

Adding a New Language to VB .NET Globalization – Making the Case for the Kurdish Language

Azad Ali
Indiana University of Pennsylvania
Indiana, PA, USA

azad.ali@iup.edu

Seever Sulaiman
Interthinx
Calabasas, CA, USA

SSulaiman@interthinx.com

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 VB .NET globalization. However, this globalization list is incomplete and is missing many languages. Among the languages that are missing from VB globalization is the Kurdish language. This paper makes a case for adding the Kurdish language to the list used in VB .NET globalization. The paper starts by explaining about VB .NET globalization, the Kurdish language and then makes a case for adding the Kurdish language. Making the case is based on factors considered in including languages and also on the features of the Kurdish language. A summary and suggested future work is included at the end.

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

Introduction

Developing programs for multiple languages used to be long and daunting tasks. Changing a program so it can be used in another language used to involve making major adjustments to the original program and sometimes used to require writing a totally different program for the new language. For example, developing a program in English and then changing it so it could be used in Arabic would take making major modifications to the original code, adding routines and introducing new tools to make the changes.

The introduction of Visual Studio .NET has changed that drastically. Application developers can write programs in one language (say English) and then with minimum recoding, the programmers can adjust them so they could be displayed in another language (like Arabic). Processing of such changes is helped through the use of a list of languages supported in VS .NET (Finch, 2005). In other words, making modifications from one language to another is made easier because of a list of classes and routines for languages that are supported in VS .NET. However, this list of languages in VS .NET is incomplete and is missing many languages. As a result, missing languages

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.

is depriving many from using programs in their own languages. Among the languages that are not available in the VS .NET globalization languages is the Kurdish language.

This paper aims to build a case for introducing the Kurdish language to the list of languages supported in VB .NET globalization. It begins by clarifying the term “Globalization” and the factors that

are usually considered when translating programs into different languages. It then explains about the Kurdish language and the features it contains to make it qualify for inclusion in the list of globalization languages. The paper then makes the case based on the factors of the languages supported in the globalization and the features of the Kurdish language. A summary and suggested future work will be included at the end.

About Globalization in VB .NET

The introduction of VS .NET marks a significant increase in the use of “Globalization” term that is used in many programming textbooks (Bradley & Millspauh, 2003; Ekedahl, 2004; Gunderloy, 2003). Newer chapters are introduced in these textbooks to explain about this emerging concept in the programming area. This increase reflects the growing need for writing programs that cover this topic and to explain the concepts associated with it.

Globalization in its simplest term means to be dealing at a global (international) level. It is often referred to when talking about dealings arise across nations at economic levels. But “Globalization” in programming terms refers to a set of roles and procedures related to designing programs for more than one language. Bradely and Millspaugh (2003) 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)”.

So the basic purpose of globalization in the programming environment is developing programs so they could be used for more than one language (Mabbut, 2005). As noted before, such writing of programs for more than one language used to be a difficult task, however VB .NET has simplified this process significantly.

The contribution of VB .NET to globalization was through the provision of classes, routines and data that made translating programs into multiple languages simpler. These classes and routines tackled many of the problems that used to make translating programs into multiple languages a difficult process. In other words, VB .NET provides programmers with classes and routines that addressed many of the problematic areas that used to make the task of translating programs into different languages a difficult and daunting undertaking.

Problematic Areas in Globalization

Writers in the programming field identify five specific areas that are potentially problematic when designing a program for multiple languages: First the display of text information. Second, date and time format. Third, formatting numeric and currency data for languages. Fourth, the sort orders for the characters in the languages to be translated to and then the fifth problem is formatting of various other data like phone numbers, postal codes (Ekedahl, 2004; Gunderloy, 2003; Kaplan, 2000).

The first problem is the display of text data for different languages. This may include changing the user interface; display the text information in a particular language, and even different modes of data entries (like right-to-left entries in the Arabic script). This problem can be helped through the “language and globalization settings” in Windows control panel. The introduction of different fonts for various languages helped with this case also. The use of Unicode increased the chances for introducing various fonts that make it possible to virtually include every character in all the

languages. VB .NET helped this further through the ability to create resource files and satellite assemblies (Gunderloy, 2003). These last terms refer to the capacity to create multiple user interface files at design time; each file is set up for a different language. Then the users can select the interface they choose for their language.

The second problem is regarding the format for date and time in different languages. This may include the display of month names, days of the week and the various formats for date display and processing. For example, in the US the date is displayed by listing the month first and then the day and year. While in the Arabic world, the day is listed first and then the month and year. The problem gets more complicated when considering different calendar format that may give different dates for starting the year, and different number of days per year. For example, the “Hi-jri” calendar (Dr. International, 2005; Kaplan, 2000) in Saudi Arabia is a lunar-based calendar and contains smaller number of days per year than the 365 days in the Georgian calendar in the US. Another example is the Baha’i calendar has 19 months each has 19 days (Ekedahl, 2003), which is different than the 12 months and days up to 31 for other calendars. The Thai calendar is another example of a calendar with a different format (Kaplan, 2000).

The third problem is about the display of currency symbols and the formatting for numbers in different languages especially when dealing with number representing monetary values. There are many currency symbols, just to name a few: Dollar in the US, Yen in Japan, and Euro in Europe. These all may need to be displayed when working with a program in different languages. Then, the use of commas and decimal point add more difficulty also. Not all cultures use the same format for numbers regarding the thousand separator and the decimal points. For example, the US uses the comma as a thousands separator and the period for decimal points. The German language uses a period as a thousands separator in numbers and uses a comma to display the decimal points. The French language uses a space as a separator for numbers with thousands and comma for decimal points (Ekedahl, 2003).

The fourth problem that used to make developing international application a difficult task is about the sort order for characters in different cultures. It is customary to use a specific sort order that refers to the order of characters in the alphabet in each of the languages. However, the problem gets complicated when languages add other characters within their alphabet. These characters are called “Diacriticals”. Kaplan (2000) lists 23 characters that are used within the alphabet in different languages. Among these characters are included the grave (ã), the acute (é), and the tilde (ñ) The French language for example has a grave accent in their alphabet and the German use of an umlaut (Ekedahl, 2004). Turkish language sometimes has up to four versions of the same letter depending on the diacriticals used with the letter (Kaplan, 2000). Further, some languages combine two characters into one letter of the alphabet, thus the term “combining character sequence” is used in some programs (Gunderloy, 2003).

The fifth and last problem regarding international application is the use of postal codes, phone number formats and social security number formats. In some of the applications developed for users in the US, programs allow the entry of five digits postal code (or sometimes nine digits). But not all countries use the same five or nine digits. Some countries do not even use postal codes. The same thing can be said about phone numbers. Not all countries use ten digits for displaying and storing phone numbers and some countries do not use social security numbers in the same way used in the US.

The problems mentioned above are made easier to solve through the use of procedures stored within VB .NET. These procedures use data regarding date, time, sorting sequence and others that are stored in VB .NET for each of the languages that are supported in VB .NET globalization. But entering data for languages can be more complicated when taking into consideration the variety of languages and countries where they are spoken. For example, the English language is spoken

in different countries (like US, Britain, Australia) and each country has its own currency symbol, date format and other differences. Another example is that a language can be written in more than one alphabet. The Azeri language in Azerbaijan can use both the Latin and Cyrillic alphabets and each of the alphabets has its own sort order and set of roles. Thus, to solve the problems resulted from these differences, new terms of culture, language and locale have been introduced to VB .NET globalization.

Language, Culture and Locale

The introduction of VB .NET simplified the development of international applications by providing classes that help with the problems listed above. VB .NET stores data about different languages that it has identified to simplify developing programs for these languages. But defining a language runs into newer problems of having multiple languages for each country, multiple alphabets for the language and other issues as noted above. So to solve these newer problems, a newer classification of language, culture and locale has been used in VB .NET.

Language is a general term and refers to the languages spoken and the alphabet they use. But a culture is a more precise term than language. Gunderloy (2003), explained about culture in VB .NET:

“A culture identifies everything that might need to be localized in an application, which require you to know more than just the language. For example, just knowing that an application uses English as its user interface language doesn’t give you enough information to completely localize it: Should you format dates and currency amounts in that application, in a way appropriate to the United States, to the United Kingdom, to Canada, to Australia or to New Zealand (p. 5572)”.

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 and 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)”. 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 a user clicks one of the cultures, the program displays the data specific to that culture.

Figure 1 displays the data about the Arabic language and the Iraqi culture after the user clicked on the language and culture combination. It shows the data about the name, currency, date/time, calendar and others. Note also that the month name is displayed in Arabic. All of these different data are processed and displayed through classes and routines stored in VB .NET for each of the language/culture listed. To standardize the storage and retrieval of data, VB .NET uses a coding for the language/culture similar to the code followed by the International Standardization Organization (ISO).

International Language Standardization

ISO is the International Standards Organizations (or International Organization for Standardization) that develops standards used in different countries. The ISO is responsible also for developing various standards that are used in the computer field. Among ISO standards are two standards for languages and cultures. These standards are termed ISO 639-1 and ISO 639-2. These two standards are used for giving codes to various cultures and languages in VB .NET globalization. Ekedahl (2004) explained about all of this:

“To identify language uniquely, standards organization, such as the International Organization for Standardization (ISO), has created a standard coding system for

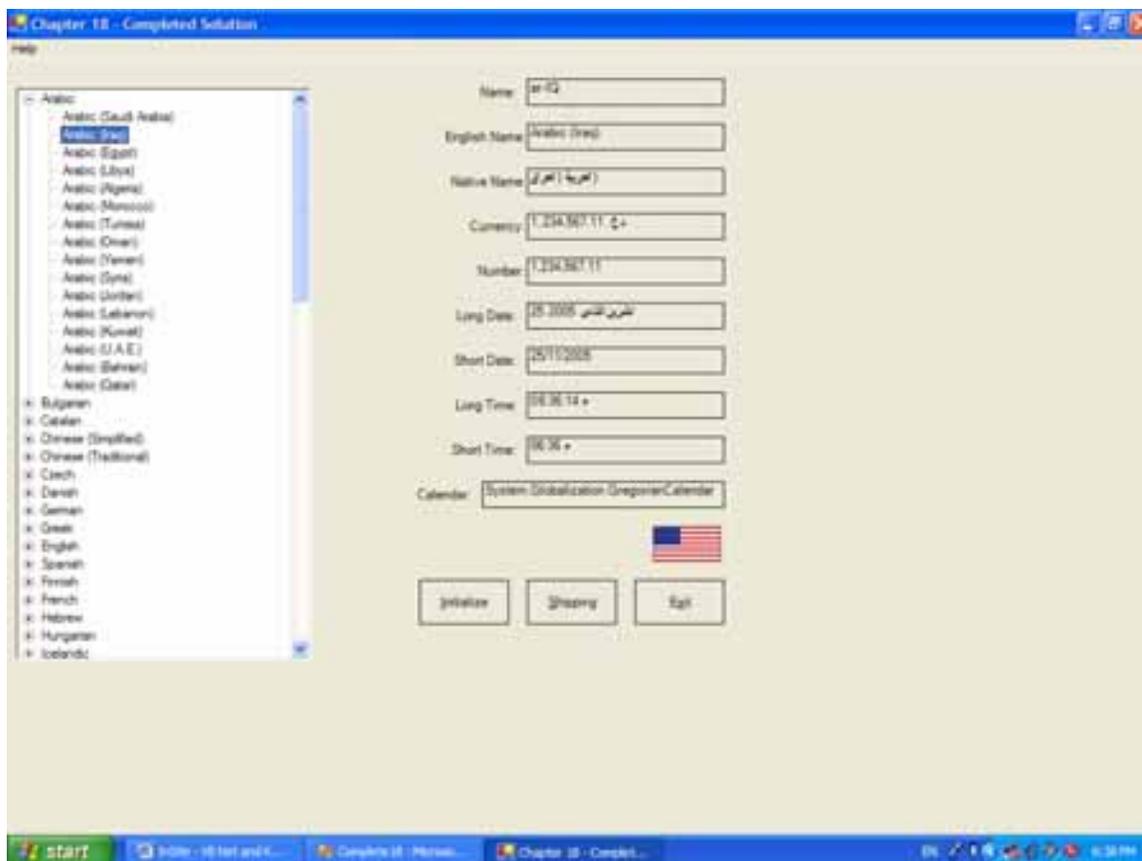


Figure 1 – Display of Data for Languages/Cultures Using a Program in VB .NET (Ekedahl, 2004)

all languages. The ISO 639-1 standard defines a two-character country code for each language. ISO refers to this two-character code as alpha-2 code. For example “en” is the alpha code for English, “fr” is the alpha code for French.

The ISO standard defines what is called the alpha-3 code. The alpha-3 code contains a 3-character specification where dialectical differences between cultures (countries) are significant” (p. 1052).

So based on the above explanation, the Arabic language is given a code of AR, while the language of Arabic that is spoken in Iraq is given the code “IQ”. Thus the code AR-IQ represents the culture code for Arabic in Iraq. In the same way, the code Az-Cyrl uses the Azeri language spoken in Azerbaijan that uses the Cyrillic alphabet.

The list of languages that are coded in ISO 639.2 is long and is listed across many pages that exceed hundreds of languages (ISO 639.2, 2005). This list also exceeds the number of languages listed in VB .NET globalization. Among these languages that are included in the ISO 639.2 list of languages is the Kurdish language that is given the code “kur-ku” in ISO.

About the Kurdish Language

Kurdish is the language that is spoken by the Kurdish people. 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 (Kurdistan, 2005). 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 (Kurdish Population, 2005) while others give a more conservative number of 26 million (Omniglot, 2005b). Most of the Kurds live in the southern part of Turkey.

The Kurdish language is of the Indo-European origin, similar to the Farsi (Persian) language. However, the different dialects use different scripts. The Kurds in Iraq use the Arabic script while the Kurds in Turkey use the Latin script and the Kurds in Azerbaijan use the Cyrillic alphabet. The main dialect among the Kurdish language is the Sorani (spoken in Iraq and Iran), and the Kermnji (spoken in Turkey and Syria). Given that the Kurds do not own an independent state, their recognition in various international settings has been often undermined and underrepresented. Among the international settings that they are underrepresented is the globalization list of languages in VB .NET. In other words, the Kurdish language is not among the languages that are supported in VB .NET. Thus, programmers cannot display data that are specific to the Kurdish language and Kurdish users cannot use VB .NET programs that display data in their language.

Inconsistencies in VB .NET Globalization Languages

The list of languages supported in VS .NET globalization does not represent all the languages spoken worldwide and it excludes many languages. We searched the languages that it supports and compared it to the languages available in ISO but did not find a clear pattern that is followed in VB .NET for including certain languages while excluding others. Basically, there are many inconsistencies among the method of including languages in VB .NET globalization list of supported languages. Lack of pattern in adding the languages to VB .NET globalization are explained in the following:

- 1- The list of languages in VB .NET is not based on the languages presented in the ISO. The ISO list of languages includes a substantially higher number of languages than those listed in VB .NET globalization.
- 2- The repeated name of countries and nations in VB .NET documentation may give the impression that the list of languages is based on list of countries or nations. However, the VB .NET globalization languages are not based on either countries or nations. The UN lists 191 member-nations (List of member states, 2005) within its charter. The CIA lists 239 countries (CIA – The World Factbook, 2005) that included both members and non-members of the (UN In Brief (2005)).
- 3- The list of languages is not based on population either. There are cultures that are listed with less population than many that are not listed.
- 4- The list of Arabic languages in VB .NET globalization includes 15 different cultures. The total number of Arabic countries is 21 (Arabic German Consulting, 2005), thus it is missing 6 countries like Sudan, Somalia, Djibouti and others. Again, we did not find a pattern for including these countries while excluding others.
- 5- There are several cultures that are listed separately, while there is minimal difference in their cultural data. For example, the Arabic culture in Iraq and Syria has similar data except for the currency. Yet these two cultures are listed separately.
- 6- The Syrian culture is listed twice, once under the Arabic language and the other under the Syriac language. But this does not include similar minorities in other countries like the Barbar in Morocco, the Kurds in Iraq, the Armenian in Turkey and the Hausa in Nigeria (Dr. International, 2005).

Making the Case for Adding the Kurdish Language

Basically, the Kurdish language is not included in the list of languages available in VB .NET globalization thus depriving the Kurdish people from displaying and using VB .NET programs in their own language. This section is to make a case for adding the Kurdish language.

Microsoft grouped the Kurdish with other ones that uses 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 the list of languages supported in VB .NET globalization reveals that there are many languages listed in the ISO while they do not appear within the languages in VB .NET globalization. Among these missing languages include Abkhazian (abk), Acoli (ch), Caddo(cad), Efik (efi), Kurdish (Kur) and other languages. So basically, people speaking these languages have to view and program in other languages. In the case of the Kurdish language, it can be implied from Microsoft statement (see the statement form Dr. International above), that the Kurds have to view their language in Arabic instead.

But the Arabic and Kurdish are two different languages. Although a portion of the Kurds uses the Arabic alphabet, but there are many differences between the two languages. Also, there are many features within the Kurdish languages that a case can be made to including it within the languages in VB .NET globalization. The case can be made based on the following facts:

- 1- The Kurdish language uses the same Arabic script but it has different set of alphabet. The number of characters in the Arabic alphabet is 28 letters (Omniglot, 2005a) and the number of letters in the Kurdish language is 36 letters (Omniglot, 2005b). This changes the entire set of text, sort order and others.
- 2- Different set of fonts are available for the Kurdish language such Ali-K-Samik, Ali-K-Sahifa, K-Dylan and others. Although these fonts can be used in Microsoft Office products, but it still cannot be used in VB .NET globalization. The availability of Kurdish fonts would make changing the display of the interface easier.
- 3- The use of dates is different from the Kurdish language than the Arabic, including the name of the months and the name of the days.
- 4- The use of the Arabic script is not limited to the Kurdish language; it is also used in the Farsi (Persian), yet the Farsi is supported in the globalization languages of VB .NET.
- 5- The population of the Kurds exceeds those of many cultures that are included in the globalization languages. Even if we take only the Kurds in Iraq, by every estimate their population exceeds four million (Kurdistan, 2005; Omniglot, 2005b) many countries (cultures) that are listed in the globalization languages.
- 6- All Kurdish scripts, whether they will be Arabic, Latin or even Cyrillic have their own set of characters and they are different from their neighboring languages.
- 7- The Kurdish language sometimes combines two characters into one. This makes sorting the data different than the Arabic characters. In other words, it cannot be assumed that

just because both languages have the same alphabet that the names or addresses will be sorted in the same way, they will not.

Summary and Future Work

This paper explained about adding a new language to the VB .NET international language settings. It made the case regarding adding the Kurdish language based on the characters of languages in the globalization settings and the features of the Kurdish language. It further explained about the globalization setting, the problems that were encountered when designing application for multiple languages and then showed how VB .NET helped solving some of these problems through storing data for different languages. We made the case for the addition of the Kurdish language based on these facts and based on a program written in VB .NET 2003.

At the time of this writing, VS .NET 2003 was the latest version of Visual Studio that was available in the market. However, we learned that a new version of VS .NET 2005 is about to be released in the near future. According to our information, the new version of visual studio still does include the Kurdish language among its set of globalization languages. Based also on the inquiries we made that Visual Studio .NET version 2005 contains new classes that enable programmers to add their own customized cultures. So if Microsoft continues on excluding Kurdish from the globalization languages, this may create a situation that it will be helpful to write another paper that shows the technical details and steps of how to add the Kurdish language as a new customized culture to VB .NET programs. It is our intention to write this new paper that will explain about the steps required and program code specifics for adding the Kurdish language.

References

- Arabic German Consulting. (2005). Arab countries and links to Arabic web sites. Retrieved November 28, 2005 from <http://www.arab.de/arabinfo/arabinfo.htm>.
- Bradley, J. C. & Millspaugh, A. C. (2003). *Advanced programming using Visual Basic.NET*. Boston: Irwin McGraw Hill.
- CIA – The World Factbook. (2005). The world factbook: Rank order – Population. Retrieved November 28, 2005 from <http://www.cia.gov/pulications/factbook/rankorder/2119rank.html>
- 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.
- Finch, T. (2005). Internationalization. Retrieved November 9, 2005 from <http://www.vb-helper.com/essay9.htm>.
- Global Development and Computing Portal (2005). Frequently asked questions: Locales & languages. Microsoft Corporation. Retrieved November 15, 2005 from <http://www.microsoft.com/globaldev/DrIntl/faqs/Locales.mspix>
- 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>
- List of Member States. (2005). Retrieved November 28, 2005 from <http://www.un.org/overview/unmember.html>
- Mabbut, D. (2005). Globalization in Visual Basic .NET. Cultural Code: Numbers and VB.NET. Retrieved November 8, 2005 from http://visualbasic.about.com/od/using_vbnet/a/blobvbnet1.htm

- Kaplan, M. S. (2000). *Internationalization with Visual Basic. The authoritative solution*. Indianapolis, IN: Sams Publishing.
- Kurdistan. (2005). Iraqi Kurdistan. Retrieved November 28, 2005 from <http://www.unpo.org/member.php?arg=34>
- Kurdish Population. (2005). The distribution of the Kurdish population. Retrieved November 28, 2005 from <http://www.geocities.com/kurdishcommunityofottawa/population.htm>.
- 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>.
- UN In Brief (2005). How the UN works. Retrieved November 28, 2005 from <http://www.un.org/overview/brief1.html>
- Visual Basic and Visual C# Concepts (2005). Introduction to international application in Visual Basic and Visual C#. Microsoft Corporation. Retrieved November 4, 2005 from <http://msdn.microsoft.com/library/en-us/vbcon/html/vbinternationalapplication.aspx>.

Biographies



Azad Ali, D.Sc., Associate Professor of Technology Support and Training at Eberly College of Business – Indiana University of Pennsylvania has 22 years of combined experience in areas of financial and information systems. He holds a bachelor degree in Business Administration from 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 from Robert Morris University. Dr. Ali's research interests include object oriented languages, web design tools, and curriculum design.



Seever Sulaiman, MBA, is a Chief Technology Officer of Interthinx which develops fraud detection and compliance tools for the mortgage industry. Interthinx has recently been formed as the merger of the equals between AppIntelligence, Inc. and Sysdome Corporation. Sulaiman possesses over eleven years of experience in the field of information technology, and eight years in the mortgage industry. He now holds the job of Chief Technology Officer and directs the product development and implementation efforts of the organization. Sulaiman holds a master's degree in business administration from Webster University and a bachelor's degree in computer engineering from the University of Baghdad.