**Only for Double-Byte Shift JIS Characters**

Kanji mode can only encode double-byte Shift JIS characters whose bytes are in the ranges 0x8140 to 0x9FFC and 0xE040 to 0xEBBF (hexadecimal). The characters in this set can be found on Rikai's Shift JIS Kanji Code Table.

**Convert to Bytes**

To encode double-byte Shift JIS characters in Kanji mode for QR codes, the characters must first be converted to bytes. For example, the characters 茗荷 can be converted as follows.

茗 → 0xE4AA

荷 → 0x89D7

**Encode the Bytes with Kanji Mode**

Kanji mode has two methods for encoding double-byte Kanji characters. One method is for characters whose bytes are in the range 0x8140 to 0x9FFC. The other method is for characters whose bytes are in the range 0xE040 to 0xEBBF. These two methods are outlined below.

For characters whose bytes are in the range 0x8140 to 0x9FFC

From the example above, the character 荷 is 0x89D7 in Shift JIS, so it is in the range 0x8140 to 0x9FFC. The first step is to subtract 0x8140 from the hex value. In this example:

0x89D7 - 0x8140 = 0x0897

The result was 0x0897. Split that number into its most significant byte and its least significant byte:

Most significant byte of 0x0897 is 0x08

Least significant byte of 0x0897 is 0x97

Next, multiply the most significant byte by 0xC0, then add the least significant byte to the result:

(0x08 \* 0xC0) + 0x97 = (0x600) + 0x97 = 0x697

Finally, convert that result into 13-bit binary:

0x697 = 0 0110 1001 0111

For characters whose bytes are in the range 0xE040 to 0xEBBF

From the example above, the character 茗 is 0xE4AA in Shift JIS, so it is in the range 0xE040 to 0xEBBF. The first step is to subtract 0xC140 from the hex value. In this example:

0xE4AA - 0xC140 = 0x236A

The result was 0x236A. Split that number into its most significant byte and its least significant byte:

Most significant byte of 0x236A is 0x23

Least significant byte of 0x236A is 0x6A

Next, multiply the most significant byte by 0xC0, then add the least significant byte to the result:

(0x23 \* 0xC0) + 0x6A = (0x1A40) + 0x6A = 0x1AAA

Finally, convert that result into 13-bit binary:

0x1AAA = 1 1010 1010 1010

**Put the 13-bit Binary Numbers Together**

The string that was to be encoded was 茗荷. These two characters were converted into 13 bit binary in the above steps. Put the two binary strings together to complete the encoding.

茗荷 encoded for QR code is 11010101010100011010010111