XLANGBOX-ARA 2.0

The Bilingual Latin/Arabic Supplement

to X Window/ Motif Applications

Technical Product Description

October 1998

1 Introduction

1.1 Background

Arabic, as a calligraphic language, presents major processing problems.

An Arabic character might take one, two, or sometimes four different shapes, yet it is represented by one code. The shape of the character is determined depending on its position in the word. This is but one problem called "**Character Shaping**".

Another problem is the direction of writing. Arabic text is written from right to left. This conflicts with Latin, which is written in the opposite direction. When mixing text languages, characters are **added** in one language and **pushed** in the other. This is called "**Bi-Directionality**".

Some users speak only Arabic. They will not accept a cursor positioned at the leftmost position of the screen. They want to have an option allowing them to start at the rightmost position of the line, i.e., in brief, a mirror image of the screen. The implication is that, in this mode, Latin characters are pushed from right to left. This is called "**Screen Mirroring**"

Yet one more complication is vocalization (or **diacritics**). These characters, like their counterparts in Latin, the vowels, are a linguistic necessity, even if they are not used in most of basic publications. Yet, in Arabic, they appear on top or below their respective consonants.

These linguistic complications -- and more -- make Arabic a difficult language to handle.

Another aspect of the problem is that standard UNIX Operating system support of Internationalization (local) supplies only support for European Languages generally based on the ISO 8859-1 character codeset. This National Language Support (NLS or XLOCALE) allows keyboard mapping, handling of collating sequences and character types, date and time format. It could also include, in the near future, a specific extension for the Japanese language, but cannot handle Arabic characters, and, in general "right-to-left" writing direction languages.

LangBox International is specialized in the design and the development of bilingual and multilingual Operating Systems. It has implemented bilingual capabilities on a large number of machines operating under UNIX, XENIX, AIX, SUN/OS, ISC, SINIX, DG/UX, EP/IX, CLIX,.SOLARIS, IRIX, OSF/1, etc…

The Arabization (i.e. implementation of the Arabic language support) of UNIX applications can be addressed in two main ways:

- For "**Character based**" applications that run generally on ASCII terminals (like ANSI, VT100, VT220, WYSE...), arabization process can be achieved on the input/output character flow of the terminal along the TTY line. The Arabic font is initially loaded on the terminal local RAM.

  The Transparency regarding (8 bits cleaned) applications can be assumed. The application manages an 8 bit codeset (ISO 8859-6) and the specific Arabic processing is done by the Operating System TTY lowest level.
  LangBox International designed and implemented "**LANGBOX for Arabic**" that provides a

global and transparent solution for this kind of applications. This product is documented separately.

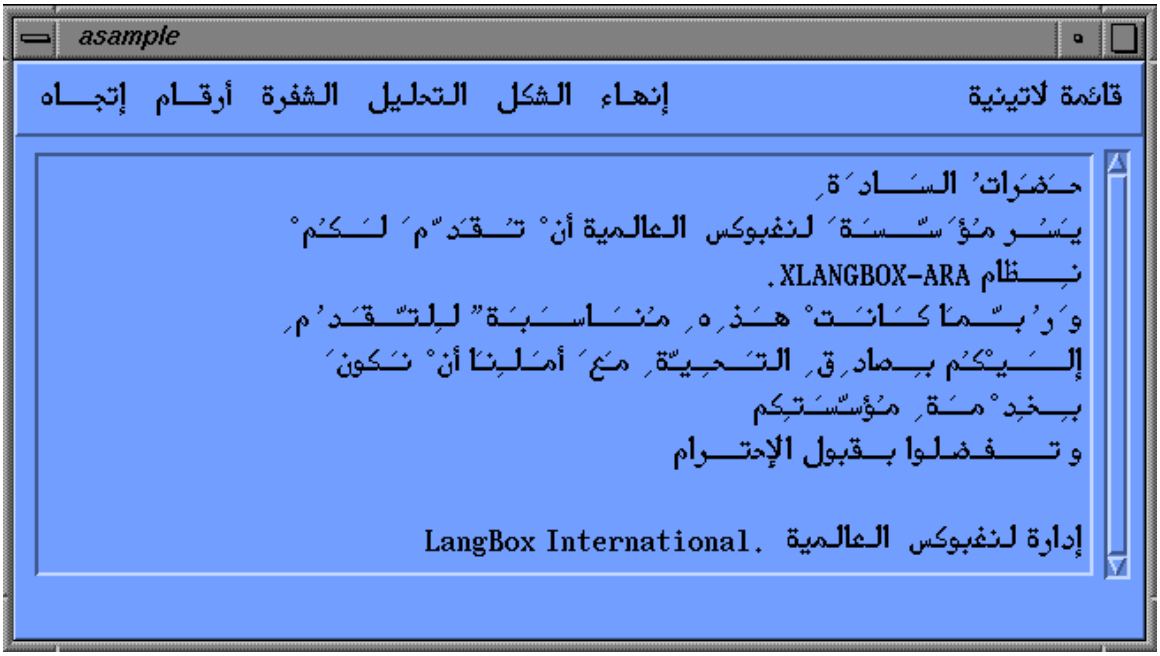
- With the development of **Graphical Interface** (like X Window) on workstations, other kinds of applications have been designed. These applications are clients that communicate with a Graphic Server through network connection facilities.

  Here, the main application routines work directly with bitmaps and the concept of character flow has disappeared. A transparent Arabization is more difficult to implement. We need at least to link the application with an Arabic Graphical library, or use an Arabic shared library at the runtime level (if the operating system allows it). Also, some application concepts must be implemented to allow a correct behavior. For example, we always need to refresh the complete line after each input character. Moreover, fonts could be selected through a Resource file or a command line option.

  Even if all these features are present, parts of the application could not behave correctly in Arabic (like cut and paste for example), and, a specific implementation must be done for the Arabic support by adding specific source code lines.

The **XLANGBOX for Arabic** package (called **XLANGBOX-ARA**) has been defined to allow Arabization of pure X Window applications and allows the following features:

- Providing a global solution to Arabic and Latin simultaneously.
- Maximum Transparency of applications.
- Total transparency to storage and display of data in national languages.
- Ease of internationalization of applications.
- Conformity with national and international standards.



This screen hardcopy shows a sample motif based application launch under the XLANGBOX-ARA environment.

2 The XLANGBOX-ARA Environment

The XLANGBOX-ARA library package is build around the MIT X Window and OSF MOTIF libraries. When added and linked to an application that uses these interfaces, XLANGBOX-ARA provides the user with a full bilingual environment in runtime application operations.

The XLANGBOX-ARA package is specially designed for software developers willing to address the Arab countries' market. With XLANGBOX-ARA, they can move a standard Latin application with minor changes. As set of a demo and X Window samples, OSF MOTIF and UIL program are arabized and included in the package.

Arabization level parameters can be selected and set at the runtime level. These parameters could be either user independent or in the application. They include "context analysis" (automatic shape determination, dual keyboard state and mapping, numeral shapes, vowels)

The Arabic character set handled by XLANGBOX-ARA is ISO 8859-6, which is the standard adopted by the International UNIX community.

XLANGBOX-ARA is composed of the following parts:

- The Arabic Context Library
- The X Window Arabic Extension
- The OSF MOTIF Arabic extension
- The Arabic fonts for the server
- The Demo Set
- The Printing Support

These parts can be used together or separately depending on the need and the internal organization of the application to be arabized.

2.1 The Arabic Context Library

This library contains the specific routines to manipulate Arabic string. These routines allow the following:

- "Contextation" of ISO 8859-6 strings coming from a storage area, and makes it readable according to the Arabic context (generally for a display purpose).
- Character position calculation: These routines allow to locate the new position value for a character after or before a contextation.
- Attribute range selection: These routines allow to calculate a new attribute range of a string after or before a contextation.
- Font name mapping management: Allows to dynamically associate Arabic fonts to Latin default font loading process.
- "Contextation" parameter management: These routines allow to get or to set the specific Arabic parameter used during the "context analysis" of strings. These parameters are mainly the following:
    - left to right / right to left main direction
    - Hindi/Arabic numeral shape
    - Vowels enabled/disabled (tashkil)

    - Automatic shape determination on-off
    - Data processing/word processing
    - Neutral character management
    - Arabic floating point symbol
- Arabic Dual Keyboard management - Key Layout customization for engraved kayboards.
- Arabic Data codeset filtering (such as for MS CP1256 codeset).

This library is an independent library. It can be used for any application that wants to handle directly Arabic strings.

2.2 The X Window Arabic extension

This library contains Arabized X11 routines for string manipulation. These routines are mainly the following:

For string display purposes:
- XDrawString( )
- XDrawImageString( )
- XDrawText( )

For keyboard management
- Xlookupstring( )

For Font loading management:
- XloadFont()
- XLoadQueryFont()

In order to be used, this library must be added to the link phase and be referenced before the standard X11 one (-lX11) on the command line or for Runtime only configuration, the LD_LIBRARY_PATH environment variable should be set to the XLANGBOX-ARA directory.

2.3 The OSF/Motif Arabic extension

This library contains Arabic complement routines to the famous Open Software Foundation "MOTIF Library". This complement allows the handling of the following primitive MOTIF widgets:

- XmText
- XmTextField
- XmString

All other widget types just benefit of this arabization (such as list, file selection box, push button, menu, etc.).
In order to be used, this library must be added to the link phase and be referenced before the standard (-lXm) on the ld command line or for Runtime only configuration, the LD_LIBRARY_PATH environment variable should be set to the XLANGBOX-ARA directory

2.4 The Arabic fonts for the server

XLANGBOX-ARA supplies a set of Arabic fonts for the X server running on the target machine. These fonts are in "pcf" format and also in "bdf" format for heterogeneous network connections. Several sizes are available and can be selected either from the application itself or from a resource file.

XLANGBOX-ARA contains fixed spacing fonts and proportional spacing fonts that gives better looking results for Arabic.
The listing of the fonts is the following :

```
ara0814_96.bdf: ara0814_96
ara0814t96.bdf: ara0814t96
ara0915_96.bdf: ara0915_96
ara0920_96.bdf: ara0920_96
ara1230_96.bdf: ara1230_96
ara0915p96.bdf: ara0915p96
naskhiBf08.bdf: -lbi-naskhi-bold-r-normal--8-80-75-75-m-50-iso8859-6
naskhiBf10.bdf: -lbi-naskhi-bold-r-normal--10-100-75-75-m-60-iso8859-6
naskhiBf12.bdf: -lbi-naskhi-bold-r-normal--12-120-75-75-m-70-iso8859-6
naskhiBf14.bdf: -lbi-naskhi-bold-r-normal--14-140-75-75-m-90-iso8859-6
naskhiBf18.bdf: -lbi-naskhi-bold-r-normal--18-180-75-75-m-110-iso8859-6
naskhiBf24.bdf: -lbi-naskhi-bold-r-normal--24-240-75-75-m-150-iso8859-6
naskhiRf08.bdf: -lbi-naskhi-medium-r-normal--8-80-75-75-m-50-iso8859-6
naskhiRf10.bdf: -lbi-naskhi-medium-r-normal--10-100-75-75-m-60-iso8859-6
naskhiRf12.bdf: -lbi-naskhi-medium-r-normal--12-120-75-75-m-70-iso8859-6
naskhiRf14.bdf: -lbi-naskhi-medium-r-normal--14-140-75-75-m-90-iso8859-6
naskhiRf18.bdf: -lbi-naskhi-medium-r-normal--18-180-75-75-m-110-iso8859-6
naskhiRf24.bdf: -lbi-naskhi-medium-r-normal--24-240-75-75-m-150-iso8859-6
naskhiRp08.bdf: -lbi-naskhi-medium-r-normal--8-80-100-100-p-12-iso8859-6
naskhiRp10.bdf: -lbi-naskhi-medium-r-normal--10-100-100-100-p-15-iso8859-6
naskhiRp12.bdf: -lbi-naskhi-medium-r-normal--12-120-100-100-p-18-iso8859-6
naskhiRp14.bdf: -lbi-naskhi-medium-r-normal--14-140-100-100-p-21-iso8859-6
naskhiRp18.bdf: -lbi-naskhi-medium-r-normal--18-180-100-100-p-28-iso8859-6
naskhiRp24.bdf: -lbi-naskhi-medium-r-normal--24-240-100-100-p-36-iso8859-6
naskhiOp12.bdf: -lbi-naskhi-medium-o-normal--12-120-75-75-p-64-iso8859-6
naskhiOp24.bdf: -lbi-naskhi-medium-o-normal--24-240-75-75-p-126-iso8859-6
naskhiOp10.bdf: -lbi-naskhi-medium-o-normal--10-100-75-75-p-54-iso8859-6
naskhiOp34.bdf: -lbi-naskhi-medium-o-normal--34-340-75-75-p-177-iso8859-6
naskhiOp20.bdf: -lbi-naskhi-medium-o-normal--20-200-75-75-p-105-iso8859-6
naskhiOp14.bdf: -lbi-naskhi-medium-o-normal--14-140-75-75-p-75-iso8859-6
naskhiBp12.bdf: -lbi-naskhi-bold-r-normal--12-120-75-75-p-63-iso8859-6
naskhiBp24.bdf: -lbi-naskhi-bold-r-normal--24-240-75-75-p-123-iso8859-6
naskhiBp10.bdf: -lbi-naskhi-bold-r-normal--10-100-75-75-p-53-iso8859-6
naskhiBp34.bdf: -lbi-naskhi-bold-r-normal--34-340-75-75-p-172-iso8859-6
naskhiBp20.bdf: -lbi-naskhi-bold-r-normal--20-200-75-75-p-102-iso8859-6
naskhiBp16.bdf: -lbi-naskhi-bold-r-normal--16-160-75-75-p-84-iso8859-6
naskhiBp14.bdf: -lbi-naskhi-bold-r-normal--14-140-75-75-p-73-iso8859-6
```

2.5 The Demo set

This set is given for X11/MOTIF developers. It allows to understand how arabization can be included in an existing application. This demo set includes samples for:

- Sample character based program
- Pure X Window (X11) interfaced program
- OSF/MOTIF interfaced application
- OSF/MOTIF and UIL build applications

Several Arabic software developer's recommendations are also included.

2.6 The Printing Support

This sub package consists in a set of printer fonts that are downloaded directly on the supported printer, using specific XLANGBOX-ARA commands.

It also includes a new specific line printer spooler that must be used instead of the standard **lp** or **lpr** ones, for printing Arabic files. Postscript printing is also supported using the **aa2ps** filter tool.

3 The XLANGBOX Standard Aspect

XLANGBOX-ARA is adapted to the standards as set forth by AT&T's SVID, X Window System and ISO.

The standards adopted under **XLANGBOX for Arabic** are related to the following:

- Character sets.
- Standard display conventions.
- Arabic level support standards.

3.1 The Character Sets

The Arabic characters are 8-bits wide and conform to the following standard:

- ISO 8859-5 (ASMO 708).
- ASMO 449+

Also, by filtering, the following data Codeset can be read and converted to ISO 8859-6:

- MS CP 1256
- ISIRI 3342

3.2 Display Conventions

Characters will be displayed according to their language specific conventions. Latin characters will always appear separately, while Arabic characters will be contexted and displayed in their composite form.
The display technique adopted depends on the base and current languages chosen by the user.

When the environment is Latin, the initial cursor position is at the left of the line and characters are added to the right as they are entered. Arabic characters are inserted, and pushed to the right as they are entered.

When the environment is Arabic, the reverse phenomenon occurs: the initial cursor position is at the right most position of the line, Arabic characters are added and Latin's are inserted.
Vocalization is fully supported on terminals capable of displaying 256 downloadable characters.

3.3 Neutral Characters

These characters are not context sensitive and do not, therefore, affect the shape of Arabic characters. Yet they have, in certain instances, opposite meaning in Arabic, due to the direction in which this language is written. Typical examples are: ( ) { } < > [ ] ...
A special command handles the meaning the user wants to assign to these special characters.

3.4 Numerals

The user is provided with a command allowing the display of numerals in Latin (Arabic shapes) or Arabic (Hindi Shapes). All known complications associated with this subject have been solved and incorporated in the package.

3.5 Automatic Shape Determination

The standard rule is to display characters in the way calligraphy requires it. It was, however, found that this feature should be optional, since system users are frequently in debug work sessions and prefer to have their characters displayed in their original, base form. A pair of

routines has been included to inhibit or restore Automatic Shape Determination.
This process is also called "context analysis".

3.6 Dual Keyboard

XLANGBOX manage a logical dual keyboard. A keystroke allows switching from one to the other (generally the <ctrl T>). Arabic layout is indicated by transparent stickers that the user disposes on his Latin Keyboard. The system's compose key or pressing both Alt key simuneatously can also be used.

Sample SGI IndigoMagic Desktop launched with XLANGBOX-ARA

4 XLANGBOX-ARA Level Support Standards

4.1 Environment Variables

The XLANGBOX-ARA working environment uses these environment variables:

AR_DIRECTION	=	latin | arabic	(set the display direction)
AR_CONTEXT	=	on | off	(enable disable auto shape determination)
AR_TASHKIL	=	on | off	(enable disable Arabic tashkil)
AR_HINDI	=	on | off	(enable disable Hindi numerals)
AR_DATA_PROC	=	on | off	(enable disable DataProc mode)
AR_NEUTRAL	=	"*string*"	(define the neutral character list)
AR_KBDLANG	=	latin | arabic	(set the initial keyboard language)
AR_KBDTOGGLEKEY	=	*code*	(set the keyboard toggle key code)
AR_KBDMAPFILE	=	*filename*	(set the keyboard file mapping)
AR_FORCEFONTNAME	=	on | off	(activate the dynamic font name mapping)
AR_DEFAULTFONTNAME	=	*fontname*	(define the default font for dynamic mapping)
AR_FONTSET	=	“sting”	(Define the Arabic fontset used)
AR_FONTMAPFILE	=	filename	(define the ouput font mapping)
AR_CODESET	=	*codeset*	(define the data input codeset for conversion)

4.2 Using dynamic linked libraries with XLANGBOX-ARA

This section tries to explain how we can use existing applications that use shared libraries, and specially X Window or/and Motif libraries

Such an executable program will load its X Window or/and Motif routines at the runtime phase, using a runtime linker. If we substitute the regular X Window or/and Motif libraries of the system with the Arabized ones, then the used program will be able to manage Arabic data without any modifications. This is a very important feature that allows to run any existing English applications with Arabic data and without any change to its binary code. This feature is called "**Transparency**".

However, all X Window or/and Motif applications will not behave correctly in Arabic at 100 per cent, and some specific parts of the application may not be covered by the Motif or the X Window Arabization. All display that don’t use the Motif or the X Window set of routines, but some proprietary routines and/or fonts. We could find these cases on GIS map design routines for example.

In order to replace regular dynamic linked libraries with the Arabized ones we could use the environment variable LD_LIBRARY_PATH. This variable allows to define the search path for shared objects at the run time.

5 Availability

XLANGBOX-ARA 2.0 is available on the following operating systems/environments:

- SUN Solaris 2.3, 2.4, 2.5.1 and 2.6 for SPARC series / OpenWindow or CDE
- SUN Solaris X86 2.6 (beta only)
- CDC's EP/IX / RISC Window
- SGI's IRIX 5.3 and 6.2 / X Window and Indigo Magic
- DEC UNIX (OSF1) 3.2 on Alpha / CDE

XLANGBOX-ARA 2.0 is available on the following target machines :

- PC/AT Architecture on the Intel 80386 and 80486 platform
- Sun's SPARC series.
- Control Data's Mips series.\
- Silicon Graphics INDY, INDIGO, CHALENGE and ONYX Series.
- DEC AlphaStation

XLANGBOX-ARA 2.0 provides full support for the following printer or compatible printer:

- IBM 4201 us (9 pin)
- IBM 4201 (9 pin)
- Epson lq1000 (24 pin)
- Fujitsu dl3400 (24 pin)
- HP deskjet 500+
- HP laserjet (PCL)
- DEC LA75+
- Postscript

Standard ZMail Motif based application dynamically linked with XLANGBOX-ARA Motif library

Note: This document includes some terms such as UNIX, X Window, OSF Motif, Postscript, IndigoMagic, Zmail, VT100, VT220, WYSE... that are trademarks or Copyrights from their respective authors.