

Arabic Tools
for
AutoCAD
by Khalid M. Khalid

CONTENTS

Part I: Arabic Tools		Page
1.1	What is the "Arabic Tools for AutoCAD"?	7
1.2	List of Files (demo release)	8
1.3	Installation, Loading and Initialization	8
1.4	Arabic Tools Interface	9
1.5	Command Reference	10
	1.5.1 ARABIC	10
	1.5.2 MSTYLE	13
	1.5.3 ATEXPL	15
	1.5.4 CRVFIT	15
	1.5.5 TXTIN	16
	1.5.6 TXTOUT	17
	1.5.7 ATFIND	19
	1.5.8 ATEDIT	21
	1.5.7 ATCFG	21
1.6	Code Page Translation	24
1.7	Arabic Tools APIs for AutoLISP Programmers	24
	1.7.1 (at_codepage)	
	1.7.2 (at_cptranslate)	
	1.7.3 (at_getstring)	
	1.7.4 (at_isat)	
	1.7.5 (at_kblayout)	
	1.7.6 (at_kbstr)	
	1.7.7 (at_makestyle)	
	1.7.8 (at_style)	
	1.7.9 (at_stylelist)	
	1.7.10 (at_text)	
	1.7.11 (at_textinput)	

Part II: VisualSHX

2.1 Overview	33
2.2 Fonts in AutoCAD	33
2.3 The Arabic Font Set	34
2.4 Designing an Arabic Font	35
2.4.1 Font Settings	37
2.4.2 Designing Arabic Characters	38
2.4.3 Font Compilation	41
2.5 VisualSHX Command Reference	42
2.5.1 FONTSET	42
2.5.2 DSTAT	44
2.5.3 SET	45
2.5.4 GET	45
2.5.6 PUT	46
2.5.7 FONTMATX	46
2.5.8 SEQ	47
2.5.9 POLREV	47
2.5.10 ORDER	47
2.5.11 PLACE	48
2.5.12 VCOMPILE	49
2.5.13 RUN	50

Appendix I: Arabic Tools Font Codes

Appendix II: Arabic Tools: Available Fonts (Registered Version)

Appendix III: Installation Trouble Shooting

Appendix IV: Arabic Keyboard Layout

Appendix V: Contact Information

Part I

Arabic Tools

1.1 What is the Arabic Tools for AutoCAD?

The Arabic Tools (AT) is an AutoCAD® plug-in which enables the addition of Arabic text to AutoCAD drawings. The full edition of AT provides a number of Arabic fonts that are suitable for most requirements. The fonts may be inserted in various styles and in any justification and rotation angle. Utilities for creating special effects, not normally available to AutoCAD text, have also been provided. Text lines created by AT may be exported to Autodesk's 3D Studio MAX for the production of still renderings and animations that contain true 3D Arabic text.

FEATURES

(Demo Release)

- 1- Contextual Analysis text insertion (automatic character concatenation).
- 2- All justification types (left, middle, center, align, fit), in any rotation angle.
- 3- Italic text.
- 4- Exploding AT text entities (vectorization).
- 5- Curve fitting.
- 6- Ease of use and control over text style and placement in a WYSIWYG manner.
- 7- Search and replace text.
- 8- Text export and import incorporating code page translation.
- 9- VisualSHX™ Fonts editor within AutoCAD.

(Registered Version)

- 10- Full font set.
- 11- Arabic spelling checker.

1.2 List of Files

The demo release of AT includes the following files:

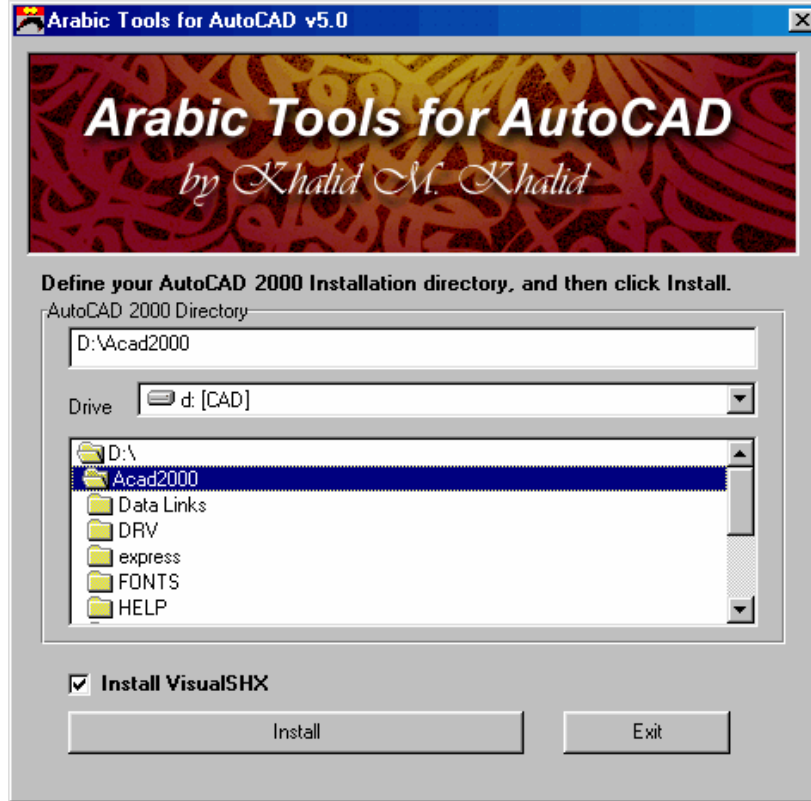
ARBTOOLS.VLX	LISP Arabization engine
ARBTOOLS.ARX	ARX Arabization engine
ARBTOOLS.DCL	AT dialogue file
ARBTOOLS.MNS	Menu file

ARBTOOLS.MNC	Compiled menu file
ARBTOOLS.MNR	Menu resource file
ARBTOOLS.INI	AT variables setting file
CODPAGE.ASC	Code page translation file
ARBTOOLS.SHX	Single-line font file
NASKH.SHX	Double-line font file
ARBTOOLS.DWG	Sample drawing
ATFONTS.DWG	Sample fonts
VSHX.VLX	VisualSHX Engine
VSHX.DWT	Arabic font design drawing template
CHARCODE.TBL	Arabic character codes and names
SETUP.EXE	Windows based installation program
ATManual.DOC	MS-Word file containing this manual

1.3 Installation, Loading and Initialization

The Arabic Tools release includes a Windows®-based installation program which can be used to install AT under the AutoCAD directory on your computer. If you are installing from the release CD the installation program will run automatically. Otherwise, you have to run the **SETUP.EXE** program from the desktop, the Windows Explorer, or the Run dialog box. Wait until the program initializes, and then designate the AutoCAD directory by navigating through the drive and directory lists. Click on the **Install** button after designating the AutoCAD 2000 directory, and that's all it takes to have the Arabic Tools installed on your computer.

The dialog has a check box to confirm the installation of the VisualSHX Arabic font editor components. If you have no intentions as all to design Arabic fonts, you can uncheck this box.

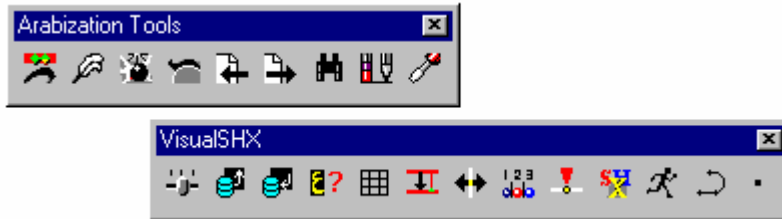


1.5 Arabic Tools Interface

When you start an AutoCAD session after installing the Arabic Tools, the AT interface elements should appear on the screen. The interface comprises the **ArbTools** and **VisualSHX** pull-down menus and toolbars. The toolbars have helping tips that appear when the cursor is placed over its buttons. This will help you identify each button. The AT interface has an Arabic and English pull-down menu. The English menus are installed by default before the Help menu on AutoCAD's menu bar. You can select the menu you want to appear

on the menu bar using AutoCAD's **MENULOAD** command. This command can also be used to control the appearance of AT toolbars.

AT commands described in section 1.5 below can be accessed by selecting them from the menu, the toolbars, or typing the command on AutoCAD's command line.



1.5 Command Reference

Command: **ARABIC**



The **ARABIC** command is used to enter Arabic text in the drawing. The command's sequence resembles AutoCAD's **DTEXT** command, and behaves similarly throughout:

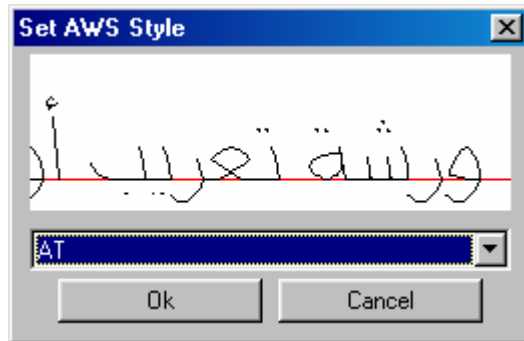
Command: arabic
Justify/Style/<Start point>:

If you select Justify by typing <j>, or the complete word, you will be asked to define the justification you desire:

Left/Center/Middle/Align/Fit:

You must then define the justification points as necessary.

Alternately, if you type <s> or <style> at the Justify/Style/<Start point>: prompt, the Style Selector dialogue box appears. This dialogue box lists all defined AT text styles. It also has an image box at the top where you can visually identify the style you want to select.



If you supply the start point, the command requests you to define the text height. However, if the current text style has a pre-defined height, then the height will not be requested at this point, and the previously defined height will be used (see the **MSTYLE** command below).

The last parameter requested before actual text insertion is rotation. You can use the pointing device to define the rotation angle or type the angle on the command line. The default values always appear between the angle brackets, and may be selected by simply pressing Return, the Space bar, or the selection button on your pointing device. (Angle -180 that appears is equal to 0)

You can enter Arabic text when the prompt: Arabic text <in Graphics Screen> appears. You do not have to change the keyboard language to Arabic. In fact, **MAKE SURE THAT THE KEYBOARD LANGUAGE IS ENGLISH AND THE CAPS LOCK IS TURNED OFF WHEN YOU USE THE ARABIC COMMAND**. A square cursor appears in the drawing area, and the shape and stance of the square cursor reflect the used style's width factor and oblique angle respectively. As you key in Arabic text AT, performs automatic character concatenation by what is known as 'Contextual Analysis';

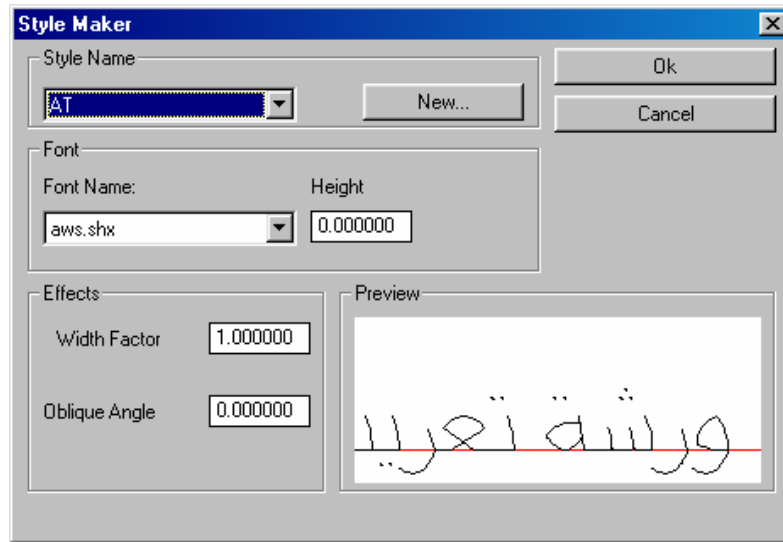
the process of choosing the correct Arabic character shape for the current position. If you press return at the end of a text line the cursor will jump to the beginning of a new line. If you pick a point using the pointing device during text insertion the cursor will jump to the selected point and you may enter text at the new location in a similar fashion to AutoCAD's **DTEXT** command. You can back up to the previous line using BackSpace. If return is pressed at the beginning of an empty line the **ARABIC** command will terminate and control returns to AutoCAD.

As you type, the cursor may move beyond the limits of the current view. It may also be large or small beyond legibility. As the text is not echoed to the command line area, it will be difficult to see what you are typing. For this purpose, a number of view control commands may be activated during text input, and you do not have to exit the **ARABIC** command to adjust the view. This is achieved using the following 'Ctrl' key combinations during text input (you press the 'Ctrl' key, and while still holding it down you press the letters indicated after the hyphen as indicated below):

Ctrl-I: Zoom In
Ctrl-O: Zoom Out
Ctrl-W: Zoom Window
Ctrl-R: Pan Right
Ctrl-L: Pan Left
Ctrl-U: Pan Up
Ctrl-D: Pan Down
Ctrl-P: Free Panning

With these commands, you have full control over the placement and typing of text.

Command: **MSTYLE**



Type **MSTYLE** at the command line, or select 'Style Maker' from the menu, and you will be presented with the Style Maker dialogue box. This dialogue box allows you to interactively define the text styles you will require for your drawing.

The image box at the bottom-right displays the style you are designing. Any adjustment you make using the buttons, menus, or input fields of this dialogue box is instantly reflected on the displayed image, and you can accurately determine the amount of adjustment you need.

The Style Maker dialog components are:

- 1- Style Name: The name of the current style is displayed in the combo-box. If you drop down the list, the names of all styles

defined in the drawing will be displayed, and you can select any of them to become the current text style. The settings for the selected style will be displayed in their respective boxes.

- 2- The New... Button: Pressing this button will open a small dialog box that allows you to define a new style name.
- 3- The Font File Drop-down List: Each style has an assigned font file. The fonts available in the AT release you have will be listed here. Choosing a font file from this list will assign this font to the style and its image will be displayed in the image box above. (The demo release includes two font files: **ARBTOOLS.SHX** and **NASKH.SHX**).
- 4- The Height Edit Box: As per AutoCAD norms, the default value of an AT style is 0.0. This means that this style is a variable-height text style. When inserting text in this style (using the **ARABIC** command), you will be prompted to provide text height. However, if you designate a definite height to a text style you create, this style will be a fixed-height text style. When you use this style later to insert text in your drawing, you will not be prompted for text height, and the height you have specified for the style upon creation will be used.
- 5- Width Factor Edit Box: Used to set the style's width factor. The width factor should be a positive real number.
- 6- Oblique Angle Edit Box: Used for setting the style's oblique angle. Negative values will cause Arabic text to slant backwards (towards your right). Positive values will create the effect of slanting in the direction of text.
- 7- The Image Box: When you open the Style Maker dialogue box the Image box displays the current style's image. If you define a new style (which assumes the status of the current style), its image will be displayed in the image box. You will notice that no change happens to the image box when you define a new style. This is because the newly defined style inherits all the settings of the previous current style. As you start

assigning new values, they are instantly reflected in the image box.

- 8- Ok Button: You can define as many styles as you want, each with its own font file, height, width factor, and oblique angle assignments. But this will not take effect in your drawing unless you exit the Style Maker dialog box by selecting Ok.
- 9- Cancel Button: If you don't want the changes you've made to take effect, exit the Style Maker by selecting the Cancel button.

Command: **ATEXPL**



The Arabic Tools provide this utility command to convert AT text entities into polylines. The command sequence is as simple as selecting an AT text entity and then the new point to position the exploded text. If you wish to keep the copy of the exploded text in the same position, you can right-click the mouse. The generated polyline copy will be drawn on the current layer. All the curves will be drawn using straight line segments. The number of segments for each curve can be set through the **ATCFG** command. Increasing the number of segments will yield smoother curves.

The shape of the generated polyline form can be edited or texture filled as desired. It is not a text entity and cannot be used for normal operations that require a text entity, such as search, editing, export, etc.

Command: **CRVFIT**



This command enables you to place an AT text entity on an ARC or a CIRCLE. Only AT text entities are valid for curve fitting with this command. The text entity will be copied as a collection of polylines. The profile of the curve will form the base line for the copied text.

Text can be placed inside or outside of curves, in either an Align or Fit mode. The Align mode scales the text according to the size of the curve. The Fit mode allows you to adjust the text height. The command sequence is as follows:

```
Command: crvfit  
Select AT text:  
Select curve:  
In/<Out>:  
Fit/<Align>:
```

The generated polyline shape can be edited or texture filled as desired. It is not a text entity and cannot be used for normal operations that require a text entity, such as search, editing, export, etc.

Command: **TXTIN**



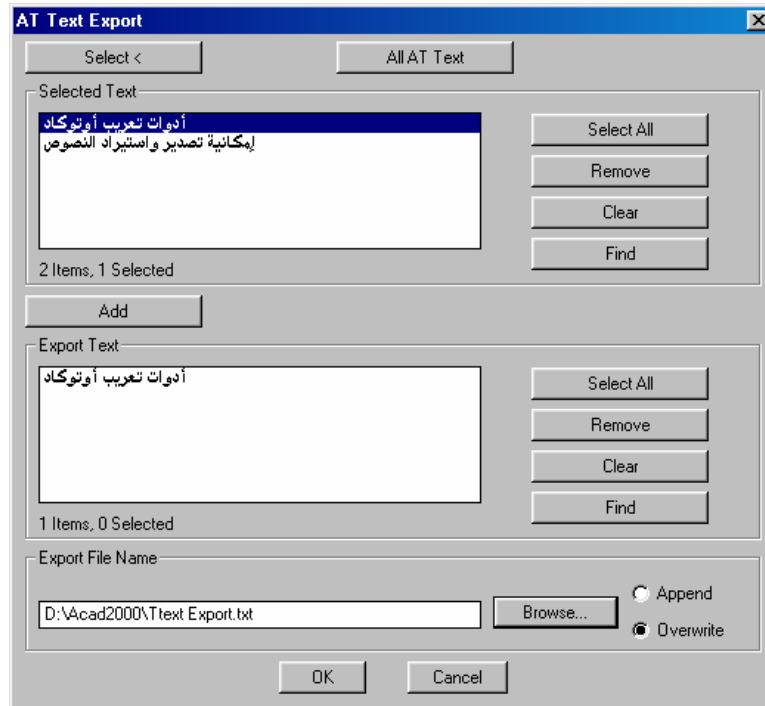
You can import Arabic text files into AutoCAD drawings using the AT **TXTIN** command. This command starts by displaying the standard file name selection dialog, and after you designate the file you wish to import, you will be prompted with the normal text placement message:

```
Justify/Style/<Start point>:  
Height:  
Rotation:
```

These options allow you to position and place imported text just as you would when using the **ARABIC** command (See the **ARABIC** command for a detailed description of the sequence of these prompts and actions of response). The file will be read in and positioned according to the parameters specified.

Importation of text is affected by the code page setting. AT's default code page is suitable for Windows users. You can change the code page setting using the **ATCFG** command. However, the code page you choose should be defined in the **CODEPAGE.ASC** file (see section 1.5, Code Page Translation on page ...).

Command: **TXTOUT**



This command allows you to export AT text to disk files. Text will be output according to the current codepage setting. After typing the command on the command line or selecting it from the menu or

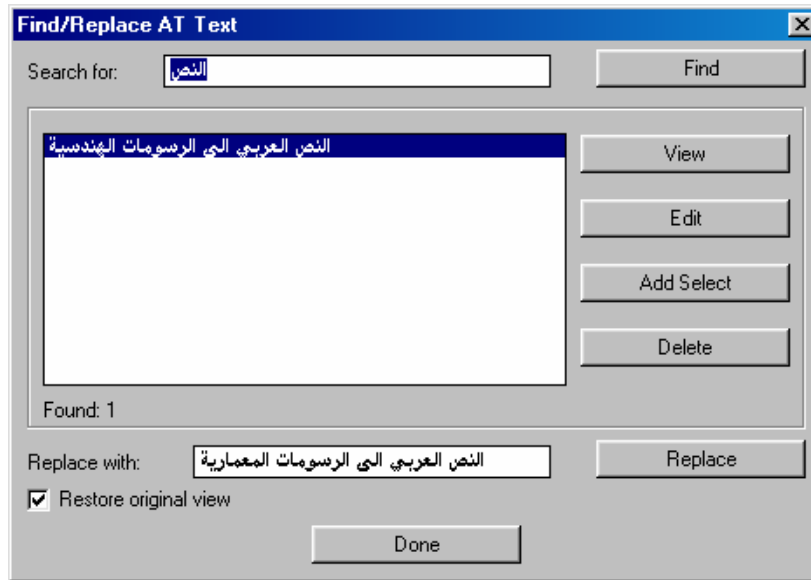
toolbar, you will be prompted to select the text entities to export. Using the standard object selection process, select the text objects you want to export. The selection can contain any object. The selection set will be filtered, and only AT text entities will be used. After you terminate your selection, the Export Text dialog will be displayed, and the text entities selected will be listed in the top list of the dialog. If you need to select additional text entities, you can do so by pressing the Select button at the top left. Pressing the All AT Text button at the top right will select all AT text present in the current drawing. You need not to worry about duplicate selection as the system will filter out any duplicate text entity.

From the selected list you can now designate the text lines to be written to the export file. Select any number of lines in the top list and then press the Add button. The selected lines will be copied to the output list below. Again duplicate selections are automatically filtered out. Only one instance of any text line can be exported.

To write the text to the disk you have to select them also in the output list. Again only the selected items in the output list will actually be written to disk. The name of the output file has to be supplied either through the edit field or through the Browse button. You can control the write mode by selecting an option of either appending exported text to the selected file (which will be created if does not exist) or overwriting the file name supplied (again creating it if does not exist). For each of the lists four buttons assist in selection as explained below:

Select All:	Selects all items in the list
Remove:	Removes the items which are currently selected
Clear:	Removes all items in the list irrespective of being selected
Find:	Allows you to search for a string in the list. The result of the find will be highlighted.

Command: **ATFIND**



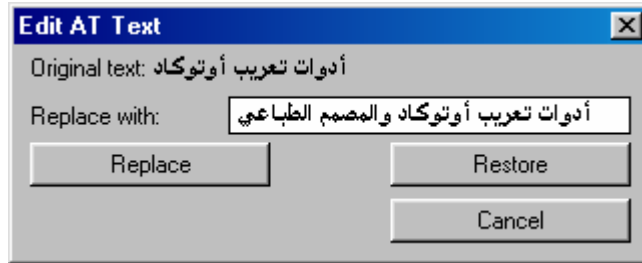
This command allows you to search for AT text in the drawing. It also provides the facility to edit the text strings found. The command brings up a dialog box that has the following components:

1. 'Search for' edit box and 'Find' button: You type the search string in this edit box. The search string can be any text pattern you wish to find. Pressing Enter or the 'Find' button will start the search process.
2. The result list: All finds are listed here. (Text lines in this are left justified in this text box. This is a known limitation.)
3. View button: You can view a text line found by selecting the text line in the list, and then pressing the View button. The dialog will be temporarily hidden, and the view zooms to reveal the actual text object in the drawing. The 'Zoom Realtime' command

will be activated and you can zoom and pan about. When you exit the 'Zoom' command the dialog will be displayed again. The 'Restore original view' check box at the bottom of the dialog will be activated to allow you to return to the same view after you terminate the dialog, in case you had changed through the 'View' option in the dialog.

4. Edit button: This button will place a copy of the string selected in the list in the 'Replace with' edit box below. This allows you to change the text string.
5. Add Select button: This will add the string selected in the list box to a selection set. When you exit this command you can perform any AutoCAD editing operation on this set, like moving, copying, erasing, etc.
6. Delete button: Will delete the selected text item in the list from the drawing (and the list as well).
7. 'Replace with' edit box and 'Replace' button: You can place a text item in the 'Replace' edit box by selecting it and pressing the 'Edit' button, or simply by double-clicking on the item in the list. After editing the string you press Enter, or the 'Replace' button. The text object in the drawing will be updated, and if the edited string contains the search pattern, it will remain the search list. However, if the new string does not contain the search pattern, it will only be updated in the drawing.
8. Done button: Exits the dialog. Changes made in the AT Find/Replace dialog are instantly executed. The 'Done' button merely closes the dialog, and no changes are initiated by this pressing this button.

Command: **ATEDIT**

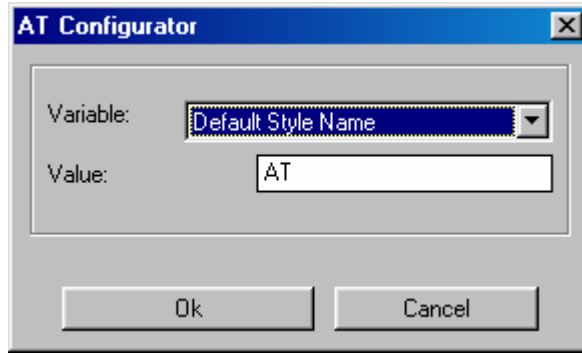


Editing an AT text object in the drawing is possible using this command. After you issue the command, you will be prompted to select the AT text object, after which the dialog above appears. The original text is displayed above, and also placed in the edit box for modification. You can edit the string and press Enter or the 'Replace' button to have the object modified, and the dialog terminated. However, if you wish to restore the edited string to the original text of the object selected, press the 'Restore' button. Pressing the 'Cancel' button will exit the dialog without change.

Command: **ATCFG**



This command allows you to adjust certain AT operation parameters to suite your particular requirements. AT operational parameters are stored in the **ARBTOOLS.INI** file. The command will activate a dialogue box that allows you to adjust the AT operation values.



The AT Configurator dialogue box has the following parts:

- 1- The Variable Name Drop Down List: A list of all AT's operation parameters.
- 2- The value of the selected variable is displayed below the variable name.
- 3- Ok Button: This button will accept your changes and store them in the AT.INI file. The settings you make will take effect upon exiting via the Ok button and will be retained in later sessions until you change them once again with the configurator. (You can manually edit the **ARBTOOLS.INI** file. This approach, however, is **NOT** recommended)
- 4- Cancel Button: If you wish to bail out without making any changes the Cancel button should be your choice.

Changeable AT variables are:

- 1- Default Style Name: (Preset: AT)
- 2- Default Style Font File: (Preset: **ARBTOOLS.SHX**)
- 3- Default Style Text Height: (Preset: 0.0. Valid values: positive real)
- 4- Default Style Text Width Factor:(Preset 1.0. Valid values: positive real)
- 5- Default Style Text Oblique Angle:(Preset 0.0. Valid values: -180.0 to 180.0)

- 6- Default Style Backwards:(Preset: No. Valid values:Yes/No)
- 7- Default Style Upside down(Preset:No. Valid values:Yes/No)
- 8- Text Height Prompt: The value for height under the **ARABIC** command (Preset: 0.2. Valid values: positive real)
- 9- Arc Segments: The number of segments generated for each text curve in the ATEXPL command. (Preset: 8. Valid values: integer more than 2)
- 10- Zoom Factor: The amount of zoom-in or zoom-out of the view control commands used under the **ARABIC** command.(Preset: 0.5. Valid values: positive real)
- 11- Pan Factor: The amount of pan-up, pan-down, pan-right, pan-left of the view control commands under the **ARABIC** command.(Preset: 0.5. Valid values: positive real)
- 12- Line Feed Scale: Controls the distance between AT text lines. The scale is with respect to text height. (Preset: 2.0. Valid values: positive real)
- 13- Polyline Welding Tolerance: The force that causes the segments of exploded Arabic text entities to weld. This force represents the distance between segment ends that will cause them to snap together. The distance will be the in the scale of the smallest segment formed. If the **ATEXPL** command fails to generate closed polylines for outline fonts, the experiment by increasing the tolerance to approach the value of 1.0.
- 14- AT UserI: AT stores the current Arabic style index in the drawing's USERI variables. AutoCAD provides 5 such variables which the user, or AutoCAD developers can use to store integer values. AT allows you to select one of the five integer variables for the storage of its current style index to avoid conflict with other AutoCAD plug-ins. The default USERI used by AT is 5. You can set it to any other between 1 and 5.

5. Code Page Translation

The Arabic Tools provides a means to translate code pages. This is basically useful for controlling text exportation and importation. Code pages are defined in a special file named **CODEPAGE.ASC**. Each code page is entered as a column in the code page table. The column header is the name of the code page, which is used for its identification by the program. The first column lists the logical code page, and the second is the keyboard code page. The Arabic Tools requires these two code pages for processing purposes. Do not edit these two columns.

You can add a code page by adding a column to the table. The code page name should be on the line starting with the word 'Letter' and the ASCII number of each letter listed below, against its respective name, which is listed in the first column. Lines starting with a semicolon are considered as comments. The file itself starts with some editing comments. Any editing you make to this file will not take effect until the Arabic Tools are loaded in a new AutoCAD session. The Arabic Tools sets the default code page upon initializing according to the setting found in **ARBTOOLS.INI**. Experienced users can 'arxunload' **ARBTOOLS.ARX** and reload it. This will invoke a fresh loading of **CODEPAGE.ASC**.

6. Arabic Tools APIs for AutoLISP Programmers

The Arabization Tools exposes a number of functions that enable AutoLISP programmers to develop applications that are 'AT' enabled. These APIs provide an alternate means to add Arabic text to AutoCAD drawings. You can add Arabic text, create Arabic text styles, extract Arabic information from AT text entities, among other things. All Arabic Tools APIs start with the 'at_' prefix.

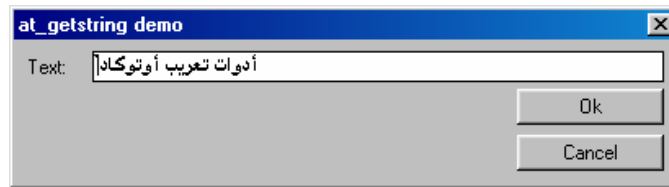
AT APIs are as below, with description, in alphabetical order:

(at_codepage [cp])

This function sets the code page or returns the current code page. If you supply the code page argument the function will set the current code page the name you supply. The code page name has to be one of those listed in the **CODEPAGE.ASC** file which is supplied by AT as described in section 5 above. If you do not supply a code page name, the function returns the name of the current code page. The Arabic Tools sets the default code page according to the setting in the **ARBTOOLS.INI** file. This default code page setting can be maintained by the Arabic Tools **ATCFG** command.

(at_cptranslate str fromcp tocp)

Translates the string provided in `str` from codepage `fromcp` to codepage `tocp`. The `fromcp` and `tocp` should be code pages listed in the **CODEPAGE.ASC** file. This function is useful for translating text strings to and from AT text entities which store Arabic text in the "Keyboard" code page (see `at_kbstr` below).

(at_getstring prompt)

You can retrieve Arabic strings from the user with this function. A dialog box appears, and the string you pass in the `prompt` argument will appear as the caption of the dialog. The keyboard switches to Arabic and the user need not press the Alt-Right Shift keys, allowing for rapid input of text. If the user presses Enter or the Ok button the typed string will be returned. An empty string ("") will be returned if the user does not provide any text in the edit box. If the user selects the Cancel button, the function returns a `nil`.

(at_isat *ename*)

You can check whether an object is an Arabization Tools text entity by passing the entity's name to this function. The function returns `T` if the object is an AT text entity. Otherwise it returns `nil`.

(at_kblayout [*kblay*])

You can change the keyboard language from your AutoLISP programs using this function. The `kblay` argument has to be a valid keyboard layout name. If the function is not successful in setting the keyboard layout name passed in the `kblay` argument, it returns `nil`.

The `kblay` argument is optional. You can call the function without supplying a keyboard layout name. The function in this case will return the name of the current keyboard layout. In the following example `(at_kblayout)` is used to set the language of the keyboard to Arabic, and then restore the original keyboard layout.

```
; Save the original keyboard layout
(setq oldkblay (at_kblayout))
(if (= (kblayout "00000401") nil)
    (alert "Setting Arabic keyboard layout failed")
    (progn
      ...
      ...
      ; Reset the keyboard layout to the original
      (at_kblayout oldkblay)
    )
  )
```

(at_kbstr *ename*)

This function returns the "keyboard" string of the Arabic Tools text entity passed to it in `ename`. The keyboard string represents the string of the code sequence of keys pressed to enter the string, and is considered by Arabic Tools as a code page (see the `at_cptranslate` function described above). If `ename` does not hold a valid AT text object name, `nil` is returned.

(at_makestyle stname fontf hyt wf obl)

Creates an Arabic Tools text style. AT Text style names are stored in AutoCAD's style table with an 'AT-' prefix. This is meant for internal use only. When Arabic styles are created using the AT **MSTYLE** command, you do not have to start style names with this prefix. This applies to the styles you make using the (at_makestyle) function in AutoLISP. The 'AT-' prefix will be added by this function and stored in AutoCAD's text style table. The arguments you pass to this function are:

stname	Style name
fontf	AT font file name
hyt	Height
wf	Width factor
obl	Oblique angle

If the function succeeds in creating the style it returns the style name passed in the first argument. The text style that was current in AutoCAD before calling this function will remain, and will not be overridden by the new style created.

(at_style [stname])

This function sets or gets the current AT text style. If you pass an AT style name in the stname parameter, this style will become the current AT style. The style name should have been created before setting either using the AT StyleMaker dialog or the (at_makestyle) function described above. If the function fails to set the style name passed, it returns nil, otherwise it returns the style name passed if it succeeds. If you do not pass a style name, (at_style) will return the name of the current AT style. If the drawing has no AT styles the function will return a nil.

(at_stylelist)

Returns a list of all Arabic Tools text styles stored in AutoCAD's text style table. Arabic Tools text styles are flagged with an 'AT-' prefix. However, all style names returned will not have this prefix, as it is meant for the internal use of the Arabic Tools only. If no AT styles are available the function will return nil. This can only happen when

the user purges the default Arabization Tools text style which AT creates upon its initialization. The styles are returned in their creation order, which is how AutoCAD saves styles in its text style table.

(at_text txtstr sp hyt rot)

This function creates an AT text entity using the arguments passed, which are:

txtstr	Text
sp	Start point
hyt	Height
rot	Rotation
style	Text style

The text string passed in the first argument has to be an Arabic string in the current code page. The style argument has to be a name of an Arabic Tools text styles previously created by the **MSTYLE** command or the (at_makestyle) function. (at_text) does not change the current style setting in AutoCAD or the Arabic Tools. If the function is successful it returns the string passed in txtstr, otherwise it returns nil.

(at_textinput sp hyt rot)

This function enables you to interactively insert an Arabic text object using the arguments passed which are as follows:

sp	Start point
hyt	Height
rot	Rotation angle
style	Text style

The user will be able to input text as in using the **ARABIC** command. The square cursor appears in the position specified by the sp argument. Only one text line can be keyed in. The function will terminate when the user presses enter. The text entered will be deleted from the drawing. You can recreate it using the (at_text) function. As in (at_text) described above, this function does not reset the text style for AutoCAD or the Arabic Tools. If

`(at_textinput)` succeeds it returns the string entered by the user, otherwise it returns `nil`.

Part II

VisualSHX™

2.1 Overview

VisualSHX is an additional plug-in that ships with the AutoCAD Arabization Tools. It provides an easy means for designing Arabic font files in the AutoCAD SHX format. In VisualSHX you use the AutoCAD editor to design your fonts. You have all the power of AutoCAD, in addition to a number of task specific tools that help you perfect your fonts.

2.2 Fonts in AutoCAD

AutoCAD supports a number of font formats, such as TrueType, Postscript, and a native SHX format that comes in a number of flavors. At present, the Arabization Tools uses only the standard SHX format.

SHX font files are the compiled form of shape (SHP) files, which represents the source code form of the fonts. A sample of code for drawing the question mark character is listed below as an example (see AutoCAD's Customization Guide for more details on SHP font coding):

```
*63,45,questionmark
2,8,(10,70),1,13,(32,30,-55),(28,-38,-59),(-17,-18,-33),
(-15,-24,36),(0,0),2,8,(-3,-10),1,13,(3,0,0),(1,-10,0),
(-5,0,0),(1,10,0),(0,0),2,8,(45,-10),0
```

This code represents a series of commands that AutoCAD uses to direct a virtual pen. This pen has two states, PEN UP and PEN DOWN. When the pen is down a line segment, whether straight or curved is drawn, while when the pen is up, it only moves to a new position, from where subsequent pen motion will start. Normally you use a text editor to code a font. Then it must be compiled to an SHX file using AutoCAD's **COMPILE** command before it can be used in AutoCAD text.

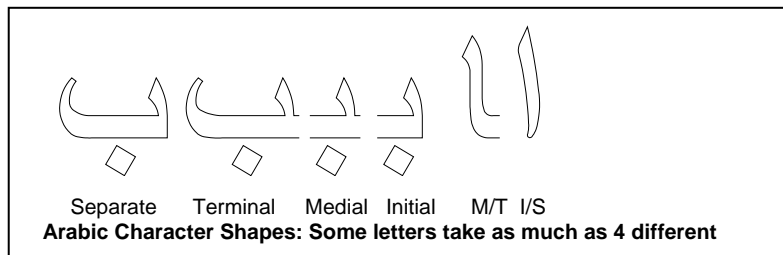
VisualSHX offers a much more convenient alternative for the design of AutoCAD fonts. With VisualSHX there is no coding at all, no guess work as how the character will look. You visually design your

characters using AutoCAD itself, and then use the VisualSHX compiler to generate the SHP code and compile it to SHX format. VisualSHX also allows you test your fonts on the fly, so that you can make any adjustments needed.

In VisualSHX each character is designed separately using polylines. The character is then saved in the drawing. The SHP code generator reads your design and generates SHP code from it. The SHP generator checks for consistency and warns you of any error. The SHP generator processes the polylines in each shape in their drawing order. POLYLINE segments represent displacements with PEN DOWN status. The distance between the end of one polyline and the beginning of the next represents a displacement with a PEN UP status. To bring the drawing pen back to the starting position of the next character in a text line, you can use a POINT object. The SHP generator will interpret this last point as a displacement from the end of the last polyline object to the point in a PEN UP status. Not every character should end with a point object. Polyines of concatenating Arabic characters extend to the starting point of the next character, and a point object to locate the pen at this point will not be necessary (you may still use it if you wish, but this will unnecessarily add to the size of the font file).

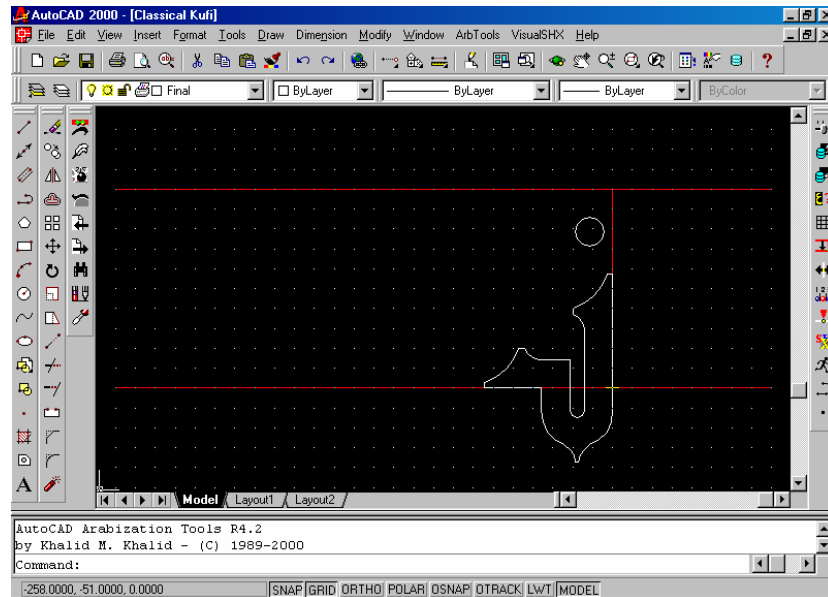
2.3 The Arabic Font Set

Arabic characters vary in shape, according to their position in a word. Most characters in an Arabic font have four shapes; initial, medial, terminal and separate. An Arabic font should, therefore, comprise all the possible shapes of the Arabic alphabet, in addition to number and punctuation marks shapes. In the font, each character shape is assigned a special code number. VisualSHX gives each character a unique name for identification. This name can be modified according to your preference.



Another special aspect of Arabic characters is that they connect at their baseline. The baseline 'thickness' should be uniform for all concatenating characters in the font set in order that they connect properly in text.

2.4 Designing an Arabic Font



VisualSHX provides a drawing template that represents a form or a sheet on which characters are designed. On this **design sheet** a structuring grid is laid out. The most important grid line is the baseline that passes through the origin point (0,0). The origin point represents the starting point AutoCAD uses for drawing each character when it is rendered in text. Grid lines are placed on separate locked layer named "Grid". You can add other grid lines or 'template' lines that help you in shaping your characters.

SHX fonts are integer based. All pen motion displacement specification should be represented by an integer value in the range of **-127** to **127**. For this reason, you will find that the snap setting in the template drawing is set to the value of 1. You can release the snap setting if you want. VisualSHX rounds all displacement values to the nearest integer, and checks for undue displacement values. However, it is recommended that you keep this setting on as the shape you design will exactly match the shape that will be generated by SHX. The rounding of values can cause vertex shifts, and may alter the shape you design.

The template drawing also has a number of preset views. The initial view displays the design grid. Another view is specified for test-running the font. The last basic view displays a matrix of designed characters. Clicking on a cell in the matrix will **get** the character on the design grid for editing if the character has been designed or **sets** the code for designing a new character.

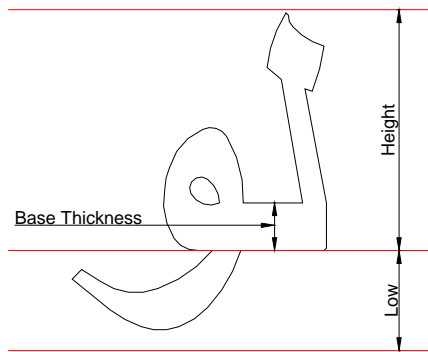
The characters you design will be stored in the drawing as blocks. In VisualSHX you do not name the blocks directly. Blocks are automatically named according to the code you set for designing. The setting of the character code can be done by a number of commands, such as SET, GET, and others.

You can choose any appropriate name to store the drawing, which represents a project file that contains all the settings and data required to generate the final font file. The name of the font file that will be generated from this drawing is also stored in the drawing itself and can be same as the drawing file name, or any other name of your choice.

The major steps involved in designing an Arabic font are:

1. Setting up the font's basic parameters. This includes setting the font name, font description (both Arabic and English), font height, font low, and base thickness. To set these variables, you access the Font Settings dialog by selecting the command from the menu, the toolbar, or typing FONTSET at the command line.

2. Designing the individual characters. The Arabic font set used in the Arabization Tools has 123 character shapes, in addition to 31 punctuation marks and 10 numbers. You can design the characters in any order you like, and you don't even have to design a full set to have the font running. However, if you use an incomplete font later, and one of the missing characters is used, AutoCAD will substitute that character with a question mark. To design the individual characters you can use any of AutoCAD drawing and editing commands, in addition to a number of specific tools provided by VisualSHX. After designing individual characters you save them in the drawing using the PUT command. You can retrieve saved characters and edit them as you wish. VisualSHX allows you to test run the font at any stage. In fact, you can run the font without even designing a single character.
3. After designing the characters, you compile the font to AutoCAD's SHX format. VisualSHX will translate the character you designed to the SHP source code first, and then compile this code to an SHX font file which can be used by AutoCAD and the Arabization Tools.



1. Font Settings

There are a number of basic font parameters that have to be set for each font. These include:

- Font name: Which is your description of the font. You may also wish to include copyright information in the description.
- Arabic title: This title appears in the Style Maker dialog.
- Font Height: The font height is used for basic scaling and formatting of text. VisualSHX uses a default font height of 100. Using a standard height among your fonts facilitates the exchange of characters between the fonts you design. This is particularly useful in the case of punctuation marks.
- Font Low: is the distance which the character strokes extend below the base line. The default in VisualSHX is 30. This value is not directly supported by the Arabization Tools but is required by AutoCAD for internal setting.
- Base Thickness: This is an Arabic Tools specific value which is stored in ASCII code No. 23. Arabic characters connect in text. The horizontal connecting line at the bottom is the character base. In outline fonts, this base has thickness. You define this thickness by setting this variable. The default value for base thickness is 10, but you will almost always have to set this value according to the font you design.

The values above can be set anytime before compilation. It is good practice however to set them at the beginning of a project. Even though these values have relation to the grid lines in the design sheet, changing them will not affect the position of the grid lines in the design sheet. Only the compiler uses these values to generate a font file.

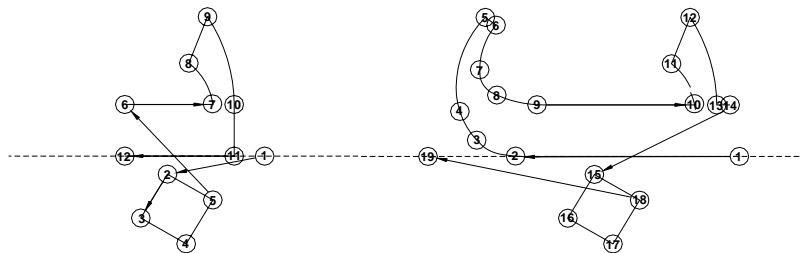
To set these values you use the Font Settings dialog box. You can access this dialog through the VisualSHX menu, the toolbar, or by issuing **FONTSET** at the AutoCAD command line. The dialog entries display the default settings which you may change as required. You can change the settings any time you wish. The values set before compilation will be those written to the SHX file.

2. Designing Arabic Characters

The following points should be observed in the design of each character:

- The character design is formed of one or more polyline or point objects, drawn on layer "Final".
- All displacements should be within the limit of +/-127. This means that the distance components (x, y values) between:
 - i) vertices of a polyline
 - ii) the origin point and the first polyline or point
 - iii) the end point of a polyline and the start point of the next polyline or point should all be within this +/-127 limit.
- All displacements should preferably have a full unit value. AutoCAD fonts are integer based, and VisualSHX rounds fraction values, if you use them in displacements, to the nearest integer value. This rounding slightly alters the shape of the character generated in the font.
- Curved segments should not encompass more than a semi-circle. You can divide curve segments that are larger than a semi-circle into a number of sub-segments.
- The generation of the character in actual text follows the drawing order and direction of the polylines and points. The generation start point is the origin point (0, 0).

The following figure illustrates the sequence AutoCAD performs to render a character. To attain this sequence, the polylines and points should be drawn accordingly.



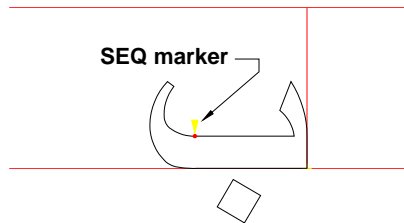
The VisualSHX compiler will check the design consistency of each character, and will generate warning and error messages if the design does not comply with the rules stated above. VisualSHX also provides a number of tools for checking the drawing order of a character. Combined with the drawing and editing tools you already use in AutoCAD, these tools offer you great control over the design of consistent characters.

To design a specific character, you first have to set its code. You can directly set the code using the **SET** command from the menu, toolbar, or the command line. The dialog box that appears informs you of the current code and allows you to input the new code. Alternatively, you can use the **GET** command. This command is used to retrieve a character that has already been designed. The command will retrieve the character, if found, and place it on the design sheet, ready for editing. If the character requested is not found, the command will simply set the code you provided for the design of the next character. After you finish designing or editing the character, you can add it to the font set using the PUT command. If the character already exists, you will be asked whether you want to overwrite it or not. If you choose to save the character design with a new code, use the **SET** command.

To check the status of the characters designed you can use the **DSTAT** (design status) command. The dialog box that appears gives you the option to list either the characters designed or those that remain to be designed. If you double-click on an entry in the designed character list the character will be retrieved and placed on the design sheet as if you had used the **GET** command. Alternately, if you double-click on an entry in the characters remaining list, the character code will be set as if you had issued a **SET** command.

Designed characters may be used in the design of other characters. Retrieve a character by issuing the **PLACE** command. The character shape you wish to place will be fetched and positioned on the base line. You can slide it left and right to position it in the exact location.

To check the drawing sequence of a character, use the **SEQ** command. A yellow marker will start from the origin point and trace the shape of the character as you press the space bar. You can later choose to reverse the drawing direction of a polyline using the **POLREV** command. To change the drawing order of the polylines and points, use the **ORDER** command. Start with the object you want to be drawn first and subsequently select objects in the order you want them to be drawn with.



VisualSHX allows you to run the font at any stage using the **RUN** command. The font that will run is the output of the last compilation performed (see Compilation below). If VisualSHX cannot find a compiled font that matches the file name specified in the font settings, it will prompt you for compilation. After you compile the font, you can use the **RUN** command to test-run the font.

3. Font Compilation

The purpose of compiling a font is to generate a font file which AutoCAD uses for rendering text in drawings. VisualSHX will generate an SHX font file from the current drawing which contains blocks that represent the characters you designed. The compiler starts by generating the source SHP code. It checks the design of each character, and will report inconsistencies. Warnings appear

when the design can be compiled to an SHX form but unwanted effects may result, such as text with a shifting baseline resulting from the print head not returning to the proper position after the drawing of a character, or curves that are larger than a semi-circle. Errors are reported when the characters have displacement components that exceed the +/-127 limit.

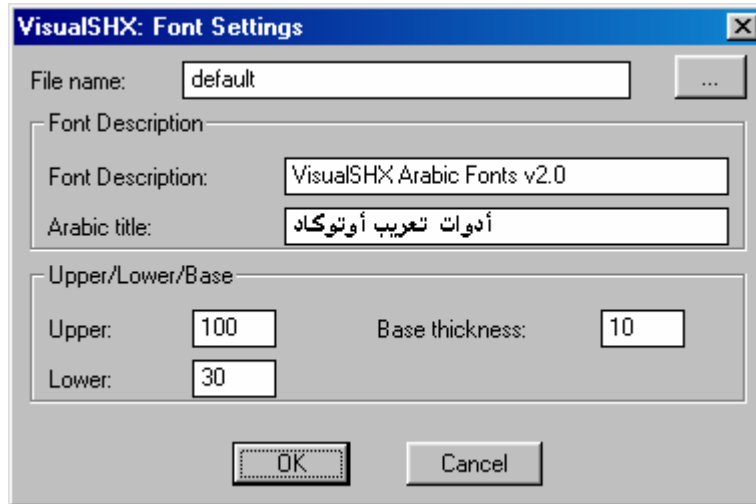
Compilation can be done at any stage. The resulting font file will have the basic information that enables AutoCAD to load and run the font. If you use the font with characters that have not been designed, AutoCAD substitutes them with the question mark symbol defined in the font. VisualSHX's basic font file can be run without designing a single character.

The name of the font to be generated can be set in the compiler dialog box. Use the edit box at the top to designate a new name for the output file. To start the compilation process, click the Compile button. If the compilation is successful and an SHX file is generated, the **Run** check box will be activated. If you check this box before pressing the Done button, you will be able to test run the font. The drawing will zoom to reveal the testing grid lines. Only one line of text is allowed. Upon pressing Enter text entry will terminate and you return to the view prior to the test run.

To use the generated SHX font file with the Arabization Tools, use the ADDFONT command provided with the Arabization Tools. (Note: this command is not available in the demo version).

2.5 VisualSHX Command Reference

Command: **FONTSET**

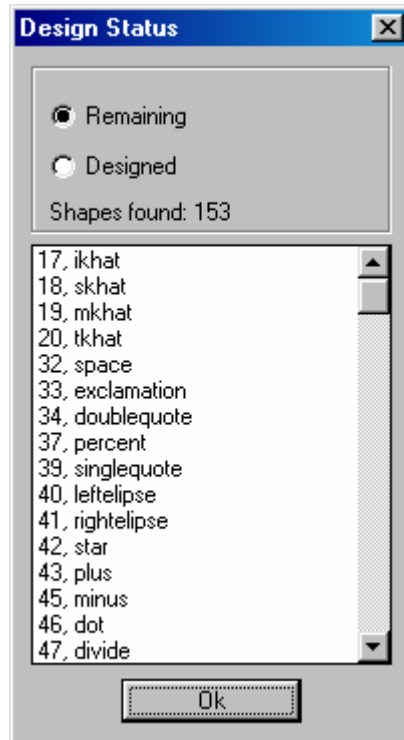


This command brings up the Font Settings dialog box. The values you set in this dialog are only used by the compiler for the generation of the font file. They do not affect the grid settings in the design sheet, or character design in any way. The font settings are:

1. Font file name: The name that will be given to the generated SHP and SHX font files. You do not have to specify an extension. Windows file naming conventions apply.
2. Font Description: The name of the font, version, copyrights, and any other title information you wish to give to your design. Any text is suitable up to the limit of 79 characters. This text will be stored in the generated SHX file.

3. Arabic Title: Change the keyboard to Arabic, and specify the Arabic title of the font. It will be displayed in the Arabization Tools' Style Maker dialog box.
4. Font Height: The height of the font that extends above the base line.
5. Font Low: The distance that the font extends below the base line. This value is not currently supported by the Arabization Tools, but is required by AutoCAD in the font file.
6. Base Thickness: This value is specific to the Arabization Tools. Character base is where Arabic characters concatenate. This value is used in curve fitting to generate catenation segments of the same thickness as the character base line.
The values set are saved in the drawing. You can reset them any time later. When you compile a font the current values saved will be those written to the font file.

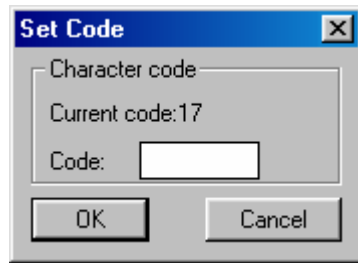
Command: **DSTAT**



To display the design status use the **DSTAT** command. The character codes and names listed in the dialog box that appears depends on the whether you check the 'Remaining' or the 'Designed' radio buttons at the top of the dialog. The 'Remaining' radio button is checked by default. All the character shapes that remain to be designed are listed. Double clicking on a list entry will set the code for that character. When you later save the character shape using the **PUT** command, the character will be saved with the code

selected from the list. If you otherwise check the 'Designed' radio button, the list will display all the characters saved. Selecting a designed character will **get** the character and place it on the design sheet for editing. When you **put** a character that already exists, the new design of the character will overwrite the existing design. You will be prompted to confirm the overwriting of any existing character (see the PUT command below).

Command: **SET**



This command sets the current character code. When you save the character using the **PUT** command, the character will be saved using the code set. The **SET** command brings up a small dialog box which displays the current code setting. You can change the code by typing the new code in the edit box. Valid code values range between 14 and 255. However, not all these values are used by the Arabization Tools. See the Arabization Tools fonts table in Appendix X for the list of valid Arabization codes. The **DSTAT** and **FONTMATX** commands can also be used to set the character shape codes.

Command: **GET**



Gets a saved character. A dialog similar to the **SET** dialog appears. Type the code of the character shape you want to get in the edit box. If the character shape has not been designed, the code you supply

will be the code set for saving the next character. Otherwise, the command will get the character shape and places it on the design sheet for editing. (see the **SET** command above for valid **GET** values)

Command: **PUT**



PUT saves the design on the design sheet. Any polyline or point drawn on layer "Final" will be saved as a character shape using the current code which has been set using any of the **SET**, **GET**, **DSTAT**, or **FONTMATX** commands. **PUT** does not validate the design for consistency. This is done on compilation (see the **VCOMPILE** command below). A dialog will appear after you issue this command which displays the current code and the associated character shape name. You can reset the code if you wish. If a character shape with the same code exists in the drawing, you will be prompted to confirm overwrite. Character shapes saved will be displayed in the Font Matrix (see **FONTMATX** command below).

Command: **FONTMATX**



This command will display the Font Matrix. This matrix has 16x16 cells. Designed characters will appear in the cell that represents its ASCII code. Clicking inside a cell will get the character if it is designed or set the code of the character for design. The view also returns to the design sheet. You can zoom in on any area of the Font Matrix to have a better look at the contents of the cell. The last view you have of the Font Matrix will be saved. If you return to the Font Matrix later, you will be in the same you view you last left the Matrix. To display the whole the matrix you can type **F** for **FULL** at the command prompt.

```
Command: fontmatx
Full view/<Click on character to get>:
```

Command: **SEQ**



The **SEQ** command is useful for checking the drawing order of your design. A marker appears on the origin point, which is the starting point for drawing every character shape. If you press the Space Bar or a mouse button the marker will jump to the next displacement to a polyline vertex or a point location. The marker disappears when all the displacements are covered. It is a good practice to check the sequence of character shape drawing before saving it with the **PUT** command.

Command: **POLREV**



The vertex sequence of a polyline can be reversed with this command. In VisualSHX, you can freely design character shapes, following the few restrictions mentioned in “Designing the Characters” above. If the drawing direction of a polyline is not suitable, you can use this command to reverse it.

```
Command: POLREV  
Select polyline:
```

This is especially useful when you want to bring the virtual drawing pen to the starting point of the next character.

Command: **ORDER**



With this command you can control the drawing order of objects defining the character shape. Object drawing order is important because it will be the same order that the virtual pen will follow when it draws your font in AutoCAD. At times you will want to bring the pen to a final location after it has drawn all the other parts of a character

shape. You can set the order of drawing by selecting the objects in the sequence you want them to be drawn with.

```
Command: order
Select object 1:
Select object 2:
...
```

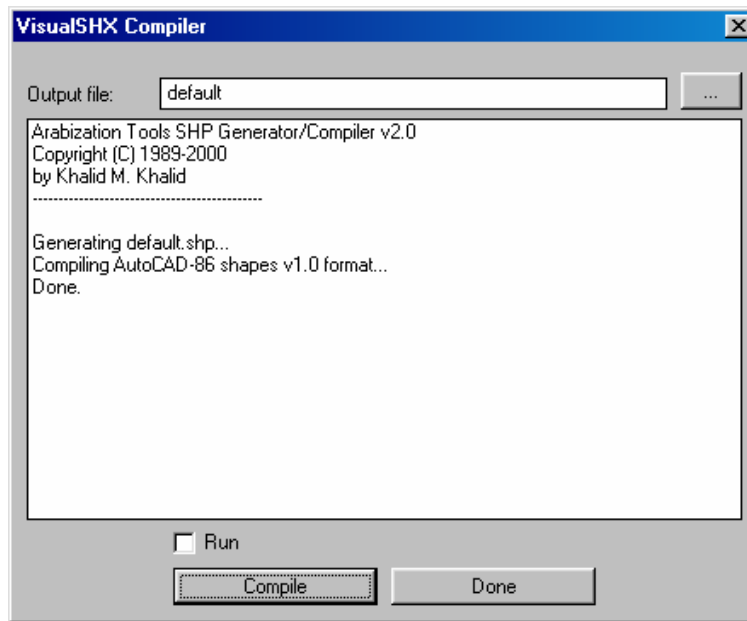
The command will continue until you right click the pointing device or make an empty selection, after which the objects will be reordered according to the sequence of your selection. You can check the new order using the **SEQ** command described above.

Command: **PLACE**



This command allows you to retrieve a designed character shape and place it on the design sheet. It does not set the character code. This is useful for using a character shape in the design of a new character. The command will display the same dialog box used in the **DSTAT** command described above. However, the **Remaining** radio button is de-activated. All designed character shapes are listed in the dialog and you can select any of them to place on the design sheet. The selected character will appear sliding on the base line, and you can pin its final position with a click.

Command: **VCOMPILE**



Use the **VCOMPILE** command to generate font files. The dialog that appears displays the name of the file that was set in the **FONTSET** command. You may change the name if you wish. The new name will be saved for subsequent compilations. Selecting the Compile button will start the compiler. Messages will be displayed in the dialog, and may include warning and error messages. Font files will be generated even when warning messages appear. However, if error messages appear, the SHX font file will not be generated. You will have to fix the characters that are generating errors before you can have a successful compilation.

The 'Run' check box below the message window in the VisualSHX compiler dialog will be activated upon the successful generation of a

font file. If you check this box, you will be able to test run the font upon selecting the 'Done' button (see **RUN** command below).

Command: **RUN**



This command enables you to test run your font design. You can only run fonts that were compiled at least once. The command will run the font generated by last compilation. Even if you add characters and 'put' them in the drawing, they will not appear in the test run until you perform a fresh compilation.

The display will zoom to reveal the test bench. The square cursor appears, and you can enter text just as you do in the Arabization Tools. Text entry is limited to one line only. If you use characters that have not been designed, the question mark appears in their place. To end the test, press Enter or Esc. The view will zoom back to the setting prior to the **RUN** command.

Appendix I

Arabization Tools Font Table

Code	Symbol Name	Symbol	Code	Symbol Name	Symbol
17	ikasheeda		167	mkha	
18	skasheeda		168	tkha	
19	mkasheeda		169	idal	
20	tkasheeda		170	mdal	
24	decimalcomma	,	171	idhal	
32	space		172	mdhal	
33	exclamation	!	173	ira	
34	doublequote	"	174	mra	
35	hash	#	175	iza	
36	dollar	\$	181	isheen	
37	percent	%	182	ssheen	
38	ampersand	&	183	msheen	
39	singlequote	'	184	tsheen	
40	leftelipse	(185	isad	
41	rightelipse)	186	ssad	
42	star	*	187	msad	
43	plus	+	188	tsad	
45	minus	-	189	idhad	
46	dot	.	190	sdhad	
47	divide	/	191	mdhad	
48	zero	0	192	tdhad	
49	one	1	193	itta	
50	two	2	194	stta	

Code	Symbol Name	Symbol	Code	Symbol Name	Symbol
51	three	3	195	mtta	
52	four	4	196	ttta	
53	five	5	197	ittha	
54	six	6	198	sttha	
55	seven	7	199	mttha	
56	eight	8	200	tttha	
57	nine	9	201	iaain	
58	colon	:	202	sain	
59	semicolon		203	main	
60	leftangle	<	204	tain	
61	equal	=	205	ighain	
62	rightangle	>	206	sghain	
63	question		207	mghain	
64	atsign	@	208	tghain	
91	leftbracket	[209	ifa	
92	backslash	\	210	sfa	
93	rightbracket]	211	mfa	
94	carat	^	212	tfa	
95	underscore	_	213	iquaf	
123	leftbrace	{	214	squaf	
124	bar		215	mquaf	
125	rightbrace	}	216	tquaf	
127	tilde	~	217	ikaf	
128	hamza		218	skaf	
129	ihamz		219	mkaf	
130	mhamz		220	tkaf	
131	ilowhamz		221	ilamalefuphamz	

Code	Symbol Name	Symb ol	Code	Symbol Name	Symbol
132	mlohamz		222	mlamalefuphamz	
133	ikursi		223	ilamaleflowhamz	
134	skursi		224	mlamaleflowhamz	
135	mkursi		225	ilamalef	
136	tkursi		226	mlamalef	
137	iuh		227	ilam	
138	muh		228	slam	
139	iaa		229	mlam	
140	maa		230	tlam	
141	ialef		231	imeem	
142	malef		232	smeem	
143	iba		233	mmeem	
144	sba		234	tmeem	
145	mba		235	inoon	
146	tba		236	snoon	
147	ita		237	mnoon	
148	sta		238	tnoon	
149	mta		239	iha	
150	tta		240	sha	
151	imarbouta		241	mha	
152	mmarbouta		242	tha	
153	itha		243	iwaw	
154	stha		244	mwaw	
155	mtha		245	iya	
156	ttha		246	sya	
157	ijeem		247	mya	
158	sjeem		248	tya	

Code	Symbol Name	Symbol	Code	Symbol Name	Symbol
159	mjeem		249	imaqsura	
160	tjeem		250	tmaqsura	
161	ihha		251	mza	
162	shha		252	iseen	
163	mhha		253	sseen	
164	thha		254	mseen	
165	ikha		255	tseen	
166	Skha				

**Appendix II: Arabic Tools: Available Fonts
(Registered version)**

Appendix III: Installation Troubleshooting

Arabic Tools relies on AutoCAD's library path to find its component files. If the Windows®-base setup program provided with this release fails to install the Arabic Tools correctly, you can try to install it manually. This involves the following steps:

1. Copying ArbTools.vlx, ArbTools.arx, VSHX.vlx, ArbTools.ini to AutoCAD's main directory
2. Copying ArbTools.dcl, ArbTools.mns, ArbTools.mnc, ArbTools.mnr, ArbTools.mnl, CodePage.asc, CharCode.tbl to the AutoCAD\Support directory.
3. Copying ArbTools.shx, Naskh.shx to the AutoCAD\Fonts directory.
4. Within AutoCAD, use the MENULOAD command to install the Arabic Tools interface. From the Menu Group tab load the ArbTools.mnc, or ArbTools.mns files. The Arabic Tools toolbars will appear on the AutoCAD screen. From the Menu Bar tab of this dialog place the Arabic Tools menus in the place of your choice on the menu bar.

This will be enough to have the Arabic Tools up and running, as AutoCAD will automatically load the ArbTools.mnl, and this file contains directives to load the Arabic Tools and the VisualSHX engines.

Appendix IV: Arabic Keyboard Layout

Appendix V: Contact Information

For inquiries, comments, and suggestions please contact:

Khalid M. Khalid
Adhamia, Sector 318, Street No. 26, H.No. 71
Baghdad
Iraq

Tel: 00964-1-4227715