

Content

1. Introduction	1
1.1. SC UniPad - The Unicode Plain Text Editor	1
1.2. Feedback & Contact	1
2. Getting Started	2
2.1. Overview	2
2.2. Opening a File	2
2.3. Creating a New Document	2
2.4. Editing a Document	3
2.5. Inserting Characters	3
2.6. Converting Files	4
3. Basic Concepts	5
3.1. Overview	5
3.2. UniPad Font	5
3.3. Virtual Keyboards	5
3.4. Handling of Combining Characters	6
3.5. Handling of Byte Order Mark	6
3.6. Handling of Line and Paragraph Separators	6
3.7. Handling of U+0000 and Noncharacters	7
3.8. Handling of Bidirectional Text	7
3.9. Glyph Shaping	8
4. Main Menu	9
4.1. File	9
4.1.1. New	9
4.1.2. Open	9
4.1.3. Save	9
4.1.4. Save As	9
4.1.5. Reload	10
4.1.6. Reload As	10
4.1.7. Page Setup	10
4.1.8. Print	10
4.1.9. Import	11
4.1.10. Export	11
4.1.11. Send	11
4.1.12. File Properties	11
4.1.13. Recent Files	12
4.1.14. Exit	12
4.2. Edit	12
4.2.1. Undo	12
4.2.2. Redo	13
4.2.3. Cut	13
4.2.4. Copy	13
4.2.5. Copy As	13
4.2.6. Paste	13

4.2.7. Paste As	14
4.2.8. Insert Character	14
4.2.9. Select All	14
4.2.10. Delete Line	14
4.2.11. Convert	14
4.2.11.1. Lowercase / Uppercase	14
4.2.11.2. Combine Characters	14
4.2.11.3. Decompose Combinations	15
4.2.11.4. UCN (\u) to Unicode	15
4.2.11.5. Unicode to UCN (\u)	15
4.2.11.6. XML/SGML Character Reference to Unicode	15
4.2.11.7. Unicode to XML/SGML Character Reference	16
4.2.11.8. XML/SGML Character Entity to Unicode	16
4.2.11.9. Unicode to XML/SGML Character Entities	16
4.3. Search	16
4.3.1. Find	16
4.3.2. Find Next	17
4.3.3. Replace	17
4.3.4. Replace Next	17
4.3.5. Go To	17
4.3.6. Find Bracket	17
4.3.7. Find Bracket Select	18
4.4. Options	18
4.4.1. Current Options	18
4.4.2. Font	18
4.4.3. Character Rendering	18
4.4.4. Document Direction	19
4.4.5. Bidirectional Text	19
4.4.6. Glyph Shaping	19
4.4.7. Toggle Keyboard	19
4.4.8. Toolbar	19
4.4.9. Default Options	20
4.4.10. Preferences	20
4.5. Keyboard	20
4.5.1. New (From Scratch)	20
4.5.2. New (Clone Current)	20
4.5.3. Load	21
4.5.4. Reload	21
4.5.5. Unload	21
4.5.6. Save	21
4.5.7. Open Definition File	22
4.6. Window	22
4.7. Help	22
5. Context Menus	23
5.1. Document	23
5.2. Virtual Keyboard	23
5.3. Character Map	24
5.4. Toolbars / Statusbar	24
5.5. Application	24
5.6. Edit Field	25

6. Dialogs	26
6.1. Open File	26
6.2. Save File	26
6.3. Import	27
6.4. Export	27
6.5. Page Setup	27
6.6. Copy As	28
6.7. Paste As	28
6.8. Go To	28
6.9. Find	28
6.10. Replace	29
6.11. Preferences	29
6.11.1. (Overview)	29
6.11.2. General	30
6.11.3. Files	30
6.11.4. Data	30
6.11.5. Errors	31
6.11.6. Editing	31
6.11.7. Keyboard	31
6.11.8. Miscellaneous	32
6.12. Options	33
6.12.1. (Overview)	33
6.12.2. File	33
6.12.3. Editing	34
6.12.4. Font	34
6.12.5. Character Rendering	34
6.12.5.1. (Overview)	34
6.12.5.2. Spaces	35
6.12.5.3. Control Codes	35
6.12.5.4. Formatting Characters	35
6.12.5.5. Unsupported Characters	36
6.12.5.6. Unassigned Codepoints	36
6.12.5.7. Unpaired Surrogates	37
6.12.5.8. Private Use Characters	37
6.12.6. Glyph Shaping	37
6.12.7. Bidirectionality	37
6.13. Reading Error	38
6.14. Assign Character Code	38
6.15. Conversion Options	38
6.16. Insert Character	39
6.17. Enter Registration Code	39
7. Toolbars / Tool Windows	41
7.1. Standard Toolbar	41
7.2. View Toolbar	41
7.3. Input Toolbar	41
7.4. Conversion Toolbar	42
7.5. Status Bar	42
7.6. Character Info Bar	42
7.7. Extended Character Info Bar	43
7.8. Character Map	43

7.9. Virtual Keyboard	44
8. Reference	45
8.1. Key Commands	45
8.1.1. (Overview)	45
8.1.2. Shortcuts	45
8.1.3. Navigation	46
8.1.4. Special Key Combinations	47
8.1.5. Aborting with [ESC]	47
8.2. Mouse Commands	47
8.3. Error Conditions	48
8.4. Command Line Parameters	48
8.5. Exit Codes	50
8.6. Limitations	50
9. Appendix	51
9.1. Character Blocks (Unicode 3.2)	51
9.1.1. (Index)	51
9.1.2. Alphabetic Presentation Forms	53
9.1.3. Arabic	53
9.1.4. Arabic Presentation Forms-A	53
9.1.5. Arabic Presentation Forms-B	54
9.1.6. Armenian	54
9.1.7. Arrows	54
9.1.8. Basic Latin	54
9.1.9. Bengali	55
9.1.10. Block Elements	55
9.1.11. Bopomofo	55
9.1.12. Bopomofo Extended	56
9.1.13. Box Drawing	56
9.1.14. Braille Patterns	56
9.1.15. Buhid	56
9.1.16. Byzantine Musical Symbols	57
9.1.17. Cherokee	57
9.1.18. CJK Compatibility	57
9.1.19. CJK Compatibility Forms	57
9.1.20. CJK Compatibility Ideographs	57
9.1.21. CJK Compatibility Ideographs Supplement	58
9.1.22. CJK Radicals Supplement	58
9.1.23. CJK Symbols and Punctuation	58
9.1.24. CJK Unified Ideographs	59
9.1.25. CJK Unified Ideographs Extension A	59
9.1.26. CJK Unified Ideographs Extension B	59
9.1.27. Combining Diacritical Marks	59
9.1.28. Combining Diacritical Marks for Symbols	60
9.1.29. Combining Half Marks	60
9.1.30. Control Pictures	60
9.1.31. Currency Symbols	60
9.1.32. Cyrillic	60
9.1.33. Cyrillic Supplementary	61
9.1.34. Dingbats	61
9.1.35. Deseret	61

9.1.36. Devanagari	61
9.1.37. Enclosed Alphanumerics	62
9.1.38. Enclosed CJK Letters and Months	62
9.1.39. Ethiopic	62
9.1.40. General Punctuation	62
9.1.41. Geometric Shapes	62
9.1.42. Georgian	63
9.1.43. Gothic	63
9.1.44. Greek and Coptic	63
9.1.45. Greek Extended	63
9.1.46. Gujarati	63
9.1.47. Gurmukhi	64
9.1.48. Halfwidth and Fullwidth Forms	64
9.1.49. Hangul Compatibility Jamo	64
9.1.50. Hangul Jamo	64
9.1.51. Hangul Syllables	65
9.1.52. Hanunoo	65
9.1.53. Hebrew	65
9.1.54. Hiragana	66
9.1.55. Ideographic Description Characters	66
9.1.56. IPA Extensions	66
9.1.57. Kanbun	66
9.1.58. KangXi Radicals	66
9.1.59. Kannada	67
9.1.60. Katakana	67
9.1.61. Katakana Phonetic Extensions	67
9.1.62. Khmer	68
9.1.63. Lao	68
9.1.64. Latin-1 Supplement	68
9.1.65. Latin Extended-A	68
9.1.66. Latin Extended-B	69
9.1.67. Latin Extended Additional	69
9.1.68. Letterlike Symbols	69
9.1.69. Malayalam	69
9.1.70. Mathematical Alphanumeric Symbols	69
9.1.71. Mathematical Operators	70
9.1.72. Miscellaneous Mathematical Symbols-A	70
9.1.73. Miscellaneous Mathematical Symbols-B	70
9.1.74. Miscellaneous Symbols	70
9.1.75. Miscellaneous Technical	70
9.1.76. Mongolian	71
9.1.77. Musical Symbols	71
9.1.78. Myanmar	71
9.1.79. Number Forms	71
9.1.80. Ogham	71
9.1.81. Old Italic	72
9.1.82. Optical Character Recognition	72
9.1.83. Oriya	72
9.1.84. Private Use Area	72
9.1.85. Runic	73
9.1.86. Sinhala	73
9.1.87. Small Form Variants	73
9.1.88. Spacing Modifier Letters	73
9.1.89. Specials	73
9.1.90. Superscripts and Subscripts	74
9.1.91. Supplemental Arrows-A	74

9.1.92. Supplemental Arrows-B	74
9.1.93. Supplemental Mathematical Operators	74
9.1.94. Supplementary Private Use Area-A	74
9.1.95. Supplementary Private Use Area-B	75
9.1.96. Syriac	75
9.1.97. Tagalog	75
9.1.98. Tagbanwa	75
9.1.99. Tags	76
9.1.100. Tamil	76
9.1.101. Telugu	76
9.1.102. Thaana	76
9.1.103. Thai	77
9.1.104. Tibetan	77
9.1.105. Unified Canadian Aboriginal Syllabics	77
9.1.106. Variation Selectors	77
9.1.107. Yi Syllables	78
9.1.108. Yi Radicals	78
9.2. Keyboards	78
9.2.1. (Index)	78
9.2.2. Albanian	79
9.2.3. Arabic	80
9.2.4. Armenian	80
9.2.5. Azeri	80
9.2.6. Bulgarian	80
9.2.7. Bulgarian (Latin)	80
9.2.8. Byelorussian	81
9.2.9. Cherokee	81
9.2.10. Croatian	81
9.2.11. Czech	81
9.2.12. Danish	81
9.2.13. Divehi	82
9.2.14. Dutch	82
9.2.15. English (British)	82
9.2.16. English (US International)	82
9.2.17. English (US)	82
9.2.18. English (US, Dvorak)	83
9.2.19. Esperanto	83
9.2.20. Estonian	83
9.2.21. Finnish	83
9.2.22. French (Belgian)	83
9.2.23. French (Standard)	84
9.2.24. Gaelic	84
9.2.25. Georgian	84
9.2.26. German (Standard)	84
9.2.27. German (Swiss)	84
9.2.28. Greek	85
9.2.29. Hawaiian	85
9.2.30. Hebrew	85
9.2.31. Hindi	85
9.2.32. Hungarian	85
9.2.33. Icelandic	86
9.2.34. Inuktitut	86
9.2.35. ISO 9995 Complementary	86
9.2.36. ISO 9995 Secondary	86
9.2.37. ISO 9995 Unified	87

9.2.38. Italian	87
9.2.39. Japanese (Kana only)	87
9.2.40. Kazakh	87
9.2.41. Kyrgyz	88
9.2.42. Latin American	88
9.2.43. Latvian	88
9.2.44. Lithuanian	88
9.2.45. Macedonian	89
9.2.46. Mongolian (Cyrillic)	89
9.2.47. Norwegian	89
9.2.48. Ogham	89
9.2.49. Persian (ISIRI 2901)	89
9.2.50. Polish	90
9.2.51. Portuguese	90
9.2.52. Punjabi	90
9.2.53. Romanian	90
9.2.54. Russian	90
9.2.55. Serbian (Latin)	91
9.2.56. Slovak	91
9.2.57. Spanish	91
9.2.58. Swedish	91
9.2.59. Syriac	91
9.2.60. Tatar	92
9.2.61. Thai	92
9.2.62. Turkish (F-Type)	92
9.2.63. Turkish (Q-Type)	92
9.2.64. Ukrainian	92
9.2.65. Urdu	93
9.2.66. Uzbek	93
9.2.67. Vietnamese	93
9.3. Formats	93
9.4. Character Sets	93
9.5. Additional Information	94
9.6. Manual Installation	94
9.7. Mirrored Characters	94
9.8. General Command Line Syntax	95
Index	97

1. Introduction

1.1. SC UniPad - The Unicode Plain Text Editor

SC UniPad is a Unicode text editor for the Microsoft Windows operating systems.

It allows you to edit Unicode plain text and to save and load it in different formats. It provides a lot of features to help you working with Unicode text.

Where to go from here...

- Getting Started
- Basic Concepts
- What's New In This Version?

1.2. Feedback & Contact

Feedback

We need your help! If you have comments, suggestions for improvements, features you would like to see, bug reports or anything else regarding SC UniPad please use the online contact form on **www.unipad.org**.

UniPad homepage

<http://www.unipad.org>

Mailing list

If you want to keep informed about new versions, bug fixes and so on, you can subscribe to the UniPad mailing list:
<http://www.unipad.org/feedback>.

2. Getting Started

2.1. Overview

Overview

UniPad generally works like any other text editor and is easy to use. However, there also some differences to other editors. The following topics give an overview of the most common tasks and their particular features.

- Opening a File – explains how to open Unicode text files and informs about the various formats.
- Creating a New File – explains how to create new Unicode text files.
- Editing a File – explains how to edit a Unicode text files and informs about special behaviour.
- Inserting Characters – informs about the various input methods.
- Converting Files – discusses import and export functions.

See Also

For more detailed description of certain Unicode-specific features please read the Basic Concepts.

2.2. Opening a File

Ways to open a file

You can open a file by

- selecting the menu command File > Open or the corresponding toolbar button,
- dragging a file from Windows Explorer and drop it onto UniPad,
- selecting a file from the history via File > Recent Files,
- specifying a name on the command line.

File, buffer, document

When opening a file, UniPad does not work directly on the file, rather it loads the file into a buffer in main memory. Most editors work this way. The content of the buffer is presented to you in a document window. A file that is loaded into the buffer is usually referred to as "document" to distinguish it from the physical file that remains unchanged on the harddisk until you save the document. UniPad can open all files irrespective of their extensions.

Formats

Unicode text files can come in various formats. The most common format nowadays is UTF-8. However, UniPad supports other formats as well. Usually UniPad is able to detect the format of a file automatically when you open it. When you open a file via menu File > Open you can also choose from the other supported formats. This is necessary if auto-detection fails or if you are sure about the format. If you open a file from the history the remembered format will be used (unless you have turned this behaviour off).

See Also

- Creating a New Document
- Converting Files
- How to abort file loading with [ESC] key

2.3. Creating a New Document

Ways to create a new document

You can create a new document by

- selecting the menu command File > New
- or by pressing the corresponding toolbar button.

If you start UniPad, a new document will be created by default (unless you have turned off this behaviour).

Format

New documents will created using the current default format. The factory setting for the default format is UTF-8, but you can change it to any of the other supported formats via the Options dialog by selecting the menu command Options > Default Options. After creating a document you can change the format of this document anytime via the Options dialog by selecting the menu command Options > Current Options.

2.4. Editing a Document

Operations

UniPad offers all common editing functions. You can

- navigate and select text with the cursor keys,
- inserting characters using different input methods,
- cut, copy and paste using the clipboard,
- perform various conversions on selected text,
- search for text and replace text.

Navigation and selection

- You may move the cursor and select text either using the mouse or by key commands. You can also go to a specific line.
- Before you work with bidirectional text you should become familiar with the peculiarities of navigating within bidirectional text.
- Some characters (control characters and formatting characters) can be invisible. When you move the cursor through invisible characters, the logical cursor position changes (which you can see on the status bar) but the physical cursor position does not change. If you feel irritated by this behaviour you should make all control characters and formatting characters visible.
- Combining marks are visible as separate characters.

Clipboard operations

UniPad provides the common clipboard operations Cut, Copy, Paste. Additionally you can use the operation Copy As to convert the Unicode text to a specific format prior to copying it to the clipboard. Or you can use the operation Paste As to convert text in a specific format prior to pasting it from the clipboard into UniPad.

Conversions

UniPad provides various conversion functions for a variety of tasks:

- uppercase and lowercase conversion,
- conversion of text to \u sequences,
- conversion of SGML character references and character entities (as used in HTML, for instance),
- combining and decomposing characters.

See Also

- Insert Characters
- Basic Concepts

2.5. Inserting Characters

Input methods

Inserting characters into a document is obviously the most important function an editor can provide. UniPad provides various input methods:

- default system keyboard,
- additionally installed system keyboards,
- native IMEs on East Asian systems or installable IMEs on non-Asian systems,
- built-in UniPad virtual keyboard,
- installable user-defined virtual keyboard,
- character map,
- hexadecimal input,
- certain third party keyboard tools.

System keyboard / IMEs

- If the keyboard combobox shows "(System)", UniPad receives input from the current system keyboard:



- Usually the system keyboard is the one that has been installed as default during the installation of the Microsoft Windows. Additional keyboard drivers can be installed under Microsoft Windows which provide input methods for various locales and languages.
- If the system keyboard is selected, UniPad cannot display the Virtual Keyboard Window, because the keyboard layout and key assignment is implemented by the Windows keyboard drivers.
- UniPad can receive input from IMEs on East Asian systems.
- IMEs that are part of additional language support on non-Asian systems are also supported.

Built-in and user-defined virtual keyboards

UniPad provides several built-in keyboards. Additionally, you can create your own keyboards by modifying built-in keyboards or creating new ones from scratch. The keyboard layout and key assignments are shown in the Keyboard Window which can also be used to "type" the characters by clicking with the mouse. Please read the Basic Concepts topic for more information about virtual keyboards. In the current version UniPad does not support complex input methods for scripts like Chinese, Yi or Ethiopic.

Character map

The Character Map is a tool window that provides access to virtually all Unicode characters. Characters can be inserted by clicking with the mouse.

Hexadecimal input

If you know the hexadecimal value of some often used Unicode character, you can use the value to insert the character into the document. Press [CTRL + Q] to open a small window where you can enter the hexadecimal value.

Third party tools

Some third party keyboard tools are able to send Unicode characters directly. UniPad recognizes this input method. If you would like to know whether a keyboard tool is compatible with UniPad, ask the manufacturer of that tool about WM_UNICHAR support (which is the technical term for this input method).

See Also

- Compatibility options for keyboard input
- Concepts: Virtual Keyboards

2.6. Converting Files

Ways to convert files

UniPad can convert non-Unicode plain text files into Unicode documents and vice versa. This can be done via import and export. Non-Unicode files are usually encoded in a single-byte or multi-byte character set (e. g. ISO 8859-1, ASCII, Big Five, etc.).

Import

When you import a file, it is converted to a new unnamed Unicode document. You can:

- save the document as a Unicode plain text file,
- re-export the file using either the same or a different character set.

Some character sets may include characters which are not encoded in the Unicode Standard (example: manufacturer's logo in the Macintosh character sets).

Upon import these characters are replaced by the Unicode replacement character or private use character codes.

Export

When you export a Unicode document, it is converted to a plain text file, using the character set you choose. Characters that cannot be encoded in the target character set are replaced by a substitution characters. For example, if you have a document containing English and Japanese text and you export to ISO 8859-1, Japanese characters will be replaced by this substitution character, because Japanese characters cannot be encoded in ISO 8859-1. As default substitution character the code 0B(hex) is used. This character is an ASCII control character that was intended to be used as a substitution characters. However, in the export dialog you may choose a different substitution character.

Character sets

UniPad supports various single-byte and multi-byte character sets. See here for a complete list.

3. Basic Concepts

3.1. Overview

The following topics discuss several basic concepts that are important for a better understanding of UniPad's capabilities.

- Font
- Virtual Keyboards
- Handling of Combining Characters
- Handling of Byte Order Mark
- Handling of Line and Paragraph Separators
- Handling of U+0000 and Noncharacters
- Handling of Bidirectional Text
- Glyph Shaping

3.2. UniPad Font

UniPad comes with its own built-in font.

It is our goal to provide a Unicode editor that works without the need for any additional fonts. However, the current version does not support all Unicode characters. Unsupported characters are depicted as a question mark in a box.

Every glyph of the fixed width variant is a multiple of 10 pixels wide. Most Western characters are 10 pixels wide, most CJK characters (ideographs) and some complex characters are 20 pixels wide. Some very complex characters are be even 30 pixels wide.

See also

- Menu: Options > Font
- Dialog: Options > Font
- Toolbar: View
- Frequently Asked Questions

3.3. Virtual Keyboards

UniPad supports character input using virtual keyboards. "Virtual" means, that the keyboard does not exist as a physical hardware keyboard. In fact any physical keyboard can be used to type characters with the virtual keyboard. If you use a virtual keyboard the assignment and the layout of the keys usually do not correspond to the physical keyboard. Therefore you may use the Virtual Keyboard window to get an overview and to "type" the keys with the mouse.

UniPad supports several built-in virtual keyboards, but you can add your own user-defined keyboards. Each user-defined virtual keyboard is associated with a keyboard definition file. You can create new keyboards, save them, load existing keyboards and unload them if you don't need them anymore. The definition files contain all necessary information for a keyboard so you can send them via email, for example.

Before you create a new keyboard you should create a new text file. This enables the Virtual Keyboard window and also allows you to test your keyboard.

You can assign a character code to each key either by using the drag-and-drop technique or by selecting the command **Assign Character Code** in the keyboard context menu. The keyboard state can be locked temporarily for easier key assignment. Please note: If you use drag-and-drop on a built-in keyboard, UniPad automatically creates a copy of that keyboard.

See also

- <http://www.unipad.org/keyboard>
- Toolbar: Input
- Tool Window: Virtual Keyboard

3.4. Handling of Combining Characters

Combining characters are handled rather simple in the current version of UniPad. They are rendered as spacing characters in the way they are depicted in the Unicode code charts. For example the accent character U+030C COMBINING CARON is depicted as ◌̣ and occupies a character position like any other character. The dots indicate a placeholder for a base character that should combine with the accent.

Normally U+0072 LATIN LETTER SMALL R and U+030C COMBINING CARON should be rendered as ṛ̌. However with UniPad the characters are displayed separated as ř ◌̣.

The current implementation does not allow automatic combination of combining characters with the base characters. This will change in a future version of UniPad.

The command Edit > Convert > Combine Characters can be used to combine pairs of a base character and a combining character into a precomposed character if possible. The command Edit > Convert > Decompose Combinations works vice versa, that is, decomposes a precomposed character into a pair of a base character (or another precomposed character) and a combining character, if possible. Please note: these functions alter the characters not just the display.

See also

- Menu: Edit > Convert > Combine Characters
- Menu: Edit > Convert > Decompose Combinations
- Toolbar: Convert

3.5. Handling of Byte Order Mark

For a short introduction of the byte order mark (or: BOM) see the topic Glossary: Byte Order Mark.

UniPad recognizes the BOM when loading a Unicode file and extracts it from the file buffer. That means you won't see the BOM as the first character in the file window. This behaviour can be turned off. Please see Dialog: Preferences > Data for more information.

All other zero width non-breaking spaces (that do not act as a BOM) are preserved.

The BOM can also be used with other formats (UTF-8, UTF-7, Standard Compression Scheme and ASCII+UCN). Since the byte order is not an issue here, the BOM acts as a signature.

To include a BOM into the current file or to remove it from the current file, change the file properties.

See also

- Menu: File > File Properties
- Dialog: Preferences > Data
- Concepts: Handling of U+0000 and Noncharacters
- Glossary: Byte Order Mark

3.6. Handling of Line and Paragraph Separators

Line separators

In good old ASCII files the C0 control characters CARRIAGE RETURN (CR) and LINE FEED (LF) are used to mark an explicit line break. Traditionally on UNIX systems only LF is used, on DOS / Microsoft Windows systems the sequence CR LF is used. Moreover, programs that do automatic line wrapping (like word processors) normally use CR LF as a paragraph separator, while in ASCII plain text an empty line is often used as a paragraph separator.

Unicode introduced the unambiguous characters U+2028 LINE SEPARATOR and U+2029 PARAGRAPH SEPARATOR.

UniPad uses U+2028 internally as a line separator. However, files can be read and written using four possible kinds of line separators:

- Unicode (U+2028)
- Unix (LF = U+000A)

- DOS (CR LF = U+000D U+000A)
- Macintosh (CR = U+000D)

When opening a file, UniPad recognizes the line separator and converts it automatically to U+2028 if necessary. UniPad treats the kind of used line separator as a file property. That means when the file is saved, UniPad uses the original line separator. This works only, if the line separator was used consistently throughout the original file. UniPad does not preserve different line separators in a file. The line separator of the current file can be changed by Menu: File > File Properties.

In the case of UTF-16 or UTF-32 the Unicode line separator should be used. Formats that are designed to be 7/8-bit compatible (UTF-7, UTF-8, ASCII+UCN) should be used with the ASCII line separator, otherwise 8-bit text editors may get in trouble.

Paragraph separator

The paragraph separator U+2029 has the same effect as the line separator: it causes a simple line break.

See also

- Menu: File > File Properties
- Menu: File > Open
- Menu: File > Save
- Unicode Standard Annex #13 (Unicode Newline Guidelines)

3.7. Handling of U+0000 and Noncharacters

Unicode reserves certain code positions as so-called noncharacters. However, these character codes may occur in text files, therefore they are handled by UniPad safely. They appear as black solid squares and are handled like regular characters. Normally, it is not required to handle these code values, but the intention was to handle them transparently to allow the editing of corrupt files and other non-conforming data.

UniPad handles U+0000 (the ASCII C0 control character NUL) as a regular character, too. The only exception is when text including U+0000 is copied to the clipboard for use in other applications. Since the system uses U+0000 as a text termination character, it is removed before.

The handling of noncharacters can be customized. See the topic Menu: Options > Preferences > Errors for a detailed discussion of the available options.

See also

- Menu: Options > Preferences > Data
- Reference: Error Conditions
- Reference: Command Line Parameters
- Concepts: Handling of Byte Order Mark
- Glossary: Replacement Character
- Unicode Standard Annex #27 (Unicode 3.1)

3.8. Handling of Bidirectional Text

UniPad is able to handle bidirectional text according to the Unicode Bidi Algorithm.

Three levels of support are provided:

- **None:** all text (even Arabic, Hebrew, etc.) is rendered from left to right. This can be very useful if you have to edit bidirectional text and you want to avoid confusion during the editing process. This mode can also be used to edit text with characters that are in visual order rather than in logical order. Please note: if this mode is active, changing the document direction affects only the alignment of text (left aligned or right aligned).
- **Simple:** bidirectional text is rendered correctly according to the Bidi algorithm but directional formatting characters are ignored.
- **Full:** bidirectional text is rendered correctly according to the Bidi algorithm including directional formatting characters. 61 embedding levels are supported (this is conforming to the Bidi Algorithm and more than sufficient for all practical purposes).

Scope

UniPad applies the Bidi algorithm line by line (lines are terminated explicitly).

Mirroring

With simple and full support, certain characters are displayed mirrored. Example: within Hebrew text the character U+005B LEFT

SQUARE BRACKET looks like a right bracket on screen, while it remains a left bracket logically (this behaviour would be clearer if the left bracket had been called "opening bracket"). If mirroring would not be done, right-to-left text would look messed up.

Cursor movement and selection

If simple or full support is activated, within bidirectional text the cursor always follows the logical reading order of the text. The function of cursor keys depends on the current document direction:

- LTR document direction: the key [Arrow Right] moves the cursor along the reading order, that is rightwards through LTR text and leftwards through RTL text; the key [Arrow Left] works vice versa; the key [POS 1] positions the cursor on the leftmost character of the line; the key [END] works vice versa.
- RTL document direction: the key [Arrow Left] moves the cursor along the reading order, that is leftwards through RTL text and rightwards through LTR text; the key [Arrow Right] works vice versa; the key [POS 1] positions the cursor on the rightmost character of the line; the key [END] works vice versa.

This works similar with other navigation keys.

Text selection works accordingly: selections are always continuous regarding the reading order of the text. However this can lead to several discontinuous segments on screen (if Bidi support is activated), especially when directional formatting characters are used. UniPad supports up to 64 discontinuous segments (this is more than sufficient for all practical purposes).

Please note: visual orientated cursor movement is not supported intentionally, because it can cause endless loops while moving the cursor and may be more confusing than logical cursor movement.

See also

- Menu: Options > Bidirectional Text
- Dialog: Options > Bidirectional Text
- Toolbar: View
- Unicode Standard Annex #9 (The Bidirectional Algorithm)
- Appendix: Mirrored Characters

3.9. Glyph Shaping

Currently, UniPad supports glyph shaping for the following scripts:

- Arabic: contextual forms of the characters are used (initial, medial, final or isolated) and the characters are connected to emulate handwriting. If Arabic shaping is activated and text is typed using an Arabic keyboard, then dead-key input method should be turned off.

Glyph shaping can be turned on and off independently from character rendering options and bidirectional behaviour.

See also

- Menu: Options > Glyph Shaping
- Dialog: Options > Glyph Shaping
- Toolbar: View

4. Main Menu

4.1. File

4.1.1. New

Creates a new unnamed Unicode plain text file in a new window.

Defaults

Default file properties and default options can be configured in the dialog Default Document Options.

See also

- Menu: File > File Properties
- Menu: Options > Default Options
- Concepts: Handling of Line and Paragraph Separators
- Concepts: Handling of Byte Order Mark

4.1.2. Open

Opens an existing Unicode plain text file. See Dialog: Open File for more information.

See also

- Dialog: Open File
- Menu: File > Save

4.1.3. Save

Saves the content of the current window as a Unicode plain text file.

SC UniPad always uses the current file properties. If you want to save the file in a different format, change the file properties or choose the menu item Save As.

You can't use the Unicode line separator U+2028 with ASCII compatible formats (UTF-7, UTF-8, ASCII+UCN, SCSU). In these cases the line separator is automatically changed to DOS line separator sequence U+000D U+000A.

See also

- Menu: File > Open
- Menu: File > Save As
- Menu: File > File Properties
- Concepts: Handling of Line and Paragraph Separators
- Concepts: Handling of Byte Order Mark

4.1.4. Save As

Saves the content of the current window in a new file.

You will get a dialog box where you can choose the new file name and the new format. You cannot change the line separators. To do this you have to change the file properties before.

You can't use the Unicode line separator U+2028 with ASCII compatible formats (UTF-7, UTF-8, ASCII+UCN, SCSU). In these cases the line separator is automatically changed to DOS line separator sequence U+000D U+000A.

Formats

You can choose between different Unicode formats. All these formats are lossless regarding the range of Unicode characters. To save files encoded in a different character set you have to export it. Because of the possible loss of information, file export is an explicit operation.

See also

- Menu: File > Export
- Menu: File > Save
- Menu: File > File Properties
- Concepts: Handling of Line and Paragraph Separators
- Concepts: Handling of Byte Order Mark

4.1.5. Reload

Reloads an already open file.

If the file has been modified you will be asked for confirmation.

See also

- Menu: File > Open
- Menu: File > Reload As

4.1.6. Reload As

Reloads an already open file using a specific format.

This is useful if the file has been erroneously opened with the wrong format or if the format has been changed outside UniPad.

If the file has been modified you will be asked for confirmation.

See also

- Menu: File > Open
- Menu: File > Reload

4.1.7. Page Setup

Opens the Page Setup dialog.

See also

- Menu: File > Print
- Dialog: Page Setup

4.1.8. Print

Prints the current file or selection.

The following options of the current document are used for printing:

- character rendering,
- glyph shaping,
- bidirectional behaviour.

Since UniPad uses its built-in bitmap font for printing, the quality is not as good as if TrueType fonts would be used. Also, temporary

printing files (spool files) can become larger than usual.

See also

- Menu: File > Page Setup
- Dialog: Options

4.1.9. Import

Opens the Import dialog to import a file encoded in a different character set.

The file is loaded into a new unnamed window. Default options are applied.

See also

- Menu: File > Export
- Dialog: Import
- Reference: Error Conditions
- Concepts: Handling of Line and Paragraph Separators

4.1.10. Export

Opens the Export dialog to export the content of the current window to a new file.

The content of the current window remains unchanged.

See also

- Menu: File > Import
- Concepts: Handling of Line and Paragraph Separators
- Dialog: Export
- Supported character sets

4.1.11. Send

Sends the current file as an email attachment using a MAPI-compliant mailer. If no MAPI-compliant mailer is installed on your system, this menu command is disabled.

The Unicode text is encoded using the Standard Compression Scheme for Unicode (SCSU), which provides a very good compression with little overhead, especially for short text.

Please note: UniPad sends a temporary copy of the file, rather than the file itself.

4.1.12. File Properties

Opens the dialog Current Options > File to change the properties of the current file.

Please note: in versions of UniPad prior to 0.99 the file properties could be changed via a separate dialog. Now the file properties are part of the Current Options dialog. This command is provided for backwards compatibility. You can alternatively use the command Options > Current Options and select the File tab in the Current Options dialog.

See also

- Menu: File > Open
- Menu: File > Save
- Dialog: Options > File
- Concepts: Byte Order Mark

- Concepts: Line and Paragraph Separators

4.1.13. Recent Files

Allows you to load files from the history of recently opened files.

The maximum number of recent files that are remembered and viewed in the menu can be changed in the Preferences dialog.

The following document-specific information will be remembered:

- File name,
- Keyboard,
- Character rendering style,
- Overwrite / insert mode,
- Font (fixed or variable),
- Last cursor position,
- Bidi support,
- Glyph shaping.

The file history will also be used when you open files using the Open command or command line parameters.

Some options are not taken from the history under certain circumstances:

- Auto-detection is used if the option Remember used file format is turned off.
- The keyboard is not used (and not changed either) if the option Keyboard is global is turned on.

See also

- Menu: File > Open
- Menu: File > File Properties
- Menu: File > Current Options
- Dialog: Options > Preferences > General
- Dialog: Options > Preferences > Miscellaneous

4.1.14. Exit

Exits SC UniPad.

If UniPad is exited this way, the exit code is always 0.

See also

- Reference: Command Line Parameters
- Reference: Exit Codes

4.2. Edit

4.2.1. Undo

Undoes the last operation.

The maximum number of undoable operations can be customized in the Preferences dialog. The default is 256.

The undo buffer is cleared after a file has been saved, unless the option Undo After Save has been enabled.

Please note the following behaviour:

Internally, UniPad knows only two operations, **delete** and **insert**. Every text replacement is done by deleting the selected text

and inserting the new text. That means: if one replaces a selection with new text, the action has to be undone twice: first to undo the insertion and then to undo the deletion. This may change in a future version.

See also

- Menu: Edit > Redo
- Menu: Options > Preferences > Editing

4.2.2. Redo

Redoes the last undone operation.

See also

- Menu: Edit > Undo
- Menu: Options > Preferences > Editing

4.2.3. Cut

Cuts the current text selection as Unicode plain text into the clipboard.

See also

- Menu: Edit > Copy
- Menu: Edit > Paste
- Reference: Mouse Actions

4.2.4. Copy

Copies the current text selection as Unicode plain text to the clipboard.

See also

- Menu: Edit > Copy As
- Menu: Edit > Cut
- Menu: Edit > Paste
- Reference: Mouse Actions

4.2.5. Copy As

Opens the Copy As dialog to copy the current text selection in a specific format or character set to the clipboard.

See also

- Menu: Edit > Copy
- Menu: Edit > Paste As
- Dialog: Copy As
- Reference: Mouse Actions

4.2.6. Paste

Pastes the content of the clipboard as Unicode plain text into the current position i.e. replaces the current text selection.

See also

- Menu: Edit > Copy
- Menu: Edit > Paste As
- Reference: Mouse Actions

4.2.7. Paste As

Open the Paste As dialog to paste the content of the clipboard into the current document.

See also

- Menu: Edit > Copy As
- Menu: Edit > Paste
- Dialog: Paste As
- Reference: Mouse Actions

4.2.8. Insert Character

Opens the Insert Character dialog to insert a character by its hexadecimal value.

4.2.9. Select All

Selects all text in the current document.

See also

- Reference: Mouse Actions

4.2.10. Delete Line

Deletes the current line in the current window.

See also

- Menu: Edit > Cut
- Reference: Key Commands

4.2.11. Convert

4.2.11.1. Lowercase / Uppercase

Converts the selected text to lower case or upper case respectively.

Characters that do not have different case forms (such as Hebrew characters or symbols) remain unchanged.

4.2.11.2. Combine Characters

Combines pairs of a base character and a subsequent combining character into a precomposed characters if possible.

For example the text `xö â ñ ló` will be converted to `x̂ â̂ l̂ó`.

See also

- Concepts: Handling of Combining Characters
- Menu: Edit > Convert > Decompose Combinations

4.2.11.3. Decompose Combinations

Decomposes precomposed characters into a pairs of a base characters (or precomposed characters) and subsequent combining characters, if possible.

For example the text `x̂ â̂ l̂ó` will be converted to `xö â ñ ló`.

See also

- Concepts: Handling of Combining Characters
- Menu: Edit > Convert > Combine Characters

4.2.11.4. UCN (\u) to Unicode

This conversion dissolves all occurrences of Universal Character Names (UCN) within the current text selection.

The UCN syntax is `\uxxxx`, where `xxxx` is a hexadecimal number specifying the Unicode character value. For example the text `\u20ac` will be converted to `€` (EURO-CURRENCY SYMBOL). For code values in the planes 1 to 17 the following syntax is used: `\Uxxxxxxxx`, where `xxxxxxxx` is a hexadecimal number specifying the Unicode character value from 10000(hex) to 10FFFF(hex).

All values beyond U+10FFFF are not allowed and will be converted into U+FFFD REPLACEMENT CHARACTER.

See also

- Menu: Edit > Convert > Unicode to UCN
- Concepts: Handling of U+0000 and Noncharacters

4.2.11.5. Unicode to UCN (\u)

Converts all characters of the selected text to UCN notation.

See also

- Menu: Edit > Convert > UCN to Unicode

4.2.11.6. XML/SGML Character Reference to Unicode

Converts XML and SGML character references (hexadecimal and decimal) within the selected text to the corresponding Unicode characters.

The syntax is `&ddd;` (decimal) and `&#hhh;` (hexadecimal). For example, both the sequences `&8364;` and `AC;` will be converted to `€` (EURO-CURRENCY SYMBOL).

All values beyond U+10FFFF are not allowed and will be converted into U+FFFD REPLACEMENT CHARACTER.

See also

- Menu: Edit > Convert > Unicode to XML Character References
- Concepts: Handling of U+0000 and Noncharacters

4.2.11.7. Unicode to XML/SGML Character Reference

Converts all characters in the selected text to XML/SGML character references (either in decimal or hexadecimal form).

See also

- Menu: Edit > Convert > XML Character References to Unicode

4.2.11.8. XML/SGML Character Entity to Unicode

Converts SGML character entities occurring in the selected text to the corresponding Unicode characters.

The ISO 8897 Character Entity Sets ISOdia, ISOgrk1, ISOgrk2, ISOGrk3, ISOgrk4, ISolat1, ISolat2, ISOnum, ISOpub, ISOtech. are supported.

See also

- Menu: Edit > Convert > Unicode to SGML Character Entities

4.2.11.9. Unicode to XML/SGML Character Entities

Converts all characters in the selected text to SGML character entities if an entity is defined for a characters. Otherwise the character remains unchanged.

Example:

Ich möchte ins Café gehen.

will be converted to

Ich möchte ins Café gehen.

The ISO 8897 Character Entity Sets ISOdia, ISOgrk1, ISOgrk2, ISOGrk3, ISOgrk4, ISolat1, ISolat2, ISOnum, ISOpub, ISOtech. are supported.

See also

- Menu: Edit > Convert > SGML Character Entities to Unicode

4.3. Search

4.3.1. Find

Opens the Find dialog to find text within the current document.

See also

- Menu: Search > Replace
- Menu: Search > Go To
- Menu: Search > Find Bracket
- Dialog: Find

4.3.2. Find Next

Finds the next occurrence of the search text that was entered in the Find dialog.

See also

- Menu: Search > Find
- Dialog: Find

4.3.3. Replace

Open the Replace dialog to replace text within the document.

See also

- Menu: Search > Find

4.3.4. Replace Next

Replaces the next occurrence of the search text that was entered in the Replace dialog.

See also

- Menu: Search > Replace
- Dialog: Replace

4.3.5. Go To

Goes to a specific line in the text.

Line numbers start with 1. If an invalid or not existing line number is entered, the button **OK** is disabled.

See also

- Menu: Search > Find

4.3.6. Find Bracket

Finds the next corresponding bracket in any direction.

This command is only available if the current character is a bracket. Multiple nested brackets are recognized.

For example if the current position is on the first bracket in the string `[[...({.})..]]` the command will position the cursor before the last bracket. Repeating the command will position the cursor back to the previous position.

"Brackets" means: all Unicode characters that are of the categories "Punctuation, Open" and "Punctuation, Close". Only unambiguous bracket pairs are supported.

See also

- Menu: Search > Find

4.3.7. Find Bracket Select

Works like Search > Find Bracket but selects the text between the brackets (including the brackets).

4.4. Options

4.4.1. Current Options

Opens the dialog Options, where you can change the options of the current document.

Please note: since UniPad edits only plain text files, most options (except the file properties) cannot be saved within a text file. However, the options are remembered in the file history (recent files).

Some options are also directly available via menu, toolbar or other means.

Directly accessible via menu

- Menu: Options > Font
- Menu: Options > Character Rendering
- Menu: Options > Document Direction
- Menu: Options > Bidirectional Text
- Menu: Options > Glyph Shaping

Directly accessible via toolbar

- Font (Toolbar: View)
- Character Rendering (Toolbar: View)
- Document Direction (Toolbar: View)
- Bidirectional Text (Toolbar: View)
- Glyph Shaping (Toolbar: View)

Accessible via keys

- Override mode (Reference: Key Commands)
- Document direction (Reference: Key Commands)

See also

- Dialog: Options
- Menu: Options > Default Options

4.4.2. Font

Allows selection between a variable width font and a fixed width font for the current document.

See also

- Dialog: Options > Font
- Concepts: Font
- Toolbar: View

4.4.3. Character Rendering

Allows you to change the current character rendering options.

See Dialog: Options > Character Rendering for more information.

See also

- [Toolbar: View](#)

4.4.4. Document Direction

Allows you to change the current document direction (LTR or RTL).

See also

- [Dialog: Options > Bidirectionality](#)
- [Concepts: Bidirectional Text](#)
- [Toolbar: View](#)

4.4.5. Bidirectional Text

Allows you to change the current Bidi implementation level.

See [Dialog: Options > Bidirectional Text](#) for more information.

See also

- [Concepts: Bidirectional Text](#)
- [Toolbar: View](#)

4.4.6. Glyph Shaping

Allows you to change the current glyph shaping options.

See [Dialog: Options > Glyph Shaping](#) for more information.

See also

- [Concepts: Glyph Shaping](#)
- [Toolbar: View](#)

4.4.7. Toggle Keyboard

Toggles between two keyboards ("Quick Toggle").

To use this feature you have to define the two keyboards in the Preferences dialog.

See also

- [Menu: Options > Preferences > Keyboard](#)

4.4.8. Toolbar

This menu allows you to control the visibility of the toolbars.

Please note: The Character Info Bar and Extended Character Info Bar behave like a toolbar. The main Status Bar is not movable.

4.4.9. Default Options

Opens the dialog Options, where you can change the default options.

The default options are used if

- a new document is created,
- a file is imported,
- a file is opened and the options are not available from the file history (recent files).

Please note: the default keyboard is the only default option that can be changed without using this dialog: if no document is open, selecting a keyboard changes the default keyboard.

See also

- Dialog: Options
- Menu: Options > Current Options

4.4.10. Preferences

Opens the Preferences dialog.

See also

- Dialog: Preferences > General
- Dialog: Preferences > Data
- Dialog: Preferences > Editing
- Dialog: Preferences > Errors
- Dialog: Preferences > Keyboard
- Dialog: Preferences > Miscellaneous

4.5. Keyboard

4.5.1. New (From Scratch)

Creates a new user-defined virtual keyboard from scratch. No characters are assigned to the keys. The keyboard is empty and unnamed.

If there is no open document, the virtual keyboard window is disabled. It is recommend to create a new empty file that can also be used for testing the keyboard. Please note: the keyboard may change if you switch to another document window (unless the option Keyboard is global is activated).

Characters can be assigned to each key by using drag-and-drop technique: you can drag a character from the Character Map and drop it onto the Virtual Keyboard window.

See also

- Menu: Keyboard > New (Clone Current)
- Menu: Keyboard > Save
- Reference: Mouse Commands

4.5.2. New (Clone Current)

Creates a new user-defined virtual keyboard. It is a copy of the currently selected keyboard and is named after the original ("Copy of ..."). However, the new virtual keyboard is treated as unnamed if you save it the first time. The original keyboard can either be built-in or user-defined.

If there is no open document, the virtual keyboard window is disabled. It is recommend to create a new empty file that can also be used for testing the keyboard. Please note: the keyboard may change if you switch to another document window (unless the option Keyboard is global is activated).

Characters can be assigned to each key by using drag-and-drop technique: you can drag a character from the Character Map and drop it onto the Virtual Keyboard window.

See also

- Menu: Keyboard > New (From Scratch)
- Menu: Keyboard > Save
- Reference: Mouse Commands

4.5.3. Load

Loads a virtual keyboard from a keyboard definition file (extension **.ukb**). You will be asked for the file name. The keyboard is registered with UniPad and will be loaded automatically upon each start.

See also

- Menu: Keyboard > Unload
- Menu: Keyboard > Save

4.5.4. Reload

Reloads the current virtual keyboard. This is useful, if you have modified the keyboard definition file manually inside or outside UniPad. Otherwise manual changes do not take effect until you restart UniPad.

Built-in keyboards cannot be reloaded.

See also

- Menu: Keyboard > Load

4.5.5. Unload

Unloads the current virtual keyboard. The keyboard is not registered anymore and will not be loaded upon start. You will be asked for confirmation. If the virtual keyboard has been modified you will be asked to save it.

Built-in keyboards cannot be unloaded.

See also

- Menu: Keyboard > Load

4.5.6. Save

Saves the current virtual keyboard. If the keyboard is still unnamed you will be asked for a name. You can enter any valid file name.

The extension **.ukb** will be appended automatically if you do not provide an extension. You can save the file at any location within the file system.

See also

- Menu: Keyboard > Load

4.5.7. Open Definition File

Opens the definition file of the current virtual keyboard with UniPad.

The file format of the keyboard definition files is based on XML.

See also

- <http://www.unipad.org/keyboard>

4.6. Window

This menu offers the standard operations for multi document (MDI) applications.

All open windows appear in a list at the end of the menu.

See also

- Menu: File > Recent

4.7. Help

Help > Introduction

Opens this help document with the Introduction topic.

Help > What's New

Opens this help document with the What's New topic.

Help > Index

Opens the help file index.

Help > Web Site

Open the UniPad Home Page www.unipad.org.

Help > About UniPad

Copyright and version information.

Help > Unicode Info

Shows some basic information about the supported Unicode version:

- Version number
- Number of defined character blocks
- Total number of characters

See also

- Appendix: Additional Information

5. Context Menus

5.1. Document

You will get the document context menu when you click with the secondary mouse button somewhere in a document window.

All the commands in this menu are also available from the main menu:

- Cut
- Copy
- Copy As
- Paste
- Paste As
- Delete
- Select All
- Convert > Lowercase / Uppercase
- Convert > Combine Characters
- Convert > Decompose Combinations
- Convert > UCN (\u) to Unicode
- Convert > Unicode to UCN (\u)
- Convert > XML Character Reference to Unicode
- Convert > Unicode to XML Character Reference
- Convert > SGML Character Entity to Unicode
- Convert > Unicode to SGML Character Entity
- Reload
- Reload As
- Current Options

5.2. Virtual Keyboard

You will get the Virtual Keyboard context menu when you click with the secondary mouse button somewhere on the Virtual Keyboard window.

Attention: the menu cannot be activated while you hold down the [AltGr] key. Please use the **Lock** command instead, as described below.

The following commands in this menu are also available from the main menu:

- New (From Scratch)
- New (Clone Current)
- Load
- Unload
- Save
- Save As

The following commands are available only in the context menu:

Lock

Temporarily locks the keyboards state. You can lock **Shift**, **AltGr** or **Shift + AltGr**. The state remains active unless you press a key on the real physical keyboard or you choose the menu item **Lock > Cancel**. This command makes assignment of keys easier, since the state remains locked during drag-and-drop key assignment.

The following commands are available only if the context menu was opened over a valid key:

Assign Character Code

Opens a dialog where you can enter the hexadecimal code of a Unicode character that you want to assign to the specific key. However, it is recommend to use the drag-and-drop technique to assign character codes from the Character Map to a specific key.

Delete Key Assignment

Deletes an assignment. The key will not produce a character anymore.

See also

- Tool Window: Virtual Keyboard
- Dialog: Assign Character Code

5.3. Character Map

You will get the Character Map context menu when you click with the secondary mouse button on a character cell in the Character Map window.

The menu commands refer to the character that was selected when the secondary mouse button was pressed. The following commands are available:

Insert

Inserts the character at the current position in the current document. This is same as clicking with the primary mouse button.

Copy to Clipboard

Copies the character as a Unicode character to the clipboard.

Append to Clipboard

Appends the character to an existing Unicode string in the clipboard.

Insert As...

Inserts the character at current position in the current document, either as a Universal Character Name, an SGML/XML character reference or a character name.

Copy to Clipboard As...

Copies the character to the clipboard, either as a UTF-8 sequence, Universal Character Name, an SGML/XML character reference or a character name.

Append to Clipboard As...

Appends the character to an existing string in the clipboard, either as a UTF-8 sequence, Universal Character Name or an SGML/XML character reference.

See also

- Tool Window: Character Map

5.4. Toolbars / Statusbar

You will get the Toolbar context menu when you click with the secondary mouse button somewhere on a toolbar.

The context menu allows you to show or hide each toolbar.

See also

- Menu: Options > Toolbar

5.5. Application

You will get the application context menu when you click with the secondary mouse button somewhere in the application window (outside of a document window).

All the commands in this menu are also available from the main menu:

- New
- Open
- Recent Files

- Default Options
- Preferences

5.6. Edit Field

This menu is available within all UniPad edit fields.

It provides almost the same commands as the context menu of standard edit fields:

- Undo (only one step)
- Cut
- Copy
- Paste
- Select All

UniPad edit fields are used in the following dialogs: Find, Replace.

6. Dialogs

6.1. Open File

This standard dialog allows you to open an existing file.

UniPad does not make any assumptions about the file name extension and allows any file to be opened.

Formats

You can choose between different Unicode formats or you can use auto-detection.

All important formats are supported: UTF-16, UTF-32, UTF-8, UTF-7, ASCII+UCN, SCSU. All these formats are lossless regarding the range of Unicode characters. To open a file encoded in a different character set you have to import it. Because of the possible loss of information (e.g. private characters) file import is an explicit operation.

SC UniPad detects the line separators used in a file and an optional byte order mark. The line separator type, the byte order mark and the format of the file are the properties of a file. These are remembered as long as the file is open in SC UniPad. The properties are used when you save the file.

Auto-detection

If you often have to change between different formats, you may want to use the automatic format detection. Simply choose the format "Unicode Text (Auto Detect)". SC UniPad tries to detect the format of the file. This should work for most files. If you often change between different byte orders of the same format you may want to use the format "UTF-16 (Auto ByteOrder)" or "UTF-32 (Auto ByteOrder)".

Errors

If an error occurs during file load, you will get a message box with error information (only if this feature is activated). If the error is a result of a wrong format you have the option to change to the format and reload the file immediately.

Aborting

Loading a file can be aborted by pressing the key [ESC].

See also

- Menu: File > Import
- Menu: File > Save
- Menu: File > File Properties
- Concepts: Handling of Line and Paragraph Separators
- Concepts: Handling of Byte Order Mark

6.2. Save File

This standard dialog allows you to save a file.

UniPad automatically chooses the file extension `.utx` as the default extension. But you are free to use any file extension you want. You can also change the default extension.

Formats

You can choose between different Unicode formats.

All these formats are lossless regarding the repertoire of Unicode characters. To save a file encoded in a different character set you have to export it. Because of the possible loss of information (e.g. private characters) file export is an explicit operation.

Aborting

Saving a file can be aborted by pressing the key [ESC]. In this case the file will not be written completely which may result in a corrupt file on disk. In this case you should save the file again (to another location if necessary). The document within UniPad always remains intact.

See also

- Menu: File > Export
- Menu: File > Open

6.3. Import

This dialog allows the import of a file encoded in a different character set.

The file is loaded into a new unnamed window. You can choose the character set from a list.

Aborting

Importing a file can be aborted by pressing the [ESC] key.

See also

- Menu: File > Export
- Menu: File > Save
- Reference: Error Conditions

6.4. Export

Exports the content of the current document to a new file.

The content of the current document remains unchanged.

You can choose the character set from a list.

Options

Additional options are:

- the type of line separator and
- the substitution character code which is used for any Unicode character that has no corresponding code in the target character set.

Aborting

Exporting a file can be aborted by pressing the key [ESC]. In this case the file will not be written completely which may result in a corrupt file on disk. In this case you should export the file again (to another location if necessary). The document within UniPad always remains intact.

See also

- Menu: File > Save As
- Menu: File > Import
- Concepts: Handling of Line and Paragraph Separators

6.5. Page Setup

Allows you to setup the page parameters for printing:

- page orientation,
- zoom factor,
- page margins,
- optional page header.

The page header includes the file name and page number.

Margins are specified either in millimeters or inches. The unit of measurement can be set in the Preferences dialog.

See also

- Menu: Options > Preferences > General
- Menu: File > Print

6.6. Copy As

Copies the current text selection as text using a specific character set to the clipboard.

In addition to the character sets this command also allows to copy the text as UTF-8, UTF-7 or ASCII+UCN.

See also

- Menu: Edit > Copy
- Menu: Edit > Paste As
- Reference: Mouse Actions

6.7. Paste As

Pastes the content of the clipboard as a specific character set.

Unicode text in the clipboard will be ignored. In addition to the character sets this command also allows to paste the text as UTF-8, UTF-7 or ASCII+UCN.

See also

- Menu: Edit > Copy As
- Menu: Edit > Paste
- Reference: Mouse Actions

6.8. Go To

Moves the text cursor to a specific line. Line numbers start with 1.

Extend selection

If this option is turned on the text from the current position up to the entered line number will be selected. This works in both directions. An existing selection will be extended.

6.9. Find

Finds a specific text in the content of the current window. The found text will be selected.

The following options are available when searching:

Allow Universal Character Names

If this option is selected, UCNs can be used in the text field. These will be converted internally before the search begins. Otherwise sequences with `\u` will be treated as normal characters.

Use view settings of the current document

If selected, the view settings of the current document are used for all characters typed in the edit field. Otherwise the following settings are used:

- formatting characters are visible,
- control characters are visible,
- unsupported characters, surrogates, private use characters and unassigned codepoints are displayed as hexadecimal value,
- bidirectional behaviour is turned off,
- glyph shaping is turned off,
- variable width font is used.

Origin

The search can start from the beginning of the file or from the current cursor position.

Entire words only

If selected, only entire words are found. A word is a text entity that is delimited by word separators.

Ignore case

Case of characters is ignored.

See also

- Menu: Search > Replace
- Menu: Search > Go To
- Menu: Search > Find Bracket
- Context Menu: Edit Field

6.10. Replace

Replaces a specific text within the current window with another text.

You can press the button **Replace All** to replace all occurrences of the text.

The following options are available when searching:

Allow Universal Character Names

If this option is selected, UCNs can be used in the texts fields for search and replace text. These will be converted internally before the replace begins.

Otherwise sequences with `\u` will be treated as normal characters.

Use view settings of the current document

If selected, the view settings of the current document are used for all characters typed in the edit field. Otherwise the following settings are used:

- formatting characters are visible,
- control characters are visible,
- unsupported characters, surrogates, private use characters and unassigned codepoints are displayed as hexadecimal value,
- bidirectional behaviour is turned off,
- glyph shaping is turned off,
- variable width font is used.

Origin

The replace can start from the beginning of the file or from the current cursor position.

Entire words only

If selected, only entire words are found. A word is a text entity that is delimited by word separators.

Ignore case

Case of characters is ignored.

Prompt on replace

If selected, you will be asked before a text will be replace.

See also

- Menu: Search > Find
- Context Menu: Edit Field

6.11. Preferences

6.11.1. (Overview)

Global settings and preferences.

Categories

The preferences are divided into the following categories:

- General
- Files
- Data
- Errors
- Editing
- Keyboard
- Miscellaneous

6.11.2. General

User language

SC UniPad currently supports English and German as the user interface language. Future versions may support more languages. The choice of the user language has no impact on the functionality. Some names (like built-in virtual keyboards, character sets) are always displayed in English.

Unit of measure

Choose whether you want to enter the page margins in the Page Setup dialog in millimeters or inches.

6.11.3. Files

Recent files

UniPad remembers the names of the recently used files. You can specify how many files shall be recorded in the history and how many of them will appear in the menu. Available options:

- **Remember used file format:** if this option is activated the file format of a file is remembered in the history and used the next time you open the file; otherwise auto-detection is used.
- **Show file format in menu:** if this option is activated the used file format will be displayed after the file name in the menu File > Recent Files.
- **Show filenames without path:** if this option is activated only the base name of the file is displayed in the menu.

Backup

If this option is activated, a backup is created upon saving the file. The suffix will be appended to the original file name. You may also enter a suffix without a dot. Example: if the suffix is a tilde ~ the backup of the file **test.utx** will be named **test.utx~**.

Please note: the backup is created by copying the original file before saving. Therefore you should turn this option off, if you have to save large files often.

Default file extension

The default file extension is used when you save a file without explicitly specifying an extension. The extension may include all valid file name characters (except the dot) and must not be longer than 6 characters.

Check files for external modification

If activated, UniPad checks a file for modification outside the editor, whenever you activate the file window within UniPad or when you want to save the file. There is no guarantee that UniPad will detect modifications on network drives. In the case you experience any strange behaviour or an erroneous detection, you should turn this option off.

See also

- Menu: File > Recent

6.11.4. Data

Default byte order

The default byte order is used for auto-detection when there is absolutely no hint of the byte order of a Unicode text file. This should occur seldom.

Auto-detection

Specifies how many bytes at the beginning of a file are used to detect the format. You may decrease this value on slow machines

or increase it to get better detection.

Miscellaneous encoding options

Offers several options for specific encodings. More information regarding UTF-7 can be found in the Glossary.

Strip initial U+FEFF

Normally UniPad strips a BOM when loading a file. If this behaviour is turned off, the character is visible in the document window and can be edited like any other character.

See also

- Menu: File > Open
- Menu: File > Import

6.11.5. Errors

Suppress errors if possible

Suppresses errors while loading a file if possible.

Replace overlong UTF-8 sequences with U+FFFD

If this option is activated, UniPad rejects overlong UTF-8 sequences and replaces them with the replacement character. Otherwise these sequences will be handled like regular sequences and converted into the corresponding character.

Replace noncharacters with U+FFFD

If this option is activated, UniPad rejects all noncharacter values and replaces them with the replacement character. Otherwise these codes will be handled like normal characters.

See also

- Menu: File > Open
- Menu: File > Import
- Concepts: Handling of U+0000 and noncharacters
- Reference: Error Conditions

6.11.6. Editing

Caret shape

Choose between a standard character shape or the UniPad caret shape. The UniPad caret indicates the width and direction of a character.

Maximum undo actions

Maximum number of actions that can be undone.

Undo after save

If this option is activated the undo buffer will not be cleared when a file is saved.

See also

- Menu: Edit > Undo
- Menu: Options > Reading Order

6.11.7. Keyboard

Show numeric keypad on Virtual Keyboard window

If this option is activated, the Virtual Keyboard window will also display a numeric keypad on the right side, similar to a real keyboard.

Show caption for Ctrl, Shift, Alt, AltGr, CapsLock, Tab, Enter, Backspace

If this option is activated, a small image is displayed on these keys indicating their function, otherwise the captions of these keys remain blank.

CapsLock applies to alphabetic keys only

If this option is activated, the CapsLock mode shifts only alphabetic keys, otherwise all keys are shifted. It is recommended to turn this option on. It should be turned off only to simulate the behaviour of the German standard keyboard of Microsoft Windows.

Keyboard window is always enabled

Normally the keyboard window is disabled when there is no open document. If this option is turned on, the window is always enabled, allowing you to use the context menu to inspect or modify the virtual keyboard. Please note: the virtual keyboard window does not respond to keystrokes if there is no open document.

Mark user-defined keyboards with a plus sign ('+')

If this option is turned on, a plus sign is appended to the name of any user-defined virtual keyboard, wherever it is displayed. This allows you to quickly recognize user-defined keyboards.

Quick toggle

Choose the two virtual keyboards you need most often. You can toggle between these keyboards using the command Options > Toggle Keyboard or the [F2] key.

See also

- Menu: Options > Toggle Keyboard
- Concepts: Keyboard

6.11.8. Miscellaneous

Acoustic signals

Enable or disable beeps that occur under several error conditions.

Accelerated Navigation in Character Map

By default, if you hold down a navigation button of the Character Map the movement will be accelerated. If this option is turned off, acceleration will be disabled.

Use origin for import file for export

If this option is activated, the origin of an imported file (i. e. filename and character set) is used as default in the Export dialog. For example, if you have imported the file "mydoc.txt" using ISO 8859-4, UniPad remembers this information and when you export the file, "mydoc.txt" and ISO 8859-4 are provided as defaults.

Always open document windows maximized

If this option is turned on, a new window is always opened in maximized state. Usually, a new window is opened by the commands File > New, File > Open, File > Import.

Allow to save unaltered files

Normally, if a file is not modified the command File > Save is disabled. If this option is activated, a file can be saved anytime, whether it has been modified or not.

Show splash screen upon program start

You can turn this option off to hide the splash screen.

Associate .utx with UniPad upon each program start

UniPad is associated with the file extension **.utx** during installation. Each time you start UniPad this association is refreshed. In case you experience any problems, e. g. like warning messages from registry watchdog programs, you should turn this off.

Allow IME character input on non-Asian versions of Windows

If this option is activated, UniPad is able to handle IME input from non-Asian version of Microsoft Windows, e. g. Windows 2000 US with Japanese language support installed. Normally this option should be turned on. You should turn this off only if you are experiencing any problems with keyboard input.

Allow direct Unicode character input from external programs

Some programs, especially certain keyboard tools, directly send Unicode characters to an application. If this option is activated, UniPad is able to handle these character messages. Normally this option should be turned on. You should turn this off only if you are experiencing any problems with keyboard input. Technical note: this input method is based on the WM_UNICHAR message.

Allow character input from Unicode locales

Normally each locale has a corresponding code page. For example: the locale "Russian" uses the code page 1251 (Cyrillic). On Windows XP, the locales of some languages (e. g. Hindi) use Unicode rather than a specialised code page. If this option is activated, UniPad is able to handle input from these locales. Normally this option should be turned on. You should turn this off only

if you are experiencing any problems with keyboard input.

Always start with a new document

If this option is activated, a new, empty and unnamed document is created whenever UniPad is started.

Also combine and decompose character triplets

In previous versions of UniPad the functions Edit > Convert > Combine Characters and Edit > Convert > Decompose Combinations were only able to handle character pairs. In the current version these functions can also handle character triplets. Normally this option should be turned on.

6.12. Options

6.12.1. (Overview)

This dialog is used for current options and default options.

Please note: the dialog also contains the file properties (in versions of UniPad prior to 0.99 file properties had to be changed via a separate File Properties dialog).

1. Current Options

The dialog is displayed to change the options and file properties of the current document. The button **Use Defaults** can be used to reset all options to the default settings (see below). A few options are not available here.

2. Default Options

The dialog is displayed to change the default options and default file properties. The button **Use Factory Settings** can be used to reset all options to the original settings.

Categories

All settings are divided into the following categories:

- File
- Editing
- Font
- Character Rendering
- Glyph Shaping
- Bidirectionality

6.12.2. File

This section contains the file properties.

Please note: in versions of UniPad prior to 0.99 this was a separate dialog.

Since all files are plain text, these properties are not stored separately. Rather they are the inherent physical properties of a plain text file.

Properties

The following properties can be changed:

- The format of a file,
- The type of line separators.
- The byte order mark / signature.

Attributes

The read-only attribute can be changed for the current document. Please note: this changes only the read-only flag of the document in UniPad and it does not change the file attributes of the physical file.

See also

- Menu: File > Open
- Menu: File > Save
- Concepts: Byte Order Mark

- Concepts: Line and Paragraph Separators

6.12.3. Editing

Keyboard

The keyboard that is used for typing characters in the document window.

Dead-key input method

If this option is activated, typing a combining character on the keyboard will produce no output until a base character is typed. If the base character combines with the combining character, the precomposed character is produced. This is the behaviour of most of the keyboards in use today. For example: Type [^] key. Nothing happens. Type the [a] key. The character â (U+00E2) is inserted into the text. If the option is not activated, combining marks are inserted immediately into the text.

Keyboard is global

If this option is activated the selected virtual keyboard is global for all open windows. Otherwise every open window uses its own virtual keyboard. Please note: this option can be changed only with the default options.

Tabulator width

Specifies the width of the tabulator. For the fixed width font the width is specified in characters (the default is 8). For the variable width font the width is specified in pixels (the default is 80).

Strip trailing spaces when new line typed

If activated, all whitespaces at the end of the line will be removed when you type a new line.

Overwrite mode

If activated, typed characters overwrite existing characters.

Auto-indent new lines

If activated, a new line is indented to the same level as the previous one whenever you press the [ENTER] key. Please note: exactly the same whitespace characters from the previous line are used. For example: if the previous line is indented by U+3000 IDEOGRAPHIC SPACE followed by U+0009 HORIZONTAL TABULATION the new line will begin with the same characters.

See also

- Dialog: Options (Overview)

6.12.4. Font

Font

Allows to choose between the variable width and the fixed width font.

See also

- Dialog: Options (Overview)
- Concepts: Font

6.12.5. Character Rendering

6.12.5.1. (Overview)

These options control how special characters are displayed.

- Spaces
- Control Characters
- Formatting Characters
- Unsupported Characters (UniPad specific)
- Unassigned Codepoints (these are not characters)

- Surrogate Codepoints
- Private Use Characters

See also

- Dialog: Options (Overview)

6.12.5.2. Spaces

Selection of character rendering style for Unicode space characters.

Standard glyph

The normal spacing glyph (only visible by the occupied space).

Picture glyph

A unique glyph depicting the specific space character using an acronym or abbreviation of the character name.

-  U+2007 FIGURE SPACE
-  U+00A0 NO-BREAK SPACE

See also

- Dialog: Options > Character Rendering (Overview)
- Glossary: Glyph

6.12.5.3. Control Codes

Selection of character rendering style for the ASCII C0 and C1 control characters and DELETE.

Picture glyph

A unique glyph depicting the specific control character using an acronym or abbreviation of the character name.

-  U+001B ESCAPE
-  U+0007 BELL

Hexadecimal

A glyph containing the hexadecimal value of the character in U+XXXX notation.

-  U+001B ESCAPE
-  U+0007 BELL

Invisible

The character is not visible at all. However it can be selected by positioning the cursor.

See also

- Dialog: Options > Character Rendering (Overview)
- Glossary: Glyph

6.12.5.4. Formatting Characters

Selection of character rendering style for Unicode formatting characters.

Picture glyph

A unique glyph depicting the specific control character using an acronym or abbreviation of the character name.

-  U+200E LEFT-TO-RIGHT MARK
-  U+200D ZERO-WIDTH JOINER

Alternative glyph

An alternative glyph using a depiction which is more related to the concrete function of the character rather than to its name. If no

alternative glyph is defined the character is invisible.

-  U+200E LEFT-TO-RIGHT MARK
-  U+200D ZERO-WIDTH JOINER

Invisible

The character is not visible at all. However it can be selected by positioning the cursor.

See also

- Dialog Options > Character Rendering (Overview)
- Glossary: Glyph

6.12.5.5. Unsupported Characters

Selection of character rendering style for unsupported characters.

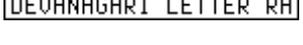
Picture glyph

A glyph containing a question mark for all characters of this category.

-  U+A490 YI RADICAL QOT
-  U+0930 DEVANAGARI LETTER RA

Character name

A glyph containing the Unicode character name.

-  U+A490 YI RADICAL QOT
-  U+0930 DEVANAGARI LETTER RA

Hexadecimal

A glyph containing the hexadecimal value of the character in U+XXXX notation.

-  U+A490 YI RADICAL QOT
-  U+0930 DEVANAGARI LETTER RA

See also

- Dialog: Options > Character Rendering (Overview)
- Glossary: Glyph

6.12.5.6. Unassigned Codepoints

Selection of character rendering style for currently unassigned code points.

Picture glyph

A glyph containing a hatched box for all unassigned code points.

-  U+036F
-  U+0530

Hexadecimal

A glyph containing the hexadecimal value of the character in U+XXXX notation.

-  U+036F
-  U+0530

See also

- Dialog: Options > Character Rendering (Overview)
- Glossary: Glyph

6.12.5.7. Unpaired Surrogates

Selection of character rendering style for unpaired surrogate characters.

Picture glyph

A glyph containing a torn ticket symbol (different for low surrogates and high surrogates).

-  U+D800
-  U+DC00

Hexadecimal

A glyph containing the hexadecimal value of the character in U+XXXX notation.

-  U+D800
-  U+DC00

See also

- Dialog: Options > Character Rendering (Overview)
- Glossary: Glyph

6.12.5.8. Private Use Characters

Selection of character rendering style for private use characters.

Picture glyph

A glyph containing an asterisk for all private use characters.

-  U+E000
-  U+F8FF

Hexadecimal

A glyph containing the hexadecimal value of the character in U+XXXX notation.

-  U+E000
-  U+F8FF

See also

- Dialog: Options > Character Rendering (Overview)
- Glossary: Glyph

6.12.6. Glyph Shaping

Arabic: contextual forms and connected characters

If this option is activated, Arabic characters are connected and their appropriate contextual forms (initial, medial, final or isolated) are used. Otherwise characters are not connected and only their isolated forms are used.

See also

- Dialog: Options (Overview)
- Concepts: Glyph Shaping

6.12.7. Bidirectionality

Implementation level

Specifies how bidirectional text is handled. For more information see: Handling of Bidirectional Text.

Document direction

Specifies the predominant direction of text: if the direction is left-to-right, the text in the document is left aligned; if the direction is right-to-left, the text is right aligned. This is independent from the implementation level.

Auto-detection

If this option is activated, UniPad tries to detect the document direction automatically upon loading and importing. Please note: this option can be changed only with the default options.

See also

- Dialog: Options (Overview)
- Concepts: Handling of Bidirectional Text

6.13. Reading Error

This dialog appears when an error occurs while opening a file (option Suppress errors if possible must be turned off).

It provides the following information:

- Type of error condition
- Current format (may be the result of auto-detection)
- Byte position in the file where the error occurred

The following actions can be taken:

- **Abort:** abort the task (opening or importing a file).
- **Ignore:** ignore the error and continue.
- **Ignore All:** ignore this error and all subsequent errors and continue.
- **Reload:** reload the file using another format.

See also

- Reference: Error Conditions

6.14. Assign Character Code

This dialog allows you to enter the hexadecimal code of a Unicode character that you want to assign to a specific key of the Virtual Keyboard.

See also

- Context Menu: Virtual Keyboard
- Tool Window: Virtual Keyboard

6.15. Conversion Options

This dialog is displayed whenever you perform one of the following conversion functions:

- Unicode to UCN (\u)
- Unicode to XML Character Reference
- Unicode to SGML Character Entity

It allows you to choose between the following options:

- convert all characters (that is, the whole Unicode range)
- convert all characters except the ASCII range (U+0000 .. U+007F)
- convert all characters except the ASCII and Latin 1 range (U+0000 .. U+00FF)

6.16. Insert Character

This dialog allows you to insert arbitrary Unicode characters quickly by typing their hexadecimal value.



Please note: the dialog is uncommon because it does not provide any buttons. But you can control it entirely via keyboard:

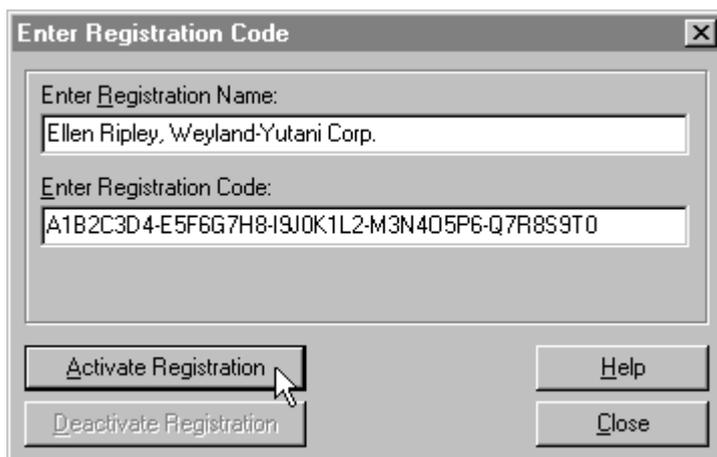
- [ENTER] confirms the input and inserts the corresponding character into the current document.
- [ESC] closes the dialog without inserting anything into the document.

6.17. Enter Registration Code

When you register UniPad you will receive a confirmation containing a registration code.

Use this dialog to enter your name and/or organisation name and the registration code. Then press **Activate Registration**.

Example:



After successful activation the dialog shows your name and/or organisation name, the serial number and the license type you have purchased. Press the **Close** button to continue.

Example:



Please note: You may press the button **Deactivate Registration** at any time to remove the registration code from the computer. If you do this, UniPad will run in unregistered mode again. You may re-activate the registration at any time by entering the registration code as described above. This is useful for easily managing multi-user licenses without deinstalling UniPad.

For further questions regarding licensing issues you may use the online contact form on www.unipad.org.

7. Toolbars / Tool Windows

7.1. Standard Toolbar



The Standard Toolbar provides access to the following functions:

- Menu: File > New
- Menu: File > Open
- Menu: File > Save
- Menu: File > File Properties
- Menu: File > Print
- Menu: Edit > Cut
- Menu: Edit > Copy
- Menu: Edit > Paste
- Menu: Edit > Undo
- Menu: Edit > Redo
- Menu: Search > Find
- Menu: Search > Find Next
- Menu: Search > Go To
- Menu: Search > Find Bracket

See also

- Menu: Options > Toolbar

7.2. View Toolbar



The View Toolbar provides access to the following functions:

- Menu: Options > Font
- Menu: Options > Character Rendering
- Menu: Options > Document Direction
- Menu: Options > Bidirectional Text
- Menu: Options > Glyph Shaping

See also

- Menu: Options > Toolbar
- Menu: Options > Default Options

7.3. Input Toolbar



The Input Toolbar provides access to the following functions:

- Selection of the currently used virtual keyboard
- Menu: Options > Toolbar > Keyboard

- Menu: Options > Toolbar > Character Map

See also

- Menu: Options > Toolbar
- Dialog: Options

7.4. Conversion Toolbar



The Conversion Toolbar provides access to the following functions:

- Convert > Lowercase / Uppercase
- Convert > Combine Characters
- Convert > Decompose Combinations
- Convert > UCN (\u) to Unicode
- Convert > Unicode to UCN (\u)
- Convert > XML Character Reference to Unicode
- Convert > Unicode to XML Character Reference
- Convert > SGML Character Entity to Unicode
- Convert > Unicode to SGML Character Entity

Please note: for easier recognition of the appropriate button, each button is as twice as wide as a normal toolbar button. The text before conversion is symbolized by black text, the text after conversion is symbolized by blue text.

See also

- Menu: Options > Toolbar

7.5. Status Bar



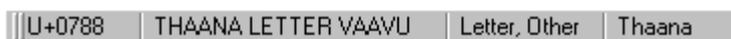
The Status Bar shows the following information:

- Field 1: short menu help (if available)
- Field 2: line and column (separated with colon)
- Field 3: current character position and total number of characters in document (separated with slash)
- Field 4: **R** if document is read-only
- Field 5: * if document has been modified
- Field 6: **O** in overwrite mode,
- Field 7: Current file format

See also

- Menu: Options > Toolbar

7.6. Character Info Bar



The Character Info Bar shows the following information:

- Field 1: character code value
- Field 2: character name (if assigned)
- Field 3: character category
- Field 4: character block (script, collection, etc.)

See also

- Menu: Options > Toolbar
- Infobar: Extended Character Info

7.7. Extended Character Info Bar



(used example: U+136D ETHIOPIC DIGIT FIVE, format: UTF-8)

The Extended Character Info Bar shows the following information:

- Field 1: decimal character code value
- Field 2: encoded byte sequence
- Field 3: bidirectional character type
- Field 4: decimal digit value (only if the character represents a decimal digit)

Please note: field 2 does not show the byte sequence as it exists in the file, but the byte sequence as it would be produced if the current single character would be encoded using the specified format. With some formats (UTF-7, SCSU) the same character may be encoded differently depending on the algorithm or current state of the encoder. This feature is most useful with UTF-8, UTF-16 and UTF-32.

See also

- Menu: Options > Toolbar
- Infobar: Character Info

7.8. Character Map



The Character Map allows an easy selection of any Unicode character by simply clicking a character cell. The character will be inserted at the current position in the current document. Clicking with the secondary mouse button opens a context menu with a variety of functions.

The arrow buttons can be used to browse through all 17 planes. The buttons have an acceleration effect. If pressed permanently the browsing speed increases.

The edit box can be used to move directly to a specific position. The behaviour is optimized for the Basic Multilingual Plane.

Example:

- Typing **1** positions the first column to U+1000
- Typing **10** positions the first column to U+1000
- Typing **10A** positions the first column to U+10A0
- Typing **10AD** positions the first column to U+10A0 and selects the character U+10AD

- Typing **10ADF** positions the first column to U+10AD0 and selects the character U+10ADF (in plane 1)
- Typing **10ADF3** positions the first column to U+10ADF0 and selects the character U+10ADF3 (in plane 16)

Additional buttons



If pressed down the unassigned codepoints are displayed with their default glyph . If released the cell remains blank. Normally you want the cells of unassigned code points to be blank, because this is clearer. You can press this button if you want to see explicitly which positions are not assigned.



Opens a short menu that allows you to select the font style used in the Character Map (automatic, variable width, fixed width). Default is automatic, depending on the font style of the current document.



Opens a help topic containing some information about the selected character block. This button is disabled if information is not available.

See also

- Context Menu: Character Map
- Reference: Mouse Commands
- Reference: Key Commands
- <http://www.unipad.org/unimap>

7.9. Virtual Keyboard



The Virtual Keyboard window has the following functions:

- It shows the assignment of current virtual keyboard (e.g. Czech).
- It allows you to enter characters by clicking the keys with the primary mouse button.
- It allows you to reassign the keys (either by drag-and-drop or explicit assignment).

Clicking with the secondary mouse button opens a context menu with a variety of functions.

Please note: if you choose the System keyboard, the window disappears. UniPad cannot display the assignment of the system keyboard driver.

See also

- Concepts: Virtual Keyboards
- Context Menu: Virtual Keyboard
- Reference: Mouse Commands
- <http://www.unipad.org/keyboard>

8. Reference

8.1. Key Commands

8.1.1. (Overview)

- Shortcuts for menu commands
- Navigation keys
- Special key combinations
- Aborting with [ESC]

8.1.2. Shortcuts

[CTRL+A]
Select all

[CTRL+C]
Copy selected text to the clipboard

[SHIFT+CTRL+C]
Copy as

[CTRL+E]
Export

[CTRL+F]
Find text

[CTRL+G]
Go to a specific line

[CTRL+I]
Import

[CTRL+N]
Create a new file

[CTRL+O]
Open an existing file

[CTRL+P]
Print the current file

[CTRL+Q]
Insert character

[CTRL+R]
Replace text

[CTRL+S]
Save the current file

[SHIFT+CTRL+S]
Save all

[CTRL+V]
Paste text from the clipboard

[SHIFT+CTRL+V]

Paste as

[CTRL+X]
Cut selected text

[CTRL+Y]
Delete the current line

[CTRL+Z]
Undo

[SHIFT+CTRL+Z]
Redo

[F1]
Shows this help file

[F2]
Toggle virtual keyboards

[F3]
Repeat last find

[F7]
Show/hide the Virtual Keyboard

[F8]
Show/hide the Character Map

[CTRL+F3]
Repeat last replacement

[CTRL+INS]
Copy selected text to the clipboard

[SHIFT+INS]
Paste text from the clipboard

[SHIFT+DEL]
Cut selected text

8.1.3. Navigation

[Arrow Left]
Move the cursor one character to the left.

[Arrow Right]
Move the cursor one character to the right.

[Arrow Up]
Move the cursor one line upwards.

[Arrow Down]
Move the cursor one line downwards.

[CTRL+Arrow Left]
Move the cursor one word to the left.

[CTRL+Arrow Right]
Move the cursor one word to the right.

[Page Up]
Move the cursor one page upwards.

[Page Down]
Move the cursor one page downwards.

[DEL]

Delete the current character or selection.

[INS]

Toggle between insert mode and overwrite mode.

[HOME]

Move the cursor to the beginning of the current line.

[END]

Move the cursor to the end of the current line.

[CTRL+HOME]

Move the cursor to the beginning of the file.

[CTRL+END]

Move the cursor to the end of the file.

[SHIFT+Arrow]

Move the cursor and expand the selection.

8.1.4. Special Key Combinations

[Left SHIFT+CTRL]

Change document direction to left-to-right

[Right SHIFT+CTRL]

Change document direction to right-to-left

8.1.5. Aborting with [ESC]

The tasks opening, saving, import and export can be aborted by pressing the [ESC] key.

Important: a short pressing of the key is not sufficient. You must hold down the [ESC] key until UniPad reacts and responds with a confirmation dialog. The dialog asks you to confirm abortion. If you do not confirm it, the task will be continued normally.

8.2. Mouse Commands

In editing window

- Double click text: select a word.

On main status bar

- Double click **Modified** field: mark the file as modified (this action is one-way, you cannot mark a file as not-modified when it has been modified).
- Double click **Format** field: open dialog Current Options > File.
- Double click **Overwrite Mode** field: toggles between overwrite mode and insert mode.
- Double click **Read Only** field: toggles between read-only and writable mode. Please note: this changes only the read-only flag of the document in UniPad and it does not change the file attributes of the physical file.

On character info bar

- Double click **Character Code** field: open Character Map and jump to the character position.
- Double click **Character Name** field: same behaviour.
- Double click **Character Block** field: open Character Map and select the character block.

On virtual keyboard

- Click primary mouse button: insert character into current document.
- Click secondary mouse button: open the context menu.

On character map

- Click primary mouse button: insert character into current document.
- Click secondary mouse button: open the context menu.

Drag-and-drop between character map and virtual keyboard

- Click on the character on the Character Map that you want to assign to a key and keep the mouse button pressed.
- Move the mouse over a key on the Virtual Keyboard window.
- Release the mouse button.

See also

- Reference: Key Commands
- Menu: Options > Toolbar
- Tool Window: Virtual Keyboard
- Tool Window: Character Map
- Infobar: Character Info
- Status Bar

8.3. Error Conditions

The following errors can occur during loading a file:

The non-character code XXXX(hex) occurred in the file.

These codes are not a valid Unicode characters. Note: this is only an error if the corresponding option is set.

An invalid byte sequence occurred in the file.

Invalid byte sequences may occur in corrupted UTF-8 files, for example.

An invalid byte value occurred in the file.

Invalid byte values may occur in a UTF-8 file, for example.

A non-canonical byte sequence occurred in the text.

A non-canonical byte sequence can be decoded correctly into a corresponding Unicode character. Its use is deprecated, however. An overlong UTF-8 sequence is an example of a non-canonical byte sequence.

An unexpected end of file occurred.

The end of the file was reached before a valid character could be decoded from the last byte sequence.

A character code greater than 10FFFF(hex) occurred in the file.

These code values are outside the scope of the Unicode standard and are not supported by UniPad.

A character code greater than 7FFFFFFF(hex) occurred in the file.

These code values are outside the scope of the Unicode standard and are not supported by UniPad. Additionally these code values are also outside the scope of ISO-10646 standard.

See also

- Concepts: Handling of U+0000 and Noncharacters
- Menu: File > Open
- Menu: File > Import
- Menu: Options > Preferences > Data
- Reference: Command Line Parameters

8.4. Command Line Parameters

UniPad recognizes several command line parameters mainly for integration in shells, batch files and scripts.

The general syntax of calling UniPad with parameters is:

up [general-options] [file-specific-options | file] ...

Most options have a short form (introduced with a - (one minus sign)) and a long form (introduced with -- (two minus signs)). Some options may have an extra argument. For a detailed discussion of the command line syntax, see the topic Appendix: General Command Line Syntax.

For example the following lines are equivalent:

```
up -r -e8 sample.txt
up --readonly --encoding=utf8 sample.txt
```

General options affect all files, file specific options affect only the following files. Options can be changed or turned on and off between different files. The following examples open the file1.txt as read-only and the file2.txt as writable:

```
up -r file1.txt -r file2.txt
up --readonly file1.txt --no-readonly file2.txt
```

The following sections discuss each option in detail.

-r

--readonly

Open the following files in read-only mode. Default is to open the files writable.

-e encoding

--encoding=encoding

Use the specified format ("encoding") when opening the following files. This option has only effect during execution of the command line parameters. Default is auto-detection. Possible values/aliases for the encoding are:

- Auto-detection: **auto a**
- UTF-16: **utf16 utf-16 16 ucs2 ucs-2 2**
- UTF-8: **utf8 utf-8 8**
- UTF-7: **utf7 utf-7 7**
- UTF-32: **utf32 utf-32 32 ucs4 ucs-4 4**
- ASCII+UCN: **ascii ascii-ucn ucn java j**
- SCSU: **scsu c**
- ASCII+XML references (hexadecimal): **xml-hex**
- ASCII+XML references (decimal): **xml-dec**

-E [m | x | i]

--onerror=[message | exit | ignore]

Control the behaviour when an error occurs during execution of the command line.

- message: Show a message box with an error message and wait for user confirmation (this is the default).
- exit: Exit immediately with an exit code.
- ignore: Ignore the error (if possible) and continue executing the command line.

-N

--nologo

Do not display a splash screen on startup.

-f [f | v]

--font=[fix | var]

Use the fixed-width font or the variable-width font when opening the following files. This option has only effect during execution of the command line parameters and does not change global settings.

-z

--maximize

Open the document windows maximized.

-y

--nohistory

Ignore the information from the file history when opening a file from the command line.

-n

--newdoc

UniPad opens a new document window upon start. This option may occur multiple times resulting in multiple new document windows.

-p

--print

Print all following files using the current printer settings.

-Y index

--recent=index

Open a file from the history of recent files specified by the index. Index 1 identifies the most recent file, 2 identifies the second most recent, and so on. Example: to open the last 3 files from your last editing session, you can create a shortcut and specify the following command line: `c:\...\lup.exe -Y3 -Y2 -Y1 .`

8.5. Exit Codes

The exit code may be useful, if you are running UniPad from a batch file or script.

The following exit codes are currently used:

- 0 (successful exit)
- 1 (unknown command line option)
- 5 (file could not be opened)
- 91 (executable file corrupt)

See also

- Reference: Command Line Options

8.6. Limitations

Fixed maxima

- Document size: 134,217,728 characters (see below)
- Line length: = max. document size
- Visible line length: 65536 characters (see below)
- Undo buffer: 8192 entries
- Tab width (variable): 320 pixels
- Tab width (fixed): 16 characters
- Keyboard name length: 127 characters (see below)
- User-defined keyboards: 256
- Find/Replace text length: 256 characters
- History: 64 entries
- Print zoom: 500 %

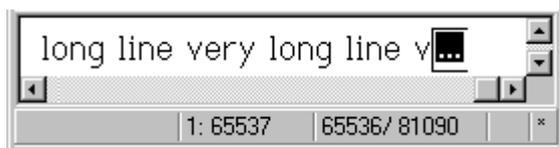
Document size

Please note: The document size is measured in characters; the actual maximum file size (in bytes) depends on the file format and can vary accordingly.

Visible line length

Rendering of a line is stopped after 65536 characters for performance reasons. That is, all the following characters up to the line end are rendered as one special replacement glyph. Cursor movement, selection and all other operations are not affected, the cursor can be moved logically through the invisible characters.

Example (shows the replacement glyph):



Keyboards

The name of a user-defined virtual keyboard is derived from the filename of the keyboard definition file (excluding directory path and extension `.ukb`). Keyboards with names longer than 127 characters can be loaded. However, only the first 127 characters of the name are used to distinguish it from other keyboards.

9. Appendix

9.1. Character Blocks (Unicode 3.2)

9.1.1. (Index)

These are the character blocks that are defined by the Unicode Standard Version 3.2. They are listed in alphabetical order.

Please note: only a rough overview can be given here. For detailed information about usage, special characters, shaping behavior and other topics please refer to the Unicode Standard.

- Alphabetic Presentation Forms
- Arabic
- Arabic Presentation Forms-A
- Arabic Presentation Forms-B
- Armenian
- Arrows
- Basic Latin
- Bengali
- Block Elements
- Bopomofo
- Bopomofo Extended
- Box Drawing
- Braille Patterns
- Buhid
- Byzantine Musical Symbols
- Cherokee
- CJK Compatibility
- CJK Compatibility Forms
- CJK Compatibility Ideographs
- CJK Compatibility Ideographs Supplement
- CJK Radicals Supplement
- CJK Symbols and Punctuation
- CJK Unified Ideographs
- CJK Unified Ideographs Extension A
- CJK Unified Ideographs Extension B
- Combining Diacritical Marks
- Combining Diacritical Marks for Symbols
- Combining Half Marks
- Control Pictures
- Currency Symbols
- Cyrillic
- Cyrillic Supplementary
- Deseret
- Devanagari
- Dingbats
- Enclosed Alphanumerics
- Enclosed CJK Letters and Months
- Ethiopic
- General Punctuation
- Geometric Shapes
- Georgian
- Gothic
- Greek and Coptic
- Greek Extended
- Gujarati
- Gurmukhi
- Halfwidth and Fullwidth Forms
- Hangul Compatibility Jamo

- Hangul Jamo
- Hangul Syllables
- Hanunoo
- Hebrew
- High Surrogates
- High Private Use Surrogates
- Hiragana
- Ideographic Description Characters
- IPA Extensions
- Kanbun
- Kangxi Radicals
- Kannada
- Katakana
- Katakana Phonetic Extensions
- Khmer
- Lao
- Latin-1 Supplement
- Latin Extended-A
- Latin Extended-B
- Latin Extended Additional
- Letterlike Symbols
- Low Surrogates
- Malayalam
- Mathematical Alphanumeric Symbols
- Mathematical Operators
- Miscellaneous Mathematical Symbols-A
- Miscellaneous Mathematical Symbols-B
- Miscellaneous Symbols
- Miscellaneous Technical
- Mongolian
- Musical Symbols
- Myanmar
- Number Forms
- Ogham
- Old Italic
- Optical Character Recognition
- Oriya
- Private Use Area
- Runic
- Sinhala
- Small Form Variants
- Spacing Modifier Letters
- Specials
- Superscripts and Subscripts
- Supplemental Arrows-A
- Supplemental Arrows-B
- Supplemental Mathematical Operators
- Supplementary Private Use Area-A
- Supplementary Private Use Area-B
- Syriac
- Tagalog
- Tagbanwa
- Tags
- Tamil
- Telugu
- Thaana
- Thai
- Tibetan
- Unified Canadian Aboriginal Syllabics
- Variation Selectors
- Yi Syllables
- Yi Radicals

See also

- Tool: Character Info Bar

9.1.2. Alphabetic Presentation Forms

Overview

- This block contains several variants of Unicode alphabetic characters. They are provided for the purpose of compatibility with existing character set standards and applications.
- The block includes Latin ligatures, Armenian ligatures and Hebrew presentation forms.

See Also

- Related blocks: Basic Latin, Armenian, Hebrew
- Term: presentation forms

9.1.3. Arabic

Overview

- The Arabic script is used to write the following languages: Arabic, Kurdish, Pashto, Persian, Sindhi, Urdu and other.
- The script is cursive, the characters are connected and may have a different form depending on its adjacent characters.
- Group: Middle Eastern scripts.
- Direction: from right to left (RTL) on a horizontal line.

Implementation

- UniPad supports both bidirectional text and shaping of Arabic contextual forms.
- Diacritical marks are automatically applied to Arabic base characters (the dead-key input method must be turned off).

Input Methods

- Built-in keyboards: Arabic, Persian

See Also

- Blocks with presentation forms: Arabic Presentation Forms-A, Arabic Presentation Forms-B
- Concepts: Handling of Bidirectional Text
- Concepts: Glyph Shaping

9.1.4. Arabic Presentation Forms-A

Overview

- This block contains several display variants of Arabic characters. They are provided for the purpose of compatibility with existing character set standards and legacy implementations.
- Characters include contextual variants of extended Arabic letters, contextual variants of ligatures, spacing forms of diacritic combinations, phrase ligatures and ornate forms of parentheses.
- Direction: RTL.

Implementation

- UniPad supports both bidirectional text and shaping of Arabic contextual forms of ligatures, if appropriate.
- Fixed width font: some ligatures occupy two or more character cells, due to complexity.

See Also

- Related blocks: Arabic, Arabic Presentation Forms-B

9.1.5. Arabic Presentation Forms-B

Overview

- This block contains several variants of Arabic characters. They are provided for the purpose of compatibility with existing character set standards and legacy implementations.
- Characters include contextual variants of basic Arabic letters, spacing forms of diacritics and the lam-alef ligature.
- Direction: RTL.

Implementation

- UniPad supports both bidirectional text and shaping of Arabic contextual forms of ligatures, if appropriate.

See Also

- Related blocks: Arabic, Arabic Presentation Forms-A

9.1.6. Armenian

Overview

- The Armenian script is used primarily to write the Armenian language.
- Letters are divided into uppercase and lowercase.
- Group: European scripts.
- Direction: from left to right (LTR) on a horizontal line.

Input Methods

- Built-in keyboard: Armenian

9.1.7. Arrows

Overview

- This block contains arrow symbols for various purposes, mainly mathematics.
- The arrows are named after their shape, not after their semantic value. Therefore a specific symbol may have more than one meaning.

Implementation

- Fixed width font: all arrows occupy only one character cell. Therefore horizontal arrows appear shorter as their vertical counterparts.

See Also

- Additional blocks: Supplemental Arrows-A, Supplemental Arrows-B

9.1.8. Basic Latin

Overview

- Most European nations (additionally some nations in Asia, all nations in America and Oceania and almost all countries in Africa) use the Latin script.
- Only a few languages can be written entirely with the Basic Latin script.
- Group: European scripts.
- Direction: from left to right (LTR) on a horizontal line.
- This character block is identical to the US-ASCII encoding (ISO/IEC 646). Functions of ASCII control codes are not defined by the Unicode Standard. However, like most applications UniPad recognizes U+0009 HORIZONTAL TABULATION as tabulator and U+000A LINE FEED and U+000D CARRIAGE RETURN as line separators.

Implementation

- U+000D and U+000A are automatically converted into U+2028 if inserted into a document.

- U+0009 is supported (visualisation of control characters must turned to "invisible").

Input Methods

- Several built-in keyboards use the Latin script.

See Also

- Additional Latin blocks: Latin-1 Supplement, Latin Extended-A, Latin Extended-B, Latin Extended Additional
- Related blocks: Alphabetic Presentation Forms, IPA Extensions

9.1.9. Bengali

Overview

- The Bengali script is used to write the following languages: Assam, Bengali, Daphla, Garo, Hallam, Khasi, Manipuri, Mizo, Munda, Naga, Rian, Santali.
- The shape of a Bengali character usually changes depending on its context.
- Group: North Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Bengali letters, nominal glyphs are displayed instead.
- Fixed width font: some characters occupy two character cells, due to complexity.

See Also

- Related scripts: Devanagari, Gurmukhi, Gujarati, Oriya

9.1.10. Block Elements

Overview

- This block contains several graphical characters. They are provided for the purpose of compatibility with existing character set standards and legacy implementations.
- These graphic characters were mainly used in semi-graphics, for example in teletext applications.

Implementation

- Fixed width font: although the block elements usually have a square form, all block elements occupy only one character cell.

See Also

- Similar graphical elements: Box Drawing, Geometric Shapes

9.1.11. Bopomofo

Overview

- Bopomofo is a syllabary used for Chinese, primarily in dictionaries and teaching books (it is not used in the writing of Chinese text).
- Group: East Asian scripts.
- Direction: from left to right (LTR) on a horizontal line, it can also be written vertically from top to bottom (TTB).

Implementation

- Fixed width font: all Bopomofo letters occupy two character cells.

See Also

- Additional Bopomofo letters: Bopomofo Extended

9.1.12. Bopomofo Extended

Overview

- This block contains additional Bopomofo syllables to cover the sounds of the Minnan and Hakka dialects.

Implementation

- Fixed width font: all Bopomofo letters occupy two character cells.

See Also

- Basic Bopomofo letters: Bopomofo

9.1.13. Box Drawing

Overview

- This block contains several graphical characters. They are provided for the purpose of compatibility with existing character set standards and legacy implementations.
- These graphic characters were mainly used in semi-graphics, for example in terminal emulations.

Implementation

- Fixed width font: although these characters usually have a square form, all of them occupy only one character cell.

See Also

- Similar graphical elements: Block Elements, Geometric Shapes

9.1.14. Braille Patterns

Overview

- Braille is a worldwide writing system used by blind people.
- All 256 possible patterns are encoded regardless of their meaning.
- On a Braille output device, black dots are rendered as tangible, white dots are rendered as non-tangible.
- Direction: from left to right (LTR).

Implementation

- The meaning of each pattern depends on the user community and also on context. A separate process is needed to convert characters of a given script into Braille patterns. No such process is defined by the Unicode Standard nor is such operation available in UniPad.

9.1.15. Buhid

Description

- Buhid is a Phillipine script that is still in use. Traditionally it was cut into bamboo.
- Direction: from left to right (LTR).

Implementation

- Automatic visual combination of Buhid vowel marks is not supported. Vowel marks are displayed as separate characters.

See Also

- Other Phillipine scripts: Tagalog, Tagbanwa, Hanunóo

9.1.16. Byzantine Musical Symbols

Description

- Byzantine Musical Symbols are used mainly to write the religious music and hymns of the Christian Orthodox Church.
- Direction: from left to right (LTR) in plain text. Display of the characters in two-dimensional layout is not defined by the Unicode Standard.

Implementation

- For legibility the Byzantine Musical Symbols are depicted larger than presented in the Unicode Standard.
- Fixed width font: some characters occupy two character cells, due to complexity.

See Also

- (Western) Musical Symbols

9.1.17. Cherokee

Description

- The Cherokee script is a syllabary used solely to write the Cherokee (or: Tsalagi) language.
- Direction: from left to right (LTR) on a horizontal line.

Input Methods

- Built-in keyboard: Cherokee

9.1.18. CJK Compatibility

Description

- This block contains symbols representing squared words (spelled in Katakana or Latin) that fill a single display cell. They are provided for the purpose of compatibility with existing character set standards and legacy implementations.

Implementation

- Fixed width font: these characters occupy two character cells.

See Also

- Other CJK non-ideographic compatibility characters: CJK Compatibility Forms, Halfwidth and Fullwidth Forms

9.1.19. CJK Compatibility Forms

Description

- This block contains vertical display variants (presentation forms) of CJK punctuation characters. They are provided for the purpose of compatibility with existing character set standards and legacy implementations.

Implementation

- Fixed width font: these characters occupy two character cells.

See Also

- Other CJK non-ideographic compatibility characters: CJK Compatibility, Halfwidth and Fullwidth Forms

9.1.20. CJK Compatibility Ideographs

Description

- This block contains CJK ideographs that are visually identical to ideographs in the CJK Unified Ideographs block. They have alternative pronunciations and were encoded separately in existing character sets. Therefore they are provided for the purpose of compatibility with existing character set standards and legacy implementations.

Implementation

- Fixed width font: these characters occupy two character cells.

See Also

- Additional compatibility ideographs: CJK Compatibility Ideographs Supplement
- Unified ideographs: CJK Unified Ideographs, CJK Unified Ideographs Extension A, CJK Unified Ideographs Extension B

9.1.21. CJK Compatibility Ideographs Supplement

Description

- This block contains additional compatibility ideographs (see also CJK Compatibility Ideographs). They are provided for the purpose of compatibility with existing character set standards and legacy implementations.

Implementation

- This block is not supported in the current version of UniPad. A glyph for unsupported characters is displayed instead: .

See Also

- Compatibility ideographs: CJK Compatibility Ideographs

9.1.22. CJK Radicals Supplement

Description

- This block contains CJK radical characters. Radicals are fragments of ideographs that are roughly related to the meaning of an ideograph. They are also used as the basis for creating new ideographs and are widely used for classification and grouping of ideographs in dictionaries.
- Radicals alone usually do not appear in normal text.

Implementation

- Fixed width font: these characters occupy two character cells.

See Also

- Other radicals: KangXi Radicals
- Ideographs: CJK Unified Ideographs

9.1.23. CJK Symbols and Punctuation

Description

- The CJK Symbols and Punctuation block contains several punctuation marks and other related symbols that are usually used only in ideographic text.

Implementation

- Fixed width font: most characters occupy two character cells.
- Automatic visual combination of non-spacing marks (here: tone marks) is not supported.

See Also

- Other punctuation: General Punctuation
- Ideographs: CJK Unified Ideographs
- Compatibility: CJK Compatibility, CJK Compatibility Forms, Halfwidth and Fullwidth Forms

9.1.24. CJK Unified Ideographs

Description

- This block contains common ideographic characters used to write Chinese, Japanese and Korean.
- Group: East Asian scripts.
- Direction: from left to right (LTR) on a horizontal line, it can also be written vertically from top to bottom (TTB).

Implementation

- Fixed width font: all ideographs occupy two character cells.
- Vertical text layout is not supported.

See Also

- Additional ideographs: CJK Unified Ideographs Extension A, CJK Unified Ideographs Extension B
- Compatibility ideographs: CJK Compatibility Ideographs, CJK Compatibility Ideographs Supplement
- Punctuation for use with ideographs: CJK Symbols and Punctuation

9.1.25. CJK Unified Ideographs Extension A

Description

- This block contains rare ideographic characters used to write Chinese, Japanese and Korean.
- Group: East Asian scripts.
- Direction: from left to right (LTR) on a horizontal line, it can also be written vertically from top to bottom (TTB).

Implementation

- Fixed width font: all ideographs occupy two character cells.
- Vertical text layout is not supported.

See Also

- Main ideographs block: CJK Unified Ideographs

9.1.26. CJK Unified Ideographs Extension B

Description

- The CJK Unified Ideographs Extension B block contains additional ideographic characters used to write Chinese, Japanese and Korean.
- Group: East Asian scripts.
- Direction: from left to right (LTR) on a horizontal line, it can also be written vertically from top to bottom (TTB).

Implementation

- This block is not supported in the current version of UniPad. A glyph for unsupported characters is displayed instead: .
- Vertical text layout is not supported.

See Also

- Main ideographs block: CJK Unified Ideographs

9.1.27. Combining Diacritical Marks

Description

- Combining Diacritical Marks are non-spacing characters that visually combine with a preceding base character.
- These marks are used in many scripts to (e. g. change the phonetic value of a letter).
- Combining marks that are specific to a certain script are usually encoded in the corresponding script block.

Implementation

- UniPad does not combine characters automatically. However, you can perform an explicit combination to produce pre-composed characters.

See Also

- Combining Diacritical Marks for Symbols
- For compatibility: Combining Half Marks

9.1.28. Combining Diacritical Marks for Symbols

Description

- Combining Diacritical Marks for Symbols are non-spacing characters that visually combine with a preceding base character.
- These marks are used especially for combination with mathematical, technical or other symbols rather than with letters.

Implementation

- UniPad does not support automatic display of combined characters.

See Also

- Combining Diacritical Marks

9.1.29. Combining Half Marks

Description

- The Combining Half Marks block contains fragments of double diacritics in the Combining Diacritical Marks block. They are provided for the purpose of compatibility with legacy implementations.

See Also

- Combining Diacritical Marks

9.1.30. Control Pictures

Description

- The Control Pictures block provides visible symbols for the otherwise invisible C0 control characters and the space character of the Basic Latin ("ASCII") block.

Implementation

- Since these characters are explicitly intended to be visible they behave not like real control characters.

9.1.31. Currency Symbols

Description

- This block contains various currency symbols.
- Currency symbols that are specific to a certain script are encoded in the corresponding script block.

9.1.32. Cyrillic

Description

- The Cyrillic script is used to write Russian and other Slavic scripts (Macedonian and Serbian languages). It is also used for several minority languages (such as Azerbaijani, Kazakh and Uzbek).
- Group: European scripts.

- Direction: from left to right (LTR) on a horizontal line.

Input Methods

- Several built-in keyboards use the Cyrillic script.

See Also

- Cyrillic Supplementary

9.1.33. Cyrillic Supplementary

Description

- This block contains additional Cyrillic characters.

See Also

- Basic Cyrillic

9.1.34. Dingbats

Description

- Dingbats are graphical symbols that are widely available in laser printers as "Zapf Dingbats" font.

See Also

- Some Dingbats symbols are already part of other blocks: Geometric Shapes, Enclosed Alphanumerics, Miscellaneous Symbols.

9.1.35. Deseret

Description

- Deseret is a phonemic alphabet devised to write the English language.
- The script was developed ca. 1850 at the University of Deseret (now the University of Utah). It was promoted by the Church of Jesus Christ of Latter-day Saints ("Mormon Church").
- Letters are divided into uppercase and lowercase.
- Direction: from left to right (LTR) on a horizontal line.

9.1.36. Devanagari

Description

- The Devanagari script is used to write the following languages: Awadhi, Bagheli, Bhatneri, Bhili, Bihari, Braj Bhasha, Chhattisgarhi, Garhwali, Gondi, Harauti, Ho, Jaipuri, Kachchhi, Kanauji, Konkani, Kului, Kumaoni, Kurku, Kurukh, Marwari, Mundari, Newari, Palpa, Santali.
- All Indian scripts are related and based on Devanagari (it is used for classical Sanskrit and Hindi).
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Devanagari letters, nominal glyphs are displayed instead.

Input Methods

- Built-in keyboard: Hindi

See Also

- Related scripts: Bengali, Gurmukhi, Gujarati, Oriya

9.1.37. Enclosed Alphanumerics

Description

- This block contains enclosed numbers and letters. They are provided for the purpose of compatibility with existing character set standards and legacy implementations.

See Also

- Other enclosed numbers: Dingbats

9.1.38. Enclosed CJK Letters and Months

Description

- This block contains enclosed CJK letters and (e.g. Katakana and Hangul) and enclosed ideographs for month names. They are provided for the purpose of compatibility with existing character set standards and legacy implementations.

Implementation

- Fixed width font: these characters occupy two character cells.

See Also

- Katakana, Hangul Syllables, CJK Unified Ideographs

9.1.39. Ethiopic

Description

- The Ethiopic script is a syllabary used to write the following languages: Amharic, Oromo, Tigre, Tigrinya.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- Fixed width font: most of the Ethiopic characters occupy two character cells.

9.1.40. General Punctuation

Description

- The General Punctuation block contains punctuation characters and similar symbols that may be used with any script.
- The block also contains several space characters and formatting characters.

Implementation

- Visualisation of special characters (formatting characters, spaces, etc.) can be changed.

See Also

- CJK specific punctuation: CJK Symbols and Punctuation

9.1.41. Geometric Shapes

Description

- The Geometric Shapes block contains various commonly used geometric shapes for several purposes.

See Also

- Similar graphical elements: Box Drawing, Dingbats

9.1.42. Georgian

Description

- The Georgian script is used to write Georgian.
- Group: European scripts.
- Direction: from left to right (LTR) on a horizontal line.

Input Methods

- Built-in keyboard: Georgian

9.1.43. Gothic

Description

- Gothic was used to write the Gothic language, which is an extinct East Germanic language.
- It was devised by the Gothic bishop Wulfila who lived in the 4th century.
- Direction: from left to right (LTR) on a horizontal line.

9.1.44. Greek and Coptic

Description

- The Greek script is used to write the following languages: Greek, Coptic.
- Group: European scripts.
- Direction: from left to right (LTR) on a horizontal line.

Input Methods

- Built-in keyboard: Greek

See Also

- Greek Extended

9.1.45. Greek Extended

Description

- The Greek Extended block contains precomposed combinations of Greek letters and one or more diacritical marks.

Implementation

- Fixed width font: some characters occupy two character cells, due to complexity.

See Also

- Basic Greek

9.1.46. Gujarati

Description

- The Gujarati script is used to write the Gujarati language.
- Group: Northern Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Gujarati letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Bengali, Devanagari, Gurmukhi, Oriya

9.1.47. Gurmukhi

Description

- The Gurmukhi script is used to write the Punjabi language. It is historically derived from Lahnda.
- Group: North Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Gurmukhi letters, nominal glyphs are displayed instead.

Input Methods

- Built-in keyboard: Punjabi

See Also

- Related scripts: Bengali, Devanagari, Gujarati, Oriya

9.1.48. Halfwidth and Fullwidth Forms

Description

- The Halfwidth and Fullwidth Forms block contains display variants of several characters from existing CJK character sets.
- In the days when double character sets for East Asian scripts evolved, usually single-byte characters (ASCII) occupied a single display cell, while double-byte characters (Katakana, Ideographs, etc.) occupied two display cells. However, certain double-byte characters (Katakana) were also provided as half-width characters (occupying a single display cell) and certain single-byte characters (e. g . Latin letters) were also provided as full-width characters (occupying two display cells).

Implementation

- These characters occupy either one character cell (halfwidth) or two character cells (fullwidth).

See Also

- Basic Latin, Katakana

9.1.49. Hangul Compatibility Jamo

Description

- Hangul is a syllabic script to write the Korean language.
- The Hangul Compatibility Jamo block contains Jamo characters that are provided for the purpose of compatibility with existing character set standards and legacy implementations.

See also

- Hangul Syllables, Hangul Jamo.

9.1.50. Hangul Jamo

Description

- Hangul is a syllabic script to write the Korean language.
- Jamo are the alphabetic components (letters) that are used to form a Hangul syllable. Two or three jamo (consonants and vocals) are arranged in a square.
- All modern syllables are available as pre-composed Hangul Syllables in the Unicode Standard.

Implementation

- Automatic arrangement of jamo components as Hangul syllables is not supported in this version of UniPad.

See Also

- Hangul Syllables, Hangul Compatibility Jamo

9.1.51. Hangul Syllables

Description

- Hangul is a syllabic script to write the Korean language.
- The Hangul Syllables block contains all modern pre-composed syllables.
- Direction: from left to right (LTR) on a horizontal line, it can also be written vertically from top to bottom (TTB).

Implementation

- Fixed width font: all syllables occupy two character cells.
- Vertical text layout is not supported.

See also

- Hangul Jamo

9.1.52. Hanunoo

Description

- Hanunóo is a Phillipine script that is still in use. Traditionally it was cut into bamboo.
- Direction: from left to right (LTR).

Implementation

- Automatic visual combination of Hanunóo combining marks is not supported. Combining marks are displayed as separate characters.

See Also

- Other Phillipine scripts: Tagalog, Tagbanwa, Buhid

9.1.53. Hebrew

Description

- The Hebrew script is used to write the following languages: Hebrew, Judezmo, Yiddish.
- Group: Middle Eastern scripts.
- Direction: from right to left (RTL) on a horizontal line.

Implementation

- UniPad supports bidirectional text.
- Automatic visual combination of Hebrew accents is not supported. Accents are displayed as separate characters.

Input Methods

- Built-in keyboard: Hebrew (including Yiddish)

See Also

- Hebrew presentation forms: Alphabetic Presentation Forms

9.1.54. Hiragana

Description

- The Hiragana script is a syllabary used to write Japanese phonetically (to indicate pronunciation of Japanese words). It is also used to write grammatical endings and conjunctions.
- Group: East Asian scripts.
- Direction: from left to right (LTR) on a horizontal line, it can also be written vertically from top to bottom (TTB).

Implementation

- Fixed width font: all syllables occupy two character cells.
- Vertical text layout is not supported.

Input Methods

- Built-in keyboard: Japanese (Kana only)

See Also

- Related syllabary: Katakana

9.1.55. Ideographic Description Characters

Description

- Ideographic Description Characters can be used with ideographs and radicals to graphically describe any ideograph whether it is encoded in the Unicode Standard or not.
- Using these characters is not a method to encode new ideographs.

Implementation

- These characters are not formatting characters, therefore they are visible.

See Also

- CJK Unified Ideographs, CJK Radicals Supplement

9.1.56. IPA Extensions

Description

- The IPA Extensions block contains symbols of the International Phonetic Alphabet (IPA), a standard system for speech sounds.
- IPA Extensions are an artificial script, most letters are derived from Latin script.
- Direction: from left to right (LTR) on a horizontal line.

See Also

- Latin blocks: Basic Latin, Latin-1 Supplement, Latin Extended-A, Latin Extended-B, Latin Extended Additional

9.1.57. Kanbun

Description

- Kanbun marks are characters that are used in Japanese texts to indicate the Japanese reading order of classical Chinese texts.

9.1.58. KangXi Radicals

Description

- The KangXi Radicals block contains CJK radical characters from 18th century KangXi dictionary. Radicals are fragments of

ideographs that are roughly related to the meaning of an ideograph. They are also used as the basis for creating new ideographs and are widely used for classification and grouping of ideographs in dictionaries.

- Radicals alone usually do not appear in normal text.

Implementation

- Fixed width font: these characters occupy two character cells.

See Also

- Other radicals: CJK Radicals Supplement
- Ideographs: CJK Unified Ideographs

9.1.59. Kannada

Description

- The Kannada script is used to write the following languages: Kannada, Tulu.
- Group: South Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Kannada letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Tamil, Telugu, Malayalam, Sinhala

9.1.60. Katakana

Description

- The Katakana script is used to write non-Japanese words phonetically in Japanese.
- Group: East Asian scripts.
- Direction: from left to right (LTR) on a horizontal line, it can also be written vertically from top to bottom (TTB).

Implementation

- Fixed width font: all syllables occupy two character cells.
- Vertical text layout is not supported.

Input Methods

- Built-in keyboard: Japanese (Kana only)

See Also

- Related syllabary: Hiragana
- Other related blocks: Katakana Phonetic Extensions, Halfwidth and Fullwidth Forms.

9.1.61. Katakana Phonetic Extensions

Description

- These Katakana syllables are used for the Japanese phonetic transcription of Ainu and other languages.

See Also

- Katakana

9.1.62. Khmer

Description

- The Khmer script (also known as Cambodian) is used to write Khmer, which is the official language in Kampuchea.
- Group: Southeast Asian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Khmer letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Thai, Myanmar

9.1.63. Lao

Description

- The Lao script is used to write Lao, it is related to Thai.
- Group: Southeast Asian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- Automatic visual combination of combining characters is not supported. These are displayed as separate characters.

See Also

- Related scripts: Thai

9.1.64. Latin-1 Supplement

Description

- The characters in this block can be used to write the following languages: Danish, Dutch, Faroese, Finnish, Flemish, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, Swedish.
- It extends the Basic Latin script with a miscellaneous set of punctuation and mathematical signs.
- Direction: from left to right (LTR) on a horizontal line.

See Also

- Basic Latin, Latin Extended-A, Latin Extended-B, Latin Extended Additional

9.1.65. Latin Extended-A

Description

- The characters in this block can be used to write the following languages: Afrikaans, Basque, Breton, Catalan, Croatian, Czech, Esperanto, Estonian, French, Frisian, Greenlandic, Hungarian, Latin, Latvian, Lithuanian, Maltese, Polish, Provençal, Rhaeto-Romanic, Romanian, Romany, Sami, Slovak, Slovenian, Sorbian, Turkish, Welsh and a couple of other languages
- By adding Latin Extended-A letters to the Basic Latin and Latin-1 Supplement blocks, you can represent most European languages that employ the Latin script.
- Group: European scripts.
- Direction: from left to right (LTR) on a horizontal line.

See Also

- Basic Latin, Latin-1 Supplement, Latin Extended-B, Latin Extended Additional

9.1.66. Latin Extended-B

Description

- The characters in this block are used to write additional languages and to extend Latin scripts.
- Direction: from left to right (LTR) on a horizontal line.

See Also

- Basic Latin, Latin-1 Supplement, Latin Extended-A, Latin Extended Additional

9.1.67. Latin Extended Additional

Description

- This block contains precomposed combinations of Latin letters with one or more diacritical marks.

See Also

- Basic Latin, Latin Extended-A, Latin Extended-B, Latin-1 Supplement

9.1.68. Letterlike Symbols

Description

- Letterlike Symbols are derived from alphabetic scripts, like Latin, Greek and Hebrew. However they are not used as letters, but as symbols in certain contexts (e. g. mathematics).

See Also

- More letterlike symbols: Mathematical Alphanumeric Symbols

9.1.69. Malayalam

Description

- The Malayalam script is used to write Malayalam.
- Group: South Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Malayalam letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Tamil, Telugu, Kannada, Sinhala

9.1.70. Mathematical Alphanumeric Symbols

Description

- This block contains stylistic variants of Latin letters, Greek letters and digits. They are intended to be used in mathematical text, for variables and other mathematical identifiers.
- These characters must not be used for writing ordinary text in a specific style.

Implementation

- Since these characters are used as symbols, case pairs are not defined. Therefore uppercase and lowercase conversion does not work on these characters. This is an intended limitation of the Unicode Standard.

See Also

- Derived from: Basic Latin, Greek
- Some letters are missing in this block because they have already been encoded in the Letterlike Symbols block.

9.1.71. Mathematical Operators

Description

- This block contains mainly mathematical operators and other mathematical symbols.
- Some arrow-like operators are encoded in the Arrows block.
- Several basic operators are encoded in the Basic Latin ("ASCII") block.

See Also

- More operators: Supplemental Mathematical Operators
- Related symbols: Miscellaneous Mathematical Symbols-A, Miscellaneous Mathematical Symbols-B

9.1.72. Miscellaneous Mathematical Symbols-A

Description

- This block contains symbols used mostly as operators or delimiters in mathematical notation.

See Also

- Operators: Mathematical Operators
- More symbols: Miscellaneous Mathematical Symbols-B

9.1.73. Miscellaneous Mathematical Symbols-B

Description

- This block contains miscellaneous symbols used for mathematical notation.

See Also

- Operators: Mathematical Operators
- More symbols: Miscellaneous Mathematical Symbols-A

9.1.74. Miscellaneous Symbols

Description

- This block contains various pictographic symbols.

See Also

- Similar graphical elements: Box Drawing, Dingbats, Geometric Shapes

9.1.75. Miscellaneous Technical

Description

- The Miscellaneous Technical block contains various technical symbols and functional symbols of APL (A Programming Language).
- Symbols that are usually used in two-dimensional diagrams only (such as electronic circuit symbols) are not included in the Unicode Standard.

See Also

- Miscellaneous Symbols

9.1.76. Mongolian

Description

- The Mongolian script is used to write a Mongolian literary language of classical origin.
- Direction: from top to bottom (TTB) on vertical columns shifting from left to right.

Implementation

- Vertical direction is not supported by UniPad. The characters are displayed from left to right instead.

9.1.77. Musical Symbols

Description

- Musical Symbols are used mainly to write Western music.
- Direction: from left to right (LTR) in plain text. Display of the characters in two-dimensional layout is not defined by the Unicode Standard.

Implementation

- Automatic visual combination of non-spacing marks is not supported.

See Also

- Byzantine Musical Symbols

9.1.78. Myanmar

Description

- The Myanmar (also known as Burmese) script is used to write the following languages: Burmese, Mon, Pali, Sanskrit, Shan.
- Group: Southeast Asian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Myanmar letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Tamil, Telugu, Kannada, Sinhala

9.1.79. Number Forms

Description

- The Number Forms block contains several number-like characters. Most of these characters are provided for the purpose of compatibility with existing character set standards and legacy implementations.

9.1.80. Ogham

Description

- The Ogham script was used in the 5th and 6th centuries in England, Ireland, Isle of Man, Scotland and Wales; it was usually written on stones.

- Group: European scripts.
- Direction: from left to right (LTR) or from bottom to top (BTT).

Input Methods

- Built-in keyboard: Ogham

Implementation

- Vertical direction is not supported by UniPad. The characters are displayed from left to right instead.

9.1.81. Old Italic

Description

- The Old Italic scripts were used to write the Etruscan language and other extinct languages located on the Italian peninsula.
- These scripts have been unified by the Unicode Standard into a single character block.
- Direction: from left to right (LTR) or from right to left (RTL) on a horizontal line. Nowadays (in research) mainly LTR direction is used, therefore this is default direction defined by the Unicode Standard. To change directionality, explicit directional formatting codes can be used.

9.1.82. Optical Character Recognition

Description

- The Optical Character Recognition block contains special symbols used in OCR-A character set and the MICR (magnetic ink character recognition) character set.

9.1.83. Oriya

Description

- The Oriya script is used to write the following languages: Khondi, Oriya, Santali.
- Group: North Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Oriya letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Bengali, Gurmukhi, Gujarati, Devanagari

9.1.84. Private Use Area

Description

- The Private Use Area is reserved for use in implementations where characters need to be encoded that are otherwise not encoded in the Unicode Standard.

Implementation

- UniPad does not support loadable fonts for the Private Use Area yet.
- All private use characters are displayed as *.

See Also

- Reserved planes for private use: Supplemental Private Use Area-A, Supplemental Private Use Area-B

9.1.85. Runic

Description

- The Runic script was used in the 3rd-17th century to write ancient German (it was also used in Scandinavian and Anglo-Saxon areas).
- Group: European scripts.
- Direction: from left to right (LTR) on a horizontal line.

9.1.86. Sinhala

Description

- The Sinhala script is used to write the following languages: Pali, Sanskrit, Sinhala.
- Group: South Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Sinhala letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Tamil, Telugu, Kannada

9.1.87. Small Form Variants

Description

- The Small Form Variants block contains small display variants of some characters that are provided for the purpose of compatibility with existing character set standards and legacy implementations.

See Also

- Basic Latin

9.1.88. Spacing Modifier Letters

Description

- Spacing Modifier Letters are generally used to indicate a modification of the preceding letter.
- The letters are typically used in phonetic alphabets (e. g. IPA) or in transcriptions.
- Many of the letters correspond to diacritical marks.

See Also

- Combining Diacritical Marks

9.1.89. Specials

Description

- The Specials block contains characters for specific purposes of software implementations, such as the Replacement Character.
- The BOM logically belongs to this block. However, due to its character code value it is encoded in the Arabic Presentation Forms-B block.

Implementation

- Formatting characters for interlinear annotation are not supported.

9.1.90. Superscripts and Subscripts

Description

- The characters in the Superscripts and Subscripts block are provided for the purpose of compatibility with existing character set standards and legacy implementations.

See Also

- Basic Latin

9.1.91. Supplemental Arrows-A

Description

- The Supplemental Arrows-A block contains additional arrow symbols to supplement the Arrows block.

See Also

- Arrows, Supplemental Arrows-B

9.1.92. Supplemental Arrows-B

Description

- The Supplemental Arrows-B block contains additional arrow symbols to supplement the Arrows block.

See Also

- Arrows, Supplemental Arrows-A

9.1.93. Supplemental Mathematical Operators

Description

- The Supplemental Mathematic Operators block contains additional operator symbols to supplement the Mathematical Operators block.

See Also

- Mathematical Operators

9.1.94. Supplementary Private Use Area-A

Description

- The entire plane 15 is reserved for private use.

Implementation

- UniPad does not support loadable fonts for the Private Use Area yet.
- All private use characters are displayed as .

See Also

- Private Use Area, Supplemental Private Use Area-B

9.1.95. Supplementary Private Use Area-B

Description

- The entire plane 16 is reserved for private use.

Implementation

- UniPad does not support loadable fonts for the Private Use Area yet.
- All private use characters are displayed as .

See Also

- Private Use Area, Supplemental Private Use Area-A

9.1.96. Syriac

Description

- The Syriac script is cursive, the characters are connected and may have a different form depending on it's adjacent characters (the script is divided into two dialects: West- and East Syriac).
- Group: Middle Eastern scripts.
- Direction: from right to left (RTL) on a horizontal line.

Input Methods

- Built-in keyboard: Syriac

Implementation

- UniPad supports bidirectional text.
- UniPad does not support Syriac shaping, the nominal glyphs are displayed instead.

See Also

- Related script: Arabic
- Concepts: Handling of Bidirectional Text

9.1.97. Tagalog

Description

- Tagalog is an extinct Phillipine script. Tradionally it was cut into bamboo.
- Direction: from left to right (LTR).

Implementation

- Automatic visual combination of Tagalog vowel marks is not supported. Vowel marks are displayed as separate characters.

See Also

- Other Phillipine scripts: Buhid, Tagbanwa, Hanunóo

9.1.98. Tagbanwa

Description

- Tagbanwa is a Phillipine script that is still in use. Tradionally it was cut into bamboo.
- Direction: from left to right (LTR).

Implementation

- Automatic visual combination of Tagbanwa vowel marks is not supported. Vowel marks are displayed as separate characters.

See Also

- Other Phillipine scripts: Buhid, Tagalog, Hanunóo

9.1.99. Tags

Description

- This block contains special invisible control characters that were intended for inserting meta-information into Unicode plain text.
- The currently only defined usage is the insertion of language and locale information into plain text. This method is known as "Plane 14 Language Tags".
- The use of Tags is deprecated now.

Implementation

- UniPad does not interpret Plane 14 Language Tags.
- Tags are formatting characters and their visibility can be controlled.

9.1.100. Tamil

Description

- The Tamil script is used to write the following languages: Badaga, Tamil.
- Group: South Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Tamil letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Malayalam, Telugu, Kannada, Sinhala

9.1.101. Telugu

Description

- The Telugu script is used to write the following languages: Ghondi, Lambadi, Telugu.
- Group: South Indian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Telugu letters, nominal glyphs are displayed instead.

See Also

- Related scripts: Malayalam, Tamil, Kannada, Sinhala

9.1.102. Thaana

Description

- The Thaana script is used to write the modern Divehi language.
- Group: Middle Eastern scripts.
- Direction: from right to left (RTL) on a horizontal line.

Implementation

- UniPad supports bidirectional text.
- Automatic visual combination of accents is not supported. Accents are displayed as separate characters.

Input Methods

- Built-in keyboard: Divehi

See Also

- Related script: Arabic
- Concepts: Handling of Bidirectional Text

9.1.103. Thai

Description

- The Thai script is used to write the following languages: Kuy, Lavna, Pali, Thai.
- Group: Southeast Asian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- Automatic visual combination of combining characters is not supported. These are displayed as separate characters.

Input Methods

- Built-in keyboard: Thai

See Also

- Related scripts: Lao

9.1.104. Tibetan

Description

- The Tibetan script is used to write the following languages: Dzongkha, Tibetan.
- Group: Southeast Asian scripts.
- Direction: from left to right (LTR) on a horizontal line.

Implementation

- UniPad does not support contextual forms of Tibetan letters, nominal glyphs are displayed instead.

9.1.105. Unified Canadian Aboriginal Syllabics

Description

- The Canadian Aboriginal Syllabics are mostly used by several aboriginal groups throughout Canada (Inuit language).
- Direction: from left to right (LTR) on a horizontal line.

Input Methods

- Built-in keyboard: Inuktitut

Implementation

- Fixed width font: some characters occupy two character cells, due to complexity.

9.1.106. Variation Selectors

Description

- Variation Selectors are special characters that behave like combining diacritical marks but have no shape on its own (i. e. they are normally invisible). They are used to change the shape of the preceding base character in a defined manner.

Implementation

- Although they are formally non-spacing marks, UniPad handles Variation Selectors like formatting characters.
- The shaping behaviour (i. e. selection of glyph variants) is currently not supported by UniPad.

See Also

- The variants are defined by the Unicode Standard (see <http://www.unicode.org/Public/UNIDATA/StandardizedVariants.html> for more information).

9.1.107. Yi Syllables

Description

- The Yi script is a syllabary used to write Yi (member of the Sino-Tibetan language).
- Group: East Asian scripts.
- Direction: from left to right (LTR) on a horizontal line, it can also be written vertically from top to bottom (TTB).

Implementation

- Vertical text layout is not supported.

See Also

- Yi Radicals

9.1.108. Yi Radicals

Description

- The Yi Radicals block contains Yi radical characters. Yi radicals are fragments of Yi syllables. They are used for classification and grouping of Yi syllables in dictionaries.
- Radicals alone usually do not appear in normal text.

See Also

- Yi Syllables

9.2. Keyboards

9.2.1. (Index)

This is an alphabetic list of the built-in keyboards.

Albanian
Arabic
Armenian
Azeri
Bulgarian
Bulgarian (Latin)
Byelorussian
Cherokee
Croatian
Czech
Danish
Divehi
Dutch
English (British)
English (US International)
English (US)
English (US, Dvorak)
Esperanto
Estonian

Finnish
French (Belgian)
French (Standard)
Gaelic
Georgian
German (Standard)
German (Swiss)
Greek
Hawaiian
Hebrew
Hindi
Hungarian
Icelandic
Inuktitut
ISO 9995 Complementary
ISO 9995 Secondary
ISO 9995 Unified
Italian
Japanese (Kana only)
Kazakh
Kyrgyz
Latin American
Latvian
Lithuanian
Macedonian
Mongolian (Cyrillic)
Ogham
Persian (ISIRI 2901)
Polish
Portuguese
Punjabi
Romanian
Russian
Serbian (Latin)
Slovak
Spanish
Swedish
Syriac
Tatar
Thai
Turkish (F-Type)
Turkish (Q-Type)
Ukrainian
Urdu
Uzbek
Vietnamese

9.2.2. Albanian

Description

- Keyboard for the Albanian language.

Implementation

- This keyboard is based on the Microsoft standard.
- U+20AC EURO SIGN and U+00A9 COPYRIGHT SIGN have been added as AltGr keys.

See Also

- Basic Latin, Latin-1 Supplement

9.2.3. Arabic

Description

- Keyboard for the Arabic language.

See Also

- Script: Arabic

9.2.4. Armenian

Description

- Keyboard for the Armenian language.

Implementation

- This keyboard is based on the "Armenian Standard Layout".
- The Latin digits 0 through 9, U+20AC EURO SIGN, and U+00A9 COPYRIGHT SIGN have been added as AltGr keys

See Also

- Script: Armenian

9.2.5. Azeri

Description

- Keyboard for the Azeri (Azerbaijani) language.
- Supports Latin and Cyrillic.

Implementation

- This keyboard is a combination of the Microsoft Azeri Latin and Azeri Cyrillic keyboards
- The Latin layout is assigned to the normal and Shift keys, while the Cyrillic layout is assigned to the AltGr and Shift + AltGr keys. This is because Azerbaijan is in the process of converting from the Cyrillic (old) to the Latin (new) script.

See Also

- Scripts: Latin, Cyrillic

9.2.6. Bulgarian

Description

- Keyboard for the Bulgarian language.
- Uses Cyrillic script.

See Also

- Script: Cyrillic

9.2.7. Bulgarian (Latin)

Description

- Keyboard for the Bulgarian language.
- Uses Latin script.

See Also

- Script: Latin

9.2.8. Byelorussian

Description

- Keyboard for the Byelorussian (Belarus) language.

See Also

- Script: Cyrillic

9.2.9. Cherokee

Description

- Keyboard for the Cherokee (or: Tsalagi) language.

Implementation

- This keyboard is based on the mapping provided by the Cherokee Nation of Tahlequah, Oklahoma, based on their "Official Cherokee Font".
- The AltGr and Shift+AltGr keys correspond to the English (US) keyboard.

See Also

- Script: Cherokee
- Cherokee Nation website: <http://www.cherokee.org>

9.2.10. Croatian

Description

- Keyboard for the Croatian language.

See Also

- Script: Latin

9.2.11. Czech

Description

- Keyboard for the Czech language.

See Also

- Script: Latin

9.2.12. Danish

Description

- Keyboard for the Danish language.

See Also

- Script: Latin

9.2.13. Divehi

Description

- Keyboard for the Divehi language.
- Uses the Thaana script.

Implementation

- This keyboard is based on the Microsoft "Divehi Typewriter" layout.

See Also

- Script: Thaana

9.2.14. Dutch

Description

- Keyboard for the Dutch language.

See Also

- Script: Latin

9.2.15. English (British)

Description

- Keyboard for the English language as used in Great Britain.

See Also

- Script: Latin
- Related keyboards: Dvorak, US, US International

9.2.16. English (US International)

Description

- Keyboard for the English language as used in the USA.
- Provides many additional Latin (accented) characters for other languages.

See Also

- Script: Latin, Latin-1 Supplement
- Related keyboards: Dvorak, British, US

9.2.17. English (US)

Description

- Keyboard for the English language as used in the USA.

See Also

- Script: Latin
- Related keyboards: Dvorak, British, US International

9.2.18. English (US, Dvorak)

Description

- The Dvorak keyboard is used to write English. The layout was designed to be more ergonomic than the original QWERTY layout.

Implementation

- This keyboard is based on the Microsoft definition for the Dvorak keyboard.

See Also

- Script: Latin
- Keyboards: US, US International, British

9.2.19. Esperanto

Description

- Keyboard for Esperanto, an artificial language that uses the Latin script.

See Also

- Scripts: Latin
- Websites: <http://lingvo.org/klavaro>, <http://esperanto.net>

9.2.20. Estonian

Description

- Keyboard for the Estonian language.

See Also

- Script: Latin

9.2.21. Finnish

Description

- Keyboard for the Finnish language.

See Also

- Script: Latin

9.2.22. French (Belgian)

Description

- Keyboard for the French language as used in Belgium.

See Also

- Script: Latin
- Related keyboard: French (Standard)

9.2.23. French (Standard)

Description

- Keyboard for the French language as used in France.

See Also

- Script: Latin
- Related keyboard: French (Belgian)

9.2.24. Gaelic

Description

- Keyboard for the Gaelic language.

Implementation

- This keyboard is based on the Microsoft standard.
- U+00A9 COPYRIGHT SIGN has been added as an AltGr key.

See Also

- Script: Latin

9.2.25. Georgian

Description

- Keyboard for the Georgian language.

Implementation

- This keyboard is based on the Microsoft standard.
- U+00A9 COPYRIGHT SIGN has been added as an AltGr key.

See Also

- Script: Georgian

9.2.26. German (Standard)

Description

- Keyboard for the German language as used in Germany.

Implementation

- The [Caps Lock] key on a German keyboard shifts all keys. To simulate this behaviour, you should turn off the option CapsLock applies to alphabetic keys only.

See Also

- Script: Latin
- Related keyboard: German (Swiss)

9.2.27. German (Swiss)

Description

- Keyboard for the German language as used in Switzerland.

See Also

- Script: Latin
- Related keyboard: German (Standard)

9.2.28. Greek

Description

- Keyboard for the Greek language.

See Also

- Script: Greek

9.2.29. Hawaiian

Description

- Keyboard for the Hawaiian language.

See Also

- Script: Latin

9.2.30. Hebrew

Description

- Keyboard for the Hebrew language.
- The keyboard can also be used to type Yiddish.

Implementation

- Based on the specification available at <http://www.yv.org/uyip/hebyidkbd.txt>.

See Also

- Website: "Understanding Yiddish Information Processing" (<http://www.uyip.org>)

9.2.31. Hindi

Description

- Keyboard for the Hindi language.

See Also

- Script: Devanagari

9.2.32. Hungarian

Description

- Keyboard for the Hungarian language.

See Also

- Script: Latin

9.2.33. Icelandic

Description

- Keyboard for the Icelandic language.

See Also

- Script: Latin

9.2.34. Inuktitut

Description

- Keyboard for the Inuktitut (Escimo) language.

Implementation

- This keyboard is based on the Inuktitut Keyboard Specification, version 1.1 of the Legislative Assembly of Nunavut. It differs from the official specification in the following ways:
- The [Caps Lock] key is not used to switch between the Latin and Inuktitut scripts, because this special use of Caps Lock is not supported by UniPad. This is an Inuktitut-only keyboard (except for LATIN LETTER L WITH STROKE). To type other Latin letters, it is necessary to select a different keyboard, e. g. with the F2 key.
- Dead keys for typing dotted syllables are not supported, because these characters are not decomposable into a base character and combining diacritical mark in Unicode.
- Key assignments from other references were added where they do not conflict with the specification.

See Also

- Script: Unified Canadian Aboriginal Syllabics
- "Writing Inuktitut with Computers: A Manual and Guide," by Jack Cain
- "Inuktitut Computing and Unicode: A New Beginning" by Gavin Nesbitt

9.2.35. ISO 9995 Complementary

Description

- This keyboard implements the "complementary Latin group" described in ISO/IEC 9995, Keyboard layouts for text and office systems - Part 3: Complementary layouts of the alphanumeric zone of the alphanumeric section

Implementation

- Keys for U+00A0 NO-BREAK SPACE and U+00AD SOFT HYPHEN are provided as recommended in ISO/IEC 9995-2

See Also

- Related keyboards: ISO 9995 Secondary, ISO 9995 Unified
- Script: Latin

9.2.36. ISO 9995 Secondary

Description

- This keyboard implements the "common secondary group" described in ISO/IEC 9995, Keyboard layouts for text and office systems - Part 3: Complementary layouts of the alphanumeric zone of the alphanumeric section

Implementation

- The character generated by key E04, Level 3 is U+00A4 CURRENCY SIGN, not U+0024 DOLLAR SIGN, which is permitted as an alternative. The rationale is that primary keyboards (with which this secondary group is intended to be used) are more likely to have a key for DOLLAR SIGN than for CURRENCY SIGN. A key for U+00A0 NO-BREAK SPACE is provided as recommended in ISO/IEC 9995-2.

See Also

- Related keyboards: ISO 9995 Complementary, ISO 9995 Unified
- Script: Latin

9.2.37. ISO 9995 Unified

Description

- This keyboard is a composite of the "complementary Latin group" and the "common secondary group" described in ISO/IEC 9995, Keyboard layouts for text and office systems - Part 3: Complementary layouts of the alphanumeric zone of the alphanumeric section

Implementation

- Group 2 is supposed to be accessible by pressing [Shift] + [AltGr] (equivalent to [Shift] + [Ctrl] + [Alt]), but in UniPad this combination accesses a "Level 4" that is not defined or sanctioned by ISO 9995. Levels 3 and 4 in this keyboard correspond to Group 2, Levels 1 and 2 in the ISO 9995 definition.
- Some character names in the ISO 9995 description did not match the Unicode name exactly (in particular "Music note," which is ambiguous and which is interpreted here as U+266A EIGHTH NOTE). U+00A4 CURRENCY SIGN is supposed to be available in Group 2, Level 3, key E04. It is not available from this keyboard.

See Also

- Related keyboards: ISO 9995 Complementary, ISO 9995 Secondary
- Script: Latin

9.2.38. Italian

Description

- Keyboard for the Italian language.

See Also

- Script: Latin

9.2.39. Japanese (Kana only)

Description

- Keyboard to type the Japanese Katakana and Hiragana syllables.

Implementation

- This keyboard is based on the Microsoft standard. The Microsoft "Japanese Hiragana" keyboard comprises 20 states, using the standard [Shift] and [AltGr] keys as well as special keys to select "English" or kana, to select Hiragana or Katakana, and to enable fullwidth Latin or halfwidth Katakana. This keyboard implements a subset of the Microsoft keyboard, with four states to cover Hiragana (plain and [AltGr]) and Katakana ([Shift] and [Shift] + [AltGr]). Input of Latin should be handled by selecting a Latin-based keyboard with the F2 key. Because the Hiragana/Katakana key on the Microsoft keyboard is a locking key, that functionality is allocated on this keyboard to the [Shift] key, while small variants and punctuation are accessed using the [AltGr] key.
- This keyboard does not interact with a Standard Windows IME.

See Also

- Scripts: Katakana, Hiragana

9.2.40. Kazakh

Description

- Keyboard for the Kazakh language.

Implementation

- This keyboard is based on the Microsoft standard.
- The Latin digits 0 through 9, U+20AC EURO SIGN, and U+00A9 COPYRIGHT SIGN have been added as [AltGr] keys.

See Also

- Script: Cyrillic

9.2.41. Kyrgyz

Description

- Keyboard for the Kyrgyz language.

Implementation

- This keyboard is based on the Microsoft standard.
- The Microsoft Kyrgyz keyboard uses the [Caps Lock] state to access U+0406 and U+0456 CYRILLIC LETTER BYELORUSSIAN-UKRAINIAN I; because UniPad does not distinguish [Caps Lock] as a separate state, these characters are provided (on the same key) in the [AltGr] and [Shift] + [AltGr] states.

See Also

- Script: Cyrillic

9.2.42. Latin American

Description

- Keyboard for the Latin American languages.

See Also

- Script: Latin

9.2.43. Latvian

Description

- Keyboard for the Latvian language.

See Also

- Script: Latin

9.2.44. Lithuanian

Description

- Keyboard for the Lithuanian language.

Implementation

- This keyboard is based on the Microsoft standard.
- U+00A9 COPYRIGHT SIGN has been added as an [AltGr] key.

See Also

- Script: Latin

9.2.45. Macedonian

Description

- Keyboard for the Macedonian language.

Implementation

- This keyboard is based on the Microsoft standard.
- U+00A9 COPYRIGHT SIGN has been added as an [AltGr] key.

See Also

- Script: Cyrillic

9.2.46. Mongolian (Cyrillic)

Description

- Keyboard to write the Mongolian language using the Cyrillic script.
- The Mongolian script is rarely used with computers.

Implementation

- This keyboard is based on the Microsoft standard.
- U+20AC EURO SIGN and U+00A9 COPYRIGHT SIGN have been added as [AltGr] keys.

See Also

- Scripts: Cyrillic

9.2.47. Norwegian

Description

- Keyboard for the Norwegian language.

See Also

- Script: Latin

9.2.48. Ogham

Description

- Keyboard to write Ogham.

Implementation

- This keyboard is derived from the Final Draft of Irish Standard 434:1997 (1997-09-20).
- The [AltGr] and [Shift] + [AltGr] keys correspond to the English (US) keyboard.

See Also

- Script: Ogham

9.2.49. Persian (ISIRI 2901)

Description

- Keyboard for the Persian language.

Implementation

- Based on the Iranian Standard ISIRI-2901.

See Also

- Script: Arabic

9.2.50. Polish

Description

- Keyboard for the Polish language.

See Also

- Script: Latin

9.2.51. Portuguese

Description

- Keyboard for the Portuguese language.

See Also

- Script: Latin

9.2.52. Punjabi

Description

- Keyboard for the Punjabi language.

See Also

- Script: Gurmukhi

9.2.53. Romanian

Description

- Keyboard for the Romanian language.

See Also

- Script: Latin

9.2.54. Russian

Description

- Keyboard for the Russian language.

See Also

- Script: Cyrillic

9.2.55. Serbian (Latin)

Description

- Keyboard for the Serbian language (using the Latin script).

See Also

- Script: Latin

9.2.56. Slovak

Description

- Keyboard for the Slovak language.

See Also

- Script: Latin

9.2.57. Spanish

Description

- Keyboard for the Spanish language.

See Also

- Script: Latin

9.2.58. Swedish

Description

- Keyboard for the Swedish language.

See Also

- Script: Latin

9.2.59. Syriac

Description

- Keyboard for the Syriac language.

Implementation

- This keyboard is based on the Microsoft "Syriac Standard" layout.
- In the Microsoft reference, the glyphs for U+00AB and U+00BB, LEFT-POINTING and RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK, are reversed. In this keyboard, these two keys are assigned according to Microsoft's textual description, not the glyphs.

See Also

- Script: Syriac

9.2.60. Tatar

Description

- Keyboard for the Tatar language.

Implementation

- This keyboard is based on the Microsoft standard.
- U+20AC EURO SIGN and U+00A9 COPYRIGHT SIGN have been added as [AltGr] keys.

See Also

- Script: Cyrillic

9.2.61. Thai

Description

- Keyboard for the Thai language.

See Also

- Script: Thai

9.2.62. Turkish (F-Type)

Description

- Keyboard for the Turkish language.

See Also

- Script: Latin
- Related keyboard: Turkish (Q-Type)

9.2.63. Turkish (Q-Type)

Description

- Keyboard for the Turkish language.

See Also

- Script: Latin
- Related keyboard: Turkish (F-Type)

9.2.64. Ukrainian

Description

- Keyboard for the Ukrainian language.

See Also

- Script: Cyrillic

9.2.65. Urdu

Description

- Keyboard for the Urdu language.

Implementation

- Please note: Nastaliq style characters are not separately encoded in the Unicode Standard.

See Also

- Script: Arabic

9.2.66. Uzbek

Description

- Keyboard for the Uzbek language.

Implementation

- This keyboard is based on the Microsoft standard.
- U+20AC EURO SIGN and U+00A9 COPYRIGHT SIGN have been added as [AltGr] keys.

See Also

- Script: Cyrillic

9.2.67. Vietnamese

Description

- Keyboard for the Vietnamese language.

See Also

- Script: Latin

9.3. Formats

UniPad supports the following Unicode formats:

- UTF-16
- UTF-32
- UTF-8
- ASCII with Universal Character Names
- UTF-7
- Standard Compression Scheme for Unicode
- ASCII with XML character references

9.4. Character Sets

UniPad supports the following character sets:

- ASCII
- ISO/IEC 8859-1 .. 8859-16 (except -12)

- Code pages 1250 .. 1258
- DOS code pages (SBCS only)
- Macintosh encodings
- KOI8-R/U
- Code page 932 (Shift-JIS)
- Several Vietnamese encodings
- EUC-KR (Korean)
- ARMSCI-8 (Armenian)
- GEOSTD8 (Georgian)
- TIS-620 (Thai)
- Big Five (Traditional Chinese)
- CNS 11643, EUC-TW (Traditional Chinese)
- GB 2312, EUC-CN (Simplified Chinese)
- JIS X 0208, EUC-JP (Japanese)

9.5. Additional Information

UniPad Home Page

<http://www.unipad.org>

Version history

<http://www.unipad.org/history>

(also lists minor corrections and all bug fixes)

Frequently Asked Questions

<http://www.unipad.org/faq>

Task List

<http://www.unipad.org/tasklist>

(informs you about known bugs, planned features, wishes by users, etc.)

Unicode Web Site

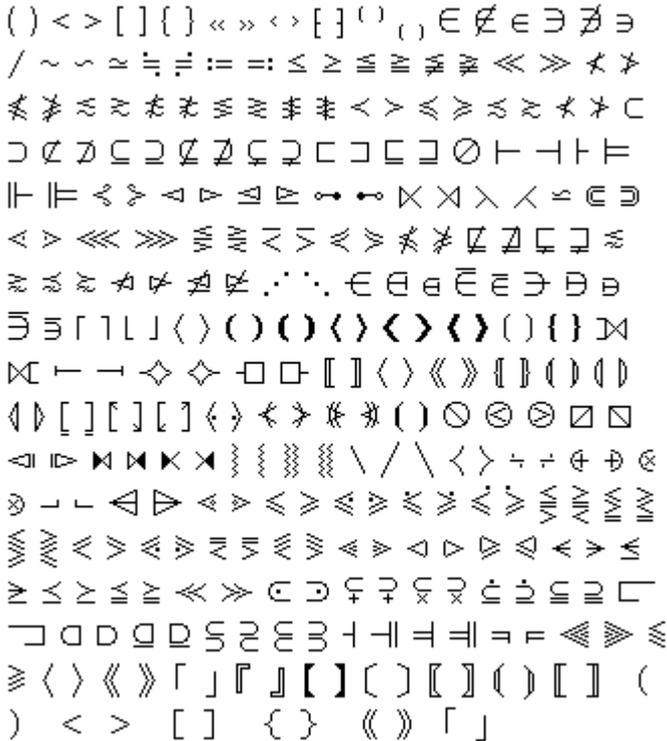
<http://www.unicode.org>

(the official web site of Unicode Consortium)

9.6. Manual Installation

Please refer to section IV of the SC Unipad Read Me file to get more information about this topic.

9.7. Mirrored Characters



9.8. General Command Line Syntax

This topic provides a detailed discussion of the general command line syntax and may be of interest if you want to call UniPad from scripts or batch files.

For an overview about the available command line parameters, refer to the topic Reference: Command Line Parameters.

Forms

Most command line switches are available as a long form and a short form. The following list shows all possible forms and their meanings.

- o** Short form without argument (semantics: turn on a boolean option).
- option** Long form of **-o** .
- o+** Short form without argument (same as **-o**, provided for symmetry to **-o-**).
- o arg** Short form with argument (semantics: use this argument for the option rather than a default).
- o=arg** Same as **-o arg** (no spaces are allowed around the equals sign).
- oa** Short form with a one-character argument (same as **-o arg** if the specific option provides a one-character form of its possible arguments).
- option arg** Long form with argument (same as **-o arg**).
- option=arg** Same as **--option arg** (no spaces are allowed around the equals sign).
- o-** Short form without argument (semantics: turn off a boolean option).
- nooption** Long form of **-o-** .
- no-option** Long form variant of **-o-** .
- nooption=arg** Theoretically possible but not used by UniPad (semantics: turn off the option, but use the argument for the next time, when it is turned on).

Switch Packing

The short forms of the switches can be packed for easier invocation on the command line. For example **-r -y -q** can be packed as **-ryq** .Switches that have an argument can only be packed if a one-character form of the argument is provided. For example **-e utf-8 -f var -q** can be packed as **-e8fvq** . Switches with turn-off-semantics can be packed, too. For example **-y -r-** can be packed as **-y-r-** .

Inhibit / Activate Switch Parsing

To temporarily inhibit recognition of switches, use **--** (two minus signs). To re-activate recognition of switches use **-+** (minus sign, plus sign). This will allow you to use strings beginning with a minus sign as normal free parameter (e.g. as a file name). For

example in the command line `-e8 -- -file -- -q` the parameter `-file` will not be treated as a switch.

Blanks

Normally, blanks (tabulator and space) act as a parameter separator. To use a space in a single parameter (e.g. a file name) enclose the parameter in double quotes like this: `up "my file"` .

Empty Strings

To provide an empty string as a parameter use two double quotes like this: `up --some-option=""` .

Index

.

.UTX 30, 32

\

\\u

convert from 15
convert to 15

2

2901 (ISIRI Standard) 89

8

8859 (ISO/IEC Standard) 93

9

9995 (ISO standard) 86-87

A

aborting with ESC 47
accents 59-60
acoustic signals 32
Afrikaans (language) 68
Albanian (keyboard) 79
Alphabetic Presentation Forms (Unicode character block) 53
Amharic (language) 62
ANS 68
Arabic (keyboard) 80
Arabic (language) 53
Arabic (script) 37, 53-54, 80, 89, 93
Arabic (Unicode character block) 53
Arabic Presentation Forms-A (Unicode character block) 53
Arabic Presentation Forms-B (Unicode character block) 54
Armenian (keyboard) 80
Armenian (language) 54
Armenian (script) 53-54, 80
Armenian (Unicode character block) 54
arrow symbols 54, 74
Arrows (Unicode character block) 54
artificial languages 83
ASCII 35, 54, 93
Assam (language) 55
assigning characters to keys 44
attributes
file 33
auto-detection 30
auto-indent 34
Awadhi (language) 61
Azerbaijani (language) 80
Azerbaijani (language) 60
Azeri (keyboard) 80
Azeri (language) 60

B

backup 30
Badaga (language) 76

- Bagheli (language) 61
- Basic Latin (Unicode character block) 54
- Basque (language) 68
- Bengali (language) 55
- Bengali (script) 55
- Bengali (Unicode character block) 55
- Bhasha (language) 61
- Bhatneri (language) 61
- Bhili (language) 61
- bidirectional text 7
 - change options 19, 37
 - mirrored characters 94
 - printing 10
- Bihari (language) 61
- Block Elements (Unicode character block) 55
- BOM 6, 30
 - strip 30
- Bopomofo (script) 55-56
- Bopomofo (Unicode character block) 55
- Bopomofo Extended (Unicode character block) 56
- Box Drawing (Unicode character block) 56
- Bra (language) 61
- bracket
 - find 17-18
- Braille (script) 56
- Braille Patterns (Unicode character block) 56
- Breton (language) 68
- Buhid (language) 56
- Buhid (script) 56
- Buhid (Unicode character block) 56
- built-in keyboards 78
- Bulgarian (keyboard) 80
- Burmese (language) 71
- Byelorussian (keyboard) 81
- byte order 6
 - change default 30
- byte order mark 6
- byte sequence 43
 - invalid 48
 - non-canonical 48
- Byzantine Musical Symbols (Unicode character block) 57

C

- Cambodian (language) 68
- Caps Lock 31
- captions
 - key 31
- caret shape 31
- Catalan (language) 68
- character
 - inserting 39
- character block 42
- character blocks 51
- character category 42
- character map
 - accelerated navigation 32
 - clicking 47
 - window 43
- character name 42
- character sets 4, 93
- characters 3, 6, 31, 51
 - byte order mark (BOM) 6
 - combining 6, 59-60
 - compatibility 53-54, 57-58, 62, 64, 71, 73-74
 - control 35, 60
 - decimal code 43
 - enclosed 62
 - formatting 35, 73, 77
 - hexadecimal code 42

- insert 3, 14
- invisible 35
- line separator 6
- mirrored 94
- non- 7, 31, 48
- NUL 7
- private use 37, 72, 74-75
- replacement and substitution 31
- space 35
- surrogate 37
- unassigned 36
- unsupported 36
- check for external modification 30
- Cherokee (keyboard) 81
- Cherokee (language) 57
- Cherokee (script) 57, 81
- Cherokee (Unicode character block) 57
- Chhattisgarhi (language) 61
- Chinese (script) 55-59, 62, 64, 66
- Church of Jesus Christ of Latter-day Saints 61
- CJK Compatibility (Unicode character block) 57
- CJK Compatibility Forms (Unicode character block) 57
- CJK Compatibility Ideographs (Unicode character block) 57
- CJK Compatibility Ideographs Supplement (Unicode character block) 58
- CJK Radicals Supplement (Unicode character block) 58
- CJK Symbols and Punctuation (Unicode character block) 58
- CJK Unified Ideographs (Unicode character block) 59
- CJK Unified Ideographs Extension A (Unicode character block) 59
- CJK Unified Ideographs Extension B (Unicode character block) 59
- clicking (mouse commands) 47
- clipboard 3
 - copy to 13, 28
 - cut to 13
 - paste from 13-14, 28
- code pages 93
- combine characters 14
- Combining Diacritical Marks (Unicode character block) 59
- Combining Diacritical Marks for Symbols (Unicode character block) 60
- Combining Half Marks (Unicode character block) 60
- command line parameters 48, 95
- context forms 37
- context menu
 - application 24
 - character map 24, 43
 - document 23
 - edit field 25
 - keyboard 23, 44
 - toolbars and statusbar 24
- control characters 35
- Control Pictures (Unicode character block) 60
- conversion 38
 - \\u to Unicode 15
 - combine characters 14
 - decompose characters 15
 - toolbar 42
 - Unicode to \\u 15
 - Unicode to XML/SGML character entities 16
 - Unicode to XML/SGML character references 16
 - uppercase and lowercase 14
 - XML/SGML character entities to Unicode 16
 - XML/SGML character references to Unicode 15
- convert 4
- Coptic (language) 63
- copy text 13, 28
- create 2
 - new file / document 2
- create new document 48
- create new file / document 9
- create new keyboard 20
- Croatian (keyboard) 81
- Croatian (language) 68
- Currency Symbols (Unicode character block) 60

- cursor movement
 - bidirectional text 7
- cursor shape 31
- cut text 13
- Cyrillic (script) 60-61, 80-81, 87-90, 92-93
- Cyrillic (Unicode character block) 60
- Cyrillic Supplementary (Unicode character block) 61
- Czech (keyboard) 81
- Czech (language) 68

D

- Danish (keyboard) 81
- Danish (language) 68
- Daphla (language) 55
- dead-key input method 34
- decompose characters 15
- default byte order 30
- default options 20
- delete line 14
- Deseret (script) 61
- Deseret (Unicode character block) 61
- Devanagari (script) 61, 85
- Devanagari (Unicode character block) 61
- diacritical marks 59-60
- digit value 43
- Dingbats (Unicode character block) 61
- Divehi (keyboard) 82
- Divehi (language) 76
- document 2-3
 - create new 2, 48
 - editing 3
 - open 2
 - size limitations 50
- document options 33
- double-clicking (mouse commands) 47
- drag and drop 44
- Dutch (keyboard) 82
- Dutch (language) 68
- Dvorak keyboard 83
- Dzongkha (language) 77

E

- editing 3
 - insert character 39
- editing options 34
- email 11
- Enclosed Alphanumerics (Unicode character block) 62
- enclosed characters 62
- Enclosed CJK Letters and Months (Unicode character block) 62
- end of file
 - unexpected 48
- English 30
- English (keyboard) 82-83
- English (language) 54, 61
- errors 38, 48
 - conditions and messages 48
 - control behaviour 31, 48
 - suppress 31
- ESC key 47
- Escimo 86
- Esperanto (keyboard) 83
- Esperanto (language) 68
- Estonian (keyboard) 83
- Estonian (language) 68
- Ethiopic (script) 62
- Ethiopic (Unicode character block) 62
- exit code 12, 50
- exit program 12, 50

export 4, 11, 27
extension
 default 30
extension (.UTX) 32
external modification 30

F

Faroese (language) 68
file 2-4
 create new 9
 creating new 2
 editing 3
 export 4, 11, 27
 import 4, 11, 27
 open 2, 9, 26
 reload 10
 save 9, 26
 save unaltered 32
file attributes 33
file extension
 default 30
file history 12
 maximum 30
file options 33
file properties 11, 33
find bracket 17-18
find text 16-17, 28
Finnish (keyboard) 83
Finnish (language) 68
fixed width font style 34
Flemish (language) 68
font 5, 34
 current 18
 in character map 43
 select style 48
format 42
formats 93
 specifying on command line 48
 using BOM 6
formatting characters 35
French (keyboard) 83-84
French (language) 68
Frisian (language) 68

G

Gaelic (keyboard) 84
Garhwali (language) 61
Garo (language) 55
General Punctuation (Unicode character block) 62
Geometric Shapes (Unicode character block) 62
Georgian (keyboard) 84
Georgian (language) 63
Georgian (script) 63, 84
Georgian (Unicode character block) 63
German 30
German (keyboard) 84
German (language) 68, 73
Ghondi (language) 76
global keyboard 34
glyph
 change shaping options 19
 shaping options 37
glyph shaping 8
go to line number 17, 28
Gondi (language) 61
Gothic (language) 63
Gothic (script) 63
Gothic (Unicode character block) 63

- Greek (keyboard) 85
- Greek (language) 63
- Greek (script) 63, 85
- Greek and Coptic (Unicode character block) 63
- Greek Extended (Unicode character block) 63
- Greenlandic (language) 68
- Gujarati (language) 63
- Gujarati (script) 63
- Gujarati (Unicode character block) 63
- Gurmukhi (script) 64, 90
- Gurmukhi (Unicode character block) 64

H

- half combining marks 60
- Halfwidth and Fullwidth Forms (Unicode character block) 64
- Hallam (language) 55
- Hangul Compatibility Jamo (Unicode character block) 64
- Hangul Jamo (Unicode character block) 64
- Hangul Syllables (Unicode character block) 65
- Hanunoo (language) 65
- Hanunoo (script) 65
- Hanunoo (Unicode character block) 65
- Harauti (language) 61
- Hawaiian (keyboard) 85
- Hawaiian (language) 54
- Hebrew (keyboard) 85
- Hebrew (language) 65
- Hebrew (script) 53, 65, 85
- Hebrew (Unicode character block) 65
- help 22
- hexadecimal input 14
- Hindi (keyboard) 85
- Hiragana (script) 66, 87
- Hiragana (Unicode character block) 66
- Ho (language) 61
- Hungarian (keyboard) 85
- Hungarian (language) 68

I

- Icelandic (keyboard) 86
- Icelandic (language) 68
- Ideographic Description Characters (Unicode character block) 66
- ideographs 57-59, 66
- IME
 - compatibility options 32
- import 4, 11, 27, 32
- inch 30
- infobar
 - character 42-43
 - clicking 47
- input method
 - dead-key 34
- input methods 3
- insert 3
 - characters 3
- insert characters 14, 39
- insert mode 34
- insert modus 42
- installation
 - manual 94
- Inuktitut (keyboard) 86
- Inuktitut (language) 77
- invalid byte sequence 48
- invisible characters 35
- IPA Extensions (Unicode character block) 66
- Irish (language) 68
- ISIRI 2901 89
- ISO 8859 93

ISO 9995 (keyboard) 86-87
Italian (keyboard) 87
Italian (language) 68

J

Jaipuri (language) 61
Jamo 64
Japanese (keyboard) 87
Japanese (language) 66-67
Japanese (script) 57-59, 62, 64, 66
Judezmo (language) 65

K

Kachchhi (language) 61
Kana (script) 66-67, 87
Kanauji (language) 61
Kanbun (Unicode character block) 66
KangXi Radicals (Unicode character block) 66
Kannada (language) 67
Kannada (script) 67
Kannada (Unicode character block) 67
Katakana (script) 57, 67, 87
Katakana (Unicode character block) 67
Katakana Phonetic Extensions (Unicode character block) 67
Kazakh (keyboard) 87
Kazakh (language) 60
key captions 31
keyboard
 assign characters 38
 change preferences 31
 clicking 47
 create new 20
 load 21
 open definition file 22
 quick toggle 19
 reload 21
 save 21
 select 34
 unload 21
 window 44
keyboard definition file
 open 22
keyboard window 31
keyboards 5
 built-in 78
 name limitations 50
 user-defined 31
keys 45
 commands 47
 ESC (abort) 47
 menu shortcuts 45
 navigation 46
Khasi (language) 55
Khmer (language) 68
Khmer (script) 68
Khmer (Unicode character block) 68
Khondi (language) 72
Konkani (language) 61
Korean (language) 64-65
Korean (script) 62, 64-65
Kului (language) 61
Kumaoni (language) 61
Kurdish (language) 53
Kurku (language) 61
Kurukh (language) 61
Kuy (language) 77
Kyrgyz (keyboard) 88

L

- Lambadi (language) 76
- Language Tags 76
- languages 51
- Lao (language) 68
- Lao (script) 68
- Lao (Unicode character block) 68
- Latin (language) 54, 68
- Latin (script) 53-54, 57, 66, 68-69, 79-93
- Latin American (keyboard) 88
- Latin Extended Additional (Unicode character block) 69
- Latin Extended-A (Unicode character block) 68
- Latin Extended-B (Unicode character block) 69
- Latin-1 Supplement (Unicode character block) 68
- Latvian (keyboard) 88
- Latvian (language) 68
- Lavna (language) 77
- Letterlike Symbols (Unicode character block) 69
- letters 62, 69
 - modifier 73
 - phonetic 66
- limitations 50
- line
 - delete 14
- line number
 - go to 17, 28
- line separator 6
- Lithuanian (keyboard) 88
- Lithuanian (language) 68
- load keyboard 21
- locales
 - Unicode 32
- lowercase
 - convert to 14
 - exceptions 69

M

- Macedonian (keyboard) 89
- Macedonian (language) 60
- Malayalam (language) 69
- Malayalam (script) 69
- Malayalam (Unicode character block) 69
- Maltese (language) 68
- Manipuri (language) 55
- manual installation 94
- MAPI 11
- Marwari (language) 61
- Mathematical Alphanumeric Symbols (Unicode character block) 69
- mathematical operators 69-70, 74
- Mathematical Operators (Unicode character block) 70
- mathematical symbols 69-70, 74
- MICR 72
- millimeters 30
- mirrored characters 7, 94
- Miscellaneous Mathematical Symbols-A (Unicode character block) 70
- Miscellaneous Mathematical Symbols-B (Unicode character block) 70
- Miscellaneous Symbols (Unicode character block) 70
- Miscellaneous Technical (Unicode character block) 70
- Mizo (language) 55
- mm 30
- modification 42
 - external 30
- modifier letters 73
- Mon (language) 71
- Mongolian (keyboard) 89
- Mongolian (language) 71
- Mongolian (script) 71
- Mongolian (Unicode character block) 71

month names 62
 Mormon script 61
 mouse commands 47
 Munda (language) 55
 Mundari (language) 61
 musical symbols 57, 71
 Musical Symbols (Unicode character block) 71
 Myanmar (script) 71
 Myanmar (Unicode character block) 71

N

Naga (language) 55
 Nastaliq font style 93
 navigation 3
 keys 46
 Newari (language) 61
 non-canonical byte sequence 48
 non-characters 7, 31, 48
 Norwegian (keyboard) 89
 Norwegian (language) 68
 NULL 7
 Number Forms (Unicode character block) 71
 numbers 71
 numeric keypad 31

O

OCR 72
 Ogham (keyboard) 89
 Ogham (script) 71, 89
 Ogham (Unicode character block) 71
 Old Italic (Unicode character block) 72
 open 2
 file 2
 open file 9, 26
 open keyboard 21
 operators
 mathematical 69-70, 74
 Optical Character Recognition (Unicode character block) 72
 options 33
 bidirectional text 19, 37
 character rendering 18
 command line 95
 current document 18
 default 20
 document direction 19
 editing 34
 file properties 11, 33
 font 34
 glyph shaping 19, 37
 preferences 20, 29
 rendering 34
 Oriya (language) 72
 Oriya (script) 72
 Oriya (Unicode character block) 72
 Oromo (language) 62
 overlong UTF-8 sequences 31
 overwrite mode 34
 overwrite modus 42

P

page setup 10, 27
 Pali (language) 71, 73, 77
 Palpa (language) 61
 parameters
 command line 48
 Pashto (language) 53

- paste text 13-14, 28
- Persian (keyboard) 89
- Persian (language) 53
- phonetic alphabet 66
- Plane 14 Language Tags 76
- Polish (keyboard) 90
- Polish (language) 68
- Portuguese (keyboard) 90
- Portuguese (language) 68
- preferences 20, 29-32
- presentation forms 53-54
- printing 10
 - from command line 48
 - inch or millimeters 30
 - page setup 10, 27
 - zoom limitations 50
- Private Use Area (Unicode character block) 72
- private use characters 37, 72, 74-75
- properties
 - file 11, 33
- Provençal (language) 68
- punctuation symbols 58, 62
- Punjabi (keyboard) 90
- Punjabi (language) 64

Q

- quick toggle 19, 31
- quotes (in command line) 95

R

- radicals
 - Chinese 58
 - Japanese 58
 - KangXi 66
 - Yi 78
- readonly
 - open file 48
- read-only modus 42
- recent files 12
 - maximum 30
 - open from command line 48
- redo 13
- registration code 39
- reload file 10
- reload keyboard 21
- rendering 18
- rendering options 34
- replace text 17, 29
- replacement character 31
- Rhaeto-Romanic (language) 68
- Rian (language) 55
- Romanian (keyboard) 90
- Romanian (language) 68
- Romany (language) 68
- Runic (script) 73
- Runic (Unicode character block) 73
- Russian (keyboard) 90
- Russian (language) 60

S

- Sami (language) 68
- Sanskrit (language) 71, 73
- Santali (language) 55, 61, 72
- save file 9, 26
- save keyboard 21
- scripts 51

- SCSU 11, 93
- search and replace 17, 29
- search in text 16-17, 28
- selecting 14
- selecting text
 - bidirectional text 7
- sending email 11
- Serbian (keyboard) 91
- Serbian (language) 60
- SGML character entities
 - convert from 16
 - convert to 16
- SGML character references
 - convert from 15
 - convert to 16
- Shan (language) 71
- shortcuts 45
- Sindhi (language) 53
- Sinhala (language) 73
- Sinhala (script) 73
- Sinhala (Unicode character block) 73
- Slovak (keyboard) 91
- Slovak (language) 68
- Slovenian (language) 68
- Small Form Variants (Unicode character block) 73
- Sorbian (language) 68
- spaces 6, 35
 - stripping 34
 - zero-width no-break 6
- spaces (in command line) 95
- Spacing Modifier Letters (Unicode character block) 73
- Spanish (keyboard) 91
- Spanish (language) 68
- Specials (Unicode character block) 73
- splash screen
 - turn off 32
 - turning off 48
- statusbar 42
 - clicking 47
- strip initial BOM 30
- strip spaces 34
- suffix
 - backup 30
- Superscripts and Subscripts (Unicode character block) 74
- Supplemental Arrows-A (Unicode character block) 74
- Supplemental Arrows-B (Unicode character block) 74
- Supplemental Mathematical Operators (Unicode character block) 74
- Supplementary Private Use Area-A (Unicode character block) 74
- Supplementary Private Use Area-B (Unicode character block) 75
- suppress errors 31
- surrogates 37
- Swedish (keyboard) 91
- Swedish (language) 68
- switch
 - command line 95
- syllables
 - Canadian Aboriginal 77
 - Cherokee 57
 - Japanese 66-67
 - Korean 64-65
 - Yi 78
- symbols 51, 60, 69-70, 74
 - arrow 54, 74
 - currency 60
 - graphical 55-56, 61-62, 70
 - mathematical 69-70, 74
 - musical 57, 71
 - OCR 72
 - phonetic 66
 - punctuation 58, 62
 - technical 70
- syntax

- command line 95
- Syriac (keyboard) 91
- Syriac (script) 75, 91
- Syriac (Unicode character block) 75

T

- tabulator
 - limitations 50
- tabulator width 34
- Tagalog (language) 75
- Tagalog (script) 75
- Tagalog (Unicode character block) 75
- Tagbanwa (language) 75
- Tagbanwa (script) 75
- Tagbanwa (Unicode character block) 75
- Tags (Unicode character block) 76
- Tamil (language) 76
- Tamil (script) 76
- Tamil (Unicode character block) 76
- Tatar (keyboard) 92
- technical symbols 70
- Telugu (language) 76
- Telugu (script) 76
- Telugu (Unicode character block) 76
- Thaana (script) 76, 82
- Thaana (Unicode character block) 76
- Thai (keyboard) 92
- Thai (language) 77
- Thai (script) 77, 92
- Thai (Unicode character block) 77
- Tibetan (language) 77
- Tibetan (script) 77
- Tibetan (Unicode character block) 77
- Tigre (language) 62
- Tigrinya (language) 62
- toggle keyboard 19, 31
- toolbar
 - conversion 42
 - input 41
 - standard 41
 - view options 41
- toolbars
 - change visibility 19
- trialware 39
- triplets 32
- Tsalagi (language) 57
- Tulu (language) 67
- Turkish (keyboard) 92
- Turkish (language) 68
- typing 5

U

- U+0000 7
- U+2028 LINE SEPARATOR 6
- U+2029 PARAGRAPH SEPARATOR 6
- U+FEFF
 - strip 30
- U+FEFF ZERO WIDTH NO-BREAK SPACE 6
- U+FFFD REPLACEMENT CHARACTER 31
- U+FFFE 6
- UCN
 - convert from 15
 - convert to 15
- Ukrainian (keyboard) 92
- unaltered files 32
- unassigned codepoints 36
- undo 12
 - limitations 50

- maximum steps 31
- undo after save 31
- unexpected end of file 48
- Unicode Language Tags 76
- Unicode locales 32
- Unified Canadian Aboriginal Syllabics 86
- Unified Canadian Aboriginal Syllabics (Unicode character block) 77
- unit of measure 30
- Universal Character Names
 - convert from 15
 - convert to 15
- unload keyboard 21
- unpaired surrogates 37
- unsupported characters 36
- uppercase
 - convert to 14
 - exceptions 69
- Urdu (keyboard) 93
- Urdu (language) 53
- US keyboard 82-83
- user language 30
- user-defined keyboards 31
- UTF-16 93
- UTF-32 93
- UTF-7 93
 - encoding options 30
- UTF-8 31, 93
 - overlong sequences 31
- UTX (extension) 30
- UTX (file extension) 32
- Uzbek (keyboard) 93
- Uzbek (language) 60

V

- variable width font style 34
- Variation Selectors (Unicode character block) 77
- Vietnamese (keyboard) 93
- virtual keyboard 31

W

- Welsh (language) 68
- Western musical symbols 71
- windows
 - arrange 22
 - character map 43
 - keyboard 44
 - open maximized 32, 48

X

- XML character entities
 - convert from 16
 - convert to 16
- XML character references
 - convert from 15
 - convert to 16

Y

- Yi (language) 78
- Yi (script) 78
- Yi Radicals (Unicode character block) 78
- Yi Syllables (Unicode character block) 78
- Yiddish (keyboard) 85
- Yiddish (language) 65