**INTRODUCTION**

The foregoing parts have described two encoding schemes for use in MARC 21 records: MARC-8 and Unicode. There are many situations in which it may be necessary or desirable to convert records from one scheme to the other. This part identifies a number of factors a successful conversion must take into account when making this conversion, and specifies techniques for converting Unicode records that contain characters not present in the MARC-8 repertoire.

[Part 5: MARC-8 Code Tables](https://www.loc.gov/marc/specifications/specchartables.html) contains tables showing the MARC-8 repertoire along with the code values of each character in the MARC-8 and Unicode schemes. It also contains links to an XML version of the table for all MARC-8 characters, and a comma-delimited file of MARC-8/Unicode correspondences for the EACC (CJK) character set only.

References in this part to individual MARC-8 characters are given in the form **0xXX**. References to individual Unicode characters are given in the form **U+XXXX**. The character value is expressed as a hexadecimal number. (For Unicode characters, this hexadecimal number is the scalar value, not the UTF-8 encoding value.)

**GENERAL CONVERSION ISSUES**

The points in this section must be considered when converting from one of the allowed encodings to the other: from MARC-8 to Unicode, or from Unicode to MARC-8. The following assumptions underlie these points:

* The starting point is a record correctly encoded according to either the MARC-8 or the Unicode scheme; the end point is a record correctly encoded according to the other scheme
* Data is to be converted so that a reversal of the conversion produces a record identical to, or as close as possible to, the original record (this is sometimes called *round-tripping*)

Different starting conditions, or a desire for a different outcome, will require adjustments to the general conversion model outlined here.

* Precomposed characters consisting of a base character and one or more combining characters in either a MARC-8 or a UTF-8 record must be replaced by their component separate characters before they can be successfully converted. See *Normalization, and the conversion from Unicode to MARC-8*.
* If a UTF-8 record is being converted to MARC-8 and round-tripping of the records is not intended, replacements with approximately equivalent characters may be made for some UTF-8 characters that are not present in the MARC-8 repertoire. (The manner in which this is done should be adjusted to meet the expected use of the MARC-8 record.) See *Characters present in Unicode but not in MARC-8*.
* A non-Unicode record may mix a part of the MARC-8 character set with elements from a different character set. See *Traps of various kinds*

**Record elements not converted**

The following MARC 21 record elements are represented in both MARC-8 and Unicode records using only characters with values below 0x80 in *Code for information interchange* (ASCII; ANSI X3.4) or its international counterpart (ISO 646). These record elements are not changed when a record is converted from MARC-8 to Unicode, or from Unicode to MARC-8.

* Record leader
* Record directory
* Field tag and indicators
* Subfield delimiter (0x1F) and subfield code
* End-of field (0x1E) and end-of-record (0x1D) markers
* Contents of control fields (tags 001-009)

**MARC 21 encoding marker**

[Leader](https://www.loc.gov/marc/bibliographic/ecbdldrd.html#mrcblea) position 9 (Character encoding scheme) identifies the scheme used to encode the data in a record.

* When converting a record from MARC-8 to Unicode, use code “a” (0x61)
* When converting a record from Unicode to MARC-8, use code "blank" (0x20)

**Escape sequences and MARC field 066**

* When converting a record from MARC-8 to Unicode, remove any [066](https://www.loc.gov/marc/bibliographic/ecbdclas.html#mrcb066) field, and convert characters encoded as escape sequences into their Unicode equivalents.
* When converting a record from Unicode to MARC-8, use escape sequences and construct an 066 field where appropriate. Field [066](https://www.loc.gov/marc/bibliographic/ecbdclas.html#mrcb066) is required in a MARC-8-encoded record whenever the record contains a type 2 escape sequence, as described in [Part 2](https://www.loc.gov/marc/specifications/speccharmarc8.html). If there are no such escape sequences, do not include an [066](https://www.loc.gov/marc/bibliographic/ecbdclas.html#mrcb066) field in the record.

**MARC subfield $6 (Linkage)**

[Subfield $6](https://www.loc.gov/marc/bibliographic/ecbdcntf.html#mrcs6) (Linkage) is used in MARC 21 records to link alternate graphic representations of the same data. In 880 fields, [Subfield $6](https://www.loc.gov/marc/bibliographic/ecbdcntf.html#mrcs6) also:

* Identifies the presence of a specific script in a field (this code identifies MARC-8 character sets, rather than scripts *per se)*
* Flags fields in which the display/print directionality of data is right-to-left (e.g., for Arabic script).

In [Subfield $6](https://www.loc.gov/marc/bibliographic/ecbdcntf.html#mrcs6) of the 880 field, a solidus (slash) character (0x2F; U+002F) precedes the script identification code; a second solidus precedes the field orientation code. [Subfield $6](https://www.loc.gov/marc/bibliographic/ecbdcntf.html#mrcs6) in the 880 field of a MARC-8 record may contain both a script identification code and a field orientation code; in a Unicode record; [Subfield $6](https://www.loc.gov/marc/bibliographic/ecbdcntf.html#mrcs6) will not contain a script identification code but may contain a field orientation code.

* When converting a record from MARC-8 to Unicode, remove the script identification code from [subfield $6](https://www.loc.gov/marc/bibliographic/ecbdcntf.html#mrcs6). Use the field orientation code if appropriate; when present, this code is separated from the [Subfield $6](https://www.loc.gov/marc/bibliographic/ecbdcntf.html#mrcs6) linking tag and occurrence number by two solidus (slash) characters (0x2F; U+0024).
* When converting a record from Unicode to MARC-8, insert the script identification code into [subfield $6](https://www.loc.gov/marc/bibliographic/ecbdcntf.html#mrcs6), typically using the same code that would be recorded in subfield $c of the [066](https://www.loc.gov/marc/bibliographic/ecbdclas.html#mrcb066) field.

The following table shows the information available for subfield $6 of an 880 field that corresponds to the 245 field in a record (the sequence numbers used here are arbitrary), and the appearance of subfield $6 using that information in a MARC-8 and a Unicode record.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field and sequence** | **Script code** | **Orientation code** | **MARC-8 subfield $6** | **Unicode subfield $6** |
| 245-02 | (B | [none] | $6 245-02/(B | $6 245-02/(B |
| 245-03 | (2 | r | $6 245-03/(2/r | $6 245-03//r |

**Combining characters (diacritics)**

In a MARC-8 record, combining characters precede their base characters; in a Unicode record, combining characters follow their base characters. If multiple combining characters are associated with a given base character they have different positions relative to each other in the two encodings. (See [Part 3](https://www.loc.gov/marc/specifications/speccharucs.html).) The best practice during conversion from one encoding to the other is to reorder multiple combining marks according to the rule for the output encoding scheme.

* When converting a record from MARC-8 to Unicode, move combining characters so that they immediately follow their base character instead of preceding it, and adjust the order of multiple combining characters on the same base character as appropriate. Special care is required when combining characters immediately precede an escape sequence.
* When converting a record from Unicode to MARC-8, move combining characters so that they immediately precede their base character instead of following it, and adjust the order of multiple combining characters on the same base character as appropriate. If the base character is the first non-combining character in an escape sequence (other than an escape sequence for the standard character set), the combining characters must move out of that escape sequence into the preceding escape sequence, or their own escape sequence.

**Directionality of numerals embedded in right-to-left text**

* When converting a record from MARC-8 to Unicode, the conversion should determine (or be told) whether multi-digit numbers used in bidirectional scripts have been entered in logical or visual order. If visual order has been used, the conversion should re-order the digits from visual order to logical order. If logical order has been used, the conversion should not re-order the digits.
* When converting a record from Unicode to MARC-8, the conversion should retain the logical order of characters already present in the data.

**Folding of certain MARC-8 and Unicode characters**

*Numbers, punctuation marks, and symbols*

The numbers, punctuation marks, and symbols represented by characters 0x21-0x3F, 0x5B, 0x5D are also, in full or in part, allocated distinct code points in the MARC-8 sets for Hebrew, Cyrillic, and Arabic. These characters are mapped (folded) into a single set of Unicode code points. (See the code tables in [Part 5](https://www.loc.gov/marc/specifications/specchartables.html).)

* When converting a record from MARC-8 to Unicode, map these characters to the appropriate Unicode code points.
* When converting a record from Unicode to MARC-8, map these characters to the standard ASCII character set instead of the characters in the Hebrew, Cyrillic or Arabic sets.

A MARC-8 record containing these characters that is converted into UTF-8 and back to MARC-8 may contain more escape sequences than the original record. This is an acceptable result that should not interfere with processing or display of the record.

*Alpha, beta, gamma*

The three lowercase characters of the custom MARC-8 Greek Symbols set (alpha, beta, gamma) are mapped to the regular Greek letters in Unicode and consequently are not reversible when converting to MARC-8. (Use of the MARC-8 Greek Symbols set is strongly discouraged for this reason.)

* When converting a record from MARC-8 to Unicode, map these characters to the normal Greek character set.
* When converting a record from Unicode to MARC-8: if any of these three characters occurs in isolation, give a textual equivalent (within brackets: [alpha], [beta], [gamma]); if any of these three characters occurs as part of a larger expression in Greek, use Numeric Character References. (See *Issues specific to the conversion from Unicode to MARC-8.*)

*Space*

The space character (0x20; U+0020) is defined only in the Unicode and ASCII sets but is recognized by all the standard graphic character sets in MARC-8 even though not included in those sets.

* When converting a record from MARC-8 to Unicode, transfer the space character unchanged.
* When converting a record from Unicode to MARC-8, transfer the space character contained within one of the standard graphic character sets unchanged; do not precede the space character by an escape sequence. (Optionally, the escape sequence to ASCII may be included before the space character.) However, if the output working set is a custom set accessed by escape technique 1, the technique 1 escape sequence to ASCII is required before the space character.

**ISSUES STEMMING FROM DIFFERENCES BETWEEN UNICODE AND MARC-8**

**Lossless and lossy conversion**

The [MARC-8 repertoire](https://www.loc.gov/marc/specifications/speccharmarc8.html) contains over 16,000 characters; at the time of writing the [Unicode repertoire](http://www.unicode.org) contains nearly 150,000 characters. Direct mappings using the tables in [Part 5](https://www.loc.gov/marc/specifications/specchartables.html) are sufficient for the Unicode to MARC-8 conversion only for a record that contains no characters that are outside the MARC-8 repertoire. Additional techniques are needed for the more general case in which non-MARC-8 characters are present in a Unicode record that is to be converted.

Two generally applicable methods for conversion from Unicode to MARC-8 encoding have been approved to aid conversion of MARC 21 records containing characters outside the MARC-8 repertoire. The two methods must not be used in the same record.

*Preferred method: lossless conversion to MARC-8 encoding*

In the *lossless* conversion method, a Unicode character that is not in the MARC-8 repertoire is replaced in a MARC-8 record by a Numeric Character Reference (NCR) identifying the specific unconvertable Unicode code point. Although this change may result in a cryptic display, this method preserves precisely in a MARC-8 record the information content of the Unicode record. A Numeric Character Reference consists only of ASCII characters, and thus can be carried in a MARC-8 record.

The structure of the NCR is **&#xXXXX;** where:

* **&** and **;** (ampersand and semicolon) mark the beginning and end of the NCR data
* **#x** indicates that the reference is expressed as a hexadecimal value
* **XXXX** is the hexadecimal representation of the scalar value of the code point for the Unicode character (not its UTF-8 equivalent), expressed in hexadecimal digits (using the characters 0-9 and A-F). The NCR can contain more than four digits if needed. (Some characters, primarily infrequently encountered CJK ideographs, may require more than four hexadecimal digits.)

Apply Numeric Character References as follows:

* When converting a record from Unicode to MARC-8, replace any character not part of the MARC-8 repertoire with a Numeric Character Reference
* When converting a record from MARC-8 to Unicode, replace all Numeric Character References with the corresponding Unicode characters

*Alternate method: lossy conversion to MARC-8 encoding*

The *lossy* conversion method is intended for use in situations in which the loss of data outside the MARC-8 repertoire is not a concern. This method is called lossy because the substitution of a generic placeholder for every unconvertible Unicode code point loses data that cannot be recovered in a reconversion to Unicode.

* When converting a record from Unicode to MARC-8, replace each character that is not in the MARC-8 repertoire with an ASCII vertical bar (0x7C).
* When converting a record from MARC-8 to Unicode, transfer the vertical bar (0x7C) as found.

**Normalization, and the conversion from Unicode to MARC-8**

Making adjustments to characters for one reason or another is referred to as *normalization.* One of the annexes to the Unicode standard describes forms of normalization used within the Unicode environment. Two of these are particularly relevant to records that are to be converted from Unicode to MARC-8: Normalization form D (NFD), also called *canonical decomposition*, and Normalization Form KD (NFKD), also called *compatibility decomposition*.

* Canonical decomposition changes composite characters into their constituent elements; the input and output forms are interconvertible. Some examples of characters with *canonical* equivalents are the lowercase *e* withthe acute accent (U+00E9), which can be decomposed into a lowercase e plus the combining acute accent (U+0065 plus U+0301); and the uppercase *U* with diaresis (U+00DC), which can be decomposed into an uppercase *U* and the combining diaresis (U+0055 plus U+0308). Once the decomposition is achieved, the resulting characters can be converted into their MARC-8 equivalents.
* Compatibility decomposition replaces one character with a near equivalent; the input and output forms are not interconvertible. Some examples of characters with *compatibility* equivalents are the ellipsis character (U+2026), which can be replaced in the MARC-8 record with a sequence of three periods (0x2E); the circled digit four (U+2463), which can be replaced with the numeral 4 (0x43); the Roman numeral IV (U+2163), which can be replaced with I (0x49) plus V (0x56); and any of the spaces of different width (U+2000-U+2008), which can be replaced with the plain space (0x20). If compatibility decomposition is employed, its precise nature will depend on the content of the Unicode records, and the uses to which the MARC-8 records are to be put.

*Limits to canonical decomposition*

The MARC-8 repertoire includes several precomposed characters that could be decomposed in Unicode, but should not be decomposed during conversion to MARC-8. These characters are specified in Table 4.1 below.

**Table 4.1**

*Characters not subject to canonical decomposition when converting a record from Unicode to MARC-8 encoding.*

All code points are shown in hexadecimal notation. MARC-8 code points are shown in the G0 range for all sets except Extended Latin.

|  |  |  |  |
| --- | --- | --- | --- |
| **Character name** | **Unicode code points (uppercase, lowercase)** | **MARC-8 G0 code points (uppercase, lowercase)** | **MARC-8 character set** |
| Cyrillic short I | U+0419, U+0439 | 0x4A, 0x6A | Basic Cyrillic |
| Cyrillic Io | U+0401, U+0451 | 0x44, 0x64 | Extended Cyrillic |
| Cyrillic Gje | U+0403, U+0453 | 0x42, 0x62 | Extended Cyrillic |
| Cyrillic Yi | U+0407, U+0457 | 0x47, 0x67 | Extended Cyrillic |
| Cyrillic Kje | U+040C, U+045C | 0x4C, 0x6C | Extended Cyrillic |
| Cyrillic Kje | U+040C, U+045C | 0x4C, 0x6C | Extended Cyrillic |
| Cyrillic short U | U+040D, U+045E | 0x4D, 0x6D | Extended Cyrillic |
| Arabic alef, madda above | U+0622 | 0x42 | Basic Arabic |
| Arabic alef, Hamza above | U+0623 | 0x43 | Basic Arabic |
| Arabic waw, Hamza above | U+0624 | 0x44 | Basic Arabic |
| Arabic alef, Hamza below | U+0625 | 0x45 | Basic Arabic |
| Arabic yeh, Hamza above | U+0626 | 0x46 | Basic Arabic |
| Latin O with horn | U+01A0, U+01A1 | 0xAC, 0xBC | Extended Latin (ANSEL) |
| Latin U with horn | U+01AF, U+01B0 | 0xAD, 0xBC | Extended Latin (ANSEL) |

|  |  |  |  |
| --- | --- | --- | --- |
| Unicode normalization form KD, the optimal for conversion from Unicode to MARC-8 repertoires, specifies that decompositions of both types, *canonical* and *compatibility*, should be done until no further decomposition is possible. The full KD normalization, however, may not be desired because of the *canonical* decompositions in the above table and other issues with the *compatibility* decompositions, such as loss of formatting with superscripts and subscripts. |  |  |  |

**Additional considerations for converting characters with combining marks from Unicode to MARC-8**

A further complication arises when a base character accompanied by a combining mark in a Unicode record cannot be normalized into components belonging to the MARC-8 repertoire. Proper treatment depends on whether it is the base character or the combining mark that is absent from the repertoire. If the base character can be converted, then the combining mark should be replaced by an NCR or placeholder properly repositioned before the base character in the output record. If the base character cannot be converted, either the lossless or the lossy technique can be applied directly, preferably to the character in its precomposed form, so that it will generate a single NCR or placeholder rather than two. This treatment is preferred whether or not the combining character is also missing from the MARC-8 repertoire.

**Traps of various kinds**

Records with code “blank” (0x20) in Leader position 9 do not necessarily follow MARC coding conventions, and do not necessarily use the full MARC-8 character repertoire. Care must be taken with MARC-8 records to Unicode, so that the finished records are acceptable. The following points describe common situations, but many other variations from the expected standard may be discovered.

* The characters in the range 0x80 to 0xFF may be used in non-standard ways. For example, 0xE9 may have been used to represent *e* with an acute accent (é), instead of the combining caron. Some sources of records may use most of the MARC-8 repertoire, but introduce variant locations for characters such as the script-l, degree sign, inverted exclamation mark and inverted question mark. It is often necessary carefully to inspect this range of characters in a set of records, to determine the proper conversion method.
* If combining characters associated with the first filing character have been included in the nonfiling zone in a MARC-8 record, the nonfiling characters indicator must be adjusted in the UTF-8 record. (This coding in the MARC-8 record is incorrect, but does no harm; if left unadjusted it will cause searching problems for the Unicode record.)