It is Time to Add Kurdish Culture to VS .NET Globalization

Azad Ali
Indiana University of Pennsylvania
Indiana, PA, USA

azad.ali@iup.edu

Frederick Kohun
Robert Morris University,
Pittsburg, PA, USA

kohun@rmu.edu

Abstract

Starting with the introduction of Visual Studio .NET (VS .NET) application developers can write programs that may be used for different languages listed in VS .NET globalization. However, the list of languages is incomplete and is missing many. One of these missing languages from VS .NET globalization is the Kurdish: a language written and spoken by millions of Kurds around the world. Previous research conducted by the authors made a case for adding the Kurdish language to the globalization component of the .NET Framework. We were hopeful that the Kurdish will be added to VS .NET in the latest version (VS .NET 2005) that was yet to be released in full version at the time of writing that paper. However, VS .NET 2005 has been released, and to our disappointment, Microsoft excluded the Kurdish language again from the languages it supports in VS .NET. This paper is to emphasize adding Kurdish to VS .NET globalization component. It shows that there are no technical, lingual or cultural hurdles for adding Kurdish to VS .NET globalization and thus it is time to add the Kurdish Culture to the newly released Microsoft .Net Framework.

Keywords: VB Globalizations, VB New languages, VB Kurdish, VS Kurdish

Introduction

This paper builds on a previously published paper by the same authors (See Ali & Sulaiman, 2006). During our previous paper we made a case for adding the Kurdish language to Visual Studio .NET (VS .NET) globalization component. Since presenting our last paper, we contacted Microsoft and conveyed the case we made regarding adding Kurdish to VS .NET globalization. An official from Microsoft promised us that the Kurdish Culture will be added as a locale but they did not specify a date. We were hopeful that Kurdish culture will be included in the new VS .NET 2005 version once it is released. To our disappointment, the full version of VS .NET 2005 has been released; it added many new languages, while the Kurdish Culture is still not included.

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact Publisher@InformingScience.org to request redistribution permission.

The introduction of Visual Studio.NET has simplified coding applications with target audience of languages other than English. Application developers can write programs that generate outputs in many cultures based on the target audience. A programmer can now write a complete code for a program in English (for example), and then with minimum recoding, the program can be translated

to another language (like Arabic). This translation of program output can be completed by utilizing classes and routines identified in the Globalization component for the specific culture as supported in VS .NET. However, this list of cultures and languages in VS .NET Globalization is missing the Kurdish culture. This paper emphasize that the time is due for adding Kurdish to VS .NET globalization. It illustrates that all requirements for adding Kurdish as a culture in VS .NET can be met thus we feel that it is time to add Kurdish language to VS .NET globalization.

VB .NET Globalization – Brief History

Although the globalization component applies to the .NET Framework in general, this paper is going to focus the discussion and illustrate the main points through examples written in Visual Basic .NET (VB .NET). Narrowing down the discussion to applications in VB .NET helps explain the examples that will be presented, clarify the technical requirements and show various routines in this technical framework.

Globalization in its simplest term means exchanges between nations at various economic levels. In cases, globalization is attributed to the increasing exchanges between countries. However, in programming in general and in VB .NET in particular, globalization means adjusting programs so they can be used and generate output suited for different cultures and languages. Kaplan (2000) uses the terms Internationalization and Globalization interchangeably and explains both:

> "*Internationalization*, also known as *globalization*, is the process that converts an existing application to be globally aware. In an internationalized application, all the issues specific to other locales – such as changes in culture, convention, the input of data, the display of data, and the user interface elements – will work properly and will not be considered incorrect or offensive to the local user. Internationalization does not require an application's user interface to be in a different language from its original form …… Internationalization does require that the user can type in data in her own language and to see it displayed properly in that language" (p. 8).

Bradely & Millspaugh (2003) were more specific regarding the technical components of globalization and explained about this term within the programming context:

> "Globalization is the process of designing your program for multiple cultures and locations. The user interface as well as the output should allow for multiple languages. This is implemented through a set of rules and data for a specific language called culture/locale. A culture/locale contains information about character sets, formatting, currency and measurement rules and methods of sorting (p. 452)".

Latest development in VB .NET globalization created classes and routines available to developers in more than 100 languages. However, these developments and the inclusion of such many languages did not take place at a single step, instead it took place gradually and across different versions of the .Net Framework that were released from the beginning until now. The remainder of this section explains the development of globalization component within all versions of the Visual Basic starting from the time before .NET was developed and through the release of Visual Basic .NET version 2005.

Globalization in Pre .NET Era

This phase represents the period of developing versions of VB prior to the introduction of the .NET version. It includes beginning with Visual Basic 1.0 and leads through the development of VB versions 6, the last version released before switching to the .NET version. O'Reilly (2006)

described the introduction of VB into the marketplace as "breath of fresh air" because it simplified programming to a great extent. Through providing different Graphical User Interface (GUI) tools, programmers are able to accomplish more regarding the development of applications with less effort. Support for globalization at this phase was scattered. The number of languages that it supported was limited. Table 1 shows a list of languages that were supported during this phase of developing Visual Basic.

Table 1 – Display of Languages supported Windows 2000 (Kaplan 2000)

Language	LangID	Language	LangID
Arabic	1025	Italian	1040
Chinese (Simplified)	2025	Japanese	1041
Chinese (Traditional)	1028	Korean	1042
Czech	1029	Norwegian	1044
Dutch	1043	Polish	1045
Finish	1035	Portuguese (Brazilian)	1046
French	1036	Portuguese (Iberian)	2070
German	1031	Russian	1049
Greek	1032	Spanish	3082
Hebrew	1037	Sweden	1053
Hungarian	1038	Turkish	1055

One of the reasons that limited the number of languages supported in VB at this stage is that languages and cultures were represented using "code pages" which uses 8 bits or a maximum of 256 representations of code pages. Thus the technology available at that time limited the number of languages that were able to be supported.

Globalization in Visual Basic .NET

This is represented by the release of VB .NET 2002 and 2003. The release of both versions showed significant progress and simplification in terms of globalization and languages it supports. The ISO code started to be used to represent languages. The International Standards Organization has developed two sets of standards for coding languages and cultures- ISO 639.1 and ISO 639.2. It gave a unique three letter code for each of the languages. For example ISO designated the codes ara for Arabic, eng for English and kur for Kurdish. So as part of distinguishing each language it supports in globalization, VB .NET used this three letter designation of the ISO code. However, giving each language a code does not define them uniquely. For instance, Arabic is spoken in different countries and each country has its own set of formats. To resolve this kind of duplicate code, the terms of language, culture and locale are further defined and used. While a language is a spoken language, a culture refers to the specific population that speaks the language. A locale is usually noted within the culture. Bradley & Millspaugh (2003) call Locale "the rules and data for a specific language" and explained it further that it "Contains information about character sets, formatting, currency and measurement rules, and methods of sorting (p. 452)".

VB .NET gave each locale a distinguished code. This locale code along with the use of ISO code given to the language gives a unique combination for each of the locales represented in VB .NET.

For example the Iraqi locale is given the code ar-IQ, and the Egyptian locale is given the code ar-EG and the US English is given the code en-US and so on.

Ekedahl (2004) uses an example in VB .NET to list the data that are stored for various languages and cultures. The display of the output from this program is shown in figure 1. As the user clicks one of the cultures, the program displays the data specific to that language/culture (locale).

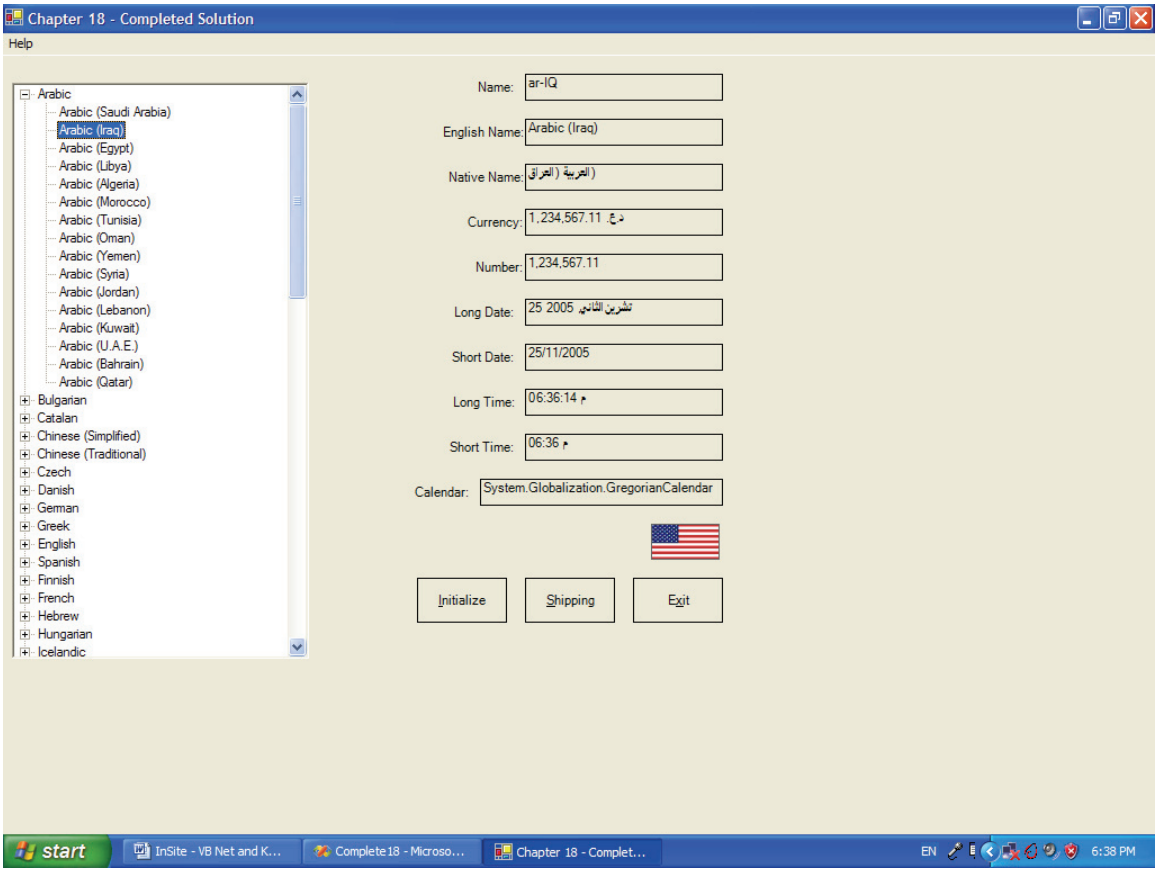

Figure 1 – Display of Data for Languages/Cultures Using a Program in VB .NET Ekedahl (2004)

A review of the example listed in Figure 1 shows that VB .NET supports 134 languages (Locales) in this new platform. The examples shows also that VB .NET added 15 locales within the Arabic language (such as Iraq, Egypt, Algeria), 13 locales within the English language (such as US, Belize, Trinidad) and 19 Spanish locales (such as Mexico, Chile, Peru). It also represented many languages within the same country such as 8 languages in India and representing less known languages such as the Syriac in Syria – However, Kurdish language is not represented.

Visual Basic .NET 2005

In this of VB .NET, new languages have been added to the list of Globalization supported languages. We noticed the addition of 25 new languages to the invariant culture of VB .NET 2005. The invariant culture was originally intended to be used for a different purpose other than to list new languages. Gunderloy (2003) explained two purposes for using the invariant culture:

- "Interacting with other software, such as system services, in which no user is directly involved.

- Sorting data in a culture-independent format that won't be displayed to end users" (P. 574).

Table 2 lists the new languages that are added to the Invariant Culture in VB .NET 2005. A quick glance of the table shows that most of these languages belong to a region of a country (such as southern Norway); languages in countries not represented before (Serbian in both Cyrillic and Latin), and to a country with a smaller population (Such as Maltese in Malta) – But Kurdish is not represented in any form in this version either.

Table 2 – Languages Added to the Invariant Culture in VB .NET 2005

Locale Name	Locale Name	Locale Name	Locale Name	Locale Name
Sami (Southern) (Norway)	Quechua (Ecuador)	Quechau (Peru)	Malayalam (India)	Sami (Lule) (Sweden)
Serbian (Cyrillic) (Bosnia and Herzegovina)	Sami (Southern) (Sweden)	Sami (Northern) (Finland)	Maoi (New Zealand)	Northern Sotho (South Africa)
Zulu (South Africa)	Sami (Northern) (Sweden)	Sami (Skolt) (Finland)	Serbian (Latin (Bosnia and Herzegovina)	Quechua (Bolivia)
Xhosa (South Africa)	Croatian (Bosnia & Herzegovina)	Welsh (United Kingdom)	Bengali (India)	Sami (Northern) (Norway)
Tswana (South Africa)	Sami (Inai) (Finland)	Bosnia (Bosnia & Herzegovina)	Sami (Lule) (Norway)	Maltese (Malta)

In addition to adding that many languages to the globalization list, VB .NET 2005 included a new class for customizing culture. Through this new class, programmers can add their own culture. The name of the new class is CultureAndRegionInforBuilder and can be used to create local custom build culture (MSDN, 2006; Symmonds, 2000). In other words, using this class, programmers can show data about cultures not represented in the .NET Framework.

Current Kurdish Representation in VB.NET

Basically, the Kurdish language is not included in the list of languages supported in VB .NET globalization thus depriving the Kurdish people from displaying and using VB .NET programs to generate output suited for their own culture and language. Microsoft grouped Kurdish with other languages that use the Arabic script. Dr. International (2005) is a forum provided by Microsoft that answers questions regarding global development and global portal. Dr. International described the process by which the Kurdish language is grouped:

> "The Arabic Windows code page (1256) is actually referring to the Arabic *language*, rather than the Arabic script (which supports many additional languages, such as Baluchi, Berber, Farsi, Kashmiri, Kazakh, Kirghiz, Kurdish, Pashto, Sindhi, Uighur, Urdu, and more). Unfortunately, there is simply not enough room on code page 1256 to support all of the additional characters that these other languages require (p. 1)".

But a closer examination of both languages reveals that the two languages, although they share the same script, they have many differences that representing Kurdish within Arabic language does not resolve the problems that VB .NET globalization intends to (Ali & Sulaiman, 2006). First, Arabic language contains 28 letters (Omniglot, 2005a) in contrast to Kurdish that contains

36 letters (Omniglot, 2005b). Second, Kurdish in some cases uses combined characters to represent one letter while Arabic does not have such feature. This changes the representation of characters and their sort orders. Gunderloy (2003) noted that VB .NET globalization has classes that address both features: combined characters and sort order. So including Kurdish within the Arabic language has two shortcomings: first, it does not represent all the letters of the Kurdish alphabet and second it does not sort the Kurdish alphabet correctly. In addition to the alphabet issue, there are many other differences between the two cultures that would limit and misrepresent the Kurdish culture by using an Arabic substitute. Calendar and day format are two additional elements that are examples of major differences between the two cultures.

Customized Kurdish Culture

In this section, we are using the CultureAndRegionInfoBuilder class that was introduced in VB .NET 2005 to create a customized Kurdish culture. This involves certain steps that begin by creating an empty customized culture and then to import data into that culture in order to assign values to the newly created Culture object's properties. It then involves executing a program to make sure that the new customized Kurdish culture has actually been added and the data are included correctly. Our goal from this section is to show how a Kurdish culture can be added to VB .NET globalization, thus demonstrating that there is no technical barries that limit the addition of Kurdish language to VB .NET globalizations. A complete listing of the program code is displayed in the Appendix at the end of this paper. The remainder of this section outlines the steps we followed in creating the curtomize Kurdish culture.

In Phase 1 of our project, we created a new culture within VB.NET to demonstrate the fact that new cultures can be introduced to the Globalization element of VB.NET and the Microsoft .Net Framework. To simplify the process, many of the properties for the new culture were imported from the currently existing .Net Framework culture, Iraq (ar-IQ).

In Phase 2, we are going to show how the new Kurdish culture can be created in full and compiled as a reusable object that would be importable into any computer with Microsoft .Net Framework 2.0. Below are the concepts and steps of how to achieve this goal:

1. Utilizing the CultureAndRegionInfoBuilder in Microsoft .net Framework 2.0, a new culture object will be instantiated
2. All properties of this culture must be set appropriately to reflect the Kurdish culture informatin, including culture name, day format, date format, calendar, etc.
3. After assigning all the properties for the culture, an XML file will be generated to include all the information described above. This XML file can be generated using the CultureAndRegionInfoBuilder.Save() method.
4. A user who choses to use the Kurdish culture, can imporat this XML file using the CultureAndRegionInfoBuilder.CreateFromLdml() method and then theRegister() method to register all the Cultuer's components within Microsoft .Net Framework 2.0

In more detail, the CultureAndRegionInfoBuilder object has several properties. Some are simple string format properties and some are objects. Exmaples of object type properties are; the number format, calendar, keyboard layout, etc. In order to create a complete culture object, all these properties must be assigned prior to generateing the importable XML file. For example, a special subroutine would be recommended just to create the calendar object for the unique Kurdish calendar and then assign properties for this newly instantiated object.

Font Issues. One of the major issues in creating a Kurdish culture in the past was the absence of a unified and standardized Kurdish font readable from any computer without having to install a

special font. Thanks to the Kurdit Group, a new universal Kurdish font has been introduced, which is called Unikurd Web. This font is compatible with all Microsoft versions and can be utilized in creating the Kurdish culture. More information about the Unikurd Web font can be obtained from the Kurdit Group at [www.kurditgroup.org].

We completed the code to create the Kurdish culture. In order to test it and show the new Kurdish culture, we used the same program that was written by Ekedahl (2004) which was shown in Figure 1 before. We show in Figure 2 a portion of the output from executing the program. This time, the program shows the new Kurdish culture that we added as a customized culture. Adding Kurdish culture is possible and there is no technological barrier that prevents adding it.

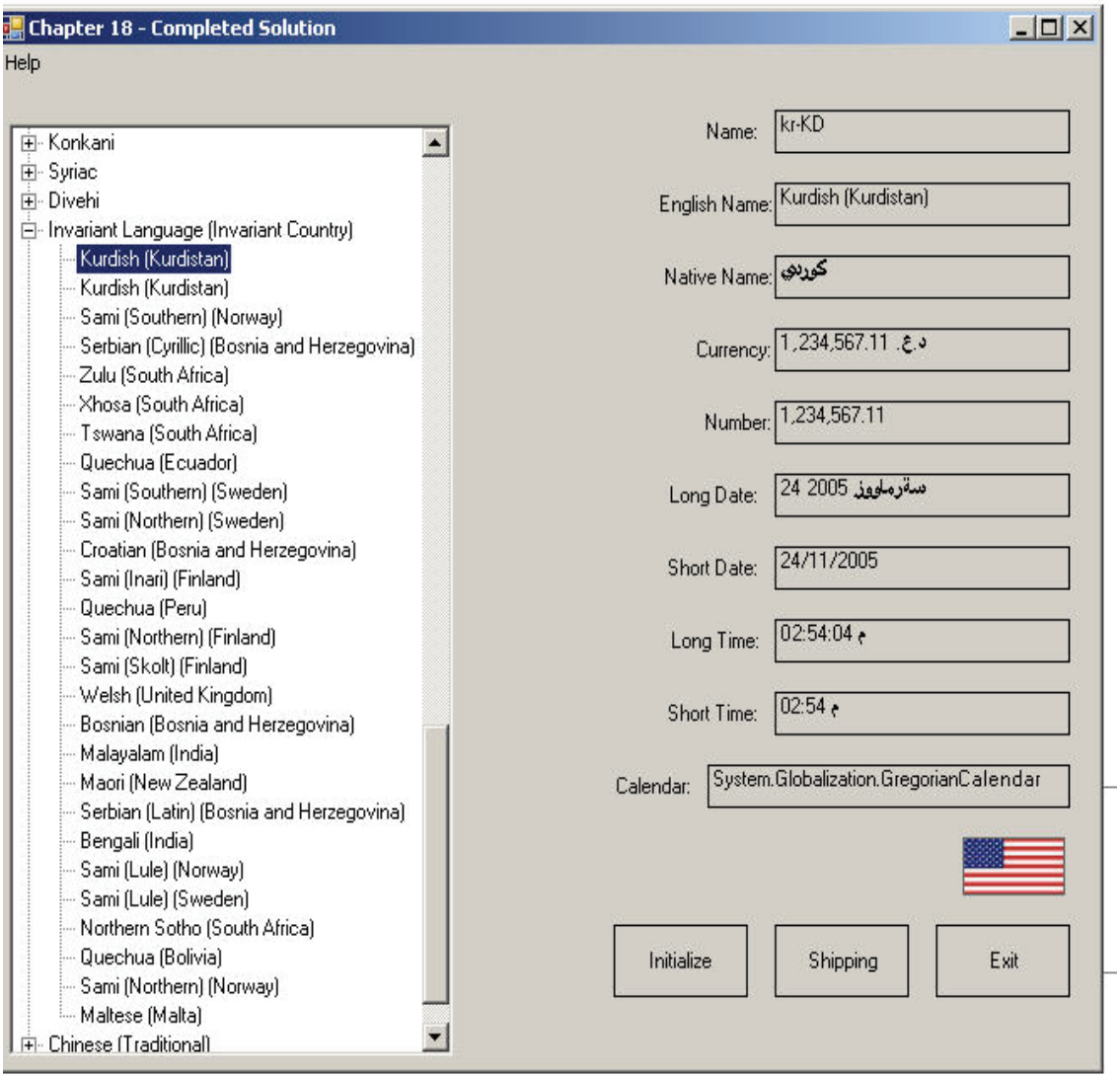

Figure 2 – Execution of a program that shows the new customized Kurdish culture

It is Time to Add Kurdish

So Kurdish can be added to VB .NET globalization. We noted earlier in this paper that the addition of Kurdish language was not possible in earlier version of VB .NET due to the limitation of the code page that was used to represent the different languages (see section **Globalization in**

Pre .NET Era in this paper). But the technology has improved significantly; the introduction of Unicode has allowed for more languages and further simplified adding more code pages. Kaplan (2000) explained about the many problems that Unicode resolved:

> Unicode was in many ways the answer to all the problems inherent in code pages. The Unicode standard is a 16-bit character designed to encode all known characters. In practice, even 2^16 or 65536 characters are still not enough, which is why the 2.0 version of the Unicode standard defines a surrogate range. This range of characters turns into a pointer to the next two bytes, this allowing a 32-bit character set." (p. 16).

So based on all the above we feel that it is time to add Kurdish language to VB .NET/VS .NET globalization for the following reasons:

1- The issue that used to limit the number languages because of restriction of code page no longer exists. Through the introduction of Unicode and the use of surrogate pair, this broke all technical barriers and enabled the addition of many languages including Kurdish.

2- ISO Code, the foundation for the VB .NET 2003 and 2005 locale coding, included Kurdish in both of its versions: ISO 603.1 and ISO 603.2.

3- We proved that adding a new Kurdish culture is feasible and can be done using available resources. We wrote an example that uses existing classes to add the Kurdish language. This eliminates any technical difficulty that may face adding the Kurdish language.

4- We made a case in our previous study (Ali & Sulaiman, 2006) for adding Kurdish as a separate culture. We based our case on issues of population, language components, cultures and other factors.

All technical and cultural barriers have removed, however the Kurdish culture is still not included in .Net Framework. In the absence of all notable barriers, one may think that there is some kind of unannounced "Political Barrier" that prevents or stops Microsoft from adding Kurdish to its globalization languages.

The Political Hurdle

In order to understand the political complexities that adding the Kurdish language may introduce, an explanation of the Kurdish culture, geography and history is warranted.

Kurdish is the language that is spoken by the people of Kurdistan. Most of the Kurds live in the area historically known as Kurdistan. The boundaries of Kurdistan extend from Iraq, Iran, Syria, Turkey and some parts of Azerbaijan. The total population of the Kurds is varied according to different statistics. Some estimate the population of the Kurds as high as 39 million people while others give a more conservative number of 26 million (Omniglot, 2005b). Most of the Kurds live in the southern part of Turkey.

The Kurds are described as "The largest ethnic group without a nation". Despite their increasing population, the Kurds are considered a minority in each of the four countries mentioned before. Given that the Kurds do not have an independent state, their recognition in various international settings have been often undermined and underrepresented. Their international representation has always been combined within the country in which they reside in. For example, the Kurds in Turkey are always been represented within the country "Turkey". Attempts to represent the Kurds without the main stream of the country are faced with resistance by the neighboring countries. Thus giving recognition of the Kurdish language in VB .NET globalization means some form of international recognition. This kind of recognition may be faced with resistance from these coun-

tries. It is our concern that Microsoft may be delaying or excluding Kurdish from its supported languages just to avoid this kind of potential political pressure. If this is the case, we proved in this paper that all the hurdles for adding Kurdish language to Globalization have been removed and that is up to Microsoft to remove the political hurdle.

Summary and Future Plan

This paper discussed adding Kurdish culture to VB .NET globalization. The Kurdish has not been included in any version of VB .NET globalization that has been released so far including Visual Basic .NET 2002, 2003 and the latest 2005. The paper explained the history of VB .NET globalization and then discussed in further detail a new class in VB .NET 2005 where developers can use to create their own customized culture. We used this new class to add the Kurdish as a customized culture. By adding Kurdish as a customized culture, we showed that Kurdish language has its own characteristics to make it eligible to be added to .Net Framework's Globalization as a separate language. Our solution was limited because it imported properties from an existing culture in .NET framework. But this solution nevertheless proved that adding the Kurdish language is possible and in our opinion. However, our solution have shortcomings, it is limited to only the platforms in which it will be executed on. We are interested in seeing the Kurdish culture installed by Microsoft as a default culture on its platform of Microsoft .NET Framework.

Our plan is to continue the dialogue with Microsoft regarding adding the Kurdish culture to VS .NET globalization. We will offer any kind of support possible if we were approached by Microsoft. But if Microsoft continues to exclude Kurdish from its globalization, then we have to conduct a more comprehensive project where we can create a reusable and importable Kurdish Culture with all its properties defined. We will also attempt to make our solution available on a broader scale such as the Internet. At the same time, we are hopeful that officials at Microsoft globalization pay attention to our studies and be convinced that "It is Time to Add Kurdish Culture to VS .NET globalization".

References

Ali, A. & Sulaiman, S. (2006). Adding a new language to Vb .NET globalization: Making the case for the Kurdish language. *Issues in Informing Science and Information Technology Education, 3,* 23-32. Available at http://informingscience.org/proceedings/InSITE2006/IISITAli198.pdf

Bradley, J. C. & Millspaugh, A. C. (2003). *Advanced programming using Visual Basic. NET.* Boston: Irwin McGraw Hill.

Dr. International. (2005). *Ask Dr. International.* Retrieved November 28, 2005 from http://www.microsoft.com/globaldev/DtIntl/columns.

Ekedahl, M. (2004). *Programming guide to developing and implementing windows-based applications with Microsoft Visual Basic .NET.* Reno, NV: Thomson Course Technology.

Global Development and Computing Portal. (n.d.). *Globalization step-by-step: Unicode enabled.* Retrieved November 20, 2006 from http://www.microsoft.com/glbaldev/getwr/steps/.

Gunderloy, M. (2003). *MCAD/MCSD developing and implementing Windows-based applications with Visual Basic .NET and Visual Studion .NET.* Indianapolis: QUE Publishing.

ISO 639.2 (2005). *Codes for representation of names of languages.* Library of Congress. Retrieved November 15, 2005 from http://www.loc.gov/standards/iso639-2/englangn.html

Kaplan, M. S. (2000). *Internationalization with Visual Basic. The authoritative solution.* Indianapolis, In: Sams Publishing.

MSDN (2006). *Custom cultures: Extend your code's global reach with new features in the .NET framework 2.0.* Retrieved November 20, 2006 from http://msdn.microsoft.com/msdnmag/

Omniglot. (2005a). *Arabic alphabet, pronunciation and language.* Retrieved November 28, 2005 from http://www.omniglot.com/writing/kurdish.htm.

Omniglot. (2005b). *Kurdish language, alphabet and pronunciation.* Retrieved November 9, 2005 from http://www.omniglot.com/writing/kurdish.htm.

O'Reilly (2006). *The history of Visual Basic revolution.* Retrieved November 2, 2006 from http://safari.oreilly.com/0321398009/ch02

Symmonds, N. (2002). *Internationalization and localization using Microsoft .NET*. Berkley, CA: Apress.

Appendix: Programming Code for Adding the Kurdish Language

```
Imports System.Globalization
Public Class frmCultureBuilder
    Private Sub cmdCreateCultureFromScratch_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
cmdCreateCultureFromScratch.Click
        CultureAndRegionInfoBuilder.Unregister("ku-KU")
        Dim cib As CultureAndRegionInfoBuilder = Nothing
        Console.WriteLine("Create the CultureAndRegionInfoBuilder..." & vbCrLf)
        cib = New CultureAndRegionInfoBuilder("ku-KU",
CultureAndRegionModifiers.None)
        cib.CultureEnglishName = "Kurdish (Kurdistan)"
        cib.RegionEnglishName = "Kurdistan"
        cib.CultureNativeName = "ßæÑÏí"
        Dim cia As CultureInfo
        cia = New CultureInfo("ar-IQ")
        cib.NumberFormat = cia.NumberFormat
        cib.GregorianDateTimeFormat = cia.DateTimeFormat
        cib.TextInfo = cia.TextInfo
        cib.CompareInfo = cia.CompareInfo
        cib.KeyboardLayoutId = cia.KeyboardLayoutId
        cib.RegionNativeName = "ßæÑÏÓÊÇä"
        cib.ThreeLetterISOLanguageName = "kur"
        cib.ThreeLetterISORegionName = "kur"
        cib.TwoLetterISORegionName = "ku"
        cib.ThreeLetterWindowsRegionName = "kur"
        cib.ISOCurrencySymbol = "USD"
        cib.CurrencyNativeName = "Dinary Kurdi"
        cib.CurrencyEnglishName = "Kurdish Dinar"
        Dim cinull As CultureInfo
        cinull = New CultureInfo("")
        cib.Parent = cinull
        cib.ThreeLetterWindowsLanguageName = "kur"
        cib.TwoLetterISOLanguageName = "ku"
        cib.IetfLanguageTag = "kurdisk"
        cib.Register()
        cib.Save("C:\temp\kurdish.xml")
        Console.WriteLine("Create and explore the custom culture..." & vbCrLf)
        Dim ci As CultureInfo
        ci = New CultureInfo("ku-KU")
        Console.WriteLine("Name: . . . . . {0}", ci.Name)
        Console.WriteLine("EnglishName:. . . . . {0}",
```

```
ci.EnglishName)
        Console.WriteLine("NativeName: . . . . . . . . . . {0}", ci.NativeName)
        Console.WriteLine("TwoLetterISOLanguageName: . . . {0}",
ci.TwoLetterISOLanguageName)
        Console.WriteLine("ThreeLetterISOLanguageName: . . {0}",
ci.ThreeLetterISOLanguageName)
        Console.WriteLine("ThreeLetterWindowsLanguageName: {0}",
ci.ThreeLetterWindowsLanguageName)
        ci = Nothing
    End Sub
    Private Sub cmdCreateCultureFromXML_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles cmdCreateCultureFromXML.Click
        Dim cib As CultureAndRegionInfoBuilder = Nothing
        CultureAndRegionInfoBuilder.Unregister("ku-KU")
        cib = CultureAndRegionInfoBuilder.CreateFromLdml("C:\temp\kurdish.xml")
        cib.Register()
    End Sub
End Class
```

Biographies

Azad Ali, D.Sc., Associate Professor of Technology Support and Training at Eberly College of Business – Indiana University of Pennsylvania has 23 years of combined experience in areas of financial and information systems. He holds a bachelor degree in Business Administration form the University of Baghdad, an M.B. A. from Indiana University of Pennsylvania, an M.P.A. from the University of Pittsburgh, and a Doctorate of Science in Communications and Information Systems form Robert Morris University. Dr. Ali's research interests include object oriented languages, web design tools, and curriculum design. His community service and academic expertise gets him in the news on Pittsburgh television and in the newspapers.

Frederick G. Kohun, Ph.D., Associate Dean and Professor in the School of Communications and Information Systems at Robert Morris University in Pittsburgh, has more that 30 years experience as a professor and administrator in the information systems field. He holds a bachelor degree in economics from Georgetown University, graduate degrees in economics and information science, from the University of Pittsburgh, and a Ph.D. in applied history in technology from Carnegie Mellon University. He had a leadership role in the design and implementation of eight technology based academic programs at the undergraduate and graduate level including a doctoral program. Most recently, he was involved in the first round of ABET-CAC information systems accreditation.