

Data Analysis

A QR code encodes a string of text. The QR code standard has four modes for encoding text: numeric, alphanumeric, byte, and Kanji. Each mode encodes the text as a string of bits (1s and 0s), but each mode uses a different method for converting the text into bits. Each method is optimized to generate the shortest possible string of bits for that data type. This page explains how to identify which mode to use.

The QR Code Modes

The four encoding modes include the following characters:

Numeric mode is for decimal digits 0 through 9.

Alphanumeric mode is for the decimal digits 0 through 9, as well as uppercase letters (not lowercase!), and the symbols \$, %, *, +, -, ., /, and : as well as a space. All of the supported characters for alphanumeric mode are listed in the left column of this alphanumeric table (alphanumeric-table).

Byte mode, by default, is for characters from the ISO-8859-1 character set. However, some QR code scanners can automatically detect if UTF-8 is used in byte mode instead.

Kanji mode is for **double-byte** characters from the Shift JIS character set. While UTF-8 can encode Kanji characters, it must use three or four bytes to do so. Shift JIS, on the other hand, uses just two bytes to encode each Kanji character, so Kanji mode compresses Kanji characters more efficiently. If the entire input string consists of characters in the double-byte range of Shift JIS, use Kanji mode. It is also possible to use multiple modes within the same QR code, as described later on this page.

Extended Channel Interpretation (ECI) mode specifies the character set (e.g. UTF-8) directly. However, some QR code readers do not support ECI mode and will not understand QR codes that use it.

Structured Append mode encodes data across multiple QR codes, up to a maximum of 16 QR codes. I will not be discussing this mode in this tutorial but may add more information at a later time.

FNC1 mode allows the QR code to function as a GS1 barcode. I will not be discussing this mode in this tutorial but may add more information at a later time.

A Note About Kanji Mode

Some QR code readers can recognize when UTF-8 is used in byte mode. Since all of the Shift JIS characters have representations in UTF-8, it is possible to use byte mode for Kanji with UTF-8 encoding.

However, Kanji in UTF-8 are encoded with three bytes (or four, in rare cases), whereas Shift JIS characters are encoded with two or one bytes. In other words, it will not be possible to fit as many characters into the QR code if using UTF-8 in byte mode for Kanji. Using Kanji mode for Shift JIS Kanji gives the highest capacity.

Therefore, it is up to you whether to use Kanji mode for Kanji or not, depending on the needs of your users.

A Note About UTF-8

Some QR code readers automatically detect if UTF-8 is used in byte mode, but those that do not may show incorrect characters if using UTF-8 in byte mode. To fix this, it may be possible to use ECI mode, which, as mentioned above, makes it possible to specify a different character set from the default ISO-8859-1 character set in byte mode. Unfortunately, not all QR code readers support ECI mode.

Another option is to put the UTF-8 byte order mark (BOM) before the input text. Some QR code readers will read the byte order mark and understand that the text is encoded in UTF-8. Not all QR code readers can interpret this correctly. The byte order mark for UTF-8 is a set of three numbers, shown here in hexadecimal: 0xEF 0xBB 0xBF

How to Choose the Most Efficient Mode

To select the most efficient mode for the QR code, examine the characters in the input string and check for the following conditions.

1. If the input string only consists of decimal digits (0 through 9), use **numeric mode**.
2. If numeric mode is not applicable, and if all of the characters in the input string can be found in the left column of the alphanumeric table (alphanumeric-table), use **alphanumeric mode**.
Lowercase letters CANNOT be encoded in alphanumeric mode; only uppercase.
3. If there is a character that is not in the left column of the alphanumeric table (alphanumeric-table) but can be encoded in ISO 8859-1, use **byte mode**. As mentioned above, QR code readers may be able to recognize UTF-8 in byte mode.
4. If all of the characters are in the Shift JIS character set, use **Kanji mode**. Shift JIS characters can be encoded in UTF-8 instead, so it is possible to use byte mode for Kanji, but it is generally more efficient to use Shift JIS and use Kanji mode for Kanji characters.

Mixing Modes and Optimization

It is possible to use multiple modes in a single QR code by including the mode indicator before each section of bytes that uses that mode. The QR code specification explains how to switch modes in the most optimal way. I will not be discussing this in the tutorial but I may add more information at a later time. This tutorial will assume that you will not mix modes in your QR codes.

Summary

By examining the characters in the input text, it is possible to choose the most optimal mode for encoding that text. Be sure to consider the limitations of QR code readers when choosing a mode, and be aware that not all QR code readers adhere to the standard. Furthermore, consider the needs of your users when deciding whether to use Kanji mode or whether to use UTF-8 in byte mode instead.

Next: Data Encoding

After choosing the encoding mode, the next step is to encode the data (data-encoding).



(<http://creativecommons.org/licenses/by-nc/4.0/>)

Thonky.com's QR Code Tutorial by Thonky (<http://www.thonky.com/>) is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>).

QR Code is registered trademark of DENSO WAVE (<http://www.denso-wave.com/en/adcd/>)
INCORPORATED.

Page last updated 2015-05-12T22:29:21-05:00

© 2018 Thonky.com (/) | Privacy Policy (/globalpriv.php) | About Thonky.com (/about/) | Support
Thonky.com (/support-thonky/)