**RDA phase 3: Qualification of 410 fields for initialisms and acronyms**

Gary L. Strawn

December 19, 2014

**Background**

RDA 11.13.1.2 provides for the addition of one or more terms to an access field when two organizations have the same name or closely similar names, or when the name does not convey the idea of a corporate body. RDA provides for several categories of addition to the name of a corporate body: a term for the type of corporate body, the name of a place associated with the body, the name of an associated institution, a date associated with the body, a term for the type of jurisdiction, and some other designation. The instructions in RDA apply equally to the authorized access point (RDA 11.13.1) and to variant access points (11.13.2) that represent a corporate body. The associated LC-PCC policy statement says that such additions should always be provided for access points that consist solely of uppercase letters, with or without intervening spaces and/or punctuation. The policy statement also defines the spelled-out form of the name as one of the possible "other designations" that can be added to a set of initials.

Under cataloging rules and LC/NACO practice in effect before the implementation of RDA, parenthetical qualifiers were not generally added to 41X fields for initialisms and acronyms. This means that most 41X fields for initials in pre-RDA authority records do not conform to LC-PCC practice under RDA. The LC-PCC policy statement for RDA 11.13.1.2 says that when reviewing an existing authority record, additions should be provided for 41X fields whenever the 41X field presents a conflict, but non-conflicting 41X fields for initials may be left unchanged.

Because 41X fields for initials without qualifiers are a frequent source of conflict (especially involving names other than corporate names),[[1]](#footnote-1) the interesting question arises as to whether or not during **Phase 3B** of the manipulation of the LC/NACO authority file for use under RDA the existing 41X fields for initials and acronyms can reliably be provided with a qualifier. The present paper describes the method devised to produce as many correctly-qualified 41X fields as possible.

**About this document**

Throughout this description, expressions such as "consisting solely of subfield $a" and "consisting solely of subfields $a and $b" should be understood to ignore subfields $w, $i, and $0-$9, as these are not part of the access string.

Each example shows the state of an authority record in the LC/NACO file at the time the example was added to this document. Subsequent updates to any of these authority records should not reduce their value as examples. The examples exclude variable fields that do not bear on the matter under discussion.

**Procedure**

*Record selection*

The program's work is limited to authority records for corporate names, excluding recognizable names of conferences. This program will only consider authority records for suitable corporate names that contain recognizable 410s for initials without qualifiers. Specifically:

* a candidate authority record must contain a 110 field consisting solely of subfields $a and $b[[2]](#footnote-2)
* a candidate authority record must contain at least one 410 field consisting solely of subfield $a containing only uppercase characters,[[3]](#footnote-3) punctuation,[[4]](#footnote-4) and spaces (this test ignores any parenthetical qualifier already present in the 410 field)

*Generation of candidate initials from 110 fields and from 410 fields that do not consist of initials*

This part of the program is only interested in 110 and 410 fields that consist solely of no more than subfields $a plus zero or more occurrences of subfield $b. Working backwards from the rightmost subfield to the leftmost, the program assembles successively longer versions of the name.[[5]](#footnote-5) For each such isolated term, the program uses an elaborate scheme to generate one or more potential sets of initials.[[6]](#footnote-6) The program uses this information to build a table that contains one entry for each distinct set of candidate initials that can be extracted from 110 and 410 fields. The table entry for one set of initials has information about each field in the record from which the program has derived those initials.[[7]](#footnote-7)

*For example, given this authority record (n 79000803):*

110 $a Université d'Abidjan. Institut de littérature et d'esthétique négro-africaines

410 $a Institut de littérature et d'esthetique négro-africaines (Universite d'Abidjan)

410 $a I.L.E.N.A.

410 $a ILENA

*The program can extract these candidate initials:[[8]](#footnote-8)*

|  |  |
| --- | --- |
| **Candidate initials** | **From** |
| ILEN  ILENA  IDLEEN  IDLEENA  IE  ILDNUD  ILDNAUD  IODLEDNUD  IDLEDNAUD  IU  ILDN  ILDNA  IDLEDN  IDLEDNA  ILENUA  ILENAUA  IDLEENUA  IDLEENAUA  IEUA | Institut de littérature et d'esthetique négro-africaines (Universite d'Abidjan)[[9]](#footnote-9) |
| UAILEN  UAILENA  UAIDLEEN  UAIDLEENA  UAIE  UDILDN  UDILDNA  UDIDLEDN  UDIDLEDNA  UI | Université d'Abidjan. Institut de littérature et d'esthétique négro-africaines[[10]](#footnote-10) |

*Initial handling of 410 fields that consist of initials*

The program creates a second table that contains one entry for each distinct set of initials (ignoring punctuation and spaces). Each such table entry points back to the 410 field that contains some form of those initials.

*For example, given this authority record (n 79000803):*

110 $a Université d'Abidjan. Institut de littérature et d'esthétique négro-africaines

410 $a Institut de littérature et d'esthetique négro-africaines (Universite d'Abidjan)

410 $a I.L.E.N.A.

410 $a ILENA

*The program creates one table entry, for "ILENA." This table entry identifies both 410 fields that contain this set of initials.*

If, in its initial examination of 410 fields that represent initials, the program discovers that one 410 field for initials already contains a qualifier, and another 410 field with the same initials (ignoring punctuation and spaces) does not contain a qualifier, the program copies the qualifier from one 410 field to another. If the program propagates a qualifier from one 410 field to another in this manner, it does not also attempt to match the initials to fuller texts that may be present elsewhere in the record.

*For example, given this authority record (n 79148123):*

110 $a Iowa. $b Office for Planning and Programming. $b Statistical Analysis Center

410 $a SAC (Statistical Analysis Center)

410 $a S.A.C.

*The program will copy the qualifier "(Statistical Analysis Center)" from the first 410 field for initials to the second.*

The program's propagation of a qualifier from one set of initials to another means that the program will, occasionally, produce results that are not exactly as expected.

*For example, given this authority record (n 82209730):*

110 $a Society of African Missions

410 $a S.M.A. (Society of African Missions)

410 $a SMA

410 $a Sociedad de Missiones Africanas

*The program will copy the qualifier "(Society of African Missions)" from the first 410 field for initials to the second, even though a qualifier based on the 410 shown in the example (or one of several other non-English 410 fields in the same record) would produce more reasonable results.*

*For example, given this authority record (n 82080281):*

110 $a Direction des musées de France

410 $a D.M.T.

410 $a DMT (Direction des musées de France)

*The program will copy the qualifier "(Direction des musées de France)" from the second 410 field for initials to the first, although there is no correspondence for the letter "T" in the qualifier text. In this case, the authority record is incorrect; the initials should be "D.M.F." and "DMF" instead, as indicated in a 670 field.*

*For example, given this authority record (n 50068436):*

110 $a Národní technické muzeum v Praze

410 $a T.M. (Technické muzeum )

410 $a TM

*The program will copy the qualifier "(Technické museum )", containing an unwanted space before the closing parenthesis, from the first 410 field for initials to the second.*

The program acts in a similar fashion when the 110 consists of initials plus a parenthetical qualifier, and a 410 field represents the same initials without a qualifier.

*For example, given this authority record (n 81122521):*

110 $a CAST (Group)

410 $a C.A.S.T.

*The program will copy the qualifier "(Group)" from the 110 field to the 410 for initials,*

*Matching 410s for initials to initials derived from the full name.*

The program uses several schemes in its attempt to find a correspondence between a 410 that contains only initials, and initials that the program has derived from other fields in the authority record. The program applies the schemes in the order given here, in decreasing order of likelihood and increasing order of complexity.[[11]](#footnote-11)

* The first scheme assumes that the 410 for initials represents true initials (with a single letter in the 410 pulled from each interesting word in the full name)
* The second scheme assumes that the 410 represents an acronym (with one or more letters pulled from the beginning of interesting words in the full name).
* The third scheme assumes that the 410 for initials represents in part an internal match (with one or more letters of the initials matching a character contained within a word which is also matched by another letter on its first letter.

If the program finds a match using any of these schemes, it adds a parenthetical qualifier to each of the 410 fields consisting of the words that correspond to the matched initials (plus any intervening words that do not participate in that set of initials). If the program cannot find a match using any of these schemes, it does nothing to the 410 fields that contains a given set of initials. The program generates two reports: one showing 410 fields for initials to which it has added a parenthetical qualifier, and one showing 410 fields for initials to which it has not added a parenthetical qualifier. (An additional report shows 410 fields for initials to which the program has added a parenthetical qualifier, when the content of the qualifier indicates that some attention may still be required.[[12]](#footnote-12))

*The following examples show records as modified by the program described in this document.*

010 $a no 88001548

110 $a American Relief Administration

410 $a A.R.A. (American Relief Administration)

410 $a ARA (American Relief Administration)

010 $a n 81102653

110 $a Université libre de Bruxelles. $b Institut de sociologie

410 $a IS (Institut de sociologie)

010 $a n 80004214

110 $a Food and Agriculture Organization of the United Nations

410 $a F.A.O. (Food and Agriculture Organization)

410 $a FAO (Food and Agriculture Organization)

410 $a UN/FAO (United Nations Food and Agriculture Organisation)

410 $a UNFAO (United Nations Food and Agriculture Organisation)

010 $a n 79117024

110 $a IIT Research Institute

410 $a IIT RI (IIT Research Institute)

010 $a n 82151094

110 $a Musée national d'art moderne (France)

410 $a M.N.A.M. (Musée national d'art moderne)

410 $a MNAM (Musée national d'art moderne)

010 $a n 85346151

110 $a Community Action Program (U.S.)

410 $a CAP (Community Action Program)

410 $a O.E.O.-C.A.P. (Office of Economic Opportunity. Community Action Program)

410 $a OEO-CAP (Office of Economic Opportunity. Community Action Program)

010 $a n 85012665

110 $a Madrasah al-Qawmīyah lil-Idārah (Tunisia. $b Markas al-Buhūth wa-al-Dirāsāt al-Idārīyah

410 $a C.R.E.A. (Centre de recherché et d'études administratives)

010 $a n 50061428

110 $a Union académique international

410 $a IUA (International Union of Academies)

410 $a U.A.I. (Union académique international)

410 $a UAI (Union académique international)

010 $a n 81020295

110 $a Los Angeles Philharmonic Orchestra

410 $a LAP (Los Angeles Philharmonic)

410 $a LAPO (Los Angeles Philharmonic Orchestra)

010 $a no2012086746

110 $a Tri-county Regional Planning Commission (Ill.)

410 $a TCRPC (Tri-county Regional Planning Commission)

010 $a no2012159323

110 $a Konfederatsiia Revoliutsionnykh Anarkho-Sindikalistskov *(ligatures omitted)*

410 $a K.R.A.S. (Konfederatsiia Revoliutsionnykh Anarkho-Sindikalistskov)

410 $a KRAS (Konfederatsiia Revoliutsionnykh Anarkho-Sindikalistskov)

410 $a К.Р.А.С. (Конфедерация Революционных Анархо-Синдлкалистсков)

410 $a КРАС (Конфедерация Революционных Анархо-Синдлкалистсков)

010 $a n 81128416

110 $a United States. $b Army Medical Department

410 $a A.M.E.D.D. (Army Medical Department)

410 $a AMEDD (Army Medical Department)

010 $a n 81040906

110 $a San Diego Association of Governments

410 $a SANDAG (San Diego Association of Governments)

010 $a n 80115554

110 $a International Council of Monuments and Sites

410 $a ICOMOS (International Council of Monuments and Sites)

010 $a n 78042824

110 $a Università commerciale Luigi Bocconi. $b Centro di studi sul commercio

410 $a CESCOM (Centro di studi sul commercio)

010 $a n 50078019

110 $a Berufsverband der Heilpädagogen in der Bundesrepublik Deutschland

410 $a B.H.D. (Berufsverband der Heilpädagogen in der Bundesrepublik Deutschland)

410 $a BHD (Berufsverband der Heilpädagogen in der Bundesrepublik Deutschland)

010 $a no2011104541

110 $a Brandenburg (Germany). $b Ministerium für Ländliche Entwicklung, Umwelt und Verbraucherschutz

410 $a MLUV (Ministerium für Ländliche Entwicklung, Umwelt und Verbraucherschutz)

010 $a n 50050006

110 $a Norges landbrukshøgskole. $b Institutt for bygningsteknik

410 $a I.B.T. (Institutt for bygningsteknik)

410 $a IBT (Institutt for bygningsteknik)

010 $a n 50047656

110 $a Universidad de Buenos Aires. $b Facultad de Agronomía

410 $a FAUBA (Facultad de Agronomía, Universidad de Buenos Aires)

*This match is based on an inversion of the subfields in the 110 field. The program attempts this match if a candidate field contains one and only one instance of subfield $b.*

010 $a no2005120028

110 $a New Zealand. $b Accounting Standards Review Board

410 $a ASRB (Accounting Standards Review Board)

410 $a ASRB NZ (Accounting Standards Review Board, New Zealand)

**Matches not found**

The program is not able to find all matches that involve letters within words, initials for which there is no reasonable correspondence to other access fields in the record, matches that require elaborate inversion,[[13]](#footnote-13) and other complicated cases. The program will attempt to not match a set of initials to the earlier or later name of a corporate body. Some of the examples here show side-effects of these limitations.

*The following examples show 410 fields for initials for which the program will* not *supply a qualifier.*

010 $a n 83030780

110 $a Companhia de Tecnologia de Saneamento Ambiental (São Paulo, Brazil)

410 $a C.E.T.E.S.B.

410 $a CETESB

*There is no obvious equivalent for the first initial "E".*

010 $a n 82097544

110 $a Science for the People (Organization)

410 $a. S.E.S.P.A.

410 $a SESPA

410 $a SftP

510 $w a $a Scientists and Engineers for Social and Political Action

*The initials "SESPA" correspond to the earlier name for the organization, in the 510 field. The program ignores "SftP" because it contains a mixture of uppercase and lowercase letters.*

010 $a no2006050815

110 $a Core University Program on Fisheries Sciences

410 $a FISCUP

*The program would only be able to find a correspondence between the initials and the heading if the words in the heading were inverted.*

010 $a no2008141250

110 $a Dyslexia Foundation

410 $a TDF

*"T" in the initials stands for "The".*

010 $a no2005032668

110 $a Centro Cultural dos Cordelistas do Nordeste

410 $a CECORDEL

*The program's acronym-matching routine does not work here, because the equivalent for the acronym does not extend to the* last *word in the heading.*

010 $a no2001332110

110 $a Kituo cha Ushauri Nasha, Lishe na Afya

410 $a Centre for Counselling, Nutrition, and Health

410 $a COUNSENUTH

*The program's acronym-matching routine does not work here, because the equivalent for the acronym does not extend to the* first *word in the heading.*

010 $a n 2004096942

110 $a Kyrgyz Committee for Human Rights

410 $a KCHR (Kyrgyz Committee for Human Rights)

410 $a KKPCh

410 $a Kyrgyzskiĭ komitet po pravam cheloveka

410 $a ККПЧ (Кыргызский комитет по правам человека)

410 $a Кыргызский комитет по правам человека

*The program does not attempt to find matching text for "KKPCh" because it contains a mixture of uppercase and lowercase characters.*

010 $a n 2003145324

110 $a Great Britain. $b National Board for Nursing, Midwifery, and Health Visiting for Northern Ireland

410 $a NBNI

*The program does not recognize a match for the initials because the distance between "National Board" (the first part of the initials) and "Northern Ireland" (the second part of the initials) are, according to its instructions, too far apart.*

010 $a n 80126110

110 $a Parti communiste français

410 $a S.F.I.C.

410 $a SFIC

*The program cannot match the initials because the equivalent text, Section française de l'Internationale communiste, is not present in any 110 or 410 fields.*

010 $a n 90683568

110 $a VLSI Technology, Inc.

410 $a VLSI

*The program will not match the initials to the 110 because the first word in the 110 is the same set of initials.*

010 $a n 78043846

110 $a American Society for German Literature of the 16th and 17th Centuries

410 $a ASGLSSC

*The program will not find a match because there is no 410 containing "Sixteenth" and "Seventeenth".*

010 $a n 82154776

110 $a Rossiĭskai︠a︡ sot︠s︡ial-demokraticheskai︠a︡ rabochai︠a︡ partii︠a︡

410 $a SDPRR (Social-Democratic Workers Party of Russia)

410 $a Social-Democratic Workers Party of Russia

410 $a Socjaldemokratyczna Partia Robotników Rosji

*The program used its "internal characters" scheme to find this match (matching "PR" to "PaRty", and skipping "Workers"). The program would have used the very same scheme to find a match with the Polish-language heading (matching "SD" to "*Socjaldemokratyczna*"), but the program stops work on a set of initials when it has found the first match. The program has no way to differentiate between the match "PR" to "PaRty" and the match "SD" to "SocjalDemokratyczna".*

1. The detection of such conflicts is made difficult by the lack of a "universal" headings search in many systems that make authority records available. [↑](#footnote-ref-1)
2. The program will not be able by this test exclude an authority record for the basic name of a conference (without $n, $d, and/or $c) that is coded 110 with subfield $b. [↑](#footnote-ref-2)
3. The program compares the uppercase and lowercase versions of the string to make this determination. This test causes the program to treat characters in scripts that do not have separate uppercase and lowercase forms as uppercase characters. (Examples: Hebrew, Arabic, Runic, Hiragana.) However, it is unlikely that the program will be able eventually to find qualifiers for initials presented in those scripts. This test also causes the program to treat 410 fields that consist of uppercase letters and numerals as initials (example: "MI6"). [↑](#footnote-ref-3)
4. Punctuation is defined for this program as any character in the Unicode character database with a character type code beginning "P". [↑](#footnote-ref-4)
5. For the 410 field "$a United States. $b Navy Department. $b Judge-Advocate-General's Department" the program will create initials for "Judge-Advocate-General's Department", "Navy Department. Judge-Advocate-General's Department" and "United States Navy Department. Judge-Advocate-General's Department". This simple description masks a feature introduced to improve efficiency, based on the assumption that most of the time corporate initials can be mapped to the *last* subfield in a given name. Under this assumption, the program first generates initials only from the last subfield in each candidate field, and tests those derived initials to initials found in 410 fields. Only if initials remain unmatched does the program add subfields that appear to the left of the last subfield, and test those additional combinations. [↑](#footnote-ref-5)
6. This scheme uses a variety of techniques to derive initials from a full name. The techniques involve the following considerations: does a hyphen (or slash) constitute a "breaking" character; if a word is broken at a hyphen (or slash), should a lowercase letter immediately following the hyphen be treated as an uppercase character; should the ampersand or the plus sign be included in the initials; should short words preceded by an apostrophe be included in the initials, instead of the term following the apostrophe; are all lowercased words to be omitted; are lowercased words following a hyphen (or slash) allowed; are lowercased words that are not on a list of "short" words allowed; should a word that consists solely of uppercase letters be included in its entirety in the initials (instead of being reduced to an initial); should the program stop at the first open parenthesis. This large number of techniques means that the program may spend a lot of time doing work that is of no ultimate value. To improve its efficiency, the program divides these techniques into two groups, a *basic* group and an *enhanced* group. The program first generates candidate initials using the basic rules, and attempts to match initials found in 410 fields to that set of candidates. The program only uses the enhanced techniques to derive additional candidate initials if it is not able to match all of the initials present in 410 fields to candidate initials derived via the basic techniques. [↑](#footnote-ref-6)
7. For authority record n 8004489 the program will generate the candidate initials "BOTB" from the 110 field and also from the last subfield $b of one of the 410 fields. The program creates one table entry for those initials; because the two texts from which the initials are derived are identical, this table entry only points back to the 110 field. For authority record n 80085318, the program generates the initials "FFN" from several different 410 fields; the one table entry for these initials points back to each of the source fields. [↑](#footnote-ref-7)
8. As explained elsewhere, the program does not derive all of these initials at once. The program will only generate many of these possibilities if it fails to dispose of all of the initials found in 410 fields using initials derived using the most likely techniques. [↑](#footnote-ref-8)
9. When the program is set to ignore information that precedes an apostrophe, the program always treats the word following the apostrophe as an uppercased word. This means that the current scheme cannot generate the initials "IUA" from this string. [↑](#footnote-ref-9)
10. When program is set to ignore information that precedes an apostrophe, the program always treats the word following the apostrophe as an uppercased word. This means that the current scheme cannot generate the initials "UAI" from this string. [↑](#footnote-ref-10)
11. The process of generating candidate initials and matching them to initials found in 410 fields is in practice more complicated than this, because (as described elsewhere in this document) there are additional variables to consider: the construction of the candidate field, and the techniques available for deriving candidate initials from full texts. The program considers the best matches to be those that involve initials derived from only a single subfield. So the program first uses its basic techniques to generate initials from only the last subfield in each heading (whether that be $a or $b), and performs routines to find simple initialism matches, acronym matches, and internal matches, on these derived initials. If any initials remain unmatched, the program applies its extended techniques to the same subfields, and again looks for simple matches. If any initials remain unmatched, the program generates initials from subfield combinations (subfield $a augmented by each occurrence of $b) using basic techniques, then performs the tests for initialisms and acronyms against those additional candidates; and uses enhanced techniques to generate additional initials and performs further tests if needed. If unmatched initials remain, the program generates yet more possibilities from candidate fields by inverting the positions of subfield $a and $b (when a candidate field contains one and only one instance of $b), then performs tests for initialisms and acronyms against those additional candidates. At any point, as soon as the program discovers that it has found matches for all of the initials found in 410 fields, it stops its work on the current authority record. [↑](#footnote-ref-11)
12. For example, the program generates a report when the qualifier text ends in a hyphen, or a full stop. [↑](#footnote-ref-12)
13. The program inverts a candidate field at the subfield level, if the candidate field contains only subfields $a and $b. [↑](#footnote-ref-13)