fhiso

CFPS 78
(Call for Papers Submission number 78)

Nine Necessities in a GEDCOM Replacement

Submitted by: Kessler, Louis

Created: 2013-06-05

URL: Most recent version: http://fhiso.org/files/cfp/cfps78.pdf
This version: http://fhiso.org/files/cfp/cfps78_v1-0.pdf

Description: This paper describes requirements for a Source Detail record and a Place record to be added, and stresses issues such as no extensions, registration and compliance testing, and regular updates, as well as the need to keep the standard as simple as possible.

Keywords: sources, conclusions, citations, formatting, places, events, facts, data model, syntax, extensions, registration, testing, updates

Nine Necessities in a GEDCOM Replacement

Contents

This paper was originally published June 6, 2013 on Louis Kessler's Behold Blog:
http://www.beholdgenealogy.com/blog/?p=1313

Abstract

GEDCOM is the existing standard for transfer and storage of genealogy data. The standard is almost twenty years old and will inevitably be updated or rewritten at some point in the future. GEDCOM has been a good standard that is now used by almost all genealogy software for data transfer. But it is twenty years later, and genealogy software does more than it used to. People expect all data to transfer correctly between programs, but more often than not find that it doesn't. This paper will describe nine necessities for a GEDCOM replacement.

Introduction

During the past 15 years of developing my genealogy software Behold, a program to read GEDCOM files and display all the information from it, I have had to travel deep into the guts of the GEDCOM standard, and discover how it works and interpret its workings.

Attempting to make Behold a very flexible GEDCOM reader, I've also had to loosen the reins of the standard and allow for the input of the various dialects of GEDCOM the hundreds of different programs have invented. This has led me to two important observations:

1. GEDCOM has served as a good standard, and all genealogists are fortunate that it was developed, but it does need some updating. However this update should not be a rewrite or even an overhaul, or too much "goodness" will be lost.

2. The number one complaint about GEDCOM is that it won't transfer all their data from one program to another. I can make the case that this is not entirely GEDCOM's fault, but is often the fault of the genealogy program developers who may have misinterpreted, misused, or incorrectly programmed some of the standard. The developers have also added their own extensions and custom tags where they saw fit, giving no other program a basis to correctly read their output.

Over the past few years, I have written many articles about GEDCOM on my Behold genealogy blog (Kessler, Louis Kessler's Behold Blog). In addition, I have been a regular contributor at the BetterGEDCOM wiki, the GEDCOM X discussion, and have been a supporter of the FHISO organization.

This article will summarize the most important considerations that I feel are necessary when the next GEDCOM replacement is written.

1 Separation of Sources from Conclusions

The existing GEDCOM standard correctly stresses the recording of the conclusions of one's research as well as the documentation of the sources of those conclusions. However, GEDCOM's data structure mixes the conclusion data with the source data causing confusion and inefficiencies.

The conclusion model and the source model must be clearly separated to allow two separate databases. This will allow source-based data entry that will not require conclusions be drawn first. This will also enable and encourage source repositories to record and index their source material in online searchable source databases.

These online source references would then be able to be linked to, or downloaded into one's genealogy program and recorded as the evidence of that person's conclusions.

Not keeping "conclusions" separate from the raw source details is the most significant deficiency in GEDCOM. Source details[1], which GEDCOM unfortunately misnames as SOURCE_CITATION[2], must either be put into its own record or into a subrecord of the source. (Kessler, My Feedback to GEDCOM X, 2012)

GEDCOM 5.5.1 places conclusion data with the source details here:

SOURCE_CITATION:=
n **SOUR** @<XREF:SOUR>@ {1:1}
+1 PAGE <WHERE_WITHIN_SOURCE> {0:1}
+1 EVEN <EVENT_TYPE_CITED_FROM> {0:1}
+2 ROLE <ROLE_IN_EVENT> {0:1}
+1 DATA {0:1}
+2 DATE <ENTRY_RECORDING_DATE> {0:1}
+2 TEXT <TEXT_FROM_SOURCE> {0:M}
+3 [CONC|CONT] <TEXT_FROM_SOURCE> {0:M}
+1 <<MULTIMEDIA_LINK>> {0:M}
+1 <<NOTE_STRUCTURE>> {0:M}
+1 QUAY <CERTAINTY_ASSESSMENT> {0:1}

The PAGE, EVEN, DATA and MULTIMEDIA_LINK are what can be called the "source detail", i.e. a description of the specific location within a given source that provides a certain set of information.

GEDCOM's mistake (other than the incorrect naming of SOURCE_CITATION) is mixing this with the user's conclusion and/or comments about the source (i.e. the NOTE_STRUCTURE) and the user's Certainty Assessment (the QUAY). Both of these items are subjective information about the source detail.

---

[1] A source detail is simply information and remains information until it gets used by the researcher to assist in the development of a conclusion. Once that happens, the source detail becomes "evidence" and can be referred to as evidence in support of the conclusion. The term "evidence" is often used incorrectly when there are only facts and no conclusion and the term "source" is meant. (Kessler, BetterGedcom - Data - Discussion - GEDCOM X, 2012)

[2] A "citation" is the formal textual way to write a reference to a source detail. It is improper to refer to the source detail or to the pointer to the source detail as a citation. (Kessler, Evidence - How Do You Transition from Person Based Genealogy to Record Based Genealogy? - Genealogy & Family History Stack Exchange, 2012)

GEDCOM then assigns the SOURCE_CITATION to an individual (INDI), family (FAM), multimedia (OBJE), note (NOTE), association structure (ASSO), event detail or name. This is intended to be the documentation about the conclusion drawn from the source detail.

What must be done instead is:

1. Separate the conclusion data (NOTE_STRUCTURE and QUAY) and leave it with individual, family, etc. that it is documenting. Possibly changing the tag NOTE in this case to CONCLUSION might make it more obvious as to what is intended.

2. Place the remaining source detail information (PAGE, EVEN, DATA and MULTIMEDIA) either in a new SOURCE_DETAIL record that is linked to the SOURCE that it belongs to.

The key thing here is that three of the records, SOURCE, SOURCE_DETAIL and REPOSITORY should not contain any conclusions. They must be just the facts.

This will allow genealogy software to implement the much needed technique of source-based data entry (Kessler, How Source Based Data Entry Should Work, 2012). This will complement the widely-used method of conclusion-based data entry that minimizes the need to properly record sources.

This will also allow Repositories to start indexing their genealogical material in a standard manner that genealogy programs can search, access, download and link to.

2 Citations and Formatting should NOT be part of the Standard

A new GEDCOM should NOT transfer citations.

Only the source information to identify the source needs to be transferred. Citations are like formatting. Once you have the source information, you can create the citation.

And each program should be allowed to create the citation any way they want. If they use Evidence Explained, then so be it. They may interpret EE differently than another program, and they should be allowed to do so their own way, and display it their own way. They may even give you, if they want, other options, e.g. Richard Lackey or even bibliographic methods such as APA or Chicago. Again, it should be up to the program, and not up to the standard to force it to one interpretation of one methodology.

The important thing is that the source data can be transferred. And GEDCOM does that reasonably.

This will be a controversial opinion, but a line must be drawn. Information should be the only thing transferred. One program should not tell another program how it should format and display that information. Structuring and formatting information should not be transferred.

The beauty in the variety of genealogy software is that they display your data in different ways. Some people like it one way. Some people like it another way. Forcing display of data in certain ways only restricts the choice.

Citations are something that most people don’t use. They are only formally required for published documents and papers. When a genealogist finally decides to publish, they can then obtain some software that will allow them to do so, and produce citations in the format

that they wish to see.

Similarly, text formatting of any type should not be transferred. There have been requests by users to transfer between programs their format codes that exist in various fields, especially note fields. Doing so is a disservice. It again forces the developer of the second program to incorporate formatting that may conflict horribly with its own screen or report display.. Formatting should be up to the program.

Part of the new standard must state that all embedded format codes in any field, especially note fields, must be removed prior to export.

To transfer some info verbatim, with formatting included, then an image of the info should be created as a graphic and included with the info as a multimedia (OBJE). Then this graphic could be shown by the receiving program to display the info in original form.

3 Places must be a Top Level Record

Currently in GEDCOM 5.5.1, a place is entered as follows:

PLACE_STRUCTURE:=
n **PLAC** <PLACE_NAME> {1:1} p.58
+1 FORM <PLACE_HIERARCHY> {0:1} p.58
+1 FONE <PLACE_PHONETIC_VARIATION> {0:M} p.59
+2 TYPE <PHONETIC_TYPE> {1:1} p.57
+1 ROMN <PLACE_ROMANIZED_VARIATION> {0:M} p.59
+2 TYPE <ROMANIZED_TYPE> {1:1} p.61
+1 MAP {0:1}
+2 LATI <PLACE_LATITUDE> {1:1} p.58
+2 LONG <PLACE_LONGITUDE> {1:1} p.58
+1 <<NOTE_STRUCTURE>> {0:M} p.37

The PLACE_STRUCTURE is *only* allowed in EVENT_DETAIL.

There is a major problem with this. The information subordinate to the place name includes FORM, FONE, ROMN, MAP and notes. Every time one particular place is entered, and one place may be entered thousands of times under thousands of different events, the subordinate information must be repeated. If it is not repeated, it is subject to being included with different values under different events, and any program reading this in will not know the correct information to assign to the place.

The proper solution to this is to create a PLACE record. Information about the place then can be recorded properly once, subordinate to the PLACE record.

Then, additional useful place information can be included to the PLACE record. One example that I recommend be considered, would be to allow events to assigned to places. These can be used to document a range of events such as fires, wars, or historic events that do not pertain to particular people, but pertain more so to the place. These events would then be usable by programs for timelines and important life events.

4 Treat Events and Facts the same

People mix up what are events and what are facts. They think that events need to happen at a certain time at a certain place. They think of facts as something that is always true.

In fact, there is very little difference between events and facts.

An event is a transition. It is the change of one fact to another fact. For example, we have a marriage event. Before the marriage, the fact is that the person is unmarried. After the marriage, the fact is that the person is married. So events separate facts, and facts separate events.

An event is not instantaneous. A marriage ceremony takes several hours. A ship's crossing can take several weeks. A war can take several years. Nor is a fact immutable. Facts have start and end dates. A person can be married for several years or just for several hours.

Some things can be both events and facts. World War II was an event. It separated the period before the war from the period after the war. It also was the state of things (a fact) between the events of the start of the war and the end of the war.

Events and facts can contain other events and facts. Within World War II were many battles.

So events and facts both can be valid for a single day or for longer periods. They both can have a starting date and an ending date. Sometimes, their exact periods cannot be precisely defined, e.g. when did you uncle have brown hair and when did he have white hair, because there may be a transition period.

The bottom line is that the same information needs to be recorded for both events and facts. Some programs don't allow dates to be stored with facts. If so, it may be impossible in that program to effect the change of certain facts, such as a person's religion, height, hair color, name or sex.

GEDCOM 5.5.1 refers to EVENT_OR_FACT_CLASSIFICATION. It puts the two together. But it then still attempts to separate the EVEN tag from the FACT tag without defining the difference in great detail. The FACT tag was only added in 5.5.1. Yet, nearly every structure under a FACT is essentially equivalent to the structure under an EVEN. (AdrianB38, 2011) What's the difference? Why complicate things.

A new standard should put events and facts together as one. Why complicate the issue by separating them? It does not matter what they are called. Probably just calling them "Facts" is best. (Kessler, Fact -- Proposed Definition, 2011)

5 The Data Model Matters. Syntax Doesn't.

There has been too much discussion on whether XML or JSON or whatever be used instead of the GEDCOM syntax. But it doesn't matter. As long as a basic tree-structure based data model is developed (Pauly, 2011), any syntax can easily and mechanically be translated into another. If so desired, the standard could even allow multiple syntaxes. Free utility programs would be able to easily translate between them, as long as they use the same data model.

In other words, the data model should be defined without worrying about the final syntax to be chosen. So the new standard should be defined in something neutral, like Backus-Naur Form (BNF)[3] . And leave the decision of the syntax to the end.

[3] Backus-Naur Form is one of the two main notation techniques for context-free grammars. (Backus-Naur Form, 2013)

6 KISS

Any complication will make the standard harder to implement. Keep it simple.

Any complication will increase the chance that the standard will fail to do its job, because programs will not correctly transfer the data between them. Keep it simple.

When there is a choice several good ways of defining a certain part of the new standard, pick the simpler one.

An example. GEDCOM implemented several nice shortcuts, easy for developers to implement, that simplified certain constructs.

For example: Names were set up with the surname between slashes,

e.g. “Sir John /Byron/ Jr.”

This clearly shows the surname between slashes, with titles and given names prior and post names after. This was simple and developers could implement this with a single string field.

GEDCOM attempted to add name pieces, and later realized the complication it caused. GEDCOM 5.5.1 states “For current future compatibility, all systems must construct their names based on the <NAME_PERSONAL> structure. Those using the optional name pieces should assume that few systems will process them, and most will not provide the name pieces.”

It is true that the future standard should accommodate worldwide name variations. But this, and everything else in the standard, needs to be done in the simplest way possible. A proper naming standard for world names could turn into its own mega-standard, even larger than the rest of the GEDCOM replacement. Few would be willing to implement such a standard, and it would only derail the entire effort to develop the genealogy standard. Keep it simple.

One other example: Places were set up as a single string, lowest to highest hierarchy, with the levels separated by commas,

e.g. “Cove, Cache, Utah, USA"

Again, what can be simpler. This sort of standard is easily implemented. But a multi-level place structure that links each level to multiple other levels, just adds complexity and difficulty for programmers, with barely a benefit that makes the extra tedium worth the effort.

7 No Extensions

GEDCOM currently allows user-defined tags. They are tags a program may define that begin with an underscore.

This is possibly the biggest mistake GEDCOM made. There is no way one program can understand what another program’s user-defined tags mean, unless the developer does the extra work to understand that program and implement the data structures and code to allow for it. (Kessler, A Plethora of Extra GEDCOM Tags, 2011)

User-defined tags result in non-transferability of data, and that’s the one thing the new standard is attempting to remedy.

GEDCOM attempted in early versions to even make its syntax extendible. It defined the SCHEMA tag, where user-defined tags could be set up so that other programs could understand them. The method was poorly implemented by Family Tree Maker and only caused more problems and complications than it was worth. The use of the SCHEMA was

eliminated in GEDCOM 5.4. The statement about the elimination was:

“Although the schema concept is valid and essential to the growth of GEDCOM, it is too complex and premature to be implemented successfully into current products. Implementing it too early could cause developers to spend a great deal of resources programming something that would be outdated very quickly. Object definition languages are likely to contribute to meeting these needs.”

This was a nice statement to say that maybe schema modification should be allowed in the future, but the complexity and ability to get programmers to accommodate it, will likely make it a non-starter for the foreseeable future. (Kessler, Tuesday, June 14, 2005)

In conclusion, there must be a hard and fast rule that no extensions should be allowed. The standard will define everything that is allowed precisely.

Even with this rule, there is still one excellent way to allow flexibility. That is through the TYPE tag. This is a tag that allows a string to represent different user-defined types for a particular tag. The context will be known by the tag it is representing, but the tag need not be treated differently because of it. Here’s a few examples from GEDCOM 5.5.1 of the TYPE tag in use:

```
1 MARR
        2 TYPE Common Law

1 EVEN
        2 TYPE Awarded BSA Eagle Rank
        2 DATE 1980

1 EVEN
        2 TYPE Land Lease
        2 DATE 2 OCT 1837

1 GRAD
        2 TYPE College

n IDNO 43-456-1899
        +1 TYPE Canadian Health Registration

1 EVEN Appointed Zoning Committee Chairperson
        2 TYPE Civic Appointments
        2 DATE FROM JAN 1952 TO JAN 1956
        2 PLAC Cove, Cache, Utah
        2 AGNC Cove City Redevelopment

1 FACT Woodworking
        2 TYPE Skills
```

Note that in the last two examples, the generic EVEN and FACT tags are followed by a textual descriptor. This would be another allowable way to extend a list of object types.

The key thing to remember is that only data values can be allowed to change. If the item intrinsically has a different purpose and cannot be represented by a descriptor or TYPE tag, then it must be specifically written into the new standard in order for it to be used.

8 Registration and Compliance Testing

GEDCOM 5.5.1 still has a field for an APPROVED_SYSTEM_ID. This is defined as:

“A system identification name which was obtained through the GEDCOM registration process.”

Yes, long ago, the various systems that used GEDCOM had to be given System IDs by FamilySearch. (Jones, GEDCOM SOUR and DEST, 2011) This overseeing no longer occurs. But it was and is a very good idea. I would hope that some independent organization, such as FHISO[4], would oversee this and assign some sort of certification that a program properly meets the new standard, once released.

To do so, the program must meet some compliance tests. The minimum requirement is that it must be able to:

1. Read a file written in the new GEDCOM standard,
2. Write out the file again, and the file should be exactly the same.

The program need not handle all the constructs in the standard, but it must pass through all the data verbatim, whether it handles it or not. Doing so will be what is necessary to ensure that data will transfer properly between programs.

The test suite will also provide the following benefits:

1. Allow developers to assess compliance of their software
2. Help developers diagnose issues
3. Assist developers in resolving issues.

Also, by certifying programs, the standard itself will have more weight and merit.

9 Updates to the Standard

Since I earlier stated that there should be no user-based extensions allowed to the standard, there must then be some mechanism put in place to allow the standard to be updated. Doing so can correct deficiencies found, and include new features needed.

This should be done on a regular basis, again by some independent organization.

It should be done at most, once every two years. It should not be done more often, because developers need time to implement changes to a standard, and then the changes to the standard need time to take hold and for compliance testing to be performed. Then a period of stability is needed to ensure that all is okay prior to the next set of changes.

All changes must have a real need and must be important, required, and desired. The changes must be kept as simple as possible. They will need to be reviewed and assessed prior to implementation. Again, this is an independent organization’s job.

There will need to be a procedure set up for this. One possibility is to use an online tool like UserVoice (e.g. http://windowsphone.uservoice.com/) to keep track of the requests and allow developers to comment and vote on them.

[4] The Family History Information Standards Organisation FHISO) is a community-driven organisation established for the purpose of developing genealogy and family history information standards. www.fhiso.org

Summary and Recommendation

There is a lot of “goodness” in the existing GEDCOM standard. This should not all be thrown away in any new standard, but should first be carefully analysed piece by piece and improved where needed.

The Header (HEAD), Individual (INDI), Family (FAM), Multimedia (OBJE), Note (NOTE), Repository (REPO), Source (SOUR) and Submitter (SUBM) records all have useful and valid purposes. I argue in this article that a Source Detail record and a Place record be added.

I avoid mentioning elimination of the Family (FAM) record as some have suggested, since it serves as an adequate placeholder for spousal and parent-child relationships. A Group record might serve better, but may be too radical a change for many programs to support.

I distinctly recommend against an Event Record as some would suggest. Instead, the Event raw data should be placed with the Source or Source Detail, and the Event conclusions should be with the Individual, Family or Place where it belongs.

I have stressed issues of no extensions, registration and compliance testing, and regular updates at not-too-frequent intervals. And above all, to keep it as simple as possible.

Hopefully these necessities in a GEDCOM replacement will be followed and will provide a new standard to serve the genealogy community for the next twenty years and longer.

Works Cited

Backus-Naur Form. (2013, May 28). Retrieved from Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Backus%E2%80%93Naur_Form

AdrianB38. (2011, February 25). *Syntax09 Define Event vs. Attribute*. Retrieved from BetterGedcom Wiki: http://bettergedcom.wikispaces.com/share/view/34875604

Jones, T. (2010, August 24). *A Gentle Introduction to GEDCOM.* Retrieved from Modern Software Experience: http://www.tamurajones.net/AGentleIntroductionToGEDCOM.xhtml

Jones, T. (2011, July 15). *GEDCOM SOUR and DEST*. Retrieved from Modern Software Experience: http://www.tamurajones.net/GEDCOMSOURandDEST.xhtml

Kessler, L. (2005, June 14). *Tuesday, June 14, 2005*. Retrieved from Louis Kessler's Behold Blog: http://www.beholdgenealogy.com/blog/?p=329

Kessler, L. (2011, November 21). *A Plethora of Extra GEDCOM Tags*. Retrieved from Louis Kessler's Behold Blog: http://www.beholdgenealogy.com/blog/?p=876

Kessler, L. (2011, January 4). *BetterGedcom - What IS BetterGEDCOM - Discussion - Thoughts on GEDCOM and Better GEDCOM.* Retrieved from BetterGedcom Wiki: http://bettergedcom.wikispaces.com/message/view/What+IS+BetterGEDCOM/32188988#32261610

Kessler, L. (2011, March 6). *Fact -- Proposed Definition*. Retrieved from BetterGedcom Wiki: http://bettergedcom.wikispaces.com/share/view/35386802?replyId=35398486

Kessler, L. (2012, January 25). *BetterGedcom - Data - Discussion - GEDCOM X.* Retrieved from BetterGedcom Wiki: http://bettergedcom.wikispaces.com/message/view/Data/48419278?o=20#49613784

Kessler, L. (2012, October 17). *Evidence - How Do You Transition from Person Based Genealogy to Record Based Genealogy? - Genealogy & Family History Stack Exchange.* Retrieved from Genealogy & Family History Stack Exchange: http://genealogy.stackexchange.com/a/1472/29

Kessler, L. (2012, May 14). *How Source Based Data Entry Should Work.* Retrieved from Louis Kessler's Behold Blog: http://www.beholdgenealogy.com/blog/?p=1094

Kessler, L. (2012, May 14). *How Source Based Data Entry Should Work*. Retrieved from Louis Kessler's Behold Blog: http://www.beholdgenealogy.com/blog/?p=1094

Kessler, L. (2012, March 29). *My Feedback to GEDCOM X.* Retrieved from Louis Kessler's Behold Blog: http://www.beholdgenealogy.com/blog/?p=1080

Kessler, L. (n.d.). *Louis Kessler's Behold Blog*. Retrieved from Behold Genealogy: http://www.beholdgenealogy.com/blog

Pauly, G. G. (2011, August 28). *xml?* Retrieved from BetterGedcom Wiki: http://bettergedcom.wikispaces.com/share/view/41734721

Thorud, G. (2011, March 1). *PFACT (Property, Fact, Attribute, Characteristic, or Trait)*. Retrieved from BetterGedcom Wiki: http://bettergedcom.wikispaces.com/share/view/35102958