Java Conversion Technical Report

19 September 2008

Prepared by: Miss VANN Sophea

Cambodia Country Component

PAN Localization project

PAN Localization Cambodia (PLC) of IDRC

Table of Contents

1. Introduction

This document will discuss about the technical details and methodology of Java Conversion of Dot Net applications. There are two possible methodologies to convert the platform, Automatic Conversion and Manual Conversion.

2. Usability of Java Conversion

2.1. Standalone Application

Standalone Application is application that can be run without installation and no need to interact any other applications for running.

After development of various Khmer applications like Khmer Encoding Conversion, Khmer Line Breaking, Khmer Spell Checker, Khmer Unicode Collation, and Khmer Unicode Find/Replace project, the requirement was to make this software available on open source platform. The accomplishment of this necessity was done by developing Java version of all applications

2.2. Open Office Plug-in

Open Office.org is a huge amount of code and provides open source code for development to update or develop application to be add in (Plug-in) on Open Office software like OO Writer. PLC also aimed to develop a plug-in for 5 localized applications. This project was possible after conversion of Dot Net applications to Java platform.

2.3. PLC Applications

There are five PLC Dot Net projects to convert into Java platform:

- **Khmer Encoding Conversion project:** a tool for converting from non-Unicode font documents to Unicode text.
- **Khmer Line breaking project:** a tool for breaking up Khmer Unicode text, into lines in such a way that words are not divided across two lines (or are hyphenated).
- **Khmer Spell Checker project:** a tool for detect and correct the spelling errors of Khmer Unicode text.
- **Khmer Unicode Collation project:** a tool for ordering Khmer Unicode text by phonetically of CHUON NATH dictionary.
- **Khmer Unicode Find/Replace project:** a tool to find and replace Khmer Unicode text.

3. Conversion Approach

3.1. Research and Exploration

After thorough research and reading the reviews of users, we found a tool for conversion from Dot Net platform to Java automatically, called Net2Java, is a plug-in of NetBeans IDE 5.5.

3.2. Methodology

3.2.1. Automatic Java Conversion

3.2.1.1. Application(s) for Dot Net to Java

There is a suitable module for converting from Dot Net to Java application that was found out, Net2Java technology. It facilitates Automatic conversion.

3.2.1.2. Net2Java Tool

NET2Java a new technology that helps to take an application written in Visual Basic or C# to the Dot Net platform, and convert it into a program written in Java source code. NET2Java is a plug-in of NetBeans 5.5 IDE [1].

3.2.1.3. Net2Java Configuration

Following are the steps to configure Net2Java application.
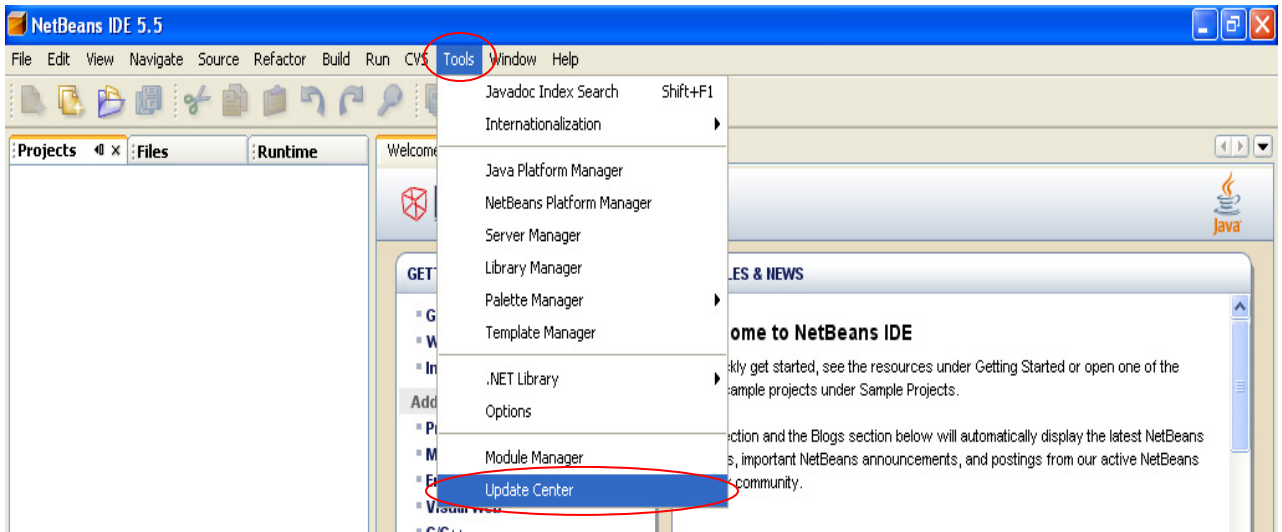
3.2.1.3.1. install the NET2Java Library

- make a directory called dnj in your user home directory
- Ex: on windows create C:\Documents and Settings\Administrator\dnj\
- Unzip the NET2Java Library Archive to the directory you just made
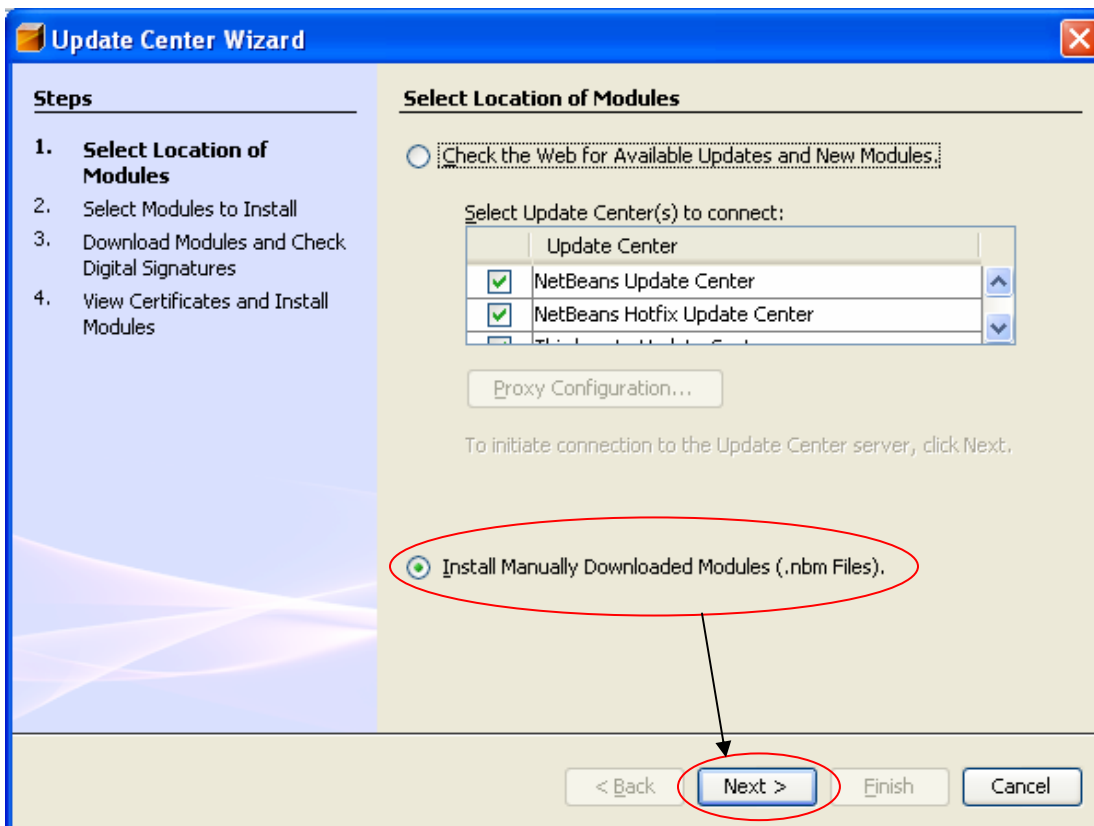
3.2.1.3.2. install the NET2Java NetBeans Plug-in

- Install Netbean 5.5 IDE
- in your Netbean 5.5, from the tools menu, open the update Center
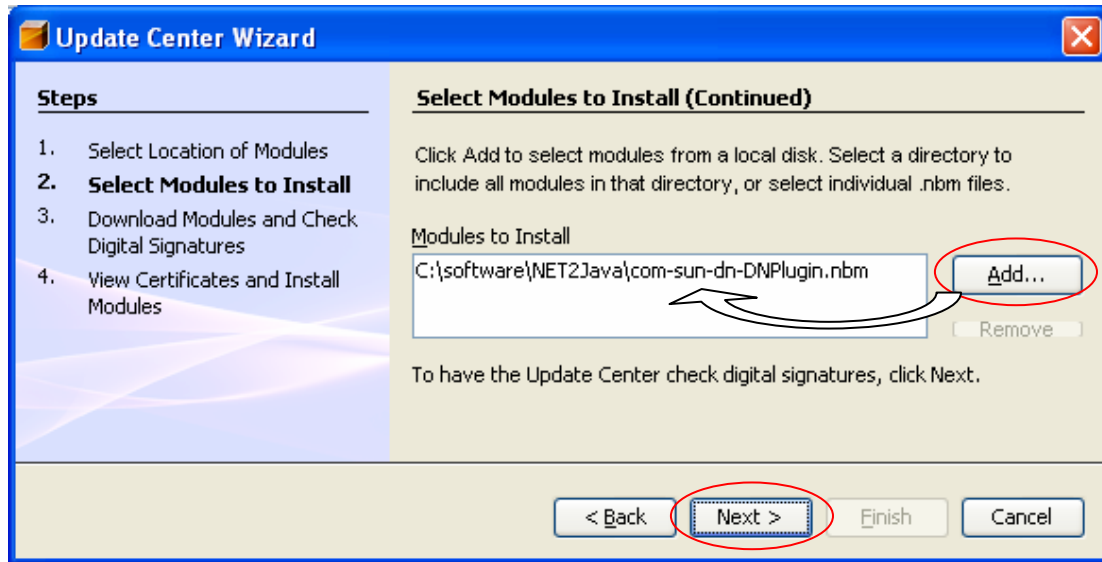
PAN localization project

- choose the option to install Manually Downloaded Files (.nbm modules),
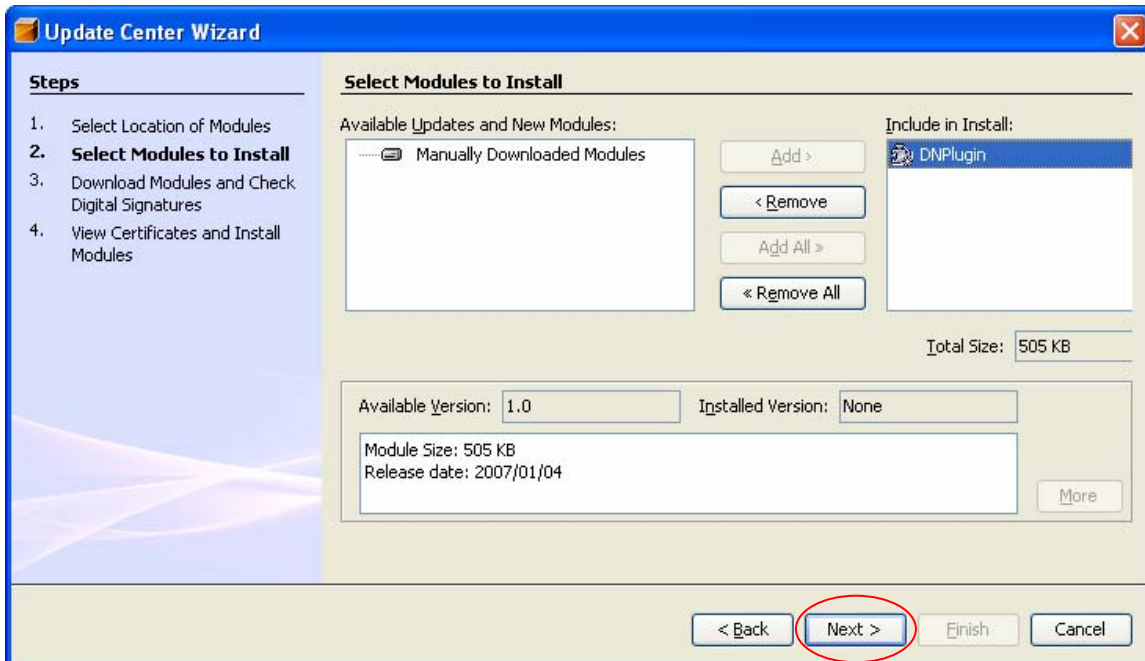hit next

- Click on "Add" Button then browse the file com-sun-dn-DNPlugin.nbm, then click on "Next" button.

- Click on "Next" button.
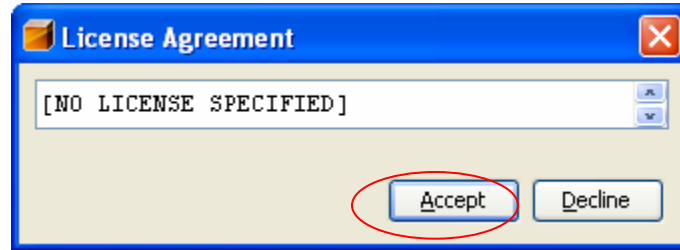
- Click "Accept" Button.
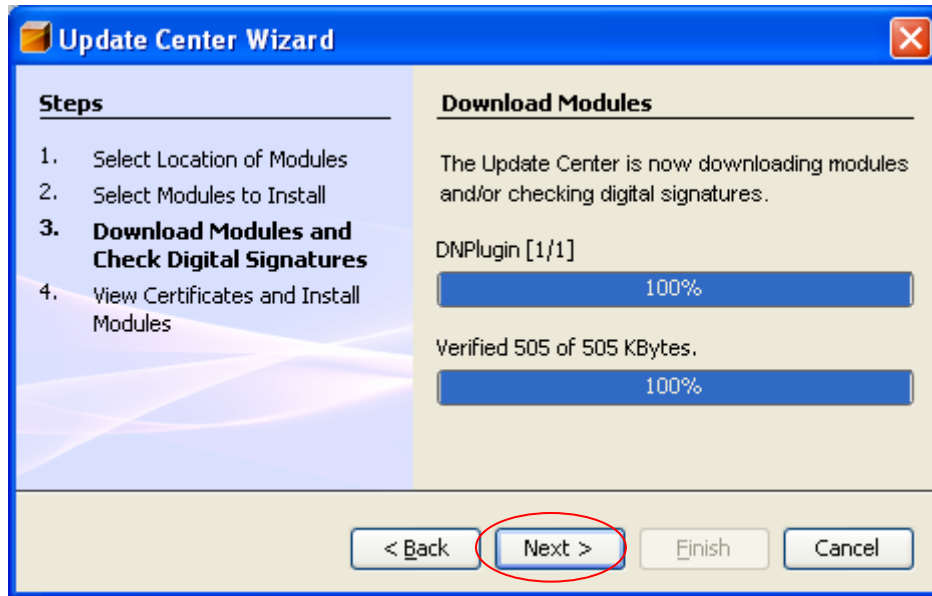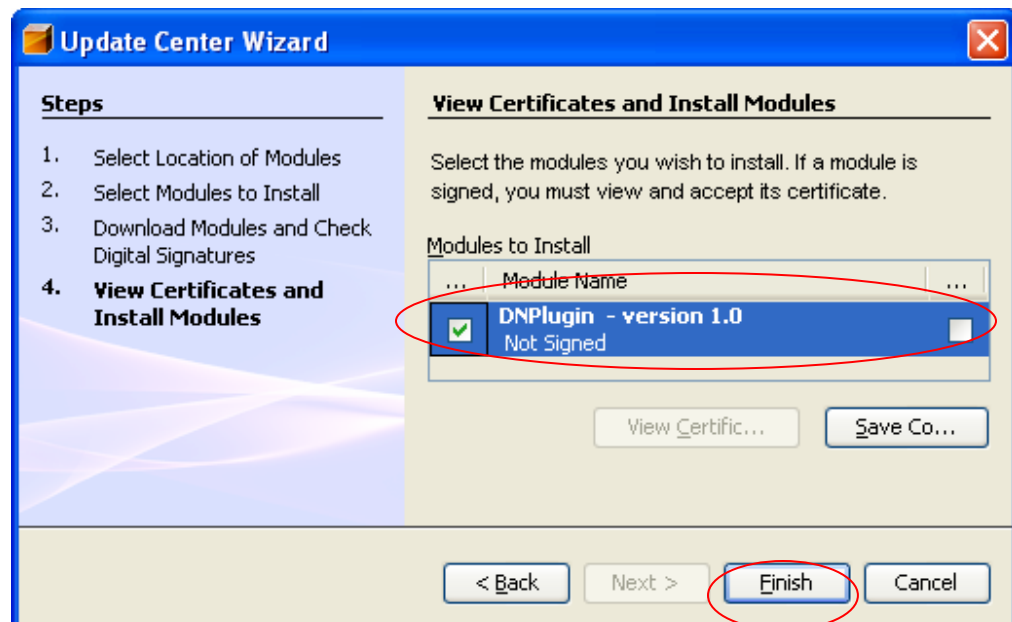

- Click "Next" Button

- Check on "DNPlugin-Version 1.0" in the Modules to install list then click on "Finish" Button.

3.2.1.4. How to Use Net2Java Tool

- Unzip these NET sample somewhere handy
- From NetBeans, create a new project (select the java Application template for example)
- Add the path to the .Net2Java library entry files and classes to your new project
- Ex: in the project Properties dialog, select the libraries category and add **C:\Documents and Settings\Administrator\dnj** by hitting the 'Add Jar/Folder' button
- Import your first .NET project, by choosing File -> ImportVB top bring up the import Wizard
- Selecting the <samples>HelloWordFromVB project directory, select 'Console app' and hit Finish
- you should see HelloWorldFromVB… in java

3.2.2. Manual Conversion

3.2.2.1. Introduction

The Manual Conversion from Dot Net Project to Java requires a lot of human effort and knowledge. We have to know the basics of both programming languages

3.2.2.2. Dot Net Project Debugging

The starting point for doing Java Conversion manually is by debugging process. The manual debugging of Dot not projects helped us in understanding the flow of programs. And hence we were able to extract the core functionality. Now the next step was to start building java codes.

3.2.2.3. Eclipse IDE

Eclipse IDE is an application that is used to build open source software comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle [ref]. The Eclipse IDE supports many programming languages such as Java Developer, Java EE, C/C++, PHP, Modeling Framework technology (EMFT), Modeling Tool (MDT), Modeling Framework (EMF), and Visual Editor (VE)...etc [2].

3.2.3. Comparison of Automatic and Manual Conversion

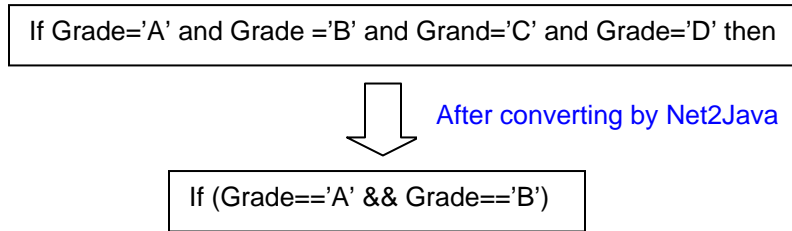#### 3.2.3.1. Advantages and Disadvantages of Automatic Conversion

Advantages of Automatic Conversion

- It takes very short time for converting whole project, so a little human involvement is required.
- The process produces readable source code, preserving class names, method names, code comments and making note <todo> on code that it can not converted.

Disadvantages of Automatic Conversion

- It uses the NetBeans Library to replace Dot Net code so the conversion project can work with NetBeans IDE only. We have to replace that NetBeans Library code to simple Java Built in code manually.
- Loosing minute details of code is another disadvantage. For example the expression in the if-else Statement having more than two if statements, it converts only the first two expressions and the other will lose.

  Ex:

  > If Grade='A' and Grade ='B' and Grand='C' and Grade='D' then

  After converting by Net2Java

  > If (Grade=='A' && Grade=='B')

- It reverse the expression and variable

  Ex:

  > If Grade='A' and Grade ='B' then

  After converting by Net2Java

  > If ('A'==Grade && 'B'== Grade)
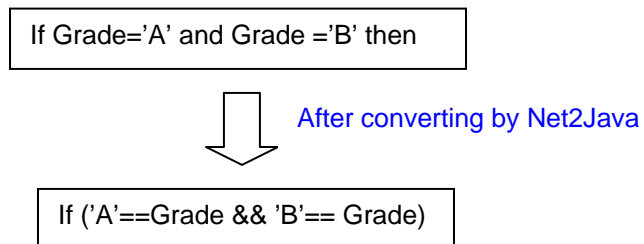
- It converts keyword "AndAlso" to keyword "&&" in Java.

  Ex:

  > If Grade='A' AndAlso Grade ='B' then

  After converting by Net2Java

  > If ('A'==Grade && 'B'== Grade)
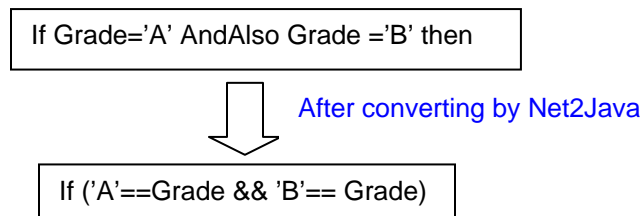
3.2.3.2. Advantages and Disadvantages of Manual Conversion

Advantages of Manual Conversion

- The accuracy of manual conversion is definitely higher than by Net2Java.

- Understanding of coding and Data structure of the project is developed in more precise way.

Disadvantages of Manual Conversion

- It takes too long for converting whole project.
- There should be a sound technical background of both platforms.

Keeping in view the pros and cons of both techniques, the ideal approach is to use the automatic conversion of the project and after that fix the errors. In our case the out put of automatic conversion was not very precise so we selected manual conversion methodology.

3.3. Problems faced in manual conversion (Technical)

The structure of Java language application and Dot Net application are similar because they are using the sane Object Oriented Programming approach.

The problem that we faced while doing the Java Conversion projects is only syntax problem. Dot net interface is more user-friendly and it provides much possible syntax that some programming language such as java language programming can't.

- **Converting Boolean value**

Dot net Input Value	Dot net output Value
0	0
1	1
False	0
True	1
0, false= 0 ; 1, true= 1	

Java Input value	Java Output Value
0	0
1	1
False	false

True	true
0 or 1 are not Boolean value in java	

- **Split Unicode String**

   Ex: String str=" +true+ + "

Dot Net Input	Dot Net Output
Str.split("+")	" ","true"," "," "

Java Input	Java Output
Str.split("+")	Syntax error, cant split the Unicode string

Our Solution

```java
w=new String[1];
int index=0;
w[0]="";
for(int l=0;l<segJ.length();l++){
      if(segJ.substring(l,l+1).equals("+")){
            String[] newW=new String[w.length+1];
            System.arraycopy(w, 0, newW, 0, w.length);
            w=newW;
            w[w.length-1]="";
            index++;
      }else{
            w[index]=w[index]+segJ.substring(l,l+1);
      }
}
```

- **Method AscW()**
   o It is used to convert Unicode String (character or string) to acscii code for Dot Net method. According to out research and study, the data structure of this method is cutting the first character of the string then converts it to ascii code.
   o In Java language, if we are using the JRE 1.6 (Java Runtime Environment), we'll convert it to acsii code only that text is character by using Integer.parseInt(text). And if the Input text is String => syntax error (cant convert from Unicode string to acsii code)

Our Solution

o Writing code for cutting a character from the string

o Converting from character to ascii code by using Integer.parseInt(cutting character).

- **The Merging String with Null Initialization String variable**

Dot net Input	Dot net Output
String str=null Str=str & "hello"	Str="hello"

Dot net Input	Dot net Output
String str=null; Str=str + "hello";	Str="nullhello";

Our Suggestion

The Initialization of String variable must be blank.

- **Initialize Array**

o In Dot Net, Array index can start from 0 or 1 (in the five of applications above, some array index is started from 0 and for other it is started from 1).

o In Java, Array index must start from 0.

- **Array Size**

o In Dot Net, If it is an array of 5, this array size is 6, which index is from 0 - 5.

o In Java, If it is an array of 5, this array size is 5, which index is from 0 - 4.

- **Keyword "AndAlso"**

o In Dot Net, keyword "AndAlso" is use for checking the first condition, if it returns true, it will continue to check another condition in the If statement and return Boolean value.

o In Java, it has only keyword "&&" that is used for checking all conditions of If statement and return the Boolean value

- **Keyword "ByRef"**
  - o In Dot Net, keyword "**ByRef**" is indicate that value type of arguments should be passed by value
  - o In Java, there is no Keyword "**ByRef**", so we have to create a reference object instead.

4. Progress

4.1. Status of Conversion

According to the PLC conversion project, we have done four of Dot Net projects conversion and 5th is almost finished.

The status of Java conversion projects:

- Khmer Unicode Encoding conversion: 100 %
- Khmer Line breaking conversion: 100%
- Khmer Spell Checking conversion: 95%
- Khmer Unicode Collation conversion: 100%
- Khmer Unicode Find/ Replace conversion: 100%

4.2. Platform

There are different jar files available for Windows and Linux platform. The programs have been tested on both platforms and working fine.

5. Conclusion

In short, the PLC has developed applications for two platforms: for Microsoft Visual Basic.Net and Java. It provides more choice for end use to use standalone applications with both platforms and plug-in applications in Microsoft Office and Open Office.

6. Reference

[1].https://net2java.dev.java.net/servlets/ProjectDocumentList?folderID=6667&expandFolder=6667&folderID=0

[2] http://www.eclipse.org/