Colorado Alliance MARC record match key generation.

March-07-2026

The Gold Rush system allows libraries to do in-depth analytics for their cataloged library holdings for either themselves or in comparison to other libraries that are already in the system. It supports a wide variety of use cases including shared print, weeding, collection building, space planning, storage facilities, etc. Due to the wide variety of cataloging sources and the long bibliographic history of most collections, no single number (e.g. OCLC, ISBN) can be used for matching purposes. This match key uses a variety of elements from the MARC record to bring together common bibliographic records. No key is perfect so this particular approach is periodically being modified to improve matching within the system. Many other systems use their own matching algorithms to accomplish the same purpose and some of these have been consulted in building this algorithm.

These MARC fields are used:

Title

Combine 245 $a $b $p (first occurrence)

General Media Description

245 $h — **DISABLED as of 11-15-22** — always returns 5 underscores "_____"

Publication Year

First check control field 008: if character 6 equals 'r' retrieve reissue date from characters 7-10; if there is a government document (086$a exists) always use date from characters 7-10; otherwise use date from characters 11-14 (if valid and >= 1200 and != 9999), falling back to 7-10. If no valid date found, check 264$c then 260$c. For 264$c and 260$c, get the rightmost 4 digits giving precedence to years preceded by 'c'. If still no valid date found return "0000". NOTE: character count starts at 0.

Pagination

300 $a

Edition Statement

250 $a — if the field is empty and the format is "Book" mark this as a 1st edition "1__".

Publisher Name

First check 264 $b then 260 $b. Ampersands are removed, then diacritics and non-alphanumeric characters (other than spaces) are automatically removed via stripPuncuation(). Convert to lowercase, remove underscores, normalize accents, and pad with underscores to 5 characters.

Type of

Character #6 of Leader

Title Part

245 $p all multi-valued $p fields

Title Number

245 $n

Author

100 $a, 110 $a, 111 $a, 130 $a

Title-Inclusive Dates (used by Daily Camera focus magazine)

245 $f

Government Document Number

086 $a — **REMOVED from match key as of 02-03-26** for HATHI compatibility.

Electronic indicator

Check 245 $h, 590 $a, 533 $a, 300 $a, 007, 337 $a, 086+856 combination, and MARC filename; appending 'e' if resource is electronic otherwise appending 'p'. See getFormatCharacter(Record) method details for filename usage.

Match Key Section description.

NOTE: Each section is padded with underscores to create the desired character count. If a field is multi-valued, take the first value unless specified. Functions (in **bold**) are described in appendix.

Chars	Element	MARC Source	Notes
95	Title	245 $a $b $p	Combine $a $b and $p applying **strip_punctuation_SPACE()** to each, then remove all spaces, **trim()**, **normalizeString()**, **pad_with_underscores(95)**. NOTES: If the first MARC 245$6 field indicates a linked 880 field, the 880 field is used instead. For non-Roman languages without an 880 field, falls back to LLC number, ISBN, or ISSN.
5	General Media Designation (GMD)	245 $h	**DISABLED (11-15-22)** — Always returns "_____". Previously: the first five contiguous alphanumeric characters, right-padded with underscores. **pad_with_underscores(5)**
4	Publication Year	008, 264 $c, 260 $c	First check 008 field: if char 6 is 'r' use reissue date (7-10); if gov doc (086$a exists) always use date 1 (7-10); otherwise prefer date 2 (11-14) if valid (>=1200, !=9999), falling back to date 1 (7-10). If no valid number, check 264$c then 260$c. Get rightmost 4 digits, preferring years preceded by 'c'. If not found return "0000". **pad_with_underscores(4)**
4	Pagination	300 $a	First four contiguous numeric characters. If fewer than four or no source field, assigned underscores. **pad_with_underscores(4)**
3	Edition Statement	250 $a	First three, two or one contiguous numeric characters. If none, use first three, two or one alphabetic characters. Diacritics removed. Convert 'fir' to 1, 'sec' to 2, 'thi' to 3, 'for' to 4, 'fif' to 5, 'six' to 6, 'sev' to 7, 'eig' to 8, 'nin' to 9, '10t' to 10. If empty and format is "Book", defaults to "1__". **pad_with_underscores(3)**
5	Publisher Name	264 $b, 260 $b	If 264 empty check 260$b. Remove ampersands, **stripPuncuation()**, remove underscores, normalize accents, convert to lowercase. **pad_with_underscores(5)**
1	Type of	Leader	Character #6. If leader contains 10 or more characters return the 6th character.
30	Title Part	245 $p	If multiple $p subfields, take first 10 characters of each. **trim()**, **stripPuncuation()**, **pad_with_underscores(30)**. NOTE: The first $p has already been added to the Title section.
10	Title Number	245 $n	**stripPuncuation()**, **pad_with_underscores(10)**
5	Author	100 $a, 110 $a, 111 $a, 130 $a	Combine contents of each field applying **stripPuncuationAndRemoveAccents()**. Remove underscores and spaces, convert to lowercase. **pad_with_underscores(5)**
15	Title-Inclusive Dates	245 $f	Remove all spaces then **stripPuncuation()**, **padWithUnderscores(15)**
—	Gov Doc Number	086 $a	**REMOVED (02-03-26)** for HATHI compatibility. Previously: **stripPuncuationAndRemoveAccents()**, **trimMAXfieldLength()**.
1	Format Character	245 $h, 590 $a, 533 $a, 300 $a, 007, 337 $a,	**getFormatCharacter()** — append 'e' for electronic, otherwise 'p'.

086+856

Post-processing: The entire match key is converted to lowercase. Any remaining colons are replaced with 'x'. Any remaining spaces are replaced with underscores.

Appendix – Method definitions

Example: 'EP 1.1/5:' becomes 'EP_1_1_5'

```
String stripPuncuation( String input )
{
        NOTE: .replace() replaces all occurrences.

        input = input.replace( "%22", "_" );
        input = input.replace( "%", "_" );
                // Remove leading 'a' or 'A' and surrounding spaces.
        input = input.replaceFirst( "^[ ]+[aA][ ]+"  , "");
                // Remove leading 'an' or 'An' and surrounding spaces.
        input = input.replaceFirst( "^[ ]+[aA]n[ ]+" , "");
                // Remove leading 'the' or 'The' and surrounding spaces.
        input = input.replaceFirst( "^[ ]+[tT]he[ ]+", "");

        input = input.replace(  "'", "" ); //39
        input = input.replace(  "{", "" ); //123
        input = input.replace(  "}", "" ); //125
        input = input.replace(  "&", "and" ); //38
        input = input.replace(  '\u0020', UNDERSCORE ); //32 ' '
        input = input.replace(  '\u0021', UNDERSCORE ); //33 '!'
        input = input.replace(  '\u0022', UNDERSCORE ); //34 '"'
        input = input.replace(  '\u0023', UNDERSCORE ); //35 '#'
        input = input.replace(  '\u0024', UNDERSCORE ); //36 '$'
        input = input.replace(  '\u0028', UNDERSCORE ); //40 '('
        input = input.replace(  '\u0029', UNDERSCORE ); //41 ')'
        input = input.replace(  '\u002A', UNDERSCORE ); //42 '*'
        input = input.replace(  '\u002B', UNDERSCORE ); //43 '+'
        input = input.replace(  '\u002C', UNDERSCORE ); //44 ','
        input = input.replace(  '\u002D', UNDERSCORE ); //45 '-'
        input = input.replace(  '\u002E', UNDERSCORE ); //46 '.'
        input = input.replace(  '\u002F', UNDERSCORE ); //47 '/'
        input = input.replace(  '\u003A', UNDERSCORE ); //58 ':'
        input = input.replace(  '\u003B', UNDERSCORE ); //59 ';'
        input = input.replace(  '\u003C', UNDERSCORE ); //60 '<'
        input = input.replace(  '\u003D', UNDERSCORE ); //61 '='
        input = input.replace(  '\u003E', UNDERSCORE ); //62 '>'
        input = input.replace(  '\u003F', UNDERSCORE ); //63 '?'
        input = input.replace(  '\u0040', UNDERSCORE ); //64 '@'
        input = input.replace(  '\u005B', UNDERSCORE ); //91 '['
        input = input.replace(  '\\',     UNDERSCORE ); //92 '\'
```

```
                input = input.replace(  '\u005D', UNDERSCORE ); //93 ']'
                input = input.replace(  '\u005E', UNDERSCORE ); //94 '^'
                input = input.replace(  '\u005F', UNDERSCORE ); //95 '_'
                input = input.replace(  '\u0060', UNDERSCORE ); //96 '`'
                input = input.replace(  '\u007C', UNDERSCORE ); //124 '|'
                input = input.replace(  '\u007E', UNDERSCORE ); //126 '~'
                input = input.replace(  '\u00A9', UNDERSCORE ); //169 '©'
                return result }
*****
* Replace punctuation in a String with spaces
****
String stripPuncuation_SPACE( String input ){
   Same as stripPuncuation() except characters are replaced with spaces
   instead of underscores. This is used for the title to handle punctuation
   at the end of the title:
   EXAMPLE:
        stripPuncuation_SPACE( "The law of primitive man /").trim()
        stripPuncuation_SPACE( "The law of primitive man;").trim()
        stripPuncuation_SPACE( "The law of primitive man : ").trim()
        All return "The law of primitive man"
}
*****
* Remove leading and trailing whitespace from a String using the
* standard Java String function.
*****
String String.trim(){
        java.lang.String.trim()
}
*****
* Replace all spaces in a String with underscores. Trim length of String
* to desired length. If String is too short pad with underscores.
*****
String padWithUnderscores( String input, Integer count ){
        replace repeating spaces with single space.
        replace all spaces in 'input' with underscores
        Trim 'input' to length to 'count' characters.
        If needed, pad the end of the string with underscores to create
        the desired character String.
        return result;
}
*****
* Trim a String to the maximum length allowed for a Solr field. This is
* used as a precaution, it is usually an error for a MARC field to be
* of this length.
```

```
*****
String trimMAXfieldLength( String input ){
        trim input to the max length of a solr field (32,000)
        return result
}
*****
* Remove all punctuation from a String
*****
String stripPuncuationAndRemoveAccents( String input ){
        tempString =  stripPuncuation( input )
        output =  normalizeStringAndRemoveAccents( tempString )
        return output
}
*****
* Normalize characters represented differently in UTF-16 and UTF-8
* to be the same character.
*****
String normalizeString( String input ){
        output = java.text.Normalizer.normalize( input, Form.NFD )
        return output
}
*****
* Normalize and remove Diacritical Marks
*****
String normalizeStringAndRemoveAccents( String input ){
        output = java.text.Normalizer.normalize( input, Form.NFD )
                 .replaceAll( "\\p{InCombiningDiacriticalMarks}" )
        return output
}
String getFormatCharacter( Record ){
  //Note: this format character is appended to matchKey
  First check for pre RDA...
    if( 245$h contains "electronic resource"
        OR 590$a contains "electronic reproduction"
        OR 533$a contains "electronic reproduction"
        OR 300$a contains "online resource"
        OR control field 007 starts with 'C' ){
      return 'e'
    }
  Next check for RDA...
    if( 337$a starts with 'c' ){
      return 'e'
    }
  Next check for electronic government document...
```

```
    if( 086 not null AND 856 not null ){
      return 'e'
    }
  Next check MARC file name...
    if( MARC file name contains "electronic" or "ebook"
        (ignore case) ){
      record 'Electronic' in format field
    }
    if( MARC file name contains "physical" or "print"
        (ignore case) ){
      remove Electronic from format field
      return p;
    }
  if( format field contains 'Electronic' ){
    return e;
  } otherwise return p;
}
```

Match Key Example

```
americancouncilofIearnedsocietiesannualreportfortheyears20062007and20052006_____________________
2008________distra_______________________________________ameri_______________e
```

```
000     02801nam a22005052u 4500
001     991034738289702766
005     20170426153237.0
006     m d f
007     cr |||||||||a
008     080101s2008 xxu|||| st ||| ||eng d
035     |a(CoDU)b49584571-01uode_inst
035     |a(ERIC)ED516193|9ExL
035     |a(MvI) 3N000000499872
040     |aericd|cMvI|dMvI|dCoDU
086     0 |aED 1.310/2:516193x
099     9|aED516193
110     2 |aAmerican Council of Learned Societies.
245     10|aAmerican Council of Learned Societies Annual Report for the Years 2006-2007 and
2005-2006|h[electronic resource].
260     |a[S.l.] :|bDistributed by ERIC Clearinghouse,|c2008.
300     |a1 online resource (66 p.)
500     |aAvailability: American Council of Learned Societies. 633 Third Avenue 8th Floor, New
York, NY 10017. Tel: 212-697-1505; Fax: 212-949-8058; Web site: http://www.acls.org.|5ericd.
500     |aAbstractor: ERIC.|5ericd.
516     |aText (Reports, Descriptive).
520     |aThe American Council of Learned Societies (ACLS) provides the humanities and related
social sciences with leadership, opportunities for innovation, and national and international
representation. The American Council of Learned Societies was created in 1919 to represent the
United States in the Union Academique Internationale. Its mission is "the advancement of
humanistic studies in all fields of the humanities and social sciences and the maintenance and
strengthening of national societies dedicated to those studies." This paper presents the annual
report of ACLS for the years 2005-2006 and 2006-2007.
524     |aAmerican Council of Learned Societies.|2ericd.
```

```
650     07|aHumanities.|2ericd.
653     0 |aUnited States.
655     7|aReports, Descriptive.|2ericd.
710     2 |aAmerican Council of Learned Societies.
730     0 |aERIC documents (Online)
856     40|uhttp://www.eric.ed.gov/contentdelivery/servlet/ERICServlet?accno=ED516193|zAccess
online
907     |a.b49584571|b05-09-16|c08-12-11
945     |aED516193|g1|h908|j0|k0|linter|o-|p$0.00|r-|sj|t0|u0|v0|w0|x0|y.i60300954|z08-12-11
998     |ain|b08-12-11|cm|da|e-|feng|gxxu|h0|i1
946     |y53824704520002766|zERIC Collection|853824704520002766
```

Key breakdown:

Title (95)	americancounciloflearnedsocietiesannualreportfortheyears20062007and20052006____________________
GMD (5)	_____ (disabled)
Pub Year (4)	2008
Pagination (4)	____
Edition (3)	___
Publisher (5)	distr
Type (1)	a
Title Part (30)	______________________________
Title Number (10)	__________
Author (5)	ameri
Title Dates (15)	_______________
Format (1)	e