

# JavaPM User Guide



Copyright © 2006 - 2012 Maxprograms



# Contents

<b>Introduction.....</b>	<b>5</b>
Supported Platforms.....	5
Requirements.....	5
Basic Concepts.....	5
Java Properties.....	5
XML Localization Interchange File Format (XLIFF).....	5
<b>User Interface.....</b>	<b>7</b>
JavaPM GUI.....	7
Menus.....	8
File Menu.....	8
Options Menu.....	8
Tasks Menu.....	8
Help Menu.....	8
<b>Localization Workflow.....</b>	<b>9</b>
Standard Workflow.....	9
Basic tasks.....	9
Create a Project.....	9
Generate XLIFF Files.....	9
Import XLIFF File.....	10
Project Maintenance.....	10
Maintenance Tasks.....	10
Export Changed or New Strings.....	10
Import Translated Strings.....	11
Mark Files as Translated.....	11
Translations Reuse.....	11
<b>Glossary.....</b>	<b>13</b>



# Introduction

---

JavaPM is a cross-platform utility that assists software developers in the localization of Java projects.

JavaPM converts Java .properties files to XLIFF 1.2 format, allowing their localization using most modern translation environments. Once the XLIFF files have been translated, JavaPM inserts the translations into the Java project, creating new .properties files or updating existing ones as necessary.

JavaPM tracks all .properties files in a Java project. It can create full translation kits or incremental ones when necessary.

All programs published by Maxprograms are localized using JavaPM and Swordfish.

## Supported Platforms

---

- Microsoft Windows XP/Vista/7
- Mac OS X 10.5/10.6/10.7 (Leopard - Snow Leopard - Lion)
- Linux with GTK2 (i386 and x86\_64)

## Requirements

---

- Java 1.6/1.7 from Sun Microsystems or Oracle on Linux/Windows. Java 1.6 from Apple on Mac OS X.
- WebKitGTK+ 1.2.x is required to run on Linux systems.

GNOME classic desktop manager is recommended for Linux. Some features may not work in other desktop managers.

## Basic Concepts

---

### Java Properties

Java and related technologies use .properties files to store strings for localization and the configurable parameters of an application. These files are also known as Property Resource Bundles.

Each entry in a .properties file (application parameter or translatable text) is stored as a pair of strings, one storing the name of the parameter (called the key), and the other storing the value.

The character set of a .properties file is ISO-8859-1, also known as Latin-1. All non-Latin-1 characters must be entered by using Unicode escape characters, e. g. \uHHHH where HHHH is a hexadecimal index of the character in the Unicode character set.

**Java Properties Viewer** is a tool specially created for viewing translated Java .properties files comprising languages not supported by the ISO 8859-1 character encoding. This program is included in Swordfish installers.

### XML Localization Interchange File Format (XLIFF)

XML Localization Interchange File Format (XLIFF) is an XML-based open format used to exchange localisation data between participants in a translation project. This special format enables translators to concentrate on the text to be translated, without worrying about text layout.

The XLIFF vocabulary has a rich set of elements and attributes that permit XLIFF-supporting applications to:

- Store source and translated text strings.
- Store alternative or suggested translations extracted from a Translation Memory system or generated by a Machine Translation engine.
- Perform revision control.
- Keep track of the different stages of the translation process.
- Carry word count calculations.

The XLIFF standard was first published by OASIS in 2002. It is supported by a large group of localisation service providers and localisation tools providers.

XLIFF specification is available at <http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html>.

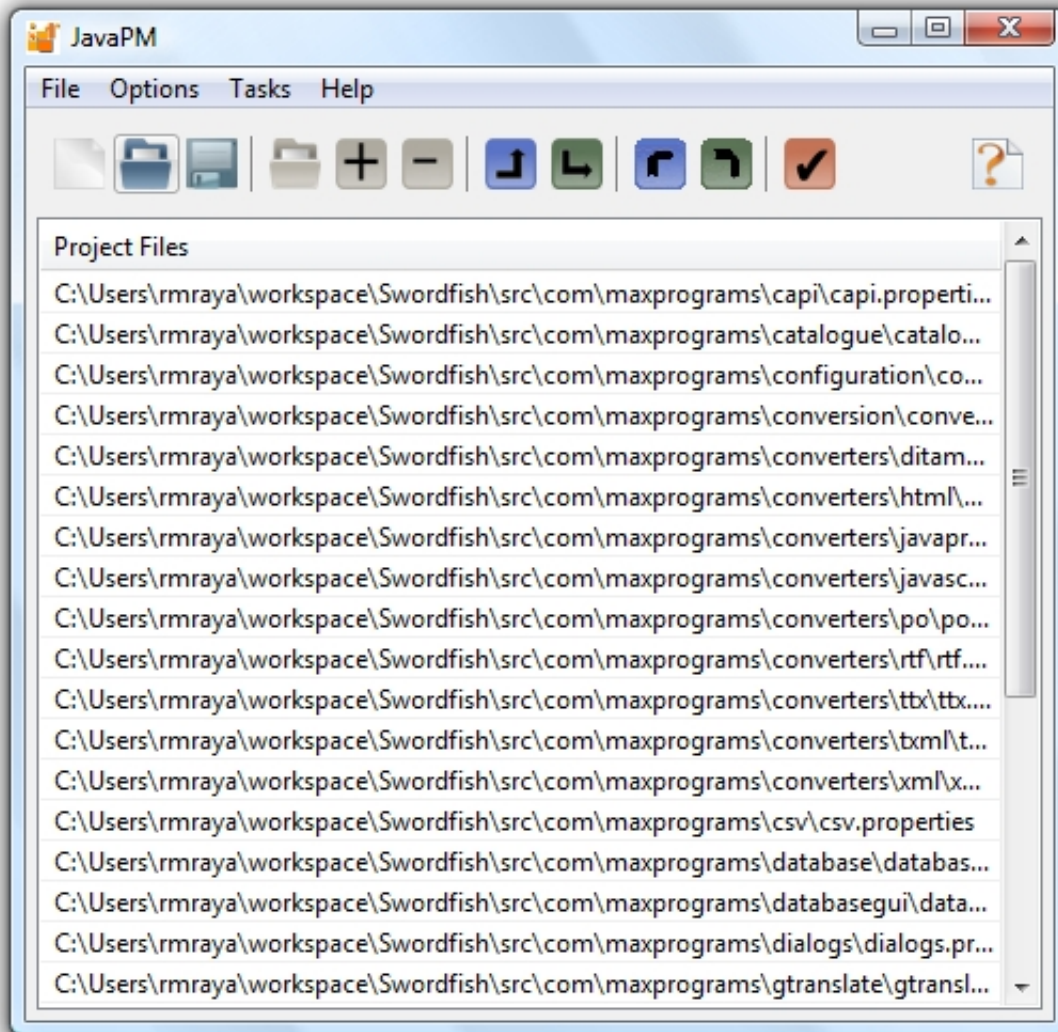
# User Interface

---

## JavaPM GUI

---




The following picture portrays JavaPM:



## Menus

---









### File Menu

	New JavaPM Project	Create a new JavaPM project.
	Open JavaPM Project	Open an existing JavaPM project.
	Save JavaPM Project	Save all changes introduced to current open JavaPM project.
	Exit/Quit	Close the program.


### Options Menu

	Language	Open a sub-menu for selecting the user interface language.
--	----------	--

### Tasks Menu

	Add Source Folder to Project	Add all .properties files from a selected Java source folder to current open project.
	Add File to Project	Add selected .properties file to current open project.
	Remove File(s) from Project	Remove selected .properties files from current open project.
	Generate XLIFF from Project	Generate an XLIFF file from all strings stored in current open project for translating into the selected target language.
	Import Translations from XLIFF	Import all approved translations from a selected XLIFF file and update the corresponding .properties files.
	Export Changed Properties	Generate a .properties file containing all translatable strings modified or added to the source code after the project was created.
	Import Translated Properties	Import all strings from a translated .properties file and update the corresponding .properties files of the source code
	Mark All Files as Translated	Review the .properties files listed in the project and incorporate all new or changed translatable strings from source code and their corresponding translations to the project.

### Help Menu

	JavaPM Help	Display JavaPM User Guide in the default PDF viewer.
	About...	Display a dialog with copyright and program version information.

# Localization Workflow

---

## Standard Workflow

---

Localization of a Java project normally requires the following steps:

1. Create a JavaPM project.
2. Generate an XLIFF file for each target language.
3. Send XLIFF files to your Localization Services Provider (LSP) for translation.
4. Import translated XLIFF files into the JavaPM project.
5. Repeat these steps until your LSP says that translations are final:
  - Build a test version of your software and send it to the LSP for review.
  - Import updated XLIFF files into the JavaPM project.

## Basic tasks

---

### Create a Project

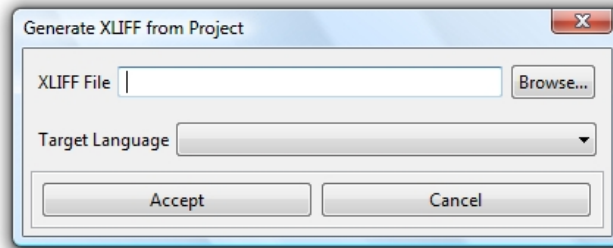
Steps for creating a JavaPM project and populating it with an initial set of files.

1. In **File** menu, select **New Java PM Project**.
2. Add .properties files to the project.
  - To add all .properties files from a given source folder at once: In **Tasks** menu, select **Add Source Folder to Project**.
  - To add individual .properties files: In **Tasks** menu select **Add File to Project**.
  - To delete .properties files from the project:
    1. Select one or more files from the file list.
    2. In **Tasks** menu, select **Remove File(s) from Project**.
3. In **File** menu, select **Save JavaPM Project** to specify a file name for the JavaPM project file and save it.

### Generate XLIFF Files

Steps for generating XLIFF files from a JavaPM project.

1. In **File** menu, select **Open JavaPM Project**.
2. Locate and open the JavaPM project to be localized.
3. In **Tasks** menu, select **Generate XLIFF from Project**.  
The following dialog appears:



4. In the **XLIFF File** text box enter the name of the XLIFF file to generate or use the **Browse...** button to select a location and file name.
5. Select the desired target language in the **Target Language** drop-down list.
6. Click the **Accept** button to generate the XLIFF file.

The steps described above explain how to generate an XLIFF file for translating a JavaPM project into a given target language. Repeat the procedure for all languages that you wish to support.

## Import XLIFF File

Steps for importing translated XLIFF files into a JavaPM project.

1. In **File** menu, select **Open JavaPM Project**.
2. Locate and open the JavaPM project to be updated.
3. In **Tasks** menu, select **Import Translations from XLIFF**.
4. Locate and open the translated XLIFF file to import.

JavaPM incorporates all approved translations from the selected XLIFF file into the corresponding .properties files of the project.

## Project Maintenance

---

In the real world, new translatable strings are added to the source code of the Java project and some strings are modified while localization is in progress. JavaPM assists you in handling unexpected changes in source. When new strings are added or changed, proceed as follows:

1. Export new or changed strings to a separate .properties file.
2. Prepare the exported file and send it out for translation. You can use Swordfish to generate XLIFF files from the changed .properties and recover existing similar translations from a Translation Memory database.
3. Import translated .properties files into the project.
4. When all changed or new strings have been translated and all generated XLIFF files have been imported, mark all files as translated.

## Maintenance Tasks

---

### Export Changed or New Strings

Steps for generating a translatable file from strings modified or added to the source code after the project was created.

1. In **File** menu, select **Open JavaPM Project**.

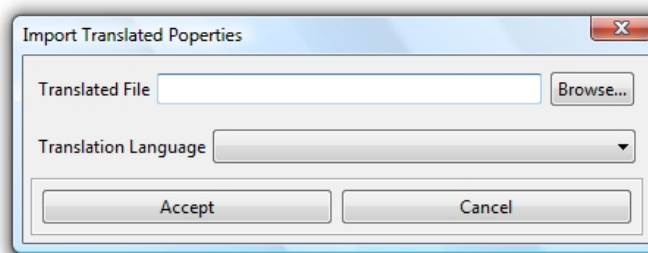
2. Locate and open the JavaPM project.
3. In **Tasks** menu, select **Export Changed Properties**.
4. Especificy a file name for the exported .properties file and save it.

A .properties file containing all translatable strings modified or added to the source code after the project was created is generated and saved in the selected location.

## Import Translated Strings

Steps for updating the translations of strings modified or added to the source code after the project was created.

1. In **File** menu, select **Open JavaPM Project**.
2. Locate and open the JavaPM project to be updated.
3. In **Tasks** menu, select **Import Translated Properties**.  
The following dialog appears:



4. In the **Translated File** text box enter the name of the translated .properties file to import or use the **Browse...** button to select a file.
5. Select the language of the .properties file in the **Translation Language** drop-down list.
6. Click the **Accept** button to import the file.

All translations imported from the selected .properties file are added to the corresponding .properties files of the source code.

## Mark Files as Translated

Steps for updating a JavaPM project, incorporating all strings modified or added to the source code after the project was created. Use this feature only after all XLIFF files generated from the project have been translated and imported.

1. In **File** menu, select **Open JavaPM Project**.
2. Locate and open the JavaPM project to be updated.
3. In **Tasks** menu, select **Mark All Files as Translated**.
4. In **File** menu, select **Save JavaPM Project** to save all changes.

JavaPM reviews the .properties files listed in the project and incorporates all new or changed translatable strings and their corresponding translations to the project.

## Translations Reuse

---

Software localization can be expensive in terms of money and/or time. Language technology helps you to cut costs by reusing translations.

After receiving a localized XLIFF file, export all approved translations to TMX format and store the generated TMX file in a Translation Memory database.

After generating XLIFF files from a JavaPM project, use the Translation Memory database that contains previous translations to populate the XLIFF files with matches that your Language Service Provider can later reuse.

New or modified strings extracted from the JavaPM project in .properties format can be processed with a CAT tool like Swordfish, generating XLIFF files with matches recovered from your Translation Memory databases.

Some large software companies, like Microsoft and Apple, publish their glossaries and translation memories. Get a copy of those glossaries and translation memories and use them for enhancing your XLIFF files with matches from well known software applications.

# Glossary

## **Character Set**

A character set (sometimes referred to as code page) is a collection of characters that are associated with a sequence of natural numbers in order to facilitate the storage of text in computers and the transmission of text through telecommunication networks.

## **Computer Aided Translation (CAT)**

Computer technology application designed to assist human translators in the translation process.

## **Localization Service Provider (LSP)**

A company or individual specialized in providing translation and localization services.

## **OASIS**

OASIS (*Organization for the Advancement of Structured Information Standards*) is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society.

## **Swordfish**

An advanced XLIFF-based CAT (Computer Aided Translation) tool published by [Maxprograms](#).

## **Source Language**

The language of a document that is to be translated.

## **Target Language**

The language into which a document is being translated.

## **TMX**

Translation Memory eXchange (TMX) is an open standard originally published by LISA (Localization Industry Standards Association). The purpose of TMX is to allow easier exchange of translation memory data between tools and/or translation vendors with little or no loss of critical data during the process.

## **Translation Memory**

Translation Memory (TM) is a language technology that enables the translation of segments (paragraphs, sentences or phrases) of documents by searching for similar segments in a database and suggesting matches that are found in the databases as possible translations.

## **XLIFF**

XLIFF (XML Localization Interchange File Format) is an open standard developed by [OASIS](#) (Organization for the Advancement of Structured Information Standards). The purpose of this vocabulary is to store localizable data and carry it from one step of the localization process to the other, while allowing interoperability between tools.

