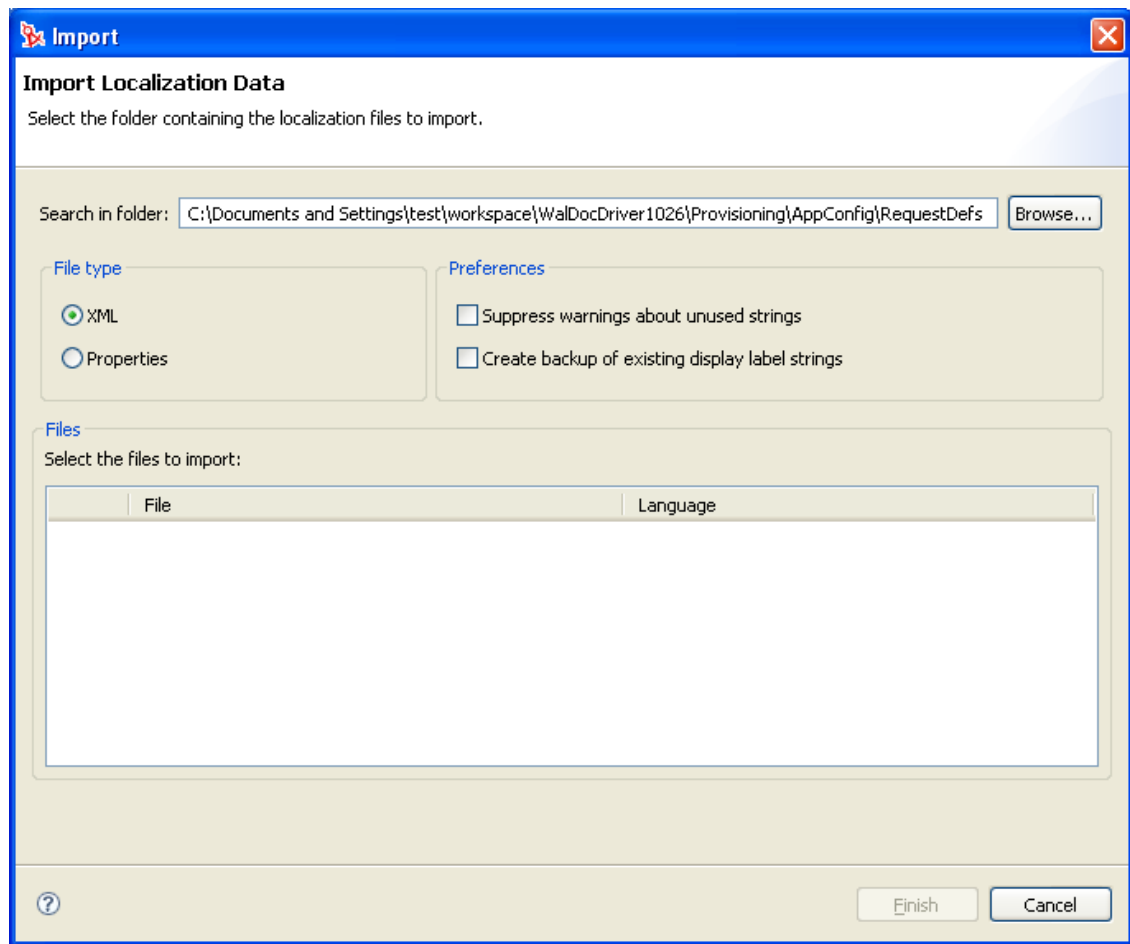
- 4 Click **Finish** to complete the import.

Designer displays a status dialog box that describes the results including any errors reading the files and any warnings about display label keys that are unused because no match was found.

Importing E-Mail Templates

To import localized e-mail templates:

- 1 To launch the Import Localization Data Wizard, right-click a container node or an object in the Default Notification Collection in the Outline view.
- 2 Select **Localization > Import Localization Data**.



3 Fill in the fields as follows:.

Field	Description
Search in folder	Specify the folder location where the files to import are located.
File Type	Select Properties .
Preferences	Select Suppress warnings about unused strings if you want the wizard to suppress warning messages. Select Create backup of existing display label strings if you want the wizard to create a backup of the existing strings before the import. Useful if you need to revert.

Field	Description
Files	Select the files to import. This table is populated with the files from the folder location and File Type specified above. If it is blank, no files of the specified type are located in the target folder. The wizard attempts to determine the language by looking at the filename. If the name cannot be determined, it defaults to English. You can change the language using the language selection list. This list is populated with the list of languages from the supported locales list. You can change the Language column if the wizard assumes the wrong language. The wizard changes the file name to reflect the language you specify and imports the display labels to the corresponding language.

- 4 Click **Finish** to complete the import.

Designer displays a status dialog box that describes the import results.

Ensuring that Dates Display Properly

If you notice that dates do not display properly in the User Application, you can check the following files to see if your language has been included in the list:

- /IDMProv/javascript/JUICE4IE.js.uncompressed.js
- /IDMProv/javascript/JUICE4IE.js.gz/JUICE4IE.js
- /IDMProv/javascript/JUICE4IE.js
- /IDMProv/javascript/3rdParty.js.uncompressed.js
- /IDMProv/javascript/3rdParty.js.gz/3rdParty.js
- /IDMProv/javascript/3rdParty.js

If you do not see your language in the list, contact Novell support for assistance.

To ensure that dates display properly, you may also need to verify that the Localized name and Localized description for a language are present in the User Application Driver in eDirectory. After adding a localization, you may find that the User Application driver is missing some localization information. To correct this, you may need to add a value for your language to the object attributes `nrfLocalizedDescrs` and `nrfLocalizedName`.

Ensuring that Dates Display Correctly in Norwegian

For language codes `no` and `nb`, you need to perform a workaround to ensure that dates display correctly in Norwegian. The `Date.js` file contains `no` but not `nb`, however, the `dmask` value (`dd/MM/yyyy`) is not correct. For both `no` and `nb`, the format should be `dd.MM.yyyy`.

To ensure that dates display correctly in Norwegian:

- 1 Copy the file `com/netiq/common/i18n/I18nDateTimeRsrc_en.properties`, modifying the locale portion of the file name to match the desired locale (for example, `I18nDateTimeRsrc_nb.properties`).

- 2 Modify the format(s) in the file to match the desired format. There are four format types: short, medium, long and full. These formats correspond to the `java.text.DateFormat.SHORT`, `.MEDIUM`, `.LONG` and `.FULL` constants.
- 3 Add the file to the `IDMProv.war` under `WEB-INF/classes/com/netiq/common/i18n` using the `jar` utility (file must be placed in a directory tree corresponding to the above path).

```
jar uvf IDMProv.war  
WEB-INF/classes/com/netiq/common/i18n/I18nDateTimeRsrc_nb.properties
```

Installing a Language Pack

This section describes how to install the newly created language pack. It includes the following sections:

- ♦ [“Customer Requirements” on page 30](#)
- ♦ [“Files for Customer Distribution” on page 30](#)
- ♦ [“Customer Setup Steps” on page 31](#)
- ♦ [“Running the Patch Installer” on page 31](#)

Customer Requirements

The customer must have:

- ♦ Designer for Identity Manager and the Roles Based Provisioning Tools plug-in (if you intend to localize Role Catalog objects).
- ♦ Identity Manager Roles Based Provisioning Module (installed but not used: clean database)

Files for Customer Distribution

You must provide the following to customers that want to use a non-standard language package:

- ♦ The patch installer:
 - ♦ Windows: `PatchUserApp.exe`
 - ♦ Linux: `PatchUserApp.bin`
 - ♦ Solaris*: `PatchUserApp.sol.bin`
- ♦ The Designer properties files containing the non-standard language translations. These files are comprised of Designer display labels, e-mail templates, and User Application driver configuration files.
- ♦ A language pack file containing the non-standard language translations. This file is the output from the User Application Localization utility. It should also contain any custom localization resource group changes necessary.
- ♦ The name of the locale to add to the User Application driver.
- ♦ The names of any custom localization resource groups that the customer must add.

Customer Setup Steps

The customer must perform the following to use a non-standard language pack:

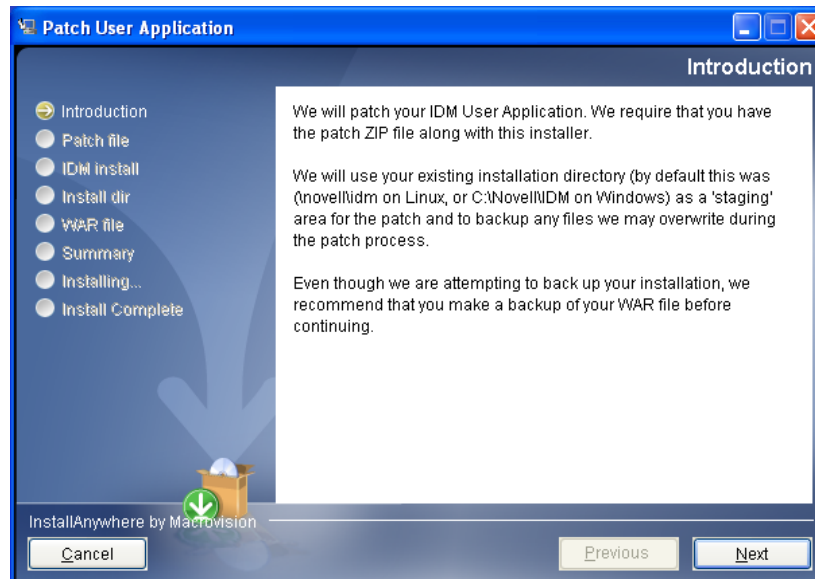
- ♦ In Designer:
 - ♦ Add the new locale to the list of supported User Application driver locales.
 - ♦ Add any new custom localization resource groups (if applicable) and the new locales.
 - ♦ Import the localized Designer components (e-mail templates, display labels, and driver configuration files).
 - ♦ If the language pack contains a custom localization resource group, add the new localization resource group.
 - ♦ Deploy (or redeploy) the User Application driver.
- ♦ Run the PatchUserApp program.
- ♦ Deploy the User Application WAR file and start (or restart) the application server.

NOTE: You must restart the application server for the locale changes to take effect. Because locale data is not cached you cannot simply deploy the User Application driver and flush the cache. The application server restart is required.

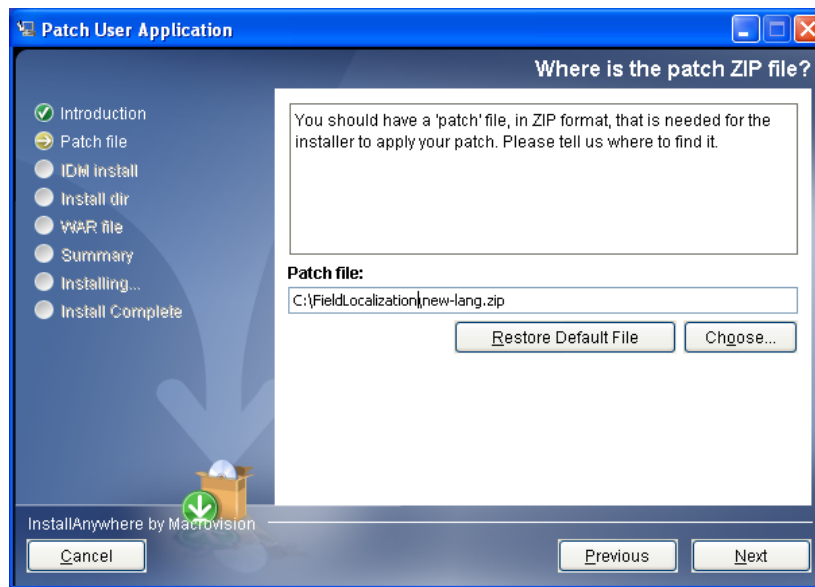
- ♦ Test the User Application.

Running the Patch Installer

- 1 Launch the installer:
 - ♦ On Linux, execute PatchUserApp.bin
 - ♦ On Solaris, execute PatchUserApp.so.bin
 - ♦ On Windows, double-click PatchUserApp.exe.



- 2 Click **Next**. You are prompted for the name and location of the language pack file.

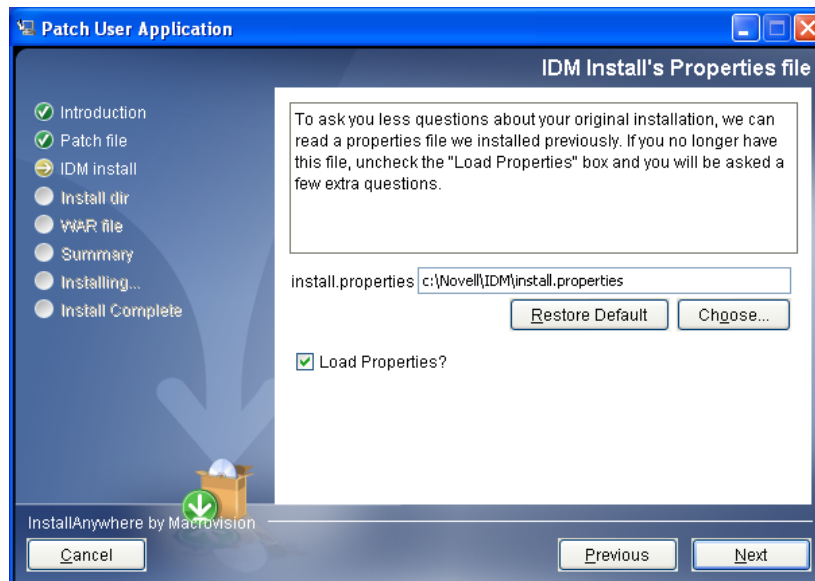


3 Fill in the fields as follows:

Field	Description
Patch file	Supply the ZIP file that contains the translated strings. This is the file generated by the Localization toolkit import described in "Importing Strings to the User Application WAR" on page 19

4 Click **Next**.

You are prompted for the name and location of the User Application's install.properties file. The install properties file is created when you initially install the User Application.

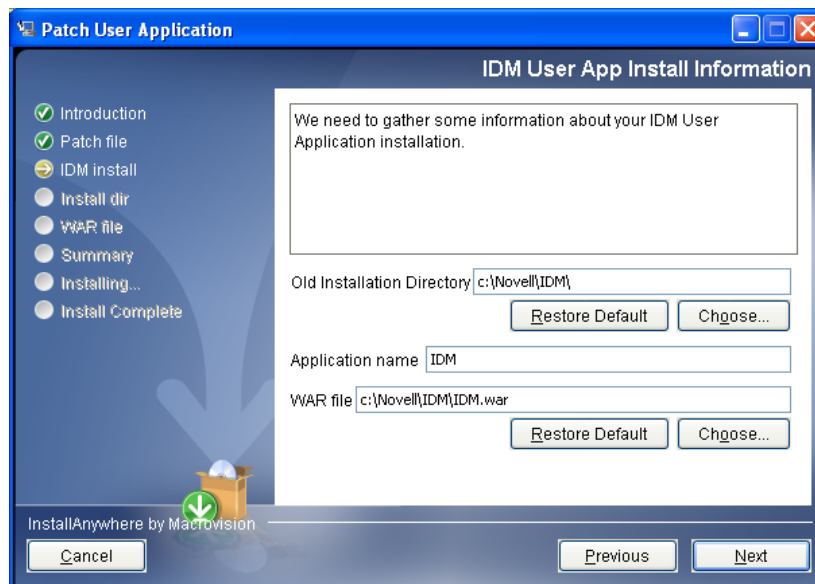


5 Fill in the fields as follows:

Field	Description
Install.properties	Provide the name and location of the install.properties file (if you know it). Otherwise, deselect Load Properties . If you can supply the install.properties location, skip to Step 8 .
Load Properties?	If selected, the patch installer detects where the User Application is deployed and it makes the necessary changes to it. After the changes are made, you must restart the application server for the changes to take effect. Deselect this check box if you do not know the location of the install.properties file or if your application server is WebSphere* or WebLogic*.

6 Click Next.

You are prompted for information about the User Application installation location.

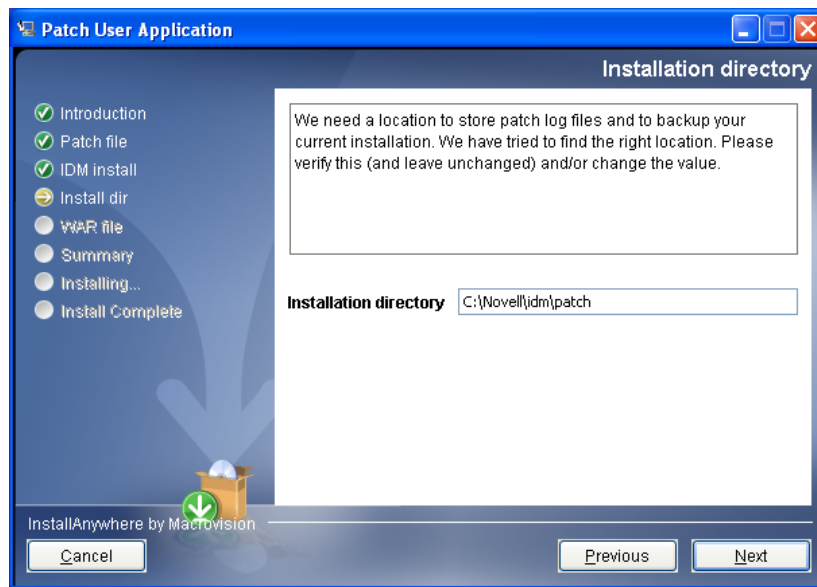


7 Fill in the fields as follows:

Field	Description
Old Installation Directory	Specify the name and location of the User Application installation directory.
Application Name	Specify the User Application context
WAR file	Specify the name and location of the User Application WAR file used during the initial installation.

8 Click Next.

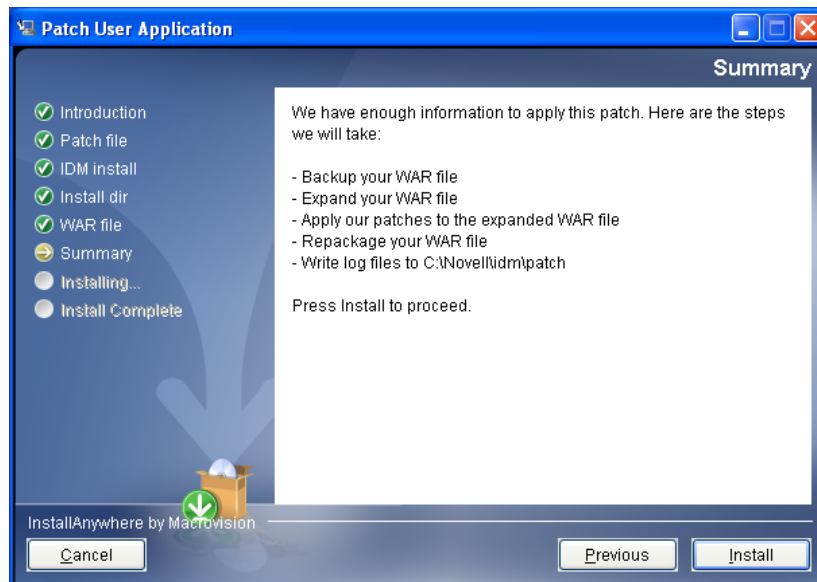
You are prompted for a directory location where the utility can backup your current installation.



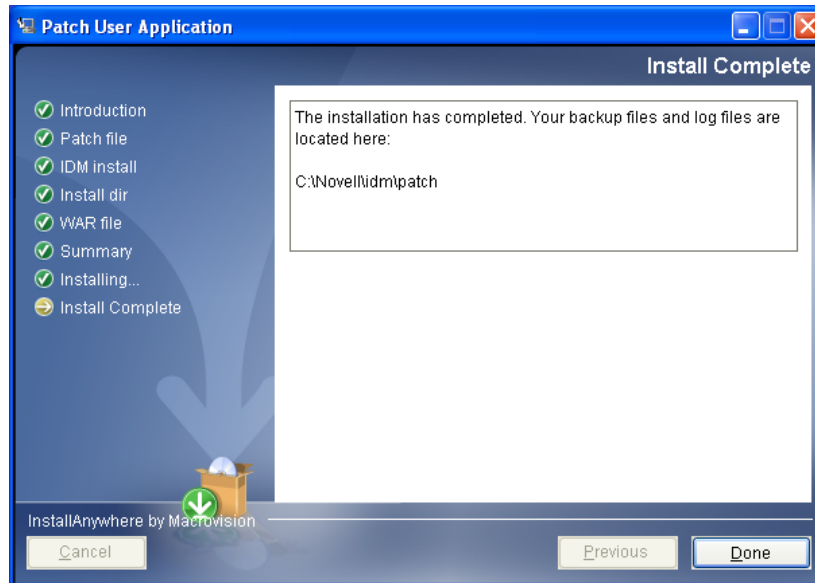
9 Fill in the fields as follows:

Field	Description
Installation directory	Specify the name and location where the patch installer can make a backup copy of your current installation before applying the patch. This is useful if you need to back out the patch at any point.

10 Click **Next**.



- 11 Review the Summary panel, click **Next**, then click **Done**.



The new language has been added to the User Application WAR. The customer can now deploy the WAR to an application server and begin testing.

3 Working with Custom Localization Resource Groups

This section describes how to define custom localization resource groups. Topics include:

- ♦ [“Creating Custom Localization Resource Groups” on page 37](#)
- ♦ [“Required Changes for Custom Localization Resource Groups” on page 38](#)
- ♦ [“Custom Localization Resource Group Example” on page 39](#)
- ♦ [“Updating the Permission Index for a New Language” on page 41](#)

See [Table 1-2 on page 8](#) for the list of standard localization resource groups.

Creating Custom Localization Resource Groups

Creating a custom localization resource group is not trivial. It requires technical expertise in these areas:

- ♦ JARs and WARs
- ♦ XML

To ensure that the additional costs associated with creating and testing a custom localization resource group are appropriate, review the `fl10n-word-count.txt` file. This file lists the word count per resource group.

The general steps for creating a custom localization resource group are described in [Table 3-1](#). You do not create a custom localization resource group for User Application driver-based strings. You must always export and translate them.

Table 3-1 Overview Steps for Creating Custom Localization Resource Groups

Step	Task	Description/Links to more information
1	Determine the localization resource group that contains the strings you want to translate then determine any dependencies.	To determine the additional dependent resources required for a particular portlet or page, use the <code>dependencies.html</code> file (located in the <code>fl10n-utility.jar</code> inside <code>fl10nutility.ZIP</code>).
2	Export the strings to translation.	“Creating an Export File” on page 40.
3	Add the new locale and the custom localization resource group to the User Application driver	For more information, see “Adding a Custom Localization Resource Group and New Locale” on page 40.
4	Import the translated strings	For more information, see “Importing Translated Files” on page 41.
5	Modify the User Application WAR to support the custom localization resource group	The modifications you have to make depend on the resource type. For more information, see “Creating Custom Localization Resource Groups” on page 37.

Step	Task	Description/Links to more information
6	Install the translated resource	For more information, see “Packaging, Installing and Testing” on page 41

Required Changes for Custom Localization Resource Groups

When you move a resource from its default localization resource group to a custom localization resource group, you must edit specific files in the User Application WAR. The following sections describe the changes you must make based on the component type. They include:

- ♦ [“Portlets” on page 38](#)
- ♦ [“Portal Artifacts” on page 38](#) (including pages, themes, categories, layouts, options, JSPs, and servlets)

Portlets

Find the `portlet.xml` file in the `WEB-INF/` directory of the User Application WAR. Locate the declaration of the target portlets and change the `<supported-locale>` tag to the new localization resource group name.

Portal Artifacts

Keep the portal artifacts in one group in general. If they must be changed, you must follow the rules outlined in each section below. In addition, when you change any of these artifacts, you must do the following:

- ♦ Have a clean database or the changes do not display correctly.
- ♦ Update the Base (`base-resgrp`), Common (`common-resgrp`), and Administrative localization resource group (`admin-resgrp`) lists of supported locales to include the new locale. For more information, see [“Adding Locales to the User Application Driver” on page 22](#).

Pages (Shared and Container)

Page definitions reside in the `profilegrouppreferences.xml` file of the `IDMFw.jar` file (located in `WEB-INF/lib` directory of the User Application WAR). Each page is an XML definition and in the definition, there is a tag for the resource-group. This can be changed to point to a different localization resource group. Because the pages are copied from the WAR to the database at application server startup, you must use a new, empty database before the effects are shown. Changing page localization resource groups for an existing database is not supported.

Themes

Theme descriptors are located in the Identity Manager User Application WAR under `themes/` resources. Each theme is located in a directory named after the theme. Each directory contains a `theme.xml` file. Change the `<resource-group>` element in each `theme.xml` file.

Categories, Layouts, Options

These reside in the IDMFw.jar (located in WEB-INF/lib directory of the User Application WAR) at /com.novell.afw.portal.artifacts.category|layout|option respectively. Each item has a <resource-group> element in the descriptor. You can change the <resource-group> element to use another localization resource group.

JSPs and Servlets

Servlet-based technologies are assigned a localization resource group through a servlet filter. The BestLocaleServletFilter? has an init param that specifies the localization resource group it is mapped to. You must make a new declaration for the BestLocaleServletFilter? with an init param specifying the new localization resource group name in the WEB-INF/web.xml file (in the IDMFw.war file). Following that, you must make a filter mapping to the appropriate resource. Overlapping BestLocaleServletFilters? causes a warning message and will not function properly at runtime. The resources that were moved to the new localization resource group must not be included in the old localization resource group's filter

Custom Localization Resource Group Example

This section describes how to create a custom resource group that contains only the bookmark accessory portlet.

- ♦ [“Finding Bookmark Portlet Dependencies” on page 39](#)
- ♦ [“Creating an Export File” on page 40](#)
- ♦ [“Creating a Bookmark Localization Resource Group JAR” on page 40](#)
- ♦ [“Adding a Custom Localization Resource Group and New Locale” on page 40](#)
- ♦ [“Importing Translated Files” on page 41](#)
- ♦ [“Modifying the User Application WAR” on page 41](#)
- ♦ [“Packaging, Installing and Testing” on page 41](#)

Finding Bookmark Portlet Dependencies

Before you begin, determine what resources the Bookmark portlet requires. You can do this by opening dependencies.html (located in the fl10n-utility.JAR). Search the file for bookmark. You see the following:

- ♦ com/novell/afw/portal/portlet/bookmark/GlobalBookmarkRsrc
- ♦ com/novell/afw/portal/portlet/bookmark/SampleBookmarkRsrc
- ♦ com/novell/afw/portal/portlet/bookmark/BookmarkRsrc

The properties files corresponding to these files must be translated for the bookmark portlet to display in the new locale.

Creating an Export File

- 1 Export the User Application driver files as described in [“Exporting User Application Driver Strings” on page 13](#) and [Exporting E-Mail Templates \(page 15\)](#).
- 2 Export the User Application WAR strings (as described in [“Exporting Strings from the User Application WAR” on page 16](#)). The Accessory portlets localization group must be selected.

Creating a Bookmark Localization Resource Group JAR

Now that you have a ZIP file containing all the strings to translate, you need to trim the Accessory portlet localization resource group to just the necessary BookMark portlet files before you send the ZIP to translation.

- 1 Open the export ZIP you created in [“Creating an Export File” on page 40](#).
- 2 Unzip the file maintaining the original file structure. You should see a subfolder for each localization resource group.
- 3 Open the accessory-resgrp\com\novell\afw\portal\portlet folder, and remove all of the folders except _include and bookmark.
- 4 Rename the accessory-resgrp folder to bookmark-resgrp.
You will use this localization resource group name when you follow the steps outlined in [“Adding a Custom Localization Resource Group and New Locale” on page 40](#).
- 5 Rezip the all the strings and send the ZIP file to translation.

NOTE: When you created the export file, you also specified a filename and location for the verifier to use on import. Make sure that you use this modified ZIP file as the file to verify against or you will encounter verification error and warning messages.

Adding a Custom Localization Resource Group and New Locale

Before you import translated strings, add both the new locale and the custom localization resource group to the driver.

- 1 Launch Designer for Identity Manager.
- 2 Select the User Application driver to which you want to add the new locale and localization resource group.
- 3 Select **Configure > Locales**.
- 4 Add the Language code and Country code.
- 5 Navigate to the **Localization Resource Group** tab.
- 6 Add the bookmark localization resource group.
- 7 Add the new locale to the list of locales for the bookmark localization group and any others that you exported.
- 8 Deploy the driver.

Importing Translated Files

After you receive the files back from translation, import the User Application driver files as described in [“Importing User Application Driver Strings” on page 22](#) and [“Importing Strings to the User Application WAR” on page 19](#).

Modifying the User Application WAR

Because you added a custom localization resource group, you need to update the User Application WAR file to make it aware of the custom localization resource group. Because the resource you changed was a portlet, you'll modify the `portlet.xml` file. For changes to other components, see [“Required Changes for Custom Localization Resource Groups” on page 38](#). For the bookmark portlet:

- 1 Open the User Application WAR's `portlet.xml` file. (Located in the WAR's WEB-INF folder).
- 2 Search for the each of the bookmark portlets (GlobalBookmark, SampleBookmark and Bookmark).
- 3 For each of the bookmark portlets (GlobalBookmark, Samplebookmark, and Bookmark), change the `<supported-locale>` tag from:

```
<supported-locale>accessory-resgrp</supported-locale>
```

to the custom localization resource group name:

```
<supported-locale>bookmark-resgrp</supported-locale>
```

- 4 Save and close the file.

Packaging, Installing and Testing

After you complete the changes to the user application WAR, you can use the patch installer and install the new WAR. See [“Installing a Language Pack” on page 30](#) for more information.

Updating the Permission Index for a New Language

You must update the permission index when a new language is added to the Identity Applications. The following procedure details steps on updating the permission index for a new language.

- 1 Stop Tomcat server.
- 2 Delete the Permission Index Directory. By default, the directory is `/opt/netiq/idm/apps/tomcat/temp/permindex`.

TIP: To find out the location of the Permission Index Directory, you can navigate to the `/opt/netiq/idm/apps/tomcat/logs/` directory and search for the Permission Index Directory string in the `catalina.out` file.

- 3 Use iManager to add the locale through the Locale configuration option. For more information, see [Adding a Language to the Identity Applications](#) in the [NetIQ Identity Manager - Administrator's Guide to the Identity Applications](#).
- 4 Start Tomcat server.

4 Glossary of Identity Manager User Application Terms

Term	Definition
abstraction layer	See directory abstraction layer
access control rights	Determines whether a user or group can read, write to, or execute (run) an object. The rights are set by the system administrator or the owner of the object. Sometimes referred to as “access control”, “access rights”, and “access permissions” or privileges.
activity	An object that represents a task or step in a workflow.
addressee	The user to whom an activity has been assigned.
anonymous user	A user that has not provided proper credentials to the system. An unauthenticated user. See authenticated user .
application	See User Application .
approval	See “ workflow ” on page 49.
approval flow	See “ workflow ” on page 49.
archive	A J2EE* file that holds the compiled classes, resources, and descriptors required by an application. The user application might be contained in one of these archive types: ♦ EAR: Enterprise archive; contains other archives. See “EAR” on page 45. ♦ WAR: Web archive; contains a Web application made up of JSP pages, HTML files, and so on. See “WAR” on page 49.
artifacts	Objects stored in the User Application driver or Java archive that are needed to support various user application features.
attribute	In eDirectory™, a property belonging to a class. An attribute consists of a type identifier together with one or more values. Also known as a property.
authenticated user	A user that has logged in and been authenticated by the system.
bottleneck	Used in performance. A part of the user application (software or hardware) that prevents the user application from running as fast as possible.
cache	1. (verb) To hold data in a temporary storage area, such as in RAM. The data can be accessed more quickly from RAM than from the hard disk. 2. (noun) A quickly accessible area of RAM or a directory or a disk that stores frequently used information. 3. (noun) On the Web, refers to the area where the browser stores downloaded graphics on the user's computer. When the user needs to view the graphics again, the browser retrieves them from the computer faster than it would from reloading.

Term	Definition
cascading relationship	A relationship that can include multiple levels within an organization.
certificate	In public-key cryptography, a digital document attesting to the binding of a public key to an owner (such as an individual or a Web server). A certificate verifies claims that a given public key does in fact belong to a given individual. It is used to prevent unauthorized users from using false keys to impersonate legitimate users. Certificates contain, at a minimum, a public key and a name. Normally they also contain the expiration date of the key, the name of the certifying authority that issued the certificate, and the serial number of the certificate. They can also contain the digital signature of the certificate issuer. Also called "digital certificate," "digital ID," "digital passport," and "public key certificate."
challenge questions	A set of questions that users answer to prove identity instead.
cleanup service	A service that removes obsolete proxy and delegate assignments.
cluster	A group of application servers that appear to run as a single unit. A cluster increases availability and fault tolerance for applications by providing the ability for servers to automatically load and run applications and services from other servers that fail. A cluster can also improve performance and scalability by allowing a greater number of concurrent users to access the system.
CN	An LDAP attribute called "common name". Typically, made from givenName and surName.
connected system	The back-end system that Identity Manager connects to through a driver (such as the User Application Driver).
container	An object that can hold, or contain, other objects. Container objects are used to logically group related objects.
container page	A special type of user application page. It has a fixed content. User Application Administrators can create container pages with images, logos, text, and navigation control to enforce their proprietary corporate identity throughout an organization. A container page can contain many shared pages
context root	Identifies a Web application in a J2EE server. The installation program and the Designer driver setup procedures prompt users for this value. It is typically IDM or IDMPProv, but can be anything.
create right	The ability to create a new object in the eDirectory tree.
data item	A data element available to the flowdata XML object. You modify a data item through the Data Item Mapping tab in the Provisioning Request Definition editor.
delegate	A user authorized to perform work for another user.
delete right	The ability to delete an object from the eDirectory tree.
digital signature	An electronic signature that someone can use to authenticate the identity of the sender of a message or the signer of a document. It can also be used to ensure that the original content of the message or document is unchanged. A digital signature can be used with any kind of message, whether it is encrypted or not, so the receiver can be sure that the sender is who he says he is and that the message arrived intact. A digital certificate contains the digital signature of the certificate-issuing authority so that anyone can verify that the certificate is real. Also called "electronic signature".

Term	Definition
directory abstraction layer	A set of XML-based files that defines a logical view of an Identity Vault for the user application.
directory object	An object stored in the Identity Vault.
directory relationship	The relationship among objects stored in the Identity Vault.
DN (distinguished name)	An LDAP entry that identifies and describes an authorized user for an LDAP server. A distinguished Name comprises the name of the LDAP entry and the names (in order from top to bottom) of the objects above the LDAP entry in the directory. For example: CN=JRoss, OU=HR, O=Acme
driverset	A container that holds Identity Manager drivers. Only one driver set can be active on a server at a time. As a result, all active drivers must be grouped into the same driver set. It is not necessary to activate all of the drivers in a driver set on every server that is using the driver set.
dynamic group	Functionality that let you specify the members of a group by using a search filter. The members of a dynamic group are computed dynamically by the eDirectory servers whenever the groups are accessed or evaluated. This makes it easier for a user to group objects together because membership can be based on a certain criterion, without manually adding each member to the Group object.
EAR	Enterprise archive A specialized type of JAR file, defined by the J2EE standard, used to deploy J2EE applications to J2EE application servers. An EAR file contains EJB* components, a deployment descriptor, and Web archive (WAR) files for individual Web applications. See also "WAR" on page 49.
eDirectory	A distributed, replicated naming service that maintains information about and provides access to a list of objects that represent network resources such as users, servers, printers, print queues, and applications. The directory is physically stored as a set of database files on a server.
effective rights	The maximum privileges that an authenticated eDirectory object is allowed to have to a specific file object. Effective rights do not include the file-specific privileges as defined by the file attributes. When a request is made to perform an operation on a file, the authorization system compares the privileges needed to perform the operation against the effective rights to determine if the operation should be allowed.
entitlement	An Identity Vault object that represents a business resource in a connected system .
entity	A directory abstraction layer object that represents an Identity Vault object. A type of activity that fulfills a provisioning request by directly manipulating an eDirectory object and its attributes.

Term	Definition
Forgot Password Management WAR	The IDMPwdMgt.war file. You can manage Forgot Password functionality through IDMPwdMgt.war rather than through the User Application. The IDMPwdMgt.war file can be copied to a remote application server deployment directory inside or outside your firewall and then renamed. It can be configured for SSL communication. It has independent theme implementation, and you can give it a new theme. For localization, this WAR uses the browser locale.
flowdata	A script object that allows workflow developers to read and write data to a workflow document that is persisted for each request instance. Flowdata can be referenced through data item mapping expressions.
form	A user interface component that allows users to input and submit information for processing. Used in provisioning requests and approvals.
grant	To confer access and/or management privileges to a trustee. The trustee represents the user receiving the privileges, and the target represents the resource that can be accessed or managed.
group	A leaf object in the eDirectory tree that represents a list of User objects. It is used to provide collective, rather than individual, network administration. The Group object differs from a container object in that instead of containing the User objects, it lists them as members. See also “dynamic group” on page 45.
guest user	See “anonymous user” .
host	A computer that provides services, applications, and information to a series of other devices.
Identity Vault	The name of the database containing objects, relationships, workflow objects, permissions, policies, and other information used by Identity Manager.
iManager	Novell iManager A browser-based management utility.
import	To retrieve information from one place into another.
inherited rights	In eDirectory, explicitly granted privileges to one object that apply to all subordinate objects, unless other rights are explicitly assigned at a lower level or the rights are blocked by an inherited rights filter (IRF).
LDAP Administrator	Lightweight Directory Access Protocol. The user who has been assigned the appropriate rights (or privileges) to manage Identity Vault. It is also a proxy user for the user application to carry out tasks on the LDAP server that an ordinary logged-in user might not have permission to execute, such as creating a new user, group, or container
LDIF	LDAP Directory Interchange Format The LDIF format is used to convey directory information (content records) or a description of a set of changes made to directory entries (change records). An LDIF file consists of a series of records separated by line separators.
locale	A unit that describes the user's language, country, and variant.
logging	The process of recording changes made to data by the system or users.

Term	Definition
metadirectory engine	An engine that synchronizes the data between the Identity Vault and other systems.
metrics system	Measurements of the user application runtime performance.
migrate	To move data files and directories from one system to another. Migration typically occurs when a system is upgraded or a different system is installed.
Modify right	A file system right that allows the trustee to rename the file or directory or to change its attributes. In the case of a directory, the trustee can also rename or modify the attributes of any subordinate file or directories. This right doesn't allow the trustee to see or modify the contents of files.
NMAS™	Novell Modular Authentication Service A flexible and expandable login framework that provides developers the ability to integrate multiple authentication services using eDirectory systems.
Organizational Unit object	OU object An optional container object in a directory tree that represents a unit within an organization, such as a department or business unit. It contains the objects that represent the network users and resources in the actual unit. It must reside either immediately under an Organization object or under another Organizational Unit.
port	The point of contact between two hardware devices or two software elements. A hardware port can be the physical connection point between a printer cable and the computer. A software port, represented by a memory address, can be the logical contact point between a LAN driver and the protocol bound to it, or the point of access to a service on a TCP/IP host computer.
portal	A Web site whose purpose is to be a starting point for user access to a broad array of resources and services.
privileges	See "rights" .
provisioned resource	The result of a successful workflow provisioning request.
provisioning application administrator	The user who has been assigned the appropriate privileges to manage the workflow-based provisioning features of the user application.
provisioning request definition	An object that maps a provisioning request to a provisioning resource.
provisioning requests	A user or system action intended to grant or revoke a resource.
provisioning resource	A resource that can be requested, granted, or revoked through a provisioning request.
provisioning team	A definition for a group of users. The team definition determines who can manage provisioning requests and approval tasks associated with this team.
provisioning team requests	A user or system action intended to grant or revoke a resource for a team.
proxy	A user authorized to perform any and all work (and also define provisioning settings) for one or more users, groups, or containers. Unlike delegate assignments, proxy assignments are independent of resource requests, and therefore apply to all work and settings actions

Term	Definition
Read right	An object right that allows the trustee to read files, view directory contents, and execute any program files located in the directory. Rights assigned at the file level can override the Read directory right. In the case of a property, the trustee can read and compare the values of the property. The Read property right implies the Compare property right.
recipient (workflow)	The beneficiary of a workflow. The object for whom the workflow is being performed.
rights	An assignment that allows access to or manipulation of a resource in some way; for example, browse, read/write, or create. Also called "privileges".
schema	The rules that define how the Identity Vault tree is constructed. These rules define specific types of information that dictate the way information is stored.
shared page	Shared pages provide a coherent set of content that is used for a specific purpose (such as updating a user's profile). They are called shared pages because they offer services used by multiple people
signature declarations	Specifies a digital signature confirmation string that confirms the user's signature
smart card	An electronic card containing a built-in microprocessor and memory used for authentication identification. To use a smart card with Novell Modular Authentication Services (NMAS), you need a smart card reader and an installed NMAS method built for that reader.
SOAP	Simple Object Access Protocol. A standard for exchanging XML-based messages over the Internet.
soap endpoint	SOAP methods are invoked against endpoints. Endpoints are the access points to an API.
special character	A character other than a digit, a letter, or \$, #, @, ., or _. For example, *, +, and % are special characters.
system administrator	A person responsible for maintaining a system.
task (workflow)	A step in a workflow. In the Identity Manager User Application, the activities that handle the approval process are referred to as tasks.
team definition	See "provisioning team" on page 47 .
team manager	Users or groups who are able to administer requests and tasks for a provisioning team.
team members	Those users who are allowed to participate on the team. Team members can be users, groups, or containers within the directory. Alternatively, they can be derived through directory relationships. For example, the list of members could be derived by the manager-employee relationship within the organization. In this case, the team members would be all users that report to the team manager.
template	For provisioning request definitions, an object that provides a base set of properties for a workflow. For e-mail, an object that provides a base set of text and tokens for an e-mail.

Term	Definition
theme	A set of visual characteristics that apply to an application. For the User Application, themes provide a simple way to ensure a consistent appearance throughout
token	A placeholder whose value is replaced at runtime.
trustee	Any object that has rights to access resources such as an object in the Identity Vault. Rights are granted to objects (making them trustees) by trustee assignments.
user	1. A person allowed to work on a specific computer or network. 2. A leaf object in the Identity Vault tree that represents a person. The object contains information about the user it represents, such as login name, telephone number, address, and group membership.
User Application	Short for Identity Manager User Application. It includes the user interface plus the underlying technologies supporting the user interface, such as the portal application, directory abstraction layer, and so on. The User Application is deployed to an application server.
user application administrator	A user who has the rights to perform administrative tasks for the User Application.
user application driver	The Identity Manager driver used to capture events (changes) in eDirectory, and then trigger a workflow process in the workflow system.
user interface	The end-user-facing Web-based user interface of the Identity Manager User Application
WAR	Web Archive A compressed file format, defined by the J2EE standard, for storing all the resources required to install and run a Web application in a single file. Having a single large file instead of many small files makes transferring the Web application from server to server easier.
white pages	A format for data returned by the Search list portlet.
wildcard (in a search)	Patterns used to generalize a search. The * is the acceptable wildcard. Users can type the * to represent zero or more of any character.
wizard	Software that leads the user and a sequence of steps through dialog boxes that results in a User Application object.
workflow	A user application component that contains activities and options that comprise the approval process. When a provisioning request requires permission from one or more individuals in an organization, the request starts one or more workflows. The workflows coordinate the approvals needed to fulfill the request.
workflow engine	A part of the Workflow subsystem that provides routing and execution services for a workflow process
workflow-based provisioning	A feature of the user application that enables users to automate the approval and revocation of user access to an organization's secure resources.
WSDL	A XML document that defines SOAP methods and parameters or data types of an API.

