# UnicodeChecker

*This release should conform to The Unicode Standard, Version 4.0*

**What is it?**

UnicodeChecker is a utility to browse the Unicode character set. For any character you select, UnicodeChecker can tell you the decimal Unicode, the hexadecimal Unicode, the hexadecimal UTF-8, UTF-16 and UTF-32 code, the Unicode name and lots more. Note that there is ToolTip help available at many places.
You can also go the other way round: By typing or pasting a character in the “Glyph” field and hitting the return key or dropping a string on the main window, UnicodeChecker will present the according character information.
You can find codepoints by their corresponding Unicode name using the “Find Codepoint” command from the "Edit" menu.

**String Utilities**

Some utilities for performing Unicode related conversions and operations on strings instead of just a single character are available from the “File” menu (“New Utility Window”). The utilities recalculate on every change to the “Input” text field. UnicodeChecker currently includes the following utilities:

*HTML*

Converts Unicode codepoints into the according HTML entity representation or the other way round. When converting to HTML entities you can specify if you want to convert codepoints from the low ASCII range (lower than U+0080) as well (this only affects the ampersand '&", lower than '<', greater than '>' and double quote '"' characters).

*Split up*

Any text you enter is split up into its Unicode codepoints.

*Normalization*

Normalizes any given string using the four Unicode Normalization Forms specified in Unicode Standard Annex #15 found at <http://www.unicode.org/unicode/reports/tr15/>. An exclamation mark ("!") is displayed to the right of any normalized form that differs from the provided input string.

*IDNA*

This is an implementation of RFC 3490 “Internationalizing Host Names In Applications (IDNA)” <http://www.ietf.org/rfc/rfc3490.txt>. Please read this file if you need further information.
In contrast to the ToUnicode conversion, the ToASCII conversion may fail for some input strings. Although UnicodeChecker informs you whether the conversion succeeded or not, the exact reason is not displayed. More verbose error information is written to StdErr and can be viewed using the “Console” application.

*URL*
Adds or replaces percent escape codes for use in URLs. Escape codes are always assumed to be UTF-8. Note that although escaping codeplains out of Plane 0 (BMP) works without any problems, unescaping those will fail due to current limitations in the according function in Apple's CoreFoundation.

**Services**
When UnicodeChecker is placed in one of the "Applications" folders or its subfolders, UnicodeChecker provides several conversion services for other applications supporting the services mechanism via the "Services" menu in the application's main menu. UnicodeChecker's services are available from the "Unicode" submenu. Note that you must log out and back in after installing UnicodeChecker in order for these services to appear.

**AppleScript**
Some UnicodeChecker features are scriptable. You can ask UnicodeChecker to convert non-ASCII codepoints in strings to their HTML entity representation and the other way round. This may be useful if you want to generate HTML pages via AppleScript using data from sources like the MacOS Address Book or you want to extract data from HTML pages.
You can have UnicodeChecker add or replace percent escape sequences (e.g. "%20") for use in URLs (UTF-8 encoding is assumed).
You can convert strings to or from IDNA. If the conversion to IDNA fails an empty string is returned.
Also, you can have UnicodeChecker display character information in its main window.
The documentation for scripting support can be viewed from Script Editor's "Open Dictionary…" menu command.
If you would like to see additional AppleScript support in UnicodeChecker please drop me a line.

**Tell me more**
The data files used in this release of UnicodeChecker are taken from the Unicode Character Database directory <http://www.unicode.org/Public/UNIDATA/> at the time of the release.
If the Unicode Consortium releases new versions of this file, new versions of UnicodeChecker should become available. However, you can just replace the files inside the folder "Unicode Data" within the application bundle with the new versions found on unicode.org yourself. If you do so, please replace all files in the "Unicode Data" folder with the according version to ensure consistent behaviour in UnicodeChecker.
Some of the "Utilities" (namely *Normalization* and *IDN*) utilize Unicode Normalization Forms. The conformance of UnicodeChecker's normalization with the Unicode Standard at the time of this release has been verified. If you replace the data files in

UnicodeChecker it is a good idea to run the normalization test from the “File” menu. You need to download the appropriate “NormalizationTest.txt” file from unicode.org in order to run the test. Note that this test may take long – depending on your configuration a few minutes – and only shows progress indication in the “Console” application.

The XHTML Entity Names are taken from the XHTML 1.0 Specification at <http://www.w3.org/TR/xhtml1/>. They are stored in three separate files inside the application bundle which can be edited or replaced if needed.
In case nothing is displayed in the “XHTML” field, there is no defined character entity available for this character and you must construct a numeric character reference yourself using the usual format from the HTML 4.01 Specification <http://www.w3.org/TR/html4/charset.html#h-5.3.1>:
&#*D*; – where *D* is the decimal character number
&#x*H*; or &#X*H*; – where *H* is the hexadecimal character number

**Extensibility**
If you are a developer familiar with Cocoa you can develop your own String Utilities Plug-Ins. Please send an e-mail to <mailto:earthlingsoft@earthlingsoft.net> to get the necessary header files and sample code.

**What’s all this in the “Description” text box?**
The “Description” text box contains the following information (if available for the selected codepoint) in the presented order. For further information, consult the mentioned Unicode Character Database file.

*Character Name*
The official Unicode character name, either taken from “UnicodeData.txt” or the algorithmically derived Hangul name.

*Unicode 1.0 Name*
The character name from Unicode Version 1.0 from “UnicodeData.txt”.

*ISO 10646 Comment*
The ISO 10646 comment from “UnicodeData.txt”.

*Jamo Short Name*
The Jamo Short Name as defined in “Jamo.txt”.

*codepoint not assigned*
When the selected codepoint does not have an entry in “UnicodeData.txt” and does not belong to one of the special ranges defined in “UnicodeData.html” it is considered as not encoded in the Unicode Standard. Note that it is possible that you see a character although UnicodeChecker says the codepoint is not assigned depending on the Fonts installed on your system.

*Extended Property*
Extended Unicode properties from “PropList.txt”.

*Script*
Script name from “Scripts.txt”.

*Block*
Character Block from “Blocks.txt”.

*Arabic/Syriac Shaping*
The arabic or syriac shaping either from “ArabicShaping.txt” or derived from the condition mentioned in that file.

*Special Casing*
Additional full casing information from “SpecialCasing.txt”.

*Designated in Unicode*
The version of Unicode the codepoint was designated, taken from “DerivedAge.txt”. Consult this file for more information.

*Other information*
Additionally UnicodeChecker displays the names found in “Index.txt” for the selected codepoint. This is often information covered before in one of the other fields, but may contain new information, too.

**What is missing?**
UnicodeChecker does not consult the following Unicode Data files for presenting codepoint information:
*BidiMirroring.txt, CaseFolding.txt, EastAsianWidth.txt, LineBreak.txt, NamesList.txt, NormalizationCorrections.txt, NormalizationTest.txt, PropertyAliases.txt, PropertyValueAliases.txt, Unihan.txt* and all *Derived*.txt* files not mentioned above.
If you think the information from one of these files should be included in UnicodeChecker please contact us and we will consider it for a future release.

**Version History**
*1.5.7*
• Uses Unicode 4.0 data files
*1.5.6*
• Updated to comply with RFC 3490 for IDNA
• AppleScript support for converting to and from IDNA
• AppleScript support, Services menu entry and new utility for adding and replacing percent escape sequences for URLs
*1.5.5*

• AppleScript support for converting to and from XHTML entities and displaying character information

*1.5.4*

• Runs in Mac OS X 10.1 again

*1.5.3*

• Fixed a bug where Hangul decompositions would not be displayed correctly

*1.5.2*

• The HTML entity conversion services now report the number of replaced characters/entities
• New Utility for converting to and from HTML entities (formerly only available from the Services menu)
• Unicode Normalization Forms are now also available from the Services menu
• The “Display Character Information” service now recognises U+nnnn, {, Ī and &abc; notations.
• Improved speed for string conversion
• The Unicode 1 names for codepoints are included for the “Find Codepoint” menu command

*1.5.1*

• In the Planes PopUp the "Supplementary Special-purpose Plane" has been corrected to be plane 14 instead of plane 16

*1.5*

• Services
• Lots of additional codepoint information
• Unicode Utilities

*1.0*

• Initial Release

**Disclaimer**

Although care was taken to ensure conformance of this release of UnicodeChecker with The Unicode Standard current at the time of release, earthlingsoft cannot guarantee that UnicodeChecker does not contain any bugs that may prevent full conformance with The Unicode Standard.

**Contact**

http://www.earthlingsoft.net
mailto:earthlingsoft@earthlingsoft.net