#[1]$[2]K[3]Uniput-- Input of Unicode characters

Quick Start:

0. (Make sure that the UCLUCIDA.TTF font file has been installed.)

1. Use the "File, Open" menu to create a new file.

2. Mouse click on characters displayed in the main window.   They will be written to the file.

3. Scroll through character point ranges with the horizontal scroll bar.

4. Create additional windows as necessary.

5. Close the file, and view the contents with UNIPAD.

More Information:

Overview   Advanced Features

Font Support   References

Known problems:

1. [Minor]   The "Font Height" and the "Characters per row" menus work to change the behavior of the active window.   However, they do not update to show the current status of the active window.

2. [Minor]   Changing the status of the "Show display range names" menu works correctly.   However, the change will not immediately be propagated to all of the child windows if there are more than one.

#⁴$⁵K⁶Overview

The Uniput program was written to provide an interim input method for entering unicode characters.   The UNIPAD application distributed with the October beta of Windows/NT is able to properly display left-to-right unicode strings.   However, it allows no practical input method.   That is the reason for Uniput.   In a sense, it serves as a crude input method for the UNIPAD application.   Notice that this is an interim solution.   It is not part of Microsoft's future direction for input methods.

The status bar currently has two fields.   The left most field displays the name of the file that the data is currently being written to.   The other field shows the stream of characters which have resulted from the squares chosen so far.   Notice that this is actually a static text window with the font set to be "Lucida Sans Unicode."   As such, it is one way to see how Windows/NT will write a unicode string.   For example, diacritical (non spacing) marks are currently handled correctly.

#[7] $[8]K[9]Advanced Features

By using the multiple document interface features fully, it is possible to tailor the application somewhat to the user's input needs.   For example, it is easy to create two windows (use the "Window, New" menu), one with the desired character range, and the other with diacritical marks.   Then, by tiling the two windows, it is easy to quickly pick characters, and modify them as needed with the diacritical marks.   It is also possible to create multiple windows, and scroll them to different character ranges independently.

The "Options" menu allows the user to set four preferences.   The later two, the font size, and the number of characters per line are self explanatory.   They may be different for each child window.   The first two preferences are Boolean, and the application starts with them being checked ON.   The first controls whether the hexadecimal value for the code point is displayed in the square with the character glyph.   The second affects the title of the child window.   With the item checked on, the title attempts to display the name of the character point range that the displayed characters are from.   Both of these Boolean options apply to ALL of the child windows in the application.

#[10]$[11]K[12]Font Support

In order for the Uniput program to work, it is necessary to have a font which covers a significant number of unicode code points.   The "Lucida Sans Unicode" font has about eleven hundred as of the October beta.   When the application first begins, it tries to create a logical font with that face name.   It then counts the number of character ranges in the realized font.   If it is less than some threshold (currently 30), the user is warned with a message box.   In this case, the user should quit the application, install the UCLUCIDA.TTF font with the control panel, and try the application again.   This same procedure is true for the NTFONTS and SHOWUNI samples distributed on the SDK.

[10]# Fonts

[11]$ Font Support

[12]K Font Support

#¹³$¹⁴K¹⁵References

Uniput was written by **Steve Firebaugh** in Microsoft's Developer Support.   Look for us in the **MSWIN32** forum of Compuserve.   Questions or bug reports related to this program should be posted to section 15/UNICODE/NLS.

Understanding the 'cmap' table is most easily possible with a specification for TrueType font files.   This is available from many sources including the following:

Microsoft **Software Library**.   (Available on Compuserve by "GO MSL" then search for keyword **TTSPEC1**.)

Anonymous ftp on **ftp.uu.net** in **vendor/microsoft/TrueType-Info.**

To learn all about unicode, including the "names" given to the different code point ranges, the best reference is the following two volume book available from Addison-Wesley:

The Unicode Standard

World-wide character encoding

Version 1.0

ISBN 0-201-56788-1