The FamilySearch GEDCOM 5.5.1 Specification

Annotated Edition

edited by
Tamura Jones

technical reviewers

Tim Forsythe	creator of Gigatrees and VGedX, the Gigatrees GEDCOM validator
Diedrich Hesmer	creator of Our Family Book and GEDCOM Service Programs
Andrew Hoyle	creator of Chronoplex My Family Tree and the Chronoplex GEDCOM Validator
Louis Kessler	creator of Behold, GEDCOM File Finder, and Double Match Triangulator
Stanley Mitchell	creator of ezGED Viewer
Nigel Munro Parker	creator of the GED-inline GEDCOM validator
Keith Riggle	blogger & genealogy software reviewer

Revision One
6 July 2018

Copyright © 2013 - 2018 Tamura Jones.
All rights reserved.

annotation

This is the first annotation in this document. It's purpose is to show what annotations look like; a box with light yellow background and a grey shadow, clearly distinct from the main text.

motivation and purpose

bringing it together

The original concept for the *Annotated Edition* was to bring helpful observations, remarks, suggestions, corrections and best practices from articles about GEDCOM into the GEDCOM specification itself, thus improving the specification and making the relevant articles easier to find. The project evolved beyond that to contain information and observations not found in previously published articles, especially concerning sections that should now be considered obsolete or deprecated.

annotations and corrections

The original idea was to add nothing but annotations, but creating an annotation box for a mere spelling correction or a simple word or phrase replacement seemed too much, so not all annotations are presented in a box like this, some are shown as edits. The entire original text is still here, and all edits are very clearly identified as such.
A benefit of this approach is that erroneous text is now clearly marked as such; it is clear something is wrong even without reading an annotation that may be there in addition to the correction.

expert review

Review by several experts did not only catch many typos and spelling errors, and a few ambiguities, but made sure that the annotations are sensible, and unlikely to be in error themselves. Their observations did not only improve existing annotations, but often sparked the creation of new ones.

improved specification

The result is a document that, although far from perfect, is considerably more informative than the un-annotated specification. It provides guidance on what's obsolete or deprecated. It explains things that FamilySearch left unexplained, like the usage of the double at-sign, or why some letters in the <<RECORD>> definition are bold. It resolves contradictions between different parts of the specifications, such as the maximum length of an identifier.
It corrects obvious errors, such as the impractically small maximum length of media file names. It notes non-obvious errors, like the omission of tab as a legal character in user text. It corrects erroneous definitions, in the GEDCOM grammar, in the lineage-linked form and the Appendix. It adds helpful observations, facts that developers may not immediately notice, like that it is legal for identifiers to contain spaces. It provides various best practices, including best practices for character sets and encodings as well as CONC and CONT usage.

page numbers

Many links in the *FamilySearch GEDCOM 5.5.1 specification* include page numbers. This *Annotated Edition* sticks with the page numbers of the unannotated edition. The original page breaks are shown, and the corresponding page numbers are shown in the right-hand margin. To avoid confusion, the pages of this document are not numbered.

THE GEDCOM STANDARD

~~DRAFT~~ Release 5.5.1

Prepared by the
Family History Department
The Church of Jesus Christ of Latter-day Saints

2 October 1999

About GEDCOM 5.5.1

5.5 versus 5.5.1

The GEDCOM 5.5.1 specification replaced GEDCOM 5.5 as the *de facto* industry standard. Although the version number difference is small, it really does matter which version a genealogy application uses. Among the new features of GEDCOM 5.5.1 are support for email addresses, web addresses, and geographic coordinates. Another significant difference is that GEDCOM 5.5.1 allows the use of UTF-8, while GEDCOM 5.5 does not.

GEDCOM 5.5.1 is *not* fully backward compatible with GEDCOM 5.5; most significantly, GEDCOM 5.5 allows encoding multimedia objects inside a GEDCOM file, GEDCOM 5.5.1 does not.

last version

The *FamilySearch GEDCOM 5.5.1 specification*, published in 1999 CE, is the last GEDCOM specification that FamilySearch published. A GEDCOM 5.6 specification created in 2000 CE surfaced in 2011 CE, but is hardly different from GEDCOM 5.5.1. The GEDCOM 5.6 specification is of historical interest only and has been widely ignored. The only interesting thing about GEDCOM 5.6 is that it introduces GEDXML, a GEDCOM alternative.

Around 2000 CE, FamilySearch abandoned development of GEDCOM, in favour of experimenting with new standards, called GEDXML, GEDCOM XML, and GEDCOM X. Those unoriginal names are misleading; GEDXML, GEDCOM XML, and GEDCOM X are neither new GEDCOM versions, nor compatible with GEDCOM. GEDXML, GEDCOM XML, and GEDCOM X are GEDCOM alternatives, which FamilySearch intended as GEDCOM replacements.

ostensible draft

GEDCOM 5.5.1 is not a draft, and never really was a draft. The *FamilySearch GEDCOM 5.5.1 specification* states that it is a draft, but actions speak louder than words.
When FamilySearch published the GEDCOM 5.5.1 specification, back in 1999, they publicly assigned it *draft* status, and told other vendors of genealogy software that they should not use it, while they themselves moved forward and implemented most of GEDCOM 5.5.1's new features in

their own Personal Ancestral File version 5.0.

When the creator of a standard uses that standard in a major product, that standard isn't a draft. What's more, by having PAF use GEDCOM 5.5.1 features and having it produce GEDCOM 5.5.1 files, FamilySearch not only used GEDCOM 5.5.1 themselves, but also practically forced other vendors to follow suit, and implement the same GEDCOM 5.5.1 features.

Last but not least, there is a remarkably hard to find statement on the LDS site that clearly states that FamilySearch (the Family History Department of the LDS) uses GEDCOM version 5.5.1. It is in the `README.TXT` file, dated 16 Feb 2000 CE, in the download directory for the *GEDCOM Specification Future Direction* draft: “For at least the near future, the Family History Department is using the GEDCOM Standard version 5.5.1.”.

GEDCOM version

FamilySearch's deception would be a historical footnote by now, were it not for another thing they did. PAF version 5.0 and later lie about the GEDCOM version used; PAF uses GEDCOM 5.5.1, but the GEDCOM header of the GEDCOM 5.5.1 files it creates lies that it is a GEDCOM 5.5 file. Several other vendors, looking to PAF for guidance on how to support GEDCOM, followed suit. This use of a truncated version number complicates GEDCOM version detection, as you can no longer simply trust the stated GEDCOM version any more. There are many GEDCOM 5.5.1 files that claim to be GEDCOM 5.5 files. The GEDCOM 5.5.1 specification does not provide any guidance on this issue.

references

- A Gentle Introduction to GEDCOM
- FamilySearch GEDCOM Specifications
- GEDCOM 5.6.
- GEDCOM Alternatives.
- GEDCOM 5.5.1 isn't a Draft.
- Truncated GEDCOM Version.
- GEDCOM Version Detection.
- ftp://ftp.ldschurch.org/genealogy/GEDCOM/Future/README.TXT

GEDCOM 5.5.x errata

The *FamilySearch GEDCOM 5.5 specification* was published on 2 Jan 1996, and includes errata relative to a draft dated 11 Dec 1995. An errata sheet dated 10 Feb 1996 enjoyed limited distributed to a few FamilySearch-favoured contacts only. FamilySearch did not release an updated GEDCOM 5.5 specification incorporating the errata.

The *FamilySearch GEDCOM 5.5.1 specification* was published on 2 Oct 1999. FamilySearch has not publicly published any errata since.

All the errata mentioned in the GEDCOM 5.5 specification and its errata sheet were incorporated in the GEDCOM 5.5.1 specification, except for “Encoding and Decoding Algorithms for Multimedia Objects”, as GEDCOM 5.5.1 dropped support for embedded media.
This annotated GEDCOM 5.5.1 specification includes a number of annotations based on the GEDCOM 5.5 errata.

FamilySearch “UNICODE” versus Unicode

Unicode is a character set for which there are multiple encodings, the best known and most common encodings are UTF-8, UTF-16 and UTF-32. GEDCOM 5.5.1 allows the use of UTF-8 and UTF-16.

The UTF-16 encoding was introduced as a legal GEDCOM encoding in the *FamilySearch GEDCOM 5.5 specification* (1995). Within the GEDCOM 5.5 specification, FamilySearch consistently misidentifies UTF-16 as “UNICODE”, *as if* UTF-16 and Unicode are the same thing. Not once does FamilySearch refer to UTF-16 as “UTF-16”; FamilySearch always refers to UTF-16 as “UNICODE”.

The GEDCOM 5.5.1 specification introduced the UTF-8 encoding in addition to the UTF-16 encoding, yet did not correct this mistake. Throughout the GEDCOM 5.5.1 specification, FamilySearch uses “UNICODE” *as if* UTF-16 and Unicode are the same thing, despite the fact that GEDCOM 5.5.1 allows the UTF-8 encoding as well.

In most cases, FamilySearch's “UNICODE” means UTF-16, in some cases it means Unicode.
It is up to reader to read through FamilySearch's misconceptions.

Microsoft “UNICODE”

FamilySearch's confused terminology seems to derive from their understanding of Microsoft's documentation for 32-bit Windows. Thus, FamilySearch “UNICODE” is really Microsoft's “UNICODE”, and that is UTF-16LE, the Little-Endian encoding of UTF-16.
It is strongly recommended that applications that support UTF-16 encoded GEDCOM files be able to read both Little-Endian ad Big-Endian UTF-16 GEDCOM files.

references

FamilySearch UNICODE

LDS extensions

GEDCOM allows the definition of extensions through so-called user-defined tags. The FamilySearch GEDCOM specification includes record types that should never have been included as part of GEDCOM proper, but have been documented separately as extensions specific to *The Company of the President of The Church of Jesus Christ of Latter-Day Saints* (LDS).
FamilySearch, originally known as *The Family History Department of The Church of Jesus Christ of Latter-day Saints* (FHD), is the genealogy data collection department of the LDS.
Many of the LDS Extensions are related to Ancestral File, an application used for a collection of data also known as Ancestral File.

FamilySearch used its position as maintainer of the GEDCOM specification to include the LDS-specific extensions as part of the standard proper. Those LDS-specific record types are not always clearly identified as such.
FamilySearch thus created the situation that, to claim full GEDCOM support, a product had to support all LDS extensions as well.

obsolete

The LDS extensions are obsolete now, as FamilySearch no longer uses Ancestral File, the application that the extensions are for. This underscores that those record types are extensions that should have been documented as such, and never have burdened and complicated the standard proper.

superfluous separatism

Another reason the LDS extensions should never have been included in the standard is that they are superfluous.
While there are baptism, confirmation and other record types for all religions already, FamilySearch added separate record types for the LDS, and then added “not LDS” to the general record type.
There is no good reason for separate LDS event records, as the LDS events can be documented using the regular event records. The duplicate record types constitute bad design; there should be one way to encode a fact such as baptism, not two different ways.

separatism backfired

Because FamilySearch included their LDS extensions in the GEDCOM specification itself, vendors that wanted to claim full GEDCOM support included the LDS extensions, but many vendors did not. Products that do not recognise the LDS extensions will not import those records, and the information recorded in them will be lost. Thus, the practical result of FamilySearch's separatism is that, when data is transferred via GEDCOM, religious events documented in general religious event records are practically sure to transfer correctly, while religious events documented in LDS extensions are likely to be lost on export & import.

best practice

New products should simply use the general religious record types, and not support LDS-specific extensions.

Products that still support LDS extensions can and should continue to do so on GEDCOM import (backward compatibility is a good thing, the product should be able to read GEDCOM files produced by earlier versions of itself), but the product's export should be improved to always use the general religious record types.
The product's user interface should also be the same for all religions.

All the LDS extensions are obsolete and therefore marked as such. Additionally, all the "not LDS" notes have been stricken through.
The explanation provided here is not repeated for every individual LDS extension.

references

GEDCOM Tags

obsolete features

The *FamilySearch GEDCOM 5.5.1* specification includes features that are now obsolete. This includes, but is not limited to, all the LDS-specific record types.

Obsolete sections are highlighted through text & background colour, like this example:

HEADER:=

```
n HEAD                                      {1:1}
+1 SOUR <APPROVED_SYSTEM_ID>                {1:1}   p.42
...
+1 SUBM @<XREF:SUBM>@                       {1:1}   p.28
+1 SUBN @<XREF:SUBN>@                       {0:1}   p.28
+1 FILE <FILE_NAME>                         {0:1}   p.50
...
```

best practice

The general guideline for obsolete features is that they should not be used. Applications should not use obsolete features in the GEDCOM files they export. The general guideline for import of obsolete features is that developers should not add support for obsolete features, but need not remove any existing support either.

deprecated features

The *FamilySearch GEDCOM 5.5.1 specification* includes features that are or should be deprecated. This includes, but is not limited to, the old address format.

Deprecated sections are highlighted through text & background colour, like this example:

HEADER:=

```
n HEAD                                      {1:1}
+1 SOUR <APPROVED_SYSTEM_ID>                {1:1}   p.42
...
+1 LANG <LANGUAGE_OF_TEXT>                  {0:1}   p.51
+1 PLAC                                     {0:1}
+2 FORM <PLACE_HIERARCHY>                   {1:1}   p.58
+1 NOTE <GEDCOM_CONTENT_DESCRIPTION>        {0:1}   p.50
...
```

best practice

The general guideline for deprecated features is that they should not be used. Applications should not use deprecated features in the GEDCOM files they export. The general guideline for import of deprecated features is that developers should not add support for deprecated features, but need not remove any existing support either.

CONC | CONT confusion

The *CONC & CONT* annotation explains the many things wrong with the explicit inclusion of CONC & CONT in the lineage-linked form.

All those explicit CONC & CONT inclusions have been highlighted through text & background colour, like this example.

SOURCE_RECORD:=

```
n @<XREF:SOUR>@ SOUR                                  {1:1}
...
+1 AUTH <SOURCE_ORIGINATOR>                           {0:1}   p.62
+2 [CONC|CONT] <SOURCE_ORIGINATOR>                    {0:M}   p.62
+1 TITL <SOURCE_DESCRIPTIVE_TITLE>                    {0:1}   p.62
```

C Style comments

In a few places, the *FamilySearch GEDCOM 5.5.1 specification* contains C-style comments; a comment in between /* and */, *inside* the syntax of the lineage-linked form or inside GEDCOM examples.

GEDCOM does not support C style comments. GEDCOM does not support comments at all.

The illegal comments have been highlighted through text & background colour, like this example:

```
+1 <<NOTE_STRUCTURE>>                                 {0:M}   p.37
```

SOURCE_CITATION:=

```
[ /* pointer to source record (preferred)*/
n SOUR @<XREF:SOUR>@                                  {1:1}
```

corrections and additions

Corrections, replacements and additions of the main text have been clearly marked as such. Erroneous text has been turned dark red on light red background, and has been stricken through. Replacement text and additions are shown in dark green on a light green background.

Example of a common corrective replacement, replacing (GEDCOM) “transmission” with (GEDCOM) “file”, from chapter 1:

> A GEDCOM ~~transmission~~ file consists of a sequence of logical records, each of which consists of a sequence of **gedcom_lines**, all contained in a sequential file or stream of characters. The following rules pertain to the **gedcom_line**:

Example of additions made to the **MULTIMEDIA_LINK** definition:

> Note: some systems may have output the following 5.5 structure. The new context above was introduced in order to allow a grouping of related multimedia files to a particular context.
>
> n OBJE

+1 FILE <MULTIMEDIA_FILE_REFERENCE>
+1 FORM <MULTIMEDIA_FORMAT>
~~——+2 MEDI <SOURCE_MEDIA_TYPE>~~
+1 TITLE <DESCRIPTIVE_TITLE>
+1 <<NOTE_STRUCTURE>>

Indented GEDCOM

The GEDCOM standard should promote valid GEDCOM through valid examples.
The GEDCOM specification does not allow GEDCOM lines to start with leading white space. The specification should set the right example, especially through its examples, yet, practically every example within the specification is indented.
Pre-releases of this *Annotated Edition* had an annotation about this on practically every bad example. These annotations have been removed in favour of simply marking the indentation as an error, like this:

0 INDI
~~—~~1 BIRT
~~——~~2 DATE 12 MAY 1920
~~—~~1 DEAT
~~——~~2 DATE 1960

GEDCOM tag words

A few times FamilySearch explains an (UPPERCASE) tag by expanding it (using lowercase) into the corresponding full word, like this: “BIRTh”. This can be confusing. It is not immediately clear whether this a typo or an intentional edit. Any possible confusion has been removed by clearly distinguishing between the tag and its meaning; the tag is shown in UPPERCASE, and its meaning within parentheses immediately after it, in lower case.

This classifies the entry as a common law marriage but the event is still a marriage event. Other descriptor values might include, for example,`stillborn' as a qualifier to ~~BIRTh~~ BIRT (birth) or `Tribal Custom' as a qualifier to ~~MARRiage~~ MARR (marriage).

illogical quoting

FamilySearch uses illogical quoting throughout the GEDCOM specification. There are style guidelines that still promote this, but it is bad practice in a computing standard.
This illogical quoting may make it seem that a full stop is part of a quoted text, while that it not intended; the full stop is merely confusingly placed.

Such instances of illogical quoting have been corrected like this:

For example, if the attribute being defined was one of the persons skills, such as woodworking, the **FACT** tag would have the value of `Woodworking', followed by a subordinate TYPE tag with the value `Skills~~.~~'.

record versus structure

The *FamilySearch GEDCOM 5.5.1 specification* talks about *structures* and *substructures*, and also about *records*, but never about *subrecords*, to mean the same.
Within the annotations, preference is given to the shorter *records* and *subrecords*, where *records* includes *subrecords* as expected, unless restricted by a modifier. Any restrictions to particular records is done by using an appropriate modifier, e.g. *top-level records*, or *INDI records*.

FamilySearch's usage possibly emerged as a result of the <<RECORD>> definition; because they defined top-level records (other than the HEAD, TRLR and SUBN records), as RECORD, they started to avoid use of the word *record* for anything but those top-level records.

subrecord notation

The annotations uses a subrecord notation as a convenient shorthand for identifying which (sub)record is being talked about. For example, INDI.NAME is the NAME subrecord of the INDI record, and HEAD.GEDC.VERS is the VERS subrecord of the GEDC subrecord of the HEAD record. The notation is always used to indicate direct subrecords.
This notation is not new. It was already being used by FamilySearch, just not as frequently. For example, the <<SOURCE_CITATION>> description mentions SOUR.PAGE, SOUR.DATA.TEXT, and SOUR.EVEN.

characters, code units, and code points

Characters, code points and code units are related but different concepts:

- The length of a string is expressed in code units.
  For the 7-bit character set (ASCII), the 8-bit character set (ANSEL), and the 8-bit Unicode encoding (UTF-8) allowed in GEDCOM, the code unit is one 8-bit byte. For the 16-bit Unicode encoding (UTF-16) allowed in the GEDCOM, the code unit is two 8-bit bytes.
- The length of a character is expressed in code points.
  Both ANSEL and Unicode allow characters to be composed from a base character and one or more combining characters. The *ANSEL Non-spacing graphic characters* table in *Appendix C ANSEL Character Set* contains examples.

The *FamilySearch GEDCOM 5.5.1 specification* often talks about characters when it should be talking about code points or code units.
This text *does not annotate nor correct each such instance*. When reading this specification, understand FamilySearch's usage of the word “character” to be uninformed and informal. Keeping that in mind, it is generally clear from context when “code point” or “code unit” is meant.

“8-bit ASCII”

Throughout the *FamilySearch GEDCOM 5.5.1 specification*, the phrase “8-bit ASCII” is used.
There is no such thing as “8-bit ASCII”.
The phrase “8-bit ASCII” is a contradiction in terms, as ASCII is a 7-bit character set.

Suggestions and Correspondence:

Email:
gedcom@gedcom.org

Telephone:
801-240-4534
801-240-3442

Mail:
Family History Department
GEDCOM Coordinator— 3T
50 East North Temple Street
Salt Lake City, UT 84150
USA

invalid address

Sending any suggestions or correspondence regarding GEDCOM to any of these addresses is useless.

FamilySearch abandoned GEDCOM back in 2000 CE. There is no GEDCOM Coordinator any more, and the email address listed above is invalid.

Copyright © 1987, 1989, 1992, 1993, 1995, 1999 by The Church of Jesus Christ of Latter-day Saints. This document may be copied for purposes of review only. It must not be used for programming of genealogical software while in draft, All other rights reserved.

copyright

As already pointed in the *About GEDCOM 5.5.1* annotation, the *FamilySearch GEDCOM 5.5.1 specification* isn't a draft, in fact never was a draft. The *FamilySearch GEDCOM 5.5.1 specification* is a publicly distributed specification, and the GEDCOM version that FamilySearch themselves since their release of PAF 5.0 in 2000 CE.

The *FamilySearch GEDCOM 5.5.1 specification* is largely identical to the *FamilySearch GEDCOM 5.5 specification*, which contains a slightly different copyright notice:

> Copyright © 1987, 1989, 1992, 1993, 1995 by The Church of Jesus Christ of Latter-day Saints. This document may be copied for purposes of review or programming of genealogical software, provided this notice is included. All other rights reserved.

Similar copyright notices are included in other versions of the GEDCOM specification. Because it isn't a draft, the GEDCOM 5.5.1 specification should contain essentially the same copyright notice as the GEDCOM 5.5 specification.

This publication, *The GEDCOM 5.5.1 Specification, Annotated Edition* is an annotated edition, an in-depth and in-place review (for purposes of programming), of the *FamilySearch GEDCOM 5.5.1 specification*. It is not only allowed by the GEDCOM 5.5 copyright notice, it is even allowed by the less permissive GEDCOM 5.5.1 copyright notice.

This publication is copyrighted too. The copyright notice is on the front page.

1

2

2

TABLE OF CONTENTS

Introduction

GEDCOM was developed by the Family History Department of The Church of Jesus Christ of Latterday Saints (LDS Church) to provide a flexible, uniform format for exchanging computerized genealogical data. GEDCOM is an acronym for **GE**nealogical **D**ata **Com**munication. Its purpose is to foster the sharing of genealogical information and the development of a wide range of inter-operable software products to assist genealogists, historians, and other researchers.

misnomer

The GEDCOM name is a misnomer. The misleading name has led to erroneous descriptions of what GEDCOM is, for example as “the language by which different genealogy software programs talk to one another”.
GEDCOM isn't a communication protocol. GEDCOM doesn't contain a communication protocol. GEDCOM is a data file format.

Purpose and Content of *The GEDCOM Standard*

The GEDCOM Standard is a technical document written for computer programmers, system developers, and technically sophisticated users. It covers the following topics:

! GEDCOM Data Representation Grammar (see Chapter 1 beginning on page 9)

! Lineage-Linked ~~Grammar~~ Form (see Chapter 2, beginning on page 19)

! Lineage-Linked GEDCOM Tags (see Appendix A, 83 and Chapter 2, beginning on page 19)

! The Church of Jesus Christ of Latter-day Saints' temple codes (see Appendix B, page 96)

! ANSEL Character Codes (see Chapter 3, beginning on page 77, and Appendix C beginning on page 97)

This document describes GEDCOM at two different levels. Chapter 1 describes the lower level,known as the ***GEDCOM data format*** . This is a general-purpose data representation language for representing any kind of structured information in a sequential medium. It discusses the syntax and identification of structured information in general, but it does not deal with the semantic content of any particular kind of data. It is, therefore, also useful to people using GEDCOM for storing other types of data, not just genealogical data.

Chapter 2 of this document describes the higher level, known as a ***GEDCOM form***. Each type of data that uses the GEDCOM data format has a specific GEDCOM form. This document discusses only one GEDCOM form: the *Lineage-Linked GEDCOM Form*. This is the form commercial software developers use to create genealogical software systems that can exchange compiled information about individuals with accompanying family, source, submitter, and note records with the Family History Department's FamilySearch Systems and with each other if desired.

Two Different Levels

GEDCOM is a two-level specification. The base level, the *GEDCOM data format* is not unlike XML; it defines the basic format and syntax. The second level, a *GEDCOM form*, is not unlike an XML Document Type Definition (DTD); it defines a particular schema on top of the basic rules.

The separation between the GEDCOM data format and GEDCOM forms is half-hearted. The *FamilySearch GEDCOM 5.5.1 specification* defines just one GEDCOM form, called the *lineage-linked form*, and FamilySearch themselves has never publicly defined another GEDCOM form. In fact, FamilySearch generally fails to distinguish between the GEDCOM data format and the lineage-linked form; often referring to either or both as just “GEDCOM”. The FamilySearch GEDCOM specification does not even properly separate the GEDCOM data format and the lineage-linked from another; the GEDCOM header (in which you specify the GEDCOM form) is actually defined as part of the lineage-linked form instead of the GEDCOM data format, where it belongs. There is also just one GEDCOM version number, while there really should be two, one for the GEDCOM data format, and one for the lineage-linked form.

This document is available on the internet at the following ftp site: ftp://gedcom.org/pub/genealogy/gedcom

availability

FamilySearch stopped providing the `gedcom.org` website around 2000 CE.

The GEDCOM 5.5.1 specification, earlier GEDCOM specifications and some related documents are available from the *FamilySearch GEDCOM specifications* page: `https://www.tamurajones.net/FamilySearchGEDCOMSpecifications.xhtml`

references

FamilySearch GEDCOM specifications

Purposes for Version 5.x

Earlier versions of *The GEDCOM Standard* were released in October 1987 (3.0) and August 1989 (4.0). Versions 1 and 2 were drafts for public discussion and were not established as a standard.

GEDCOM 1 and 2

The brief statement FamilySearch makes about GEDCOM 1 and 2 is misleading. Neither was merely a draft for discussion. Both GEDCOM 1 and GEDCOM 2 have been implemented.

FamilySearch proposed GEDCOM 1 as a standard, and at least one product, the *Family History System* (FHS), an MS-DOS application created by Phillip Brown, implemented it. The GEDCOM 1.0 format is quite different from GEDCOM 2.0 and later, so much so that a typical GEDCOM reader will not even recognise it as GEDCOM. Two current genealogy applications that can read GEDCOM 1.0 files are Louis Kessler's Behold, and Chronoplex My Family Tree.

GEDCOM 2.0 was implemented by FamilySearch themselves in Personal Ancestral File 2.0 and later.

There seem to be no remaining copies of GEDCOM 2.x specifications. I asked FamilySearch about early GEDCOM specifications years ago, and FamilySearch was unable to find their own specifications. It seems that GEDCOM 2.x versions correspond to PAF 2.x releases; PAF 2.0 supports GEDCOM 2.0, PAF 2.1 supports GEDCOM 2.1.and PAF 2.2 supports GEDCOM 2.2.

references

- GEDCOM 1.0
- GEDCOM Version Detection
- FamilySearch GEDCOM Specifications
- Behold
- Chronoplex My Family Tree

The 5.x series of drafts includes both the first standard definition of the Lineage-Linked GEDCOM Form and also the first major expansion of the Lineage-Linked Form since its initial use in GEDCOM 3.0. The GEDCOM 5.x compatible systems should be able to read previous GEDCOM versions. See "Compatibility with Previous GEDCOM Releases", (starting on page 67) for compatibility specifics.

reading previous versions

The statement that GEDCOM 5.x compatible systems should be able to read previous GEDCOM versions is not a command or instruction; it is a statement about compatibility that isn't entirely true. Few genealogy applications are able to read GEDCOM 4.0 or earlier, and most genealogy applications do not even recognise GEDCOM 1.0 as GEDCOM.

best practice

The GEDCOM 5.5.1 specification has been the *de facto* standard for so long (since 1999 CE) that there is no pressing need to support any older GEDCOM versions, not even GEDCOM 5.5. However, applications should take care to recognise GEDCOM 5.5.1 files that claim to be GEDCOM 5.5 files correctly, as GEDCOM 5.5.1 files, even if they still support GEDCOM 5.5. Additionally, every application should of course be able to read the GEDCOM files it writes, including all GEDCOM files produced by earlier versions of that application.

Applications should read older GEDCOM versions for compatibility's sake. Applications should, again for compatibility's sake, not write those older formats.
Applications should not try to read GEDCOM versions they do not support, but honestly report unsupported versions as unsupported.

Modifications in Version 5.5.1

! Editorial corrections.

! Added ~~continuation tags~~ CONC (concatenation) and CONT (continuation) tags to the **header** copyright tags (see <<HEADER>>, page 23.)

header CONC and CONT

The explicit inclusion of CONC and CONT record in the lineage-linked form is a mistake that promotes misinterpretation and confusion.

! Added **ADR3** to the **address** structure (see <<ADDRESS_STRUCTURE>>, page 31.)

ADR3

GEDCOM 5.5. introduced ADR1 and ADR2, but not ADR3, and it was not added through the GEDCOM errata sheet either. The ADR3 record was introduced in GEDCOM 5.5.1.

! Added **email, fax, and web page** addresses to the **address** structure (see <<ADDRESS_STRUCTURE>>, page 31.)

! Added a **status** tag to the **child to family link** (see <<CHILD_TO_FAMILY_LINK>>, page 31.)

! Added a **restriction** notice tag to the **family record** to allow a source database to indicate why data may not have been supplied in the ~~transmission~~ GEDCOM file. (see <<FAMILY_RECORD>>, page 24.) Also added a restriction notice tag to the <<EVENT_DETAIL>> structure page 32 to allow an event to be flagged so that it can be treated in special ways such as not to be printed on reports.

<RESTRICTION_NOTICE> is specific to Ancestral File

The brief description provided here mentions “source databases”, thus suggesting that the restriction notice is a general mechanism, for use by any application. However, the <RESTRICTION_NOTICE> definition leaves no doubt that it is actually specific to exports from Ancestral File. The first two sentences of the <RESTRICTION_NOTICE> definition are:

> The restriction notice is defined for Ancestral File usage. Ancestral File download GEDCOM files may contain this data.

Ancestral File is obsolete, so the RESN record type is obsolete too.

Product-specific tags do not belong in the GEDCOM specification, but in a product-specific document detailing its GEDCOM extensions. If the RESN record wasn't obsolete already, it would have been deprecated.

FAMILY_RECORD does not exist

This item in the list of modifications made in GEDCOM 5.5.1 mentions a <<FAMILY_RECORD>>, but the specification does not define a <<FAMILY_RECORD>>, it defines a <<FAM_RECORD>>.
This is an editing oversight; it was still called <<FAMILY_RECORD>> in GEDCOM 5.3.

! Added additional subordinate structure to the **personal name structure** (see <<PERSONAL_NAME_STRUCTURE>>, page 38.) These changes are in preparation for handling more varied cultures as we move into the Unicode character set environment.

! Added subordinate map coordinates and other additional changes to the **place structure** (see <<PLACE_STRUCTURE>>, page 38.) These changes are in preparation for handling more varied cultures as we move into the Unicode character set environment and to allow recording of map coordinates to places such as burial cites.

! Added a subordinate affiliated **religion** tag to the **event detail** substructure (see <<EVENT_DETAIL>>, page 32.)

! Added a generic **FACT** tag to the individual attribute structure. Previously, the generic EVEN tag was used. The FACT was added to give a semantic difference between generic events and generic facts or characteristics (see <<INDIVIDUAL_ATTRIBUTE_STRUCTURE>>, page 33.)

! **Changed** the <DATE_PHRASE> definition to no longer require enclosing parentheses *as part of* the date phrase.
The <DATE_VALUE> definition continues to require parentheses *around* date phrases.

! **Removed** the option for **encoding embedded multimedia** objects. A file reference to a multimedia file and its subordinate format and media types were added to the multimedia record. Multiple file references can now be used to group related multimedia objects. This changed the multimedia link by placing the FORM tag subordinate to the FILE tag rather than at the same level. The BLOB tag was eliminated. See FILE tag and its subordinate FORM tag used in the <<MULTIMEDIA_RECORD>> page 26 and the <<MULTIMEDIA_LINK>> page 37.

6

7

BLOB removed

The BLOB tag is only supported by GEDCOM 5.4 and 5.5; it was introduced in GEDCOM 5.4 and removed in GEDCOM 5.5.1.
GEDCOM 5.5.1 allows media to be referenced by file name. This reflects the design of most genealogy databases; typically, media files are kept outside the database, and referenced by file name.

The FamilySearch GEDCOM specification does not specify any bundling mechanism for transfer of a GEDCOM file together with the referenced files. It is not uncommon to use ZIP files and nowadays, many users rely on synchronisation technology to transfer media files between their desktop database and their on-line tree.

BLOB usage

There is no pressing need to support the BLOB record in GEDCOM readers, not even if you support GEDCOM 5.5 in addition to GEDCOM 5.5.1.
Very few applications ever used BLOB records in GEDCOM files, the ones I know about are GEDitCOM, Family Historian, and Ancestral Quest. There is just a very small number of GEDCOM 5.5 files that contain BLOB records.

references

GenealogyTools.com: Why All Genealogy Apps Should Support GEDCOM 5.5.1 (Updated 29 Apr 2016)

! The following tags were added:

EMAIL	electronic mailing address
FAX	FAX address
FACT	A fact or characteristic.
FONE	Phonetic variation of a text.
ROMN	Romanized variation of a text.
WWW	Web Home page address.
MAP	Pertaining to maps.

LATI value of a latitudinal coordinate pertaining to the place of an event

LONG value of a longitudinal coordinate pertaining to the place of an event.

! The following tag was removed:

BLOB

Chapter 1 Data Representation Grammar

Introduction

This chapter describes the core GEDCOM data representation language.
The generic data representation language defined in this chapter may be used to represent any form of structured information, not just genealogical data, using a sequential stream of characters.

GEDCOM files are text files

This chapter fails to state that GEDCOM files are text files. This seems obvious to anyone who has seen a few GEDCOM files, but it is not obvious from reading this chapter. There is a significant error in this chapter. It does *not* define a text file as intended...

The specification should point out that GEDCOM files are text files. The practical upshot of this fact is that you can examine and edit GEDCOM files in a text editor.
This is both a boon and a bane. The ability to edit GEDCOM files using a regular text editor can be very handy. However, it is very easy to mess up the encoding of a GEDCOM file while doing so, because regular text editors do not support ANSEL, and UTF-8 detection isn't perfect.

references

GEDCOM Lines

Concepts

A GEDCOM ~~transmission~~ file represents a database in the form of a sequential stream of related records. A record is represented as a sequence of tagged, variable-length lines, arranged in a hierarchy. A line always contains a hierarchical level number, a tag, and an optional value. A line may also contain a cross-reference identifier or a pointer. The GEDCOM line is terminated by a carriage return, a line feed character, or any combination of these.

The tag in the GEDCOM line, taken in its ~~hierarchial~~ hierarchical context, identifies the information contained in the line, in the same sense that a field-name identifies a field in a database record. This means that the data is self-defining. Tags allow a field to occur any number of times within a record, including zero times. They also allow the use of different or new fields to be included in the GEDCOM data without introducing incompatibility, because the receiving system will ignore data which it does not understand and process only the data that it does understand.

The hierarchical relationships are indicated by a level number. Subordinate lines have a higher level number. The hierarchy allows a line to have sub-lines, which in turn may have their own sub-lines, and so forth. A line and its sub-lines constitute a context or enclosure, that is, a cluster of information pertaining directly to the same thing. This hierarchical arrangement corresponds with the natural hierarchy found in most structured information.

A series of one or more lines constitutes a record. The beginning of a new record is indicated by a *line whose level number is 0 (zero).*

In addition to hierarchical relationships, GEDCOM defines the inter-record relationships that allow a record to be logically related to other records, without introducing redundancy. These relationships are represented by two additional, but optional, parts of a line: a cross-reference pointer and a cross-reference identifier. The cross-reference pointer "points at" a related record, which is identified by a required, matching unique cross-reference identifier. The cross-reference identifier is analogous to a primary key in relational database terminology.

Grammar

This chapter defines the grammar for the GEDCOM format. The grammar is a set of rules that specify the character sequences that are valid for creating the GEDCOM line. The character sequences are described in terms of various combinations of elements (variables and/or constants). Elements may be described in terms of a set of other elements, some of which are selected from a set of alternative elements. Each element in the definition is separated by a plus sign (+) signifying that both elements are required. When there is a choice of different elements that can be used, the set of alternatives are listed between opening and closing square brackets ([]), with each choice separated by a vertical bar([alternative_1 | alternative_2]). When there is only one alternative shown then the choice is optional, that is, it is the same as [alternative_1 | <NULL>]. The user can read the grammar components of the selected element by substituting any sub-elements until all sub-elements have been resolved.

A GEDCOM ~~transmission~~ file consists of a sequence of logical records, each of which consists of a sequence of **gedcom_lines**, all contained in a sequential file or stream of characters. The following rules pertain to the **gedcom_line**:

Grammar Rules

! Long values can be broken into shorter GEDCOM lines by using a subordinate CONC or CONT tag. The CONC tag assumes that the accompanying subordinate value is concatenated to the previous line value ~~without saving the carriage return prior to the line terminator~~ without the line terminator. If a concatenated line is broken at a space, then the space must be carried over to the next line. The CONT tag assumes that the subordinate line value is concatenated to the previous line, after inserting a ~~carriage return~~ line terminator.

> ### Carriage Return confusion
>
> The phrase “the carriage return prior to the line terminator” is seriously confused. In FamilySearch GEDCOM, a line terminator consists of a carriage return, a line feed, or a combination of both. A carriage return outside a line terminator is a fatal error.
> The instruction to concatenate to the previous line “after inserting a carriage return” is almost as wrong. A GEDCOM writer should insert the line terminator appropriate for the receiving system (`HEAD.DEST`), and that line terminator need not be a carriage return. A GEDCOM reader should insert the line terminator appropriate for the platform it is running on; that line terminator need not be a carriage return, and it need not to be equal to the line terminator used by the GEDCOM file either.

! The beginning of a new logical record is designated by a line whose **level** number is 0 (zero).

! Level numbers must be between 0 to 99 and must not contain leading zeroes, for example, level one must be 1, not 01.

! Each new level number must be no higher than the previous line plus 1.

! All GEDCOM lines have either a **value** or a **pointer** unless the line contains subordinate GEDCOM lines. The presence of a level number and a tag alone should not be used to assert data (i.e. **1 FLAG Y** not just 1 FLAG to imply that the flag is set).

Records must have value

The sentence “All GEDCOM lines have either a **value** or a **pointer** unless the line contains subordinate GEDCOM lines.” is easily misinterpreted as saying that each GEDCOM record should have either a line value or subrecords, and may not have both or neither. However, a record may have both a line value and subrecords. All this sentence says is that *if there are no subrecords, there has to be a line value.*
The simple notion underlying this rule is that a record may not be empty, but must provide some data, either through its line value or through subrecords.

The two GEDCOM records that are an exception to this rule are the CONT and TRLR records.
You need empty CONT records to encode a sequence of empty lines, and the mandatory TRLR may not have a line value.
The CONC record *isn't* an exception to this rule. Each CONC record must have a line value.

! Logical GEDCOM record sizes should be constrained so that they will fit in a memory buffer of less than 32K. GEDCOM files with records sizes greater than 32K run the risk of not being able to be loaded in some programs. Use of pointers to records, particularly NOTE records, should ensure that this limit will be sufficient.

maximum logical GEDCOM records size

The notion of maximum record size of (32K-1)B made sense back in the early 1980s, when a home computer with 64 KB of RAM was uncommon.
With today's typical desktop computers containing gigabytes of RAM, there is no need to for such a low maximum size any more. The notion of a maximum logical record size is essentially obsolete now.

The existence of maximum logical record size hardly matters, as `NOTE` records are the only record type remotely likely to contain larger values. The size of notes should not be unnecessarily constrained, but it is not really necessary to allow very large notes either; notes should not be very long. Users entering more than (32K-1)B of text in a note field are almost certainly doing something wrong, and should probably attach a document instead.

The notion of a maximum logical record size is seriously dated. This requirement is obsolete.

The stated maximum size of “less than 32K” is tied to the use of 16-bit signed integers. The largest value a 16-bit signed integer can handle is 32K-1.
Applications should use 32-bit integers for size. The largest value a signed 32-bit integer can handle is 2G-1.

references

GEDCOM Lines

! Any length constraints are given in characters, not bytes. When wide characters (characters

10

11

wider than 8 bits) are used, byte buffer lengths should be adjusted accordingly.

code units, not characters

FamilySearch is trying to do the right thing here, by recognising that so-called wide characters take up two bytes instead of one, but still gets it wrong.
The length of line values is not measured in characters, but in code units.

The *characters, code units, and code points* annotation provides a brief explanation.

! The cross-reference ID has a maximum of 22 characters, including the enclosing ‘at ’ signs (@), and it must be unique within the GEDCOM ~~transmission~~ file.

GEDCOM Identifier Length

Here, the GEDCOM grammar states that GEDCOM identifiers are limited to 22 characters (code units), including the enclosing at signs. Further on, the lineage-linked form states that identifiers are at most 22 characters, but that's 22 *without* the enclosing at signs. Thus, the GEDCOM grammar and the lineage-linked form contradict each other. The GEDCOM grammar takes precedence over any GEDCOM form.

- The maximum length of a bare record identifier is 20 characters.
- The maximum length of an enclosed record identifier is 22 characters.

references

↗ GEDCOM Identifiers: Length

! Pointers to records ~~imply that the record pointed to does actually exist~~ must be to records that exist within the ~~transmission~~ GEDCOM file. Future pointer structures may allow pointing to records within a public accessible database as an alternative.

Referential Integrity

This particular statement demands *referential integrity*; pointers must be valid, i.e. must point to a record within the GEDCOM file.
The existence of a pointer does not imply the existence of a corresponding record, it demands the existence of corresponding record.

The FamilySearch GEDCOM specification does not explicitly state whether GEDCOM readers should check referential integrity, but it is implied. A GEDCOM reader has to make sense of all pointers by matching them to corresponding records, and then translating that reference into its own database format. The specification does not state what to do when a pointer is invalid (no matching record), but it is not hard to understand that the most reasonable action is to reject the GEDCOM file as corrupt; an invalid pointer implies a corrupt database or erroneous GEDCOM export.

orphaned records

Note that the specification demands that each pointer is valid, but *does not demand* that each record is pointed to. It is not illegal for a GEDCOM file to have so-called orphaned records, and they do occur in practice. A GEDCOM reader must import all records, including the orphaned ones. Orphaned records do not constitute a GEDCOM error of any kind, but it is a good idea for a GEDCOM reader to warn the user about orphaned records.

! The length of the GEDCOM TAG is a maximum of 31 characters, with the first 15 characters being unique.

! The total length of a GEDCOM line, including level number, cross-reference number, tag, value, delimiters, and terminator, must not exceed 255 (wide) characters.

Maximum Line Length

The FamilySearch GEDCOM 5.5.1 specification is crystal clear that a GEDCOM line may not be longer than 255 characters (code units) including the line terminator. This information is here in this chapter about the GEDCOM grammar and is repeated in the *Primitive Elements of the Lineage-Linked Form* section of chapter 2, *Lineage-Linked ~~Grammar~~ Form*.
Yet there are still are genealogy applications that produce GEDCOM lines longer than 255 code units.

GEDCOM writer best practice

- Respect the maximum length, for compatibility with GEDCOM readers that use 256-element buffers
- Use CONC and CONT to record large values

GEDCOM reader best practice

- whether you accept longer lines or not, issue an error for each oversized line

! ~~Leading white space (tabs, spaces, and extra line terminators) preceding a GEDCOM line should be ignored by the reading system.~~ Empty lines and leading white space are illegal. Systems generating GEDCOM should not ~~place any white space in front of the GEDCOM line~~ create empty lines or start any lines with leading white space.

No Leading White Space

The GEDCOM grammar (below) leaves no doubt that leading white is illegal; a line starting with leading white space isn't a GEDCOM line. Thus, the statement that GEDCOM readers should ignore (read: accept as legal!) leading white space contradicts the grammar directly. The suggestion that empty lines are acceptable is even worse.

The mere notion that it is okay to skip leading white space in GEDCOM files is dangerously inconsistent. Developers expect and tend to assume consistent treatment of white space, and skipping leading white space is what a GEDCOM reader should most definitely *not do* when dealing with CONC & CONT line values.

GEDCOM files do not contain empty lines or lines starting with white space. That is one of the characteristics of GEDCOM files by which they can be recognised. GEDCOM readers can and should reject any file containing any empty lines or lines starting with white space as not a GEDCOM file.

Grammar Syntax

A **gedcom_line** has the following syntax:

gedcom_line:=

level + delim + [optional_xref_ID] + tag + [optional_line_value] + terminator

for example:
1 NAME Will /Rogers/

The components used in the pattern above are defined below in alphabetical order. Some of the components are defined in terms of other primitive patterns. The spaces used in the patterns below are only to set them apart and are not a part of the resulting pattern. Character constants are specified in the hex form (0x20) which is the ASCII hex value of a space character. Character constants that are separated by a (-) dash represent any character with in that range from the first constant shown to and including the second constant shown.

alpha:=

[(0x41)-(0x5A) | (0x61)-(0x7A) | (0x5F)]
where:
(0x41)-(0x5A)=A to Z
(0x61)-(0x7A)=a to z

(0x5F)=(_) underscore

A through Z

This definition contains two obviously erroneous statements:

- The range 0x41 through 0x5A does not specify A *to* Z, but A *through* Z; Z is included.
- The range 0x61 through 0x7A does not specify a *to* z, but A *through* z; z is included.

alphanum:=

[alpha | digit]

any_char:=

[alpha | digit | otherchar | (0x23) | (0x20) | (0x40)+(0x40)]
where:
(0x23)=#
(0x20)=space character
(0x40)+(0x40)=@@

not really any character

Notice that this definition of **any_char** does *not* allow every character; **regular_char** would have been a better name.

double at-sign

Note the last item in the specification of **any_char**: `| (0x40)+(0x40).`
What it says there is that regular text is not allowed to contain a single at sign (@), but must always include two consecutive at signs (@@) instead.
The definition states this, but does not explain this.

A GEDCOM writer must export a double at-sign instead of single at-sign, and a GEDCOM reader must import the double at-sign as a single one. This is because GEDCOM uses the at sign to start and end an **escape**.

email

This rule affects all line values, but particularly the EMAIL line value, <ADDRESS_EMAIL>, introduced with GEDCOM 5.5.1, as email addresses always contain a single at-sign.

dates

GEDCOM writers should pay special attention to at-signs in date fields. When an application does not provide support for a particular calendar, a user may choose to enter a date complete with the <DATE_CALENDAR_ESCAPE> for the calendar, e.g. “@#JULIAN@ 12 Oct 1492” (without the quotes). The at-signs within such dates should *not* be doubled.

best practice

Many vendors fail to notice this rule, and their products fail to double the at-sign on export. So in practice, GEDCOM readers must recognise both the correct line value `user@@email.com` and the illegal line value `user@email.com` as `user@email.com`. A GEDCOM reader that accept an illegal value should still issue an error for that illegal line value.

not deprecated

This feature complicates GEDCOM handling, so it is desirable to mark this GEDCOM feature as deprecated, but as long as some records can contain GEDCOM pointers as well as free text, this feature allows GEDCOM readers to distinguish between an at-sign that starts a pointer, and one that does not.

delim:=

[(0x20)]

where:

(0x20)=space_character

digit:=

[(0x30)-(0x39)]
where:
(0x30)-(0x39) = One of the digits 0, 1,2,3,4,5,6,7,8,9

escape:=

[(0x40) + (0x23) + escape_text + (0x40) ~~+ non_at~~]
where:
(0x40)=@
(0x23)=#

non_at is non-part

An escape sequence begins and ends with an at-sign.
An escape sequence must be followed by a non_at (or rather a delim), but that character *is not a part of the escape sequence.*

escape_text:=

[any_char | escape_text + any_char]
The escape_text is coded to meet the rules of a particular GEDCOM form.

level:=

[digit | digit + digit]
(Do not use non-significant leading zeroes such as 02.)

line_item:=

~~[any_char | escape | line_item + any_char | line_item + escape]~~
[| escape | line_text | escape + delim + line_text]

line_item definition errors

The way FamilySearch defines line_item (recursively):

- line_item may consist of nothing but a single escape sequence.
- line_item may consist of a some text followed by an escape sequence.
- line_item may consist of a some text followed by multiple escape sequence.
- line_item may consist of a some text both preceded and followed by an escape sequence.
- line_item may consist of nothing but multiple escape sequences.

Most of that is not as intended. Most of these possibilities are wrong.

escape in line_item

escape only

FamilySearch's definition of line_item includes

> escape

This is explicit inclusion of nothing but an escape sequence as possible line value is intended. It was used in GEDOM 5.0 to switch character sets in the middle of a GEDCOM file, with escape sequences such as `@#ASCII@`.
The grammar needs some correction, but the possibility of an escape-sequence only line value was a deliberate decision, not some error. Although the GEDCOM 5.5.1 lineage-linked form doesn't use it, the replacement definition retains this possibility, merely marked as deprecated.

escape + text

FamilySearch's definition of line_item includes

> line_item + escape

However, escape sequences always appear at the start of a line value, not the end, so the only correct order is the other way round:

> escape + line_item

That still isn't the correct syntax. There must be a delimiter (space) between the escape sequence and the value that follows it. The FamilySearch GEDCOM 5.5.1 grammar includes that space *as part of* the escape sequence. That is a conceptual error. An escape sequence starts and ends with an at-sign. The escape definition has been corrected, and the mandatory delimiter has added to the line_item definition:

> escape + delim + line_item

It has been corrected further to allow just one escape sequence per line value, as intended:

> escape + delim + line_text

line_text:=

[any_char | line_text + any_char]

line_text is line value text that neither is nor contains a pointer or an escape sequence.

line_text

line_text was introduced to correct mulitple errors in the line_item definition.

line_value:=

[pointer | line_item]

non_at:=

[alpha | digit | otherchar | (0x23) | (0x20)]
where:
(0x20)=space character
(0x23)=#

null:= nothing

optional_line_value:= delim + line_value

optional_xref_ID:= xref_ID + delim

otherchar:=

[(0x21)-(0x22) | (0x24)-(0x2F) | (0x3A)-(0x3F) | (0x5B)-(0x5E) | (0x60) | (0x7B)-(0x7E) | (0x80)-(0xFE)]
where, respectively:
(0x21)-(0x22)=! "
(0x24)-(0x2F)=$ % & ' ( ) * + , - . /
(0x3A)-(0x3F)=: ; < = > ?
(0x5B)-(0x5E)=[\ ] ^
(0x60)=`
(0x7B)-(0x7E)={ | } ~
(0x80)-(0xFE)=~~ANSEL~~ characters above 127
Any ~~8-bit ASCII~~ character except control characters (0x00–0x1F), alphanum, space ( ), number sign (#), at sign (@), _ underscore, and the DEL character (0x7F).

any_char and Unicode

This definition completely fails to account for the addition of Unicode support, despite the fact that Unicode was added in GEDCOM 5.5 already, four years earlier.
This definition, when take literally - as you are supposed to do with definitions like this - excludes all characters not supported by ANSEL! - and that surely isn't the intention. It makes no sense to add Unicode support and then forbid applications to take advantage of Unicode. The result is that is not really clear which characters are and aren't allowed.

This definitions should not be read as a statement of which characters are allowed, but as a statement of which character are disallowed; the last line of this definition is the most important one.

best practice

Applications are allowed to take advantage of Unicode's rich character set. When doing so, they should follow Unicode rules.

Horizontal Tab

This syntax does not allow the use of a Horizontal Tab anywhere. Thus, this specification demands that genealogy applications disallow the use of tabs in notes...
This is an unintentional error. The **otherchar** definition excludes the *C0 Control Set* (0x00-0x1F), and most of these control characters should indeed be illegal. Three characters within the *C0 Control Set* are commonly found in GEDCOM; the Carriage Return, the Line Feed, and the Horizontal Tab. The Carriage Return and Line Feed are explicitly included in the syntax by the **terminator** definition.
FamilySearch simply forgot to include the Horizontal Tab as a legal value. However, most vendors never even noticed that tabs are illegal, and all genealogy applications allow it.

Developers from different background have different ideas about default tab settings. The indented text in the GEDCOM 5.5.1 specification uses a tab setting of 4 spaces.

best practice

For compatibility with existing practice, the Horizontal Tab must be treated as legal within NOTE records (and their CONC & CONT subrecords).

GEDCOM does not provide a mechanism for specifying tab settings; when displaying text, applications should assume tabs are set at intervals of four spaces.

Users may use tabs in notes, but it should also be easy for users to link documents that allow rich format, such as RTF, PDF or HTML files.

pointer:=

[(0x40) + alphanum + pointer_string + (0x40)]
where:
(0x40)=@

pointers may include spaces

Notice that this definition of pointers allows a lot more than just letters and digits. It even allows spaces inside pointers!

best practice

- GEDCOM writers should not use spaces in identifiers
- GEDCOM readers should be careful to handle spaces in identifiers

pointer_char:=

[non_at]

pointer_string:=

[null | pointer_char | pointer_string + pointer_char]

tag:=

[[(0x5F)] + alphanum | tag + alphanum]
where:
(0x5F)=_

tags may contain underscores

The GEDCOM specification demands that non-standard (user-defined) tags start with an underscore, but according to FamilySearch's grammar, tags aren't allowed to start with an underscore!
The definition has been amended to allow a tag to start with an single underscore, without allowing a tag consisting of just one underscore.

GEDCOM tags are case-sensitive

The tag definitions uses alphanum, which includes both uppercase and lowercase letters. The FamilySearch GEDCOM specification fails to explicitly specify this, but GEDCOM tags are case-sensitive.

All lineage-linked tags are UPPERCASE.

tags may start with digits

According to this definitions, tags may start with digits. Tags may even be all-digits. That the GEDCOM grammar does not demand that tags start with a letter is remarkable, and probably an oversight. It is a demand commonly placed on keywords and identifiers of formal languages, to ease parsing and error-detection.

None of the tags defined in the GEDCOM grammar or the lineage-linked form start with a digit, and user-defined tags must start with an underscore.

13

14

terminator:=

[carriage_return | line_feed | carriage_return + line_feed | line_feed + carriage_return]

GEDCOM line terminator specification errors

There are two issues with this line terminator definition.

An experienced programmer will immediately notice that the definition tries to account for the fact that different operating systems use different line terminators. Classic Mac System (the predecessor of MacOS) uses Carriage Return (CR), Unix uses Line Feed (LF), and Windows uses Carriage Return followed by Line Feed (CR/LF), and this definition allows them all. That this definition additionally allows Line Feed followed by Carriage Return (LF/CR) is a serious mistake, a problem instead of a solution. Text editors and GEDCOM readers expect CR, LF and CR/LF, and are likely to get seriously confused when presented with LF/CR.

The LF/CR line terminator remains a mostly theoretical case, that only needs to be supported by GEDCOM validators. There is no practical need to support import of GEDCOM files using LF/CR, as no genealogy applications uses LF/CR as a GEDCOM line terminator. A rare special case is that Famberry, a genealogy hosting site that has used LF/CR/LF as its GEDCOM line terminator.

The second and arguably more serious issue is that the syntax as presented here allows each line to have a line terminator, independent of the line terminator used for other lines. According to this syntax, it is legal for one line to be terminated by CR, for another line to be terminated with LF, and for yet another line to be terminated with CR/LF. That too, is a specification error.

This chapter, and indeed the entire specification, fails to state it, but a GEDCOM file is supposed to be a text file, which you can load into an text editor. Like any other text file, a GEDCOM file must use a single line terminator throughout the file, and the occurrence of either CR or LF outside the line terminator is a fatal error: not a GEDCOM file, not even a text file.

GEDCOM Writer best practice

- Use a single line terminator consistently
- Never use LF/CR.
- When not sure which line terminator to use, use CR/LF.

references

GEDCOM Lines
Famberry GEDCOM

xref_ID:=

[pointer]

Description of Grammar Components

alpha:=

The alpha characters include the underscore, which is used to link word pieces together in forming tag names or tag labels.

any_char:=

Any ~~8-bit ASCII~~ character except the control characters found in the range of 0x00–0x1F and 0x7F.

delim:=

The **delim** (delimiter), a single space character, terminates both the variable-length **level** number and the variable-length **tag**. Note that space characters may also be present in a **value**.

escape:=

The **escape** is a character sequence in the grammar used to specify special processing, such as for switching character sets or for indicating an inclusion of a non-GEDCOM data form into the GEDCOM structure. The form of the escape sequence is:

@+#+**escape_text**+@+ ~~**non_at**~~ **delim**~~.~~

Receiving systems should discard any space character which follows the escape sequence ’s closing at-sign (@). If the character following the escape sequence's closing at-sign (@) is not a space character then it should be kept as a part of the text following the escape. Systems writing escape sequences should always output a space character following the escape sequence.

The specific format of the escape sequence is defined for the specific GEDCOM form being defined.

escape full stop

The FamilySearch GEDCOM 5.5.1 specification adds a full stop after the escape syntax shown here. This creates possible confusion about whether or not the full stop is part of the escape sequence. The full stop has been removed to avoid possible confusion.

escape sequence must be followed by space

Notice that the syntax demands that the closing at-sign not be followed by another at-sign, but otherwise allows any character to follow it:

```
@+#+escape_text+@+non_at
```

mandatory isn't optional

However, the description clearly states that a GEDCOM writer should always follow the closing at-sign with a space.
The description specifically states that it should be a space character, not a tab, nor a newline character or any other white space. so the corrected syntax uses **delim** (a space) instead of **non_at**:

```
@+#+escape_text+@+delim
```

The description specifies to treat the absence of the mandatory space, or the presence of multiple spaces (and presumably tabs as well) the same as a the presence of the mandatory delimiter. That is a specification mistake, as it effectively turns the mandatory delimiter into an optional delimiter.

delimiter isn't part of escape sequence

The character following the escape sequence (**delim** instead of **non_at**) isn't part of the escape sequence, so it should not be defined as part of the escape sequence. The mandatory **delim** has been moved from this **escape** definition to the **line_item** definition, where it belongs, like this:

```
escape + delim + line_item
```

usage

The FamilySearch Lineage-Linked Form uses escape sequences to specify calendars.
It is not used anywhere else.

best practice

Vendors should not use escape sequences in their GEDCOM extensions.

A GEDCOM validator *must* issue and a GEDCOM reader should issue a warning whenever the closing at-sign of an escape sequence is not followed by a space character.

escape_text:=

The escape_text is defined to meet the requirements of a particular GEDCOM form.

level:=

The **level** number works the same way as the level of indentation in an indented outline, where indented lines provide detail about the item under which they are indented. A line at any level L is

enclosed by and pertains directly to the *nearest preceding line* at level L-1. The Level L may increase by 1 at most. Level numbers must not contain leading zeroes, for example level one must be (1), not

(01).

The enclosed subordinate lines at level L are said to be in the context of the enclosing superior line at level L-1. The interpretation of a **tag** must be in the context of the **tag**s of the enclosing line(s) rather than just the tag by itself.
Take the following record about an individual's birth and death dates, for example:

```
0 INDI
—1 BIRT
——2 DATE 12 MAY 1920
—1 DEAT
——2 DATE 1960
```

In this example, the expression DATE 12 MAY 1920 is interpreted within the INDI (individual) BIRT (birth) context, representing the individual's birth date. The second DATE is in the INDI.DEAT (individual's death) context. The complete meaning of DATE depends on the context.

> *Note:* The above example is indented according to the level numbers to make the concept more obvious. *In the actual GEDCOM data, the level numbers are lined up vertically, meaning they are the first character(s) of the GEDCOM line.*

Some systems output indented GEDCOM data for better readability by putting space or tab characters between the terminator and the **level** number of the next line to visibly show the hierarchy. Also, some people have suggested allowing extra blank lines to visibly separate physical records. GEDCOM files produced with these features are not to be used when transmitting GEDCOM to other systems.

line_value:=

The **line_value** identifies an object within the domain of possible values allowed in the context of the **tag**. The combination of the tag, the **line_value**, and the hierarchical context of the supporting **gedcom_lines** provides the understanding of the enclosed **values**. This domain is defined by a specific grammar for representing a given GEDCOM form. (See Chapter 2, starting on page 19 for Lineage-Linked GEDCOM Form grammar.)

Values whose source information contains illegible parts of the value should be indicated by replacing the illegible part with an ellipsis (...).

Values are generally not encoded in binary or other abbreviation schemes for reducing space requirements, and they are generally constrained to be understandable by a typical user without decoding. This is intended to reduce the decoding burden on the receiving software. A

GEDCOM-optimized data compression standard will be defined in the future to reduce space requirements. Meanwhile, users may agree to compress and decompress GEDCOM files using any compression system available to both sender and receiver.

line value ellipsis

The instructions to use ellipsis for illegible parts is problematic, because original text may contain ellipsis already.

FamilySearch probably meant this instruction for names in particular; use ellipsis for illegible parts of a name.

GEDCOM compression

GEDCOM version 5.5 supports encoding of media objects inside a GEDCOM file, in addition to linking to a media file.
GEDCOM 5.5.1 does not allow encoding of media objects inside a GEDCOM file.

FamilySearch's promise of a future “GEDCOM-optimized data compression standard” was never realised.
A GEDCOM-optimised data compression standard is not needed; GEDCOM files are text files with a fair amount of repetition, so they compress well using common file compression methods.
It is not unusual to zip GEDCOM files before mailing or uploading them.

The **line_value** within the context of a tag hierarchy of **gedcom_line**s represents one piece of information and corresponds to one field in traditional database or file terminology.

otherchar:=

Any ~~8-bit ASCII~~ character except control characters (0x00–0x1F), **alphanum**, space ( ), number-sign (#), the at sign (@), and the DEL character (0x7F).

pointer:=

A **pointer** stands in the place of the record or context identified by the matching **xref_ID**. Theoretically, a receiving system should be prepared to follow a **pointer** to find **any** needed **value** in a manner that is transparent to the logic of the subsystem that is looking for specific **tag**s. This highly flexible facility will probably be used more in the future. For the time being, however, the use of **pointer**s is explicitly defined within the GEDCOM form, such as the Lineage-Linked GEDCOM Form defined in Chapter 2 (see page 19).

The **pointer** represents the association between two objects that usually reside in different records. Objects within a logical record can be associated. If this need exists, the pointer record composition contains an exclamation point (!) that separates the parent record's cross-reference ID from the specific substructure's cross-reference ID, which is at some subordinate level to the logical record at level zero. The cross-reference ID of the substructure subordinate to a zero level record, for inter-record associations is always composed of the Record ID number and the Substructure ID number, such as @I132!1@. Including the Record ID number in the pointer that associates objects within a record will allow the GEDCOM processors to build the index only at the record level and then search sequentially for the appropriate substructure cross-reference ID. The parent record ID is assumed when the cross-reference ID begins with a exclamation point (!) signifying an intra-record association.

cross-reference exclamation mark syntax

Typical GEDCOM cross-references are pointers from top-level records to other top-level records. Here, the FamilySearch GEDCOM specification defines an exclamation mark syntax for pointers to subrecords.

This is a completely theoretical feature that does not occur in GEDCOM practice; pointers can only be created to records or subrecords that have identifiers, and the syntax of *the lineage-linked form does not allow identifiers on or references to anything but top-level records*.
There is no need to support the exclamation mark syntax, on the contrary; there are no real GEDCOM files containing this syntax.

The cross-reference exclamation mark syntax should be considered an obsolete theoretical feature.
In practice, any cross-reference containing an exclamation mark should be reported as an error.

Complex logical record structures are divided into small physical records to accommodate memory constraints, many-to-many relationships, and independent record creation and deletion.

The **pointer** must match a corresponding unique **xref_ID** within the ~~transmission~~ GEDCOM file, unless the colon (:) character is present (which will be used in the future as a network reference to a permanent file record). ~~A **pointer** is given instead of duplicating an object, though the logical result is equivalent.~~ An expanded traversal of a record tree includes following the **pointer** to related records to some depth, and splicing those records (logically) into the resultant expanded tree. *Pointers may refer to either records which have not yet appeared in the ~~transmission~~ GEDCOM file (forward reference) or to records that have already appeared earlier in the ~~transmission~~ GEDCOM file (backward reference).* This arrangement usually requires a preliminary pass to construct a look up table to support random access by **xref_ID** during subsequent passes.

pointers versus record duplication

FamilySearch's statement that having pointers and duplicating records are logically equivalent is incorrect. These aren't two different ways to express the same thing, these are two conceptually different things. There are real, relevant differences between multiple pointers to the same object, and multiple identical records, *especially in the case of circular references*.

cross-reference colon syntax

The FamilySearch GEDCOM specification reserved the colon syntax for a future feature that never came to be, in which a cross-reference consists of a <REGISTERED_RESOURCE_IDENTIFIER> and <RECORD_IDENTIFIER> within that resource, separated by a colon.

There is no need to support the colon syntax, on the contrary:

- There are no registered resource identifiers, so a GEDCOM writer can not create a GEDCOM file using this syntax.
- There are no registered resource identifiers, so a GEDCOM reader cannot make sense of a GEDCOM using this syntax.
- There are no real GEDCOM files containing this syntax.

The cross-reference colon syntax should be considered an *obsolete future feature*.
In practice, any cross-reference containing a colon should be reported as an error.

16

17

tag:=

A **tag** consists of a variable length sequence of **alphanum** characters. All user-defined tags that have not been defined in the GEDCOM standard, must begin with an underscore character (0x95).

tag represents the meaning of the **line_value** within the context of the enclosing tags, and contributes to the meaning of the enclosed subordinate lines. Specific **tags** are defined in Appendix A (starting on page 83). The presence of a tag together with a value represents an assertion which the submitter wishes to communicate to a receiver. A tag without a value does not represent an assertion. ~~If a tag is absent, no assertion is made.~~ If an optional record (the entire line; level, *tag*, line value and terminator) is absent, no assertion is made. Information of a negative nature (such as knowing positively an event did not occur) is handled through the semantic definition of a different tag and its accompanying value that assert the information explicitly.

Absent tag

It is illegal for a GEDCOM line to not contain a tag. FamilySearch sometimes uses “tag” to mean “record”.
The statement about absent tags has been corrected to be statement about optional records.

Although ~~formally defined tagss~~ all the tags of the lineage-linked form are only three or four or five characters long, systems should prepare to handle user tags of greater length. Tags will be unique within the first 15 characters.

only three or four characters long

Not all tags defined in the FamilySearch GEDCOM specification are only three or four character long. This very specification introduces the five-character EMAIL tag.

More importantly, it is odd and arguably inappropriate for the GEDCOM grammar to make such a sweeping claim about all tags, in all GEDCOM forms. The text has been edited to restrict the claim to the lineage-linked form only.

Valid combinations of specific **tags**, **line_values**, **xref_IDs**, and **pointers** are constrained by the GEDCOM form defined for representing a given kind of information. (See Chapter 2, starting on page 19, for the Lineage-Linked GEDCOM Form grammar.)

box around “Appendix A (starting on page 83)”

There is a box around the text “Appendix A (starting on page 83)” above. It is not clear why.

terminator:=

The **terminator** delimits the variable-length **line_value** and signals the end of the **gedcom_line**. The valid terminator characters are:
[carriage_return |
line_feed |
carriage_return + line_feed |
line_feed + carriage_return]

GEDCOM writers must pick one line terminator, and use that line terminator throughout the entire GEDCOM file.

terminator LF/CR

The LF/CR line terminator should never have been included as an option.
See GEDCOM line terminator specification errors.

xref_ID:=

(See **pointer**, page 16)
The **xref_ID** is formed by any arbitrary combination of characters from the **pointer_char** set. The first character must be an alpha or a digit. The **xref_ID** ~~is not~~ need not be retained in the receiving system, and it may therefore be formed from any convenient combination of identifiers from the sending system. No meaning is attributed by the receiver to any part of the **xref_ID**, other than its **unique** association with the associated record. The use of the colon (:) character is also reserved.

Examples:
The following are examples of valid but unrelated GEDCOM lines:

0 @1234@ INDI

```
. . .
1 AGE 13y
```

17

18

```
. . .
1 CHIL @1234@
. . .
1 NOTE This is a note field that is
2 CONT continued on the next line.
```

The first line has a **level** number 0, a **xref_ID** of @1234@, an INDI **tag**, and no **value**.
The second line has a **level** number 1, no **xref_ID**, an AGE **tag**, and a **value** of 13.
The third line has a **level** number 1, no **xref_ID**, a CHIL **tag**, and a **value** of a **pointer** to a **xref_ID** named @1234@.

“not retained in the receiving system”

FamilySearch's statement that “The **xref_ID** is not retained in the receiving system” is well-intentioned, but not entirely true. The numbers used within identifiers for INDI and FAM records often *are* retained, on purpose. This allows searching for individuals and family groups using the same numbers as used in the source system.

cross-reference reserved characters

The xref_ID description states that use of the colon is reserved. In actual fact, the GEDCOM grammar reserves both the colon and exclamation mark.
The colon is reserved for the cross-reference colon syntax and the exclamation mark is reserved for the cross-reference exclamation mark syntax.
Both are obsolete non-features, so actual GEDCOM cross-references should not contain any colons or exclamation marks.

box around “See pointer, page 16”

There is a box around the text “See pointer, page 16” above. It is not clear why.

18

CONC & CONT

Chapter 1 Data Representation Grammar defines two records types, CONC & CONT. That CONC & CONT are part of the GEDCOM grammar is significant; it means they can be used with any line value in any GEDCOM form.

Chapter 1 Data Representation Grammar defines these two record types, but does not define or describe them very well. In fact, most of the things you should know about CONC & CONT is found elsewhere within the specification.
This annotation brings all the sections about CONC & CONT together.

This is all chapter 1 itself says about CONC & CONT:

> ! Long values can be broken into shorter GEDCOM lines by using a subordinate CONC or CONT tag. The CONC tag assumes that the accompanying subordinate value is concatenated to the previous line value ~~without saving the carriage return prior to the line terminator~~ without the line terminator. If a concatenated line is broken at a space, then the space must be carried over to the next line. The CONT tag assumes that the subordinate line value is concatenated to the previous line, after inserting a ~~carriage return~~ line terminator.

The Primitive Elements of the Lineage-Linked Form section of Chapter 2 Lineage-Linked ~~Grammar~~ Form starts with the following paragraph:

> The field sizes show the minimum recommended field length within a database that is constrained to fixed length fields. The field sizes are in addition to the GEDCOM level and tag overhead. GEDCOM lines are limited to 255 characters. However, the ~~CONCatenation~~ CONC (concatenation) or ~~CONTinuation~~ CONT (continuation) tags can be used to expand a field beyond this limit. CONT line implies that a new line should appear to preserve formatting. CONC implies concatenation to the previous line without a new line. This is used so that a text note or description can be processed (word wrapped) in a text window without fixed carriage returns. The CONT and CONC tags are being used to extend specified textual values.

The lineage-linked form chapter further explains usage of CONC & CONT in the <<NOTE_STRUCTURE>> definition:

> **NOTE_STRUCTURE**:=
>
> [
> n **NOTE** @<XREF:NOTE>@ {1:1} p.27
> |
> n **NOTE** [<SUBMITTER_TEXT> | <NULL>] {1:1} p.63
> +1 [CONC|CONT] <SUBMITTER_TEXT> {0:M}
>]
>
> Note: There are special considerations required when using the CONC tag. The usage is to provide a note string that can be concatenated together so that the display program can do its own word wrapping according to its display window size. The requirement for usage is to either break the text line in the middle of a word, or if at the end of a word, to add a space to the first of the next CONC line. Otherwise most operating systems will strip off the trailing space and the space is lost in the reconstitution of the note.

And finally, there are the CONC & CONT definitions in Appendix A:

> **CONC {CONCATENATION}**:=
> An indicator that additional data belongs to the superior value. The information from the CONC value is to be connected to the value of the superior preceding line without adding a space and without adding a ~~carriage return and/or~~ new line character. Values that are split for a CONC tag ~~must~~ should always be split at a nonspace. If the value is split on a space the space ~~will~~ may be lost when concatenation takes place. This is because of the treatment that spaces get as a GEDCOM delimiter, many GEDCOM values are trimmed of trailing spaces and some systems look for the first non-space starting after the tag to determine the beginning of the value.

> **CONT {CONTINUED}**:=
> An indicator that additional data belongs to the superior value. The information from the CONT value is to be connected to the value of the superior preceding line ~~with a carriage return and/or~~ after first appending a new line character. Leading spaces could be important to the formatting of the resultant text. When importing values from CONT lines the reader should assume only one delimiter character (the mandatory space) following the CONT tag. Assume that the rest of the leading spaces are to be a part of the value.

carriage return outside line terminator

There are issues with all the quoted fragments. The phrases “the carriage return prior to the line terminator” & “a carriage return and/or new line character” do not make sense.
A GEDCOM line terminator *consists of* a carriage return, a line feed or a combination of both. Carriage returns are not allowed outside line terminators. A carriage return anywhere else is a fatal error: not GEDCOM.
More detailed comments are provided on the cited definitions.

not just spaces, tabs too

Be aware that long text may not only include spaces, but tabs as well. It is a little-known fact that the GEDCOM grammar excludes tabs from appearing in GEDCOM files. That is a specification error which genealogy application vendors have ignored *en masse*, mostly because they did not even notice the restriction...
The reality is that products do not stop users from entering tabs, that many GEDCOM files contain tabs within text, and that users expect these tabs to transfer correctly.

GEDCOM writers should treat both spaces and tabs as white space, and avoid splitting on any white space, as GEDCOM readers are likely to treat both tabs and spaces differently from other characters.
Unicode has many more characters in the white space category, but there is no need to treat those as white space, as no GEDCOM reader takes any special action for those.

respect character boundaries

Text-splitting must respect character boundaries; text must be split between characters, not inside characters. The three most important encodings allowed by GEDCOM (ANSEL, UTF-8 & UTF-16) all support characters made up of multiple code points. Splitting between code points that make up a single character will generally result in lines that aren't ANSEL, UTF-8 or UTF-16 any more and that is a fatal error.
Use operating system or programming library functions to make sure you split on character boundaries.

CONC problem & solutions

Several genealogy applications that use CONC to record long values split text at white space, resulting in GEDCOM line with trailing white space, and line values with leading white space. As the GEDCOM specification notes, a GEDCOM reader may decide to trim the trailing white space, the leading white space, or both. These two facts combine to produce erroneous transfers, in which spaces and tabs between words are lost.

Vendors have tried to solve this problem by doing one or more of the following things on GEDCOM import:

- maintain an internal list of which genealogy application versions split at white space, and which ones don't, and process accordingly
- prompt the user for input: present the user with a dialog box, and ask which output looks right
- provide an import option to add a space to CONC lines

The first solution has the benefit of not bothering the user, but is not perfect. It is practically impossible to include all genealogy applications in the list, and, because the behaviour is version-dependent, the list must be updated for every new version of every application included in the list. The second approach bothers the user with a dialog box during import, which is wrong, and the question this dialog box poses is surprising at best, many users are understandably confused by the question. The third solution let's the user specify the desired behaviour before the import. This avoids having to pop-up a dialog box, and allows saving the setting for repeated imports, but expects the user to know the right setting for an advanced import option.

The article *GEDCOM CONC and CONT* presented a very simple solution that does not require maintaining any tables or asking the user for input. The detailed best practice this article presents can be summarised as *do not trim leading or trailing white space from CONC and CONT lines.*

trimming trailing white space

The CONC definition warns against trailing spaces; it states that long line must be split at a non-space and warns that GEDCOM readers may trim trailing spaces.
The statement made there strongly suggests that it is okay for GEDCOM readers to trim trailing white space, that this is just something that GEDCOM readers do, but that is a serious mistake. A GEDCOM reader *should not* trim trailing white space from NOTE, TEXT, CONC or CONT line values, because *GEDCOM writers cannot always avoid trailing white space*!

unavoidable

It is true that GEDCOM writers should split long lines smartly, i.e. split in the middle of words instead of between words, to avoid problems with trailing white space. However, it is also true that trailing white space cannot always be avoided. There are situations where trailing white space is unavoidable.
GEDCOM writers must use CONT to encode a line break, so if there is white space before that line break, the previous CONC or CONT record must end with that trailing white space. Similarly, if a line starts with white space, leading white space is unavoidable.

Both leading white space and trailing white space are unavoidable, and in the pathological case of really long stretches of white space, or the more likely case of lines that contain nothing but a single space, CONC records with a line value that is nothing but white space unavoidable.

preserve white space

White space in user text should be preserved, even if it is white space at the end or beginning of line.
A GEDCOM reader should trim leading white space from the line values of most lineage-linked record types, and may trim trailing white space from most. However, a GEDCOM reader should *not* trim leading or trailing white space from NOTE, TEXT, CONC or CONT line values.
Simply not trimming leading or trailing white space from CONC & CONT line values solves most problems users experience with the transfer of long text.

GEDCOM writer best practice

- When using CONC or CONT records, avoid splitting before or after tabs and spaces, split in the middle of a word
- if you must split at white space, prefer splitting before the white space over after the white space: this prefers leading white space on next line over trailing white space on this line
- Use operating system or programming library functions to make sure you split on character boundaries, not inside characters

GEDCOM reader best practice

- do not trim trailing white space from TEXT or NOTE line values
- do not trim leading white space from TEXT or NOTE line values
- do not trim trailing white space from CONC or CONT line values
- do not trim leading white space from CONC or CONT line values

references

- GEDCOM CONC and CONT
- Gaenovium Presentations: Louis Kessler: Reading Wrong GEDCOM Right
- Behold blog 2010-01-10: CONC Me on the Head

CONC and CONT in the lineage-linked form

Because the CONC & CONT records are part of the GEDCOM grammar, they need never be specified in a GEDCOM form. In fact, explicitly including them in a GEDCOM form is a serious mistake and sadly one that FamilySearch makes repeatedly. Every time they do so, they always include both CONC & CONT, even when only CONC makes sense, thus explicitly (but often erroneously) allowing line breaks, even for line values where allowing line breaks does not make sense.

SOURCE_RECORD:=

```
n @<XREF:SOUR>@ SOUR                              {1:1}
...
+1 AUTH <SOURCE_ORIGINATOR>                       {0:1}   p.62
+2 [CONC|CONT] <SOURCE_ORIGINATOR>                {0:M}   p.62
+1 TITL <SOURCE_DESCRIPTIVE_TITLE>                {0:1}   p.62
+2 [CONC|CONT] <SOURCE_DESCRIPTIVE_TITLE>         {0:M}   p.62
```

This extract of the **SOURCE_RECORD** definition shows the misguided explicit inclusion of CONC & CONT for both the SOUR.AUTH (source author) record and the SOUR.TITLE (source title) record. Neither line value should contain line breaks. FamilySearch makes the rather fundamental error of explicitly including CONC & CONT, while neither should ever be explicitly referenced in any GEDCOM form. On top of that, they make the mistake of always including both CONC & CONT, even for line values that should not contain line breaks. That is an error on top of another error...
The unintended result is that FamilySearch specifies that these line values may contain line breaks...

HEAD and TRLR

Chapter 1 Data Representation Grammar defines just two records types: CONC & CONT. The HEAD & TRLR record are not even mentioned. That seems a mistake.

The GEDCOM grammar allows multiple GEDCOM forms. Chapter 2 Lineage-Linked ~~Grammar~~ Form defines the so-called lineage-linked form. The HEAD & TRLR record are defined within this chapter. That seems a mistake.
It is definitely wrong that the HEAD.GEDC.FORM line value supposedly specifies the GEDCOM form, but that the entire GEDCOM header is defined within the lineage-linked form. While it is defined as part of the lineage-linked form, the HEADER definition *can not* allow any other possible line value for HEAD.GEDC.FORM record than LINEAGE-LINKED, and that in turn implies that the HEAD.GEDC.FORM record is superfluous and redundant.

The GEDCOM header & trailer record should both be part of the GEDCOM grammar, not of any particular GEDCOM form.

Chapter 2 Lineage-Linked ~~Grammar~~ Form

Introduction

This chapter describes the specific tag, value, and pointer combinations used for exchanging familybased lineage-linked genealogical information in the GEDCOM format. Lineage-linked data pertains to individuals linked in family relationships across multiple generations. The chapter also addresses specific compatibility issues pertaining to previous Lineage-Linked GEDCOM Form releases and contains a simple lineage-linked GEDCOM ~~transmission~~ file example.

The Lineage-Linked GEDCOM Form defined in this chapter is based on the general framework of the GEDCOM data representation grammar defined in Chapter 1. Commercial genealogical software systems are invited to use the Lineage-Linked GEDCOM Form to exchange data. It is the only form approved for exchanging data with Ancestral File, TempleReady and other Family History resource files maintained for this purpose.

Organization

The basic description of the **Lineage-Linked GEDCOM Form's** ~~**grammar**~~ is presented in the following three major sections:

! "Record Structures of the Lineage-Linked Form" (beginning on page 23)
! "Substructures of the Lineage-Linked Form" (beginning on page 31)
! "Primitive elements of the Lineage-Linked Form" (beginning on page 41)

The definition of the tags used in defining the lineage-linked structures are contained in Appendix A.

Symbols Used in Chapter 2

The following symbols are used in Chapter 2:

<<double_angle bracket>>

Indicates that a subordinate GEDCOM structure pattern of either a record, structure, or substructure is to be substituted in place of the line containing the enclosing double angle brackets. The substitute structure pattern is found subordinate to the LINEAGE_LINKED_GEDCOM beginning on page 24 for record pattern definition or in alphabetical order under the "Substructures of the Lineage-Linked Form" section, beginning on page 31.

<Single_angle bracket>

Indicates the name of the appropriate value for this GEDCOM line— <Primitive>. The specific definition of this value is found in alphabetical order in "Primitive Elements of the Lineage-Linked Form," beginning on page 41.

{braces}

Indicates the minimum to maximum occurrences allowed for this structure or line— {Minimum:Maximum}. Note that minimum and maximum occurrence limits are defined relative to the enclosing superior line. This means that a required line (minimum = 1) *is not required if the optional enclosing superior line is not present*. Similarly, a line occurring only once (maximum = 1) may occur multiple times as long as each occurs only once under its own multiple-occurring superior line.

[Square brackets]

Indicates a choice of one or more options— [Choice of].

| vertical bar |

Separates the multiple choices, for example [Choice 1 | Choice 2].

n level number

A level number which assumes the level number of the line which referenced the substructure name.

+1, **+2** ...

A **+1** level number is 1 greater than the level number assumed by the superior **n** level. A **+2** level number is 2 greater, and so forth.

0xHH

Indicates an allowable hexadecimal character value where HH is that value, for example, 0x20 (decimal 32) indicates the space character.

Lineage-Linked Form Usage Conventions

! The order in which GEDCOM lines are written to a GEDCOM file is controlled by the context and level number. When the lines are of equal level number but have a different tag name then the order is not significant. ~~The occurrence of equal level numbers and equal tags within the same context imply that multiple opinions or multiple values of the data exist. The significance of the order in these cases is interpreted as the submitter's preference.~~ The occurrence of equal level numbers and equal tags within the same context happens in two situations: events that may happen more than once (e.g. marriage), and events that happen just once, but for which there are multiple possible conflicting values (e.g. birth date). In the latter case, the order of the records is interpreted as the submitter's preference. *The most preferred value being the first with the least preferred data listed in subsequent lines by order of decreasing preference.* For example, a researcher who discovers conflicting evidence about a person's birth event would list the most credible information first and the least credible or least preferred items last.

! Systems that support multiple fields or structures should allow their users to indicate their preference opinion. Systems that only store single value structures should use the preferred information (the first occurrence listed) and store the remaining information as an exception, preferably *within an appropriate NOTE field* or in some way that the patron has ready access to the less-preferred data when viewing the record.

! Conflicting event dates and places should be represented by placing them in separate event structures with appropriate source citations rather than by placing them under the same enclosing event.

! The Lineage-Linked GEDCOM Form uses the TYPE tag to classify its superior tag for the viewer.

The value portion given by the TYPE tag is not intended to inform a computer program how to process the data, unless there is a list of standardized or controlled line_value choices given by the definition of the line value in this standard. The difference between an uncontrolled line value and a note value is that displaying systems should always display the *type value* when they display the data from the associated context. This gives the user flexibility in further defining information in a compatible GEDCOM context and the reader to understand events or facts which have not been classified by a specific tag. For example:

```
—1 EVEN
———2 TYPE Awarded BSA Eagle Rank
———2 DATE 1980
```

! All controlled line_value choices should be considered as case insensitive.
This means that the values should be converted to all uppercase or all lowercase prior to comparing. The terms UPPERCASE and UpperCase are considered equal. TAGS are always UPPERCASE.

All lineage-linked tags are UPPERCASE

The GEDCOM grammar defines which characters are allowed in tags. The GEDCOM grammar allows both uppercase and lowercase characters in tags. GEDCOM tags are case-sensitive.

The statement here, that all GEDCOM tags are UPPERCASE is merely a statement about the lineage-linked form; all the tags of the lineage-linked form are UPPERCASE.
GEDCOM readers will convert so-called controlled line values to upper or lowercase, but will not convert tags either way. A GEDCOM writer must use the actual lineage-linked tags, which are always ALL-UPPERCASE.

As this statement is listed in a section about lineage-linked form *conventions*, it applies to user-defined tags as well; user-defined tags should be UPPERCASE.

! All GEDCOM lines have either a **value** or a **pointer** unless the line contains subordinate GEDCOM lines. In other words the presence of a level number and a tag alone should not be used to assert data (i.e. **1 DEAT Y** should be used to imply a death known to have happened but date and place are unknown, not **1 DEAT**). The Lineage-linked form **does not allow** a GEDCOM line with both a value and a pointer on the same line.

21
22
22
23

Record Structures of the Lineage-Linked Form

LINEAGE_LINKED_GEDCOM:=

This is a model of the lineage-linked GEDCOM structure for submitting data to other lineage-linked GEDCOM processing systems. A header and a trailer record are required, and they can enclose any number of data records. Tags from Appendix A (see page 83) must be used in the same context as shown in the following form. User defined tags (see <NEW_TAG> on page 56) are discouraged but when used must begin with an under-score. Tags that are required within a desired context have been bolded. Note that some contexts are not required but if they are used then the bolded tags are required.

0 <<HEADER>>	{1:1}	p.23
0 <<SUBMISSION_RECORD>>	{0:1}	p.28
0 <<RECORD>>	{1:M}	p.24
0 **TRLR**	{1:1}	

TRLR record not superfluous

The TRLR record has neither a line value nor subrecords, and may seem superfluous. However, FamilySearch GEDCOM 5.5.1 still supports multi-volume GEDCOM files; a series of files (volumes) that must be read as a single GEDCOM file. For these multi-volume GEDCOM files, the TRLR record indicates the end of the volumes and the GEDCOM file, the absence of the TRLR record indicates that there should be at least one more volume to process.
Multi-volume GEDCOM files are obsolete.

The TRLR record is reassuring; it confirms that the end of a complete GEDCOM file has been reached.

line terminator

The TRLR record ends with a line terminator just like any other record. The TRLR record may not be followed by anything else.
Some old GEDCOM files contain a Ctrl-Z after the TRLR record. That is an error, and a GEDCOM reader should report it as such.

submission record

The FamilySearch GEDCOM specification defines both a mandatory <<SUBMITTER_RECORD>> and an optional <<SUBMISSION_RECORD>>. This top-level definition of GEDCOM file singles out the SUBMISSION_RECORD, and the **RECORD** definition below lists all top-level records, except the header, the trailer, *and* <<SUBMISSION_RECORD>>.
What it says here is that the <<SUBMISSION_RECORD>>, if present, must come directly after the header record, while the <<SUBMITTER_RECORD>> may occur anywhere before the trailer record.

record order

There a few issues with this demand on record order.

- These statements about record order being important or not are made in the chapter about the lineage-linked form. There is no statement about the relevance or irrelevance of record order in the definition of the GEDCOM grammar.
- The first item in this list of *Lineage-Linked Form Usage Conventions* clearly states that “When the lines are of equal level number but have a different tag name then the order is not significant.”. Demanding that one record type is listed before all other records anyway

  is almost a direct contradiction of that statement.
- The demand the FamilySearch GEDCOM specification makes here, and FamilySearch's own actual practice are not in accordance with each other.

PAF order

FamilySearch's own product, Personal Ancestral File does not comply with this instruction. PAF always places the submitter record directly after the header, and the submission record, if present, after the submitter record. It makes sense to place the submitter record directly after the header; conceptually it is a part of the header, just in a separate record. The submission record was introduced in the GEDCOM 5.4 specification. In the GEDCOM 5.3 specification, the above still looked like this:

```
0 <<HEADER>>                                         {1:1}
0 <<RECORD>>                                         {1:M}
0 TRLR                                               {1:1}
```

When FamilySearch added the <<SUBMISSION_RECORD>> to the GEDCOM 5.4 specification, and changed this definition to explicitly include the <<SUBMISSION_RECORD>> directly after the header, they apparently simply forgot about the <<SUBMITTER_RECORD>> record.

To correspond with FamilySearch's own practice, the <<RECORD>> definition should exclude both the <<SUBMISSION_RECORD>> and the <<SUBMITTER_RECORD>>, while the top-level definition should look like this:

```
0 <<HEADER>>                                         {1:1}
0 <<SUBMITTER_RECORD>>                               {1:1}
0 <<SUBMISSION_RECORD>>                              {0:1}
0 <<RECORD>>                                         {1:M}
0 TRLR                                               {1:1}
```

The SUBMISSION_RECORD is obsolete.

GEDCOM 5.4 Lineage-Linked Form

In GEDCOM 5.5.1, the TRLR record signifies the end of the GEDCOM file.
In GEDCOM 5.4, the TRLR record may be followed by embedded multi-media objects, in which case the end of the file is marked by an the END record:

```
0 <<HEADER>>                                         {1:1}
0 <<SUBMISSION_RECORD>>                              {0:1}
0 <<RECORD>>                                         {1:M}
0 TRLR                                               {1:1}
0 <<MULTIMEDIA_RECORD>>                              {0:M}
0 END                                                {0:1}
```

GEDCOM 5.5 supports embedded multimedia records like GEDCOM 5.4, but they appear before the TRLR record, just like all other records.

HEADER:=

```
n HEAD                                               {1:1}
  +1 SOUR <APPROVED_SYSTEM_ID>                       {1:1}     p.42
    +2 VERS <VERSION_NUMBER>                         {0:1}     p.64
    +2 NAME <NAME_OF_PRODUCT>                        {0:1}     p.54
    +2 CORP <NAME_OF_BUSINESS>                       {0:1}     p.54
      +3 <<ADDRESS_STRUCTURE>>                       {0:1}     p.31
    +2 DATA <NAME_OF_SOURCE_DATA>                    {0:1}     p.54
      +3 DATE <PUBLICATION_DATE>                     {0:1)     p.59
      +3 COPR <COPYRIGHT_SOURCE_DATA>                {0:1)     p.44
      +4 [CONT|CONC] <COPYRIGHT_SOURCE_DATA>        {0:M}   p.44
  +1 DEST<RECEIVING_SYSTEM_NAME>                    {0:1*}  p.59
  +1 DATE <TRANSMISSION_DATE>                       {0:1}   p.63
    +2 TIME <TIME_VALUE>                            {0:1}   p.63
  +1 SUBM @<XREF:SUBM>@                             {1:1}   p.28
  +1 SUBN @<XREF:SUBN>@                             {0:1}   p.28
  +1 FILE <FILE_NAME>                               {0:1}   p.50
  +1 COPR <COPYRIGHT_GEDCOM_FILE>                   {0:1}   p.44
  +1 GEDC                                           {1:1}
    +2 VERS <VERSION_NUMBER>                        {1:1}   p.64
    +2 FORM <GEDCOM_FORM>                           {1:1}   p.50
  +1 CHAR <CHARACTER_SET>                           {1:1}   p.44
    +2 VERS <VERSION_NUMBER>                        {0:1}   p.64
  +1 LANG <LANGUAGE_OF_TEXT>                        {0:1}   p.51
  +1 PLAC                                           {0:1}
    +2 FORM <PLACE_HIERARCHY>                       {1:1}   p.58
  +1 NOTE <GEDCOM_CONTENT_DESCRIPTION>              {0:1}   p.50
```

23

24

```
    +2 [CONC|CONT] <GEDCOM_CONTENT_DESCRIPTION>     {0:M}
```

* NOTE:
Submissions to the Family History Department for Ancestral File submission or for clearing temple ordinances must use a ~~DESTination~~ DEST (destination) of **ANSTFILE** or **TempleReady**, respectively.

The header structure provides information about the entire ~~transmission~~ GEDCOM file. The ~~SOURce~~ SOUR (source) system name identifies which system sent the data. The ~~DESTination~~ DEST (destination) system name identifies the intended receiving system.

Additional GEDCOM standards will be produced in the future to reflect GEDCOM expansion and maturity. This requires the reading program to make sure it can read the **GEDC.VERS** and the **GEDC.FORM** values to insure proper readability. The **CHAR** tag is required. ~~All character codes greater than 0x7F **must be converted to ANSEL**. (See Chapter 3, starting on page 77.)~~ The entire header must, just like the rest of the file, be encoded using the encoding specified in the header.

must check GEDCOM version

This is the only spot where the GEDCOM specification states that a GEDCOM reader must check the GEDCOM version of a GEDCOM file, to make sure it can read it.

> This requires the reading program to make sure it can read the **GEDC.VERS** and the **GEDC.FORM** values to insure proper readability.

This is an important instruction. Too many genealogy applications do not do so, and that is a serious error.
A GEDCOM reader must check the GEDCOM version of a file, and only try to read GEDCOM versions it explicitly supports. For GEDCOM versions it does not support, it should state so and not try to import the GEDCOM file.
To do this correctly, a GEDCOM reader must be able to recognise GEDCOM 5.5.1 files that

are mislabelled as GEDCOM 5.5 files.

GEDCOM reader best practice

- Always check the GEDCOM version
- Be sure to detect GEDCOM 5.5.1 files correctly (see bonus chapter *GEDCOM 5.5.1 Version Detection*)
- Only try to import a GEDCOM file if its version is supported
- **do not** try to import unsupported GEDCOM versions, clearly state the GEDCOM version is not supported

references

GEDCOM Version Detection

WieWasWie GEDCOM Header

The ostensible GEDCOM files produced by WieWasWie have an empty GEDCOM header; there is only the `0 HEAD` line, all the mandatory subrecords are missing.
This major defect was never fixed. The WieWasWie newsletter of 10 Mar 2015 announced that the family tree functionality would be retired, and that the GEDCOM export had been improved to make migration to another product easier, but the WieWasWie GEDCOM files still did not have a real GEDCOM header.
The WieWasWie trees were small, and WieWasWie GEDCOM files are rare, and otherwise problematic. It is okay to simply reject WieWasWie's ostensible GEDCOM files as not really GEDCOM files, but if you want to try read them, assume GEDCOM version 5.5.1 and UTF-8 encoding.

references

WieWasWie Nieuwsbrief 2015 No. 1: Verhuis je stamboom
Blog Coret 2015-03-06: Het gezwalk van WieWasWie
Blog Coret 2015-03-08: Hoe het CBG gegevensverlies veroorzaakt
WieWasWie GEDCOM
Gaenovium Presentations: Louis Kessler: Reading Wrong GEDCOM Right

Legacy Family Tree GEDCOM Version

Legacy Family Tree version 3 and 4 GEDCOM files lack the mandatory GEDCOM version number.
Legacy Family Tree version 3 and 4 use GEDCOM 5.5.

~~“All character codes greater than 0x7F must be converted to ANSEL”~~

catch-22

A GEDCOM writer specifies the character encoding used through the `HEAD.CHAR` line value. A GEDCOM reader must process the GEDCOM file using the specified encoding. There is a catch-22 here: A GEDCOM reader must read the GEDCOM header to discover the character encoding used, but it cannot read the GEDCOM header until it knows the encoding of the file.

FamilySearch “solution”

FamilySearch's preposterous “solution” to this catch-22 is to demand that the header is encoded using ANSEL, regardless of the encoding the header specifies for the file. That demand is both impossible and impractical.
It is generally impossible because ANSEL supports only a tiny subset of Unicode; most Unicode characters cannot be converted to ANSEL. It is immensely impractical because a

GEDCOM reader detects the end of a record by detecting the beginning of the next record, and now it would have to do that while the next record may use a different character encoding - that imposes considerable complexity.
Perhaps the most fundamental issue of all is that if you put an ANSEL-encoded header in front of an UTF-16 encoded GEDCOM file, it is no longer an UTF-16 encoded GEDCOM file. In fact, if you put an ANSEL-encoded header in front of anything but an ANSEL-encoded file, it probably isn't a text file any more.

By the way, you may be tempted to assume that this ANSEL demand is some holdover from a pre-Unicode version of GEDCOM, but it is not. Unicode support was added in GEDCOM 5.3, and this ANSEL demand was added in GEDCOM 5.4.

FamilySearch surely meant their ANSEL-demand to say that the entire header should be using ANSEL, but what they actually wrote is even stranger; only characters above 0x7F must be converted to ANSEL, so a header for a UTF-16 GEDCOM should use UTF-16 for code points 0x00 through 0x7F, and use ANSEL for all other code points, resulting in an mix of UTF-16 and ANSEL encoded characters...

contradiction

The ANSEL demand made here is contradicted by Chapter 3 Using Character Sets in GEDCOM:

> If the Unicode environment is used to produce a GEDCOM ~~transmission~~ file, the header record would also be in Unicode, requiring receiving systems to determine whether the ~~transmission~~ GEDCOM file is Unicode or ASCII before they could interpret the GEDCOM header.

That sentence is sloppy in more ways than one, but still clearly communicates that a GEDCOM file encoded in UTF-16 has a GEDCOM header encoded in UTF-16, not ANSEL.

Another sentence in Chapter 3 Using Character Sets in GEDCOM is even clearer:

> The character set for an entire ~~transmission~~ GEDCOM file is specified in the character set line of the header record.

That statement leaves no doubt: the entire file (“transmission” is confused FamilySearch-speak for file) uses one encoding, the one specified in the GEDCOM header.

one encoding throughout

GEDCOM files are supposed to be text files. To be text files, they must use a single encoding and single line terminator throughout the file.
The GEDCOM header must use the same encoding as the rest of the file.

real-world practice

FamilySearch's defective ANSEL demand is universally ignored. Not one vendor has ever tried to implement it in any product. All vendors use the encoding specified in the header for the entire file, including the header itself.
Even FamilySearch themselves do do not what FamilySearch demands! FamilySearch's Personal Ancestral File encodes each GEDCOM header the same way as the rest of the GEDCOM file.

recognising the encoding

There *is* a catch-22. The key to reading a GEDCOM file correctly is to read the header first, and the key to reading a GEDCOM header correctly is to figure out the encoding first. There is much to say about reading a GEDCOM header correctly, but to keep this brief: the solution to the catch-22 is to start with an analysis pass through the header which only figures out the encoding, before actually reading the header.

GEDCOM writer best practice

- choose one legal encoding
- use that encoding for the entire GEDCOM file
- specify the encoding used through the HEAD.CHAR line value

references

GEDCOM header encoding

HEAD.CHAR.VERS

The mandatory HEAD.CHAR line value specifies the character encoding used. The optional HEAD.CHAR.VERS is a version number, so presumably specifies a version number for that character encoding.

The HEAD.CHAR.VERS value is never used (but it is abused), and few GEDCOM readers support it.

support

Character sets can have version numbers, but are designed such that regular applications need not be aware which version they are using. Different versions of Windows support different versions of Unicode, but most Windows developers are not even aware of that.
FamilySearch never made it clear just why they included an optional HEAD.CHAR.VERS record. The most likely reason is simply that they mistakenly assumed that it would be a good idea. The truth is that is completely superfluous.

real-world HEAD.CHAR.VERS abusage

The HEAD.CHAR.VERS record has been creatively abused by GEDitCOM, MacFamilyTree and Ahnenblatt.

Reunion

Reunion is a MacOS application that supports the MacRoman character set. Reunion specifies the use of this illegal character set just like any other character set, legal or not, through the HEAD.CHAR line value, like this:

```
1 CHAR MACINTOSH
```

The MacRoman character set should not be used, but when it is used anyway, that is how it should be specified.

GEDitCOM

GEDitCOM is another MacOS application that supports the MacRoman character set. GEDitCOM specifies the usage of this illegal character set like this:

```
1 CHAR ASCII
2 VERS MacOS Roman
```

This is completely wrong. GEDitCOM starts by lying that the character set used is ASCII. Then, it claims that MacRoman is a particular version of ASCII (it is not), and we are supposed to interpret that claim as the statement that MacRoman is being used.
GEDitCOM should not use MacRoman at all, but when it does, it should be honest about it and specify it in a straightforward, non-confusing way, like Reunion.

This self-contradictory abusage of HEAD.CHAR.VERS does not deserve to be supported.
GEDitCOM itself has to support it, but only because genealogy applications should be able to read GEDCOM and not-quite-GEDCOM files produced by itself, including earlier versions of itself.
If the character set specified is ASCII, the GEDCOM file should be interpreted as ASCII, and processing should end with a fatal error as soon as a non-ASCII code is encountered.

MacFamilyTree

MacFamilyTree up to version 6 or so specifies MacRoman in exactly the same erroneous ways as GEDitCOM does. From around version 6 up to and including version 8.3.4, MacFamilyTree uses a minor variation of the same syntax:

```
1 CHAR ASCII
```

```
2 VERS MACINTOSH
```

This was fixed in MacFamilyTree 8.3.5, released in April 2018; MacFamilyTree 8.3.5 and later uses `1 CHAR MACINTOSH` as it should.

Ahnenblatt

This is how Ahnenblatt abuses the HEAD.CHAR.VERS value:

```
1 CHAR ANSI
2 VERS 1252
```

The HEAD.CHAR value specifies that Ahnenblatt is using Windows ANSI. That is an illegal choice, but this is the right way to specify it.
However, the HEAD.CHAR.VERS value below it is nonsense. There is no Windows ANSI version 1252.
The intention is to specify code page 1252, but the HEAD.CHAR.VERS value is not for code pages, it is for version numbers.
What's more, if there were a way to specify the Windows ANSI code page, it would not be necessary to specify code page 1252. There is no way to specify a particular Windows ANSI code page. GEDCOM does not allow Windows ANSI. In practice, Windows ANSI is assumed to be code page 1252.

This HEAD.CHAR.VERS abusage does not deserve to be supported either. Applications should not be using Windows ANSI at all, it is illegal. GEDCOM readers that do support Windows ANSI, should always assume code page 1252.

Ahnenblatt 3.0, expected to release this year (2018 CE), will export to UTF-8 exclusively.

GEDCOM reader best practice

- read the HEAD.CHAR value
- upon encountering a HEAD.CHAR.VERS record, issue the warning that HEAD.CHAR.VERS is not supported
- report an error if the HEAD.CHAR.VERS line value does not look like a version number (demand at least two numbers separated by a dot)

GEDCOM writer best practice

- use only legal encodings
- specify the encoding used through the HEAD.CHAR value
- do not use the HEAD.CHAR.VERS record to specify a character set version
- do not abuse the HEAD.CHAR.VERS record for anything else either

GEDCOM validator best practice

- upon encountering a HEAD.CHAR.VERS record, warn that it has never been used, only abused
- report an error if the HEAD.CHAR.VERS line value does not look like a version number (demand at least two numbers separated by a dot)

references

GEDCOM Character Encodings

GEDCOM version number

The HEAD.GEDC.VERS value is defined as a <VERSION_NUMBER>, which is defined as free-form text. As noted in the *version number* annotation, that does not make sense. The GEDCOM version number isn't an arbitrary string.

The GEDCOM version number is a proper version number with a straightforward format. The GEDCOM version number consists of three parts; a major version number, a minor version number, and an optional revision number.

The revision number has been used only once, namely for FamilySearch GEDCOM 5.5.1. The differences between GEDCOM 5.5 and 5.5.1 are significant, so the revision number is significant.

It is allowed, and indeed expected, to leave off the revision number when it's zero. In actual practice, some vendors also leave the minor version number off when it is zero. So, some products specify GEDCOM version 5.0 as `VERS 5`, while other products specify it as `VERSION 5.0`.
For example, Phil Sapiro's FamTree 4.21 uses `VERS 5`, while FormalSoft's Family Origins 1.1 uses `VERS 5.0`.

A number of products emulate Personal Ancestral File 5.0+, which produces GEDCOM 5.5.1 files that claim to be GEDCOM 5.5. files. This is an erroneous practice that presents an unnecessary challenge for GEDCOM readers; GEDCOM readers must figure out whether an ostensible GEDCOM 5.5 file really is a GEDCOM 5.5 file, or actually a GEDCOM 5.5.1 file.

best practice

GEDCOM writer

- Use GEDCOM 5.5.1 (or later)
- Honestly specify the GEDCOM version
- Default to UTF-8

GEDCOM reader

This is simplified version of the algorithm in bonus chapter *GEDCOM 5.5.1 Version Detection*.
It is presented here to get the idea across. It is recommended that you implement the full detection algorithm.

- Read the GEDCOM version
- If the claimed version is 5.5, check whether it is actually 5.5.1
  - If the encoding is UTF-8, it is 5.5.1
  - If it is PAF 5.0 or later, it is 5.5.1
  - If the GEDCOM header contains HEAD.SOUR.CORP.EMAIL, it is 5.5.1
  - If the GEDCOM header contains HEAD.SOUR.CORP.WWW, it is 5.5.1
  - If it is actually 5.5.1, issue an informational message
- If it is a GEDCOM version you do not support, state that implementation limitation and abort

references

Truncated GEDCOM Version
GEDCOM Version Detection

RECORD:=

[		
n <<FAM_RECORD>>	{1:1}	p.24
|		
n <<IINDIVIDUAL_RECORD>>	{1:1}	p.25
|		
n <<MULTIMEDIA_RECORD>>	{1:1}	p.26
|		
n <<NOTE_RECORD>>	{1:1}	p.27
|		
n <<REPOSITORY_RECORD>>	{1:1}	p.27
|		
n <<SOURCE_RECORD>>	{1:1}	p.27
|		
n <<SUBMITTER_RECORD>>	{1:1}	p.28
]		

<<RECORD>>

The name of this syntax definition, <<RECORD>>, is ill-chosen. It does not define all records. It only defines top-level records, but <<TOPLEVEL_RECORD>> would not be a good name either, as it does not include all top-level records.
This <<RECORD>> definition includes all the top-level record types, except the header, the trailer, and the obsolete optional <<SUBMISSION_RECORD>>. <<MAIN_RECORD>> would be a better name.

unexplained bold letters

FamilySearch does not explain the odd use of bold letters within this definition. This is related to the Common GEDCOM Identifier Naming Convention.
Most genealogy applications generate cross-reference identifiers that consist of a single capital letter followed by a number, e.g. `@S123@` or `@F456@`. The emboldened letters are a suggestion to use that particular capital letter for records of that type. Most vendors use a slightly different set of letters, which typically includes using the letter N for <<**N**OTE_RECORD>> identifiers.

references

Common GEDCOM Identifier Naming Convention

user-defined top-level records

GEDCOM allows vendors to define additional records types, and that includes top-level record types.
For example, multiple vendors use top-level _PLACE or _LOC records to deal with GEDCOM 5.5.1's *place record design error*.

FAM_RECORD:=

```
n @<XREF:FAM>@ FAM                                 {1:1}
   +1 RESN <RESTRICTION_NOTICE>                    {0:1)      p.60
   +1 <<FAMILY_EVENT_STRUCTURE>>                   {0:M}      p.32
   +1 HUSB @<XREF:INDI>@                           {0:1}      p.25
   +1 WIFE @<XREF:INDI>@                           {0:1}      p.25
   +1 CHIL @<XREF:INDI>@                           {0:M}      p.25
   +1 NCHI <COUNT_OF_CHILDREN>                     {0:1}      p.44
   +1 SUBM @<XREF:SUBM>@                           {0:M}      p.28
   +1 <<LDS_SPOUSE_SEALING>>                       {0:M}      p.36
   +1 REFN <USER_REFERENCE_NUMBER>                 {0:M}      p.63, 64
```

24

25

```
     +2 TYPE <USER_REFERENCE_TYPE>                 {0:1}      p.64
   +1 RIN <AUTOMATED_RECORD_ID>                    {0:1}      p.43
   +1 <<CHANGE_DATE>>                              {0:1}      p.31
```

```
  +1 <<NOTE_STRUCTURE>>                                    {0:M}    p.37
  +1 <<SOURCE_CITATION>>                                   {0:M}    p.39
  +1 <<MULTIMEDIA_LINK>>                                   {0:M}    p.37, 26
```

The ~~FAMily~~ FAM (family) record is used to record marriages, common law marriages, and family unions caused by two people becoming the parents of a child. There can be no more than one HUSB ~~/father~~ record and one WIFE ~~/mother~~ record listed in each <<FAM_RECORD>>. If, for example, a man participated in more than one family union, then he would appear in more than one <<FAM_RECORD>>. ~~The family record structure assumes that the HUSB/father is male and WIFE/mother is female.~~ No assumption about gender should be made based on the usage of the FAM.HUSB or FAM.WIFE record.

The preferred order of the ~~CHILdren~~ CHIL (children) pointers within a ~~FAMily~~ FAM (family) structure is chronological by birth.

empty families

Notice that FAM.HUSB, FAM.WIFE and FAM.CHIL are *all* optional. Thus, the lineage-linked form allows a family group without any family members.
Most genealogy applications do not create empty family groups, and automatically delete FAM records when you remove the last family member from a family.
It is okay to have multiple children in a family without parents; that states that the children are siblings.

references

- Marriage in GEDCOM

FAM.SUBM deprecated

The ability to indicate separate submitters for each family group makes limited sense and has remained widely unsupported. Consider this feature deprecated.

FAM.HUSB and FAM.WIFE are not roles

The subrecord names FAM.HUSB and FAM.WIFE *suggest but do not specify* particular roles within a family group. The FamilySearch text says that the gender of the INDI records that the FAM.HUSB and FAM.WIFE point to is *assumed*; that means that their gender is not verified against the suggested role. That statement may originally have been meant as an optimisation; no GEDCOM reader should go to the trouble of verifying that the gender of the linked records matches the suggested roles. Today, leading developers of genealogy applications understand it to mean that the gender of the linked records may be anything.

No application should assume an individual linked from FAM.HUSB is a husband and an individual linked from FAM.WIFE is a wife. The names of the lineage-linked form records suggest that interpretation, but the lineage-linked form does not specify that interpretation. Thus, although FamilySearch's definition tried to exclude same-sex couples from the standard, genealogy applications use the FAM record for all couples.

The optional FAM.HUSB and FAM.WIFE subrecords both link to an INDI record, *without any gender restrictions*. The FAM.HUSB will often link to a male individual, and FAM.WIFE will often link to a female individual, but that need not be the case.

Gender is specified by INDI.SEX, and by INDI.SEX only. The FAM.HUSB and FAM.WIFE records should be treated as awkward synonyms for a genderless FAM.SPOU records.

best practice

The terms husband and wife are gender-based names for partners. Any usage of these terms on screen and in reports should always be based on the gender on the individual (INDI.SEX), never on assumed FAM.HUSB or FAM.WIFE roles.

references

- Marriage in GEDCOM
- Same-Sex Marriage in GEDCOM

order of children

The FamiySearch GEDCOM specification states that “The preferred order of the ~~CHILdren~~ CHIL (children) pointers within a ~~FAMily~~ FAM (family) structure is chronological by birth.”, and that statement is neither clear enough nor strong enough.

Children within a family group should be ordered chronologically, i.e. in order of birth, and not anti-chronologically, and the order of multiplets born on the same day should be preserved.

Although it would certainly make some sense to let applications put children in order, it is current practice to let the user decide the order of children. Users expect genealogy applications to respect the order they choose, and not change it on import on export.

genealogy application best practice

- encourage users to enter dates, including approximate dates
- order children within a family group chronologically by birth
    - *when ordering*, treat dates consisting of just a year as January 1 of that year
    - *when ordering*, treat dates consisting of just a year and a month as the first day of that month
- for multiplets born on the same day, respect the order chosen by the user
- Encourage users to put and keep children in order:
    - when you show children, show their birth (or baptism) dates
    - visually indicate problems with the order of children
    - alert users when they insert a child out of order
    - include a children-in-order check as a basic genealogy consistency check
- Preserve the order of children across GEDCOM import and export
- optionally: make ordering children easy with a reorder children function

GEDCOM 5.5 <<FAM_RECORD>> erratum missing

The GEDCOM 5.5 errata sheet specifies that two lines should be added to the <<FAM_RECORD>> definition, below the <<SOURCE_CITATION>> line, like this:

```
+ 1 <<SOURCE_CITATION>>
    + 2 <<NOTE_STRUCTURE>>
    + 2 <<MULTIMEDIA_LINK>>
```

These additional lines are not present in GEDCOM 5.5.1 specification, because *the GEDCOM 5.5 errata sheet is wrong*!
The <<SOURCE_CITATION>> definition already includes the <<NOTE_STRUCTURE>> and <<MULTIMEDIA_LINK>>, so there is no need (and indeed, it would be wrong) to include them here.

INDIVIDUAL_RECORD:=

n @<XREF:INDI>@ **INDI**	{1:1}	
+1 RESN <RESTRICTION_NOTICE>	{0:1}	p.60
+1 <<PERSONAL_NAME_STRUCTURE>>	{0:M}	p.38
+1 SEX <SEX_VALUE>	{0:1}	p.61
+1 <<INDIVIDUAL_EVENT_STRUCTURE>>	{0:M}	p.34
+1 <<INDIVIDUAL_ATTRIBUTE_STRUCTURE>>	{0:M}	p.33
+1 <<LDS_INDIVIDUAL_ORDINANCE>>	{0:M}	p.35, 36
+1 <<CHILD_TO_FAMILY_LINK>>	{0:M}	p.31
+1 <<SPOUSE_TO_FAMILY_LINK>>	{0:M}	p.40
+1 SUBM @<XREF:SUBM>@	{0:M}	p.28

```
  +1 <<ASSOCIATION_STRUCTURE>>                  {0:M}    p.31
  +1 ALIA @<XREF:INDI>@                         {0:M}    p.25
  +1 ANCI @<XREF:SUBM>@                         {0:M}    p.28
  +1 DESI @<XREF:SUBM>@                         {0:M}    p.28
  +1 RFN <PERMANENT_RECORD_FILE_NUMBER>         {0:1}    p.57
  +1 AFN <ANCESTRAL_FILE_NUMBER>                {0:1}    p.42
  +1 REFN <USER_REFERENCE_NUMBER>               {0:M}    p.63, 64
    +2 TYPE <USER_REFERENCE_TYPE>               {0:1}    p.64
  +1 RIN <AUTOMATED_RECORD_ID>                  {0:1}    p.43
  +1 <<CHANGE_DATE>>                            {0:1}    p.31
  +1 <<NOTE_STRUCTURE>>                         {0:M}    p.37
  +1 <<SOURCE_CITATION>>                        {0:M}    p.39
  +1 <<MULTIMEDIA_LINK>>                        {0:M}    p.37, 26
```

The individual record is a compilation of facts, known or discovered, about an individual. Sometimes these facts are from different sources. This form allows documentation of the source where each of the facts were discovered.

The normal lineage links are shown through the use of pointers from the individual to a family through either the FAMC tag or the FAMS tag. The FAMC tag provides a pointer to a family where this person is a child. The FAMS tag provides a pointer to a family where this person is a spouse or parent. The **<<CHILD_TO_FAMILY_LINK>>** (see page 31) structure contains a FAMC pointer which is required to show any child to parent linkage for pedigree navigation. The <<CHILD_TO_FAMILY_LINK>> structure also indicates whether the pedigree link represents a birth lineage, or an adoption lineage~~, or a sealing lineage~~.

Linkage between a child and the family they belonged to at the time of an event can also be shown by a FAMC pointer subordinate to the appropriate event. For example, a FAMC pointer subordinate to an adoption event indicates a relationship to family by adoption. Biological parents can be shown by a FAMC pointer subordinate to the birth event(optional).

Other associations or relationships are represented by the ~~ASSOciation~~ ASSO (association) tag. The person's relation or association is the person being pointed to. The association or relationship is stated by the value on the subordinate RELA line. For example:

```
0 @I1@ INDI
1 NAME Fred/Jones/
1 ASSO @I2@
2 RELA Godfather
```

order of parents

The INDI record may contain more than one FAMC subrecord, with each INDI.FAMC record pointing to a family group the individual was part of at some time. These should be listed chronologically; thus, the official parents should be listed first, and the current legal parents should be listed last.

references

Behold blog: Sex in GEDCOM

alias ALIA

The ALIA record has a remarkable history.
FamilySearch GEDCOM 3.0 and 4.0 defined the ALIA (alias) record as way to record alternate names. FamilySearch GEDCOM 5.0 and later specify that alternate names must be recorded using multiple NAME records. At the same time, FamilySearch GEDCOM 5.0 defined a new ALIA record, namely as a way to record that another INDI is possibly for the same person.

There was no reason to create confusion by calling this the ALIA record. In fact, there was no reason to create this record type at all, because GEDCOM already supports the <<ASSOCIATION_STRUCTURE>> (ASSO record).

In practice, many products kept creating ALIA record to record alternate names. All versions of Ancestry.com Family Tree Maker kept creating ALIA records. Software MacKiev Family Tree Maker 2014.1 stopped creating ALIA records; it uses multiple NAME records as it should.

The new ALIA record, a link to another INDI record, has never been used. It only occurs in GEDCOM test files.

best practice

GEDCOM writer

- do not use ALIA to record alternate names
- uses multiple NAME to record alternate names
  - Export the main name first
  - Use additional NAME records to support alternate names
  - Use slashes to delimit the surname on every NAME record
- Do not use INDI.ALIA for linking to other INDI; consider it a deprecated feature
- You can use the ASSO record to link an INDI record to another INDI record

GEDCOM reader

- Read multiple NAME records containing alternate names
- Assume the first NAME record is the main, preferred name
- Optionally, read INDI.ALIA and INDI.NAME.ALIA records containing alternate names
- If the INDI.ALIA line value appears to be a reference, do not import it, but warn that this legal usage is considered deprecated and not supported

references

GEDCOM ALIA
FTM 2017 GEDCOM

INDI.SUBM deprecated

The ability to indicate separate submitters for each individual makes limited sense and has remained widely unsupported.
Consider this feature deprecated.

ANCI and DESI

The ANCI and DESI records remain widely unsupported and are rarely used.
Consider these records deprecated.

GEDCOM 5.5 <<INDIVIDUAL_RECORD>> erratum missing

The GEDCOM 5.5 errata sheet specifies that two lines should be added to the <<INDIVIDUAL_RECORD>> definition, below the <<SOURCE_CITATION>> line, like this:

```
+ 1 <<SOURCE_CITATION>>
    + 2 <<NOTE_STRUCTURE>>
    + 2 <<MULTIMEDIA_LINK>>
```

These additional lines are not present in GEDCOM 5.5.1 specification, because *the GEDCOM 5.5 errata sheet is wrong*!
The <<SOURCE_CITATION>> definition already includes the <<NOTE_STRUCTURE>> and <<MULTIMEDIA_LINK>>, so there is no need (and indeed, it would be wrong) to include them here.

MULTIMEDIA_RECORD:=

n @<XREF:OBJE>@ **OBJE**	{1:1}	
~~+1 **FILE** <MULTIMEDIA_FILE_REFN>~~	{1:M}	p.54
+1 **FILE** <MULTIMEDIA_FILE_REFERENCE>	{1:M}	p.54
+2 **FORM** <MULTIMEDIA_FORMAT>	{1:1}	p.54
+3 TYPE <SOURCE_MEDIA_TYPE>	{0:1}	p.62
+2 TITL <DESCRIPTIVE_TITLE>	{0:1}	p.48
+1 REFN <USER_REFERENCE_NUMBER>	{0:M}	p.63, 64
+2 TYPE <USER_REFERENCE_TYPE>	{0:1}	p.64
+1 RIN <AUTOMATED_RECORD_ID>	{0:1}	p.43
+1 <<NOTE_STRUCTURE>>	{0:M}	p.37
+1 <<SOURCE_CITATION>>	{0:M}	p.39
+1 <<CHANGE_DATE>>	{0:1}	p.31

The BLOB context of the multimedia record was removed in version 5.5.1. A reference to a multimedia file was added to the record structure. The file reference occurs one to many times so that multiple files can be grouped together, each pertaining to the same context. For example, if you wanted to associate a sound clip and a photo, you would reference each multimedia file and indicate the format using the FORM tag subordinate to each file reference.

non-existent <MULTIMEDIA_FILE_REFN>

The <<MULTIMEDIA_RECORD>> syntax references <MULTIMEDIA_FILE_REFN>, but the GEDCOM lineage-linked form doesn't define <MULTIMEDIA_FILE_REFN>, it defines <MULTIMEDIA_FILE_REFERENCE>.

multiple multimedia files deprecated

The GEDCOM 5.5.1 <<MULTIMEDIA_RECORD>> allows a single OBJE record to link to multiple related files.
This feature is not widely supported; many genealogy application will read only the first OBJE.FILE. To avoid loss of data, it is recommended to have just one OBJE.FILE subrecord per OBJE record.

GEDCOM 5.5 <<MULTIMEDIA_RECORD>>

The FamilySeach GEDCOM 5.5.1 <<MULTIMEDIA_RECORD>> is different from the GEDCOM 5.5 <<MULTIMEDIA_RECORD>>. Here is the GEDCOM 5.5 <<MULTIMEDIA_RECORD>> for comparison:

MULTIMEDIA_RECORD:=

```
n @<XREF:OBJE>@ OBJE                                              {1:1}
   +1 FORM <MULTIMEDIA_FORMAT>                                    {1:1}
   +1 TITL <DESCRIPTIVE_TITLE>                                    {0:1}
   +1 <<NOTE_STRUCTURE>>                                          {0:M}
   +1 <<SOURCE_CITATION>>                                         {0:M}
   +1 BLOB                                                        {0:1}
      +2 CONT <ENCODED_MULTIMEDIA_LINE>                           {1:M}
   +1 OBJE @<XREF:OBJE>@    /* chain to continued object */       {0:1}
   +1 REFN <USER_REFERENCE_NUMBER>                                {0:M}
      +2 TYPE <USER_REFERENCE_TYPE>                               {0:1}
   +1 RIN <AUTOMATED_RECORD_ID>                                   {0:1}
   +1 <<CHANGE_DATE>>                                             {0:1}
```

Large whole multimedia objects embedded in a GEDCOM file would break some systems. For this purpose, large multimedia files should be divided into smaller multimedia records by using the subordinate OBJE tag to chain to the next <MULTIMEDIA_RECORD> fragment. This will allow GEDCOM records to be maintained below the 32K limit for use in systems with limited resources.

Within the definition above, the BLOB subrecord has been marked as deprecated (in GEDCOM 5.5), because on second thoughts, FamilySearch considered BLOBs a bad idea, so they discontinued BLOB support in GEDCOM 5.5.1.
However, it is best to think of this entire GEDCOM 5.5 record as obsolete. Developers should pay attention to the small differences, and make sure that they are using the GEDCOM 5.5.1 definition.

SOURCE_CITATION not new

The <<SOURCE_CITATION>> line is not new in GEDCOM 5.5.1. It was added to GEDCOM 5.5 through the GEDCOM 5.5 errata sheet.

chain pointer

The <XREF:OBJE> pointer within the BLOB subrecord (on the line highlighted because it includes an illegal C-style comment) isn't a circular reference. As the GEDCOM 5.5 definition explains, it's there because encoded media file are likely to be larger than the (now obsolete) maximum logical record size of 32 KB. A GEDCOM 5.5 writer must break the encoded media files into pieces of less than 32KB, and then chain those pieces together. A GEDCOM 5.5 reader must follow the chain pointers and put the encoded media file together again.

no standard for multimedia file transfer

As long as the source and destination system are two different applications on the same computer or mirrored systems, the multimedia file links are efficient, but for sharing multimedia files to another system, a locally valid link is inadequate.
Links to web resources do remain valid between different systems, but their permanence cannot be guaranteed, and performance is likely to suffer.

The FamilySearch GEDCOM specification does not provide any standard, any rules or guidelines for bundling multimedia files with a GEDCOM file. It is not uncommon to bundle a GEDCOM file and media together in a ZIP file, but that only takes care of the media transfer, not the file paths, which are almost sure to be different on different systems, especially if these are managed by different users.

Some desktop applications have built-in smarts to try and repair files references, and several desktop applications provide syncing of desktop databases with web databases.

best practices

Applications should maintain links to media, *not* store media inside their database. Storing media inside the database bloats the database, and makes the media inaccessible to other tools. Applications should encourage users to collect and keep all related media in one directory (with subdirectories).

```
NOTE_RECORD:=

  n @<XREF:NOTE>@ NOTE <SUBMITTER_TEXT>                  {1:1}    p.63
    +1 [CONC|CONT] <SUBMITTER_TEXT>                      {0:M}
    +1 REFN <USER_REFERENCE_NUMBER>                      {0:M}    p.63, 64
      +2 TYPE <USER_REFERENCE_TYPE>                      {0:1}    p.64
    +1 RIN <AUTOMATED_RECORD_ID>                         {0:1}    p.43
    +1 <<SOURCE_CITATION>>                               {0:M}    p.39
    +1 <<CHANGE_DATE>>                                   {0:1}    p.31
```

```
REPOSITORY_RECORD:=

  n @<XREF:REPO>@ REPO                                   {1:1}
    +1 NAME <NAME_OF_REPOSITORY>                         {1:1}    p.54
    +1 <<ADDRESS_STRUCTURE>>                             {0:1}    p.31
    +1 <<NOTE_STRUCTURE>>                                {0:M}    p.37
    +1 REFN <USER_REFERENCE_NUMBER>                      {0:M}    p.63, 64
      +2 TYPE <USER_REFERENCE_TYPE>                      {0:1}    p.64
    +1 RIN <AUTOMATED_RECORD_ID>                         {0:1}    p.43
    +1 <<CHANGE_DATE>>                                   {0:1}    p.31
```

```
SOURCE_RECORD:=

  n @<XREF:SOUR>@ SOUR                                   {1:1}
    +1 DATA                                              {0:1}
      +2 EVEN <EVENTS_RECORDED>                          {0:M}    p.50
        +3 DATE <DATE_PERIOD>                            {0:1}    p.46
        +3 PLAC <SOURCE_JURISDICTION_PLACE>              {0:1}    p.62
      +2 AGNC <RESPONSIBLE_AGENCY>                       {0:1}    p.60
      +2 <<NOTE_STRUCTURE>>                              {0:M}    p.37
    +1 AUTH <SOURCE_ORIGINATOR>                          {0:1}    p.62
      +2 [CONC|CONT] <SOURCE_ORIGINATOR>                 {0:M}    p.62
    +1 TITL <SOURCE_DESCRIPTIVE_TITLE>                   {0:1}    p.62
      +2 [CONC|CONT] <SOURCE_DESCRIPTIVE_TITLE>          {0:M}    p.62
    +1 ABBR <SOURCE_FILED_BY_ENTRY>                      {0:1}    p.62
    +1 PUBL <SOURCE_PUBLICATION_FACTS>                   {0:1}    p.62
      +2 [CONC|CONT] <SOURCE_PUBLICATION_FACTS>          {0:M}    p.62
    +1 TEXT <TEXT_FROM_SOURCE>                           {0:1}    p.63
      +2 [CONC|CONT] <TEXT_FROM_SOURCE>                  {0:M}    p.63
    +1 <<SOURCE_REPOSITORY_CITATION>>                    {0:M}    p.40
    +1 REFN <USER_REFERENCE_NUMBER>                      {0:M}    p.63, 64
    +2 TYPE <USER_REFERENCE_TYPE>                        {0:1}    p.64
    +1 RIN <AUTOMATED_RECORD_ID>                         {0:1}    p.43
    +1 <<CHANGE_DATE>>                                    {0:1}     p.31
    +1 <<NOTE_STRUCTURE>>                                 {0:M}     p.37
    +1 <<MULTIMEDIA_LINK>>                                {0:M}     p.37, 26
```

Source records are used to provide a bibliographic description of the source cited. (See the <<SOURCE_CITATION>> structure, page 39, which contains the pointer to this source record.)

SUBMISSION_RECORD:=

```
n @<XREF:SUBN>@ SUBN                                      {1:1}
    +1 SUBM @<XREF:SUBM>@                                 {0:1}     p.28
    +1 FAMF <NAME_OF_FAMILY_FILE>                         {0:1}     p.54
    +1 TEMP <TEMPLE_CODE>                                 {0:1}     p.63
    +1 ANCE <GENERATIONS_OF_ANCESTORS>                    {0:1}     p.50
    +1 DESC <GENERATIONS_OF_DESCENDANTS>                  {0:1}     p.50
    +1 ORDI <ORDINANCE_PROCESS_FLAG>                      {0:1}     p.57
    +1 RIN <AUTOMATED_RECORD_ID>                          {0:1}     p.43
    +1 <<NOTE_STRUCTURE>>                                 {0:M}     p.37
    +1 <<CHANGE_DATE>>                                    {0:1}     p.31
```

The sending system uses a submission record to send instructions and information to the receiving system. TempleReady processes submission records to determine which temple the cleared records should be directed to. The submission record is also used for communication between Ancestral File download requests and TempleReady. Each GEDCOM ~~transmission~~ file should have only one submission record. Multiple submissions are handled by creating separate GEDCOM ~~transmission~~ files.

> The <<SUBMISSION_RECORD>> is obsolete.
> The SUBMISSION_RECORD is not merely LDS-specific but even product-specific; it exists for communication between Personal Ancestral File (PAF), a desktop genealogy application, and TempleReady, a proprietary administrative application. PAF and TempleReady are two legacy products of FamilySearch, that even FamilySearch themselves neither uses nor supports any more, so even vendors eager to support LDS-specific extensions need not support <<SUBMISSION_RECORD>> any more.
>
> There is no need for any genealogy application to read submission records, it is okay to ignore a <<SUBMISSION_RECORD>> when encountering one. There is no need for GEDCOM writers to produce a <<SUBMISSION_RECORD>> either, but products that do create a <<SUBMISSION_RECORD>> should not place it right after the header (and in front of the <<SUBMITTER_RECORD>>) as the specification says, but right after the *after* the <<SUBMITTER_RECORD>>, as that is what FamilySearch PAF does.

SUBMITTER_RECORD:=

```
n @<XREF:SUBM>@ SUBM                                      {1:1}
    +1 NAME <SUBMITTER_NAME>                              {1:1}     p.63
    +1 <<ADDRESS_STRUCTURE>>                              {0:1}*    p.31
    +1 <<MULTIMEDIA_LINK>>                                {0:M}     p.37, 26
    +1 LANG <LANGUAGE_PREFERENCE>                         {0:3}     p.51
    +1 RFN <SUBMITTER_REGISTERED_RFN>                     {0:1}     p.63
    +1 RIN <AUTOMATED_RECORD_ID>                          {0:1}     p.43
    +1 <<NOTE_STRUCTURE>>                                 {0:M}     p.37
    +1 <<CHANGE_DATE>>                                    {0:1}     p.31
```

The submitter record identifies an individual or organization that ~~contributed information contained in~~ created the GEDCOM ~~transmission~~ file. All records in the ~~transmission~~ GEDCOM file are assumed to be submitted by the <<SUBMITTER>> referenced in the ~~HEADer~~ <<HEADER>>, unless a ~~SUBMitter~~ SUBM (submiter) reference inside a specific record points at a different <<SUBMITTER>> record.

* Note: submissions to the ancestral file **require the name and address** of the submitter.

more than one <<SUBMITTER_RECORD>>

A typical GEDCOM 5.5.1 file contains exactly one <<SUBMITTER_RECORD>>, referenced from HEAD.SUBM. However, the lineage-linked allows multiple submitter records.

Both the <<INDIVIDUAL_RECORD>> and the <<FAM_RECORD>> optionally reference a submitter record. This allows multiple submitters to be associated with different records.

This is widely unsupported. It makes little sense for single-user applications, and even collaborative applications (social genealogy sites) do not use it. Practically all genealogy applications, including FamilySearch PAF support only one submitter per GEDCOM file.
The editor and reviewers are not aware of any genealogy application that supports more than submitter per GEDCOM file. Even if there is one, other genealogy applications are not able to import and process that data.
Consider this feature strongly deprecated.

GEDCOM writer best practice

- create exactly one submitter record
- have HEAD.SUBM reference this submitter record
- write the submitter record immediately after the GEDCOM header

GEDCOM reader best practice

- Remember the HEAD.SUBM cross-reference identifier
- Keep track of submitter record identifiers like you keep track of other identifiers
  - if any cross-reference identifier is reused, issue a fatal error (identifier already used) and abort the import
  - if the referenced submitter record is not found, issue a fatal error (referenced record not found) and abort the import

- Until you have the right submitter record:
  - Accept each submitter record you encounter (overwriting the previous one, if any)
  - issue an error if it isn't the one referenced from HEAD.SUBM

- Once you have the right submitter record, for each additional submitter record:
  - do *not* overwrite the submitter record you already found
  - issue an *implementation limitation* error (only one submitter record supported) and move on

SUBM.RFN obsolete

The SUBM.RFN record should never have been part of the GEDCOM specification. It should have been documented separately, as an LDS-extension. The SUBM.RFN record is not only LDS-specific, but even product-specific; it is a user identifier for Ancestral File. Ancestral File is obsolete now, so SUBM.RFN is obsolete too.

29

30

30

31

Substructures of the Lineage-Linked Form

ADDRESS_STRUCTURE:=

n **ADDR** <ADDRESS_LINE>	{1:1}	p.41
+1 CONT <ADDRESS_LINE>	~~{0:3}~~ {1:3}	p.41
+1 ADR1 <ADDRESS_LINE1>	{0:1}	p.41
+1 ADR2 <ADDRESS_LINE2>	{0:1}	p.41
+1 ADR3 <ADDRESS_LINE3>	{0:1}	p.41
+1 CITY <ADDRESS_CITY>	{0:1}	p.41
+1 STAE <ADDRESS_STATE>	{0:1}	p.42
+1 POST <ADDRESS_POSTAL_CODE>	{0:1}	p.41
+1 CTRY <ADDRESS_COUNTRY>	{0:1}	p.41
n PHON <PHONE_NUMBER>	{0:3}	p.57
n EMAIL <ADDRESS_EMAIL>	{0:3}	p.41
n FAX <ADDRESS_FAX>	{0:3}	p.41
n WWW <ADDRESS_WEB_PAGE>	{0:3}	p.42

The address structure should be formed as it would appear on a mailing label~~ using the ADDR and the CONT lines to form the address structure~~. ~~The ADDR and CONT lines are required for any address. The additional subordinate address tags such as STAE and CTRY are provided to be used by systems that have structured their addresses for indexing and sorting. For backward compatibility these lines are not to be used in lieu of the required ADDR and CONT line structure.~~ The ADDR line value should not be used; it is available for backward compatibility with older versions of GEDCOM only. For forward compatibility, GEDCOM writers should be using subrecords such as STAE and CTRY to provide structured addresses. Applications should not use the ADDR line value (and its CONT records) in lieu of a structured address.

Address structure

Notice that the <<ADDRESS_STRUCTURE>> isn't a single GEDCOM record, but one record optionally followed by a few more.
The <<ADDRESS_STRUCTURE>> always contains and starts with the ADDR record, optionally followed by PHON, EMAIL, FAX and WWW records.

PHONE, EMAIL, FAX & WWW

The PHON, EMAIL, FAX, and WWW records are *not* subrecords of the ADDR record, nor does the GEDCOM specification define another record that all these records are a subrecord of. The PHON, EMAIL, FAX, and WWW records simply appear at the same level as the ADDR record.

5.5 versus 5.5.1

GEDCOM 5.5 specifies the multiplicity of the ADDR record as {0:1}.
Thus, GEDCOM 5.5 allows the PHON record to appear *without* an ADDR record.
FamilySearch apparently considered that a mistake, so they changed this in GEDCOM 5.5.1.

The GEDCOM 5.5.1 specifies the multiplicity of the ADDR record as {1:1}.
The PHON, EMAIL, FAX, and WWW records may only appear when there is an ADDR record.

home, work and mobile

The <<ADDRESS_STRUCTURE>> allows up to three PHON records, EMAIL, FAX & WWW records. This allows inclusion of say a home, work and mobile phone number, but GEDCOM does not provide a mechanism for indicating which phone number is which.

best practice

- Although records on the same level may appear in any order, GEDCOM writers should put the ADDR record first.

references

GEDCOM ADDR

PAF addresses use URL instead of WWW

FamilySearch PAF uses GEDCOM 5.5.1, but PAF addresses do not use the WWW tag specified here, they use the illegal tag URL instead. That is an error in PAF, and FamilySearch should have fixed PAF. Instead, FamilySearch “fixed” the GEDCOM specification: in GEDCOM 5.6, the tag has changed from WWW to URL.

GEDCOM reader best practice

- Accept URL as a synonym for WWW
- Still issue an error stating that the URL tag is illegal

ADDR.CONT multiplicity contradiction

This FamilySearch definition contradicts itself.

- The <<ADDRESS_STRUCTURE>> syntax states that the multiplicity of the ADDR.CONT line is {0:3}.
- The <<ADDRESS_STRUCTURE>> description states that “The ADDR and CONT lines are required”, so there must be at least one ADDR.CONT line.

The multiplicity has been corrected to reflect the demand made by the description.

explicit CONT

That said, it should be noted that CONC or CONT are part of the GEDCOM grammar, and that the explicit inclusion of a CONC or CONT in a GEDCOM form is a mistake that promotes misinterpretation and confusion.
The text should simply have said that the (now deprecated) ADDR line value must consist of multiple lines of text. That a text which consists of multiple lines is recorded using CONT records is a given within GEDCOM.

deprecated

More important than the multiplicity correction itself is the fact that this correction is for old style addresses, which should no longer be used. See the *old style versus new style addresses* annotation.

old style versus new style addresses

GEDCOM 5.5.1 supports two address styles; old style and new style. The old style address is the ADDR line value (and its subordinate CONT line values): a single value split over several lines. The new style address, introduced in GEDCOM 5.4, is a structured format that uses the ADDR subrecords. A single address can be specified both ways at once.

The FamilySearch GEDCOM 5.5.1 specification demands the use of old-style addresses, by stating that, for backward compatibility, the new style should not be used in lieu of the required ADDR and CONT line structure. That requirement made sense back in 1999, when GEDCOM 5.5 was only four years old yet, but makes no sense any more. Today (2018 CE), GEDCOM writers should be using the new style address exclusively. The ADDR line value should not be used, but considered deprecated.

best practice

- Use structured addresses exclusively.

references

GEDCOM ADDR

Correspondence between Old Style and New Style Addresses

Part of the <<ADDRESS_STRUCTURE>> specification is provided in the <ADDRESS_LINE1>, <ADDRESS_LINE2> & <ADDRESS_LINE3> definitions; these definitions demands a correspondence between the lines that make up the ADDR line value on the one hand, and the ADR1, ADR2 & ADR3 line values on the other hand.
This correspondence demand isn't an issue for applications that use structured addresses exclusively.

ASSOCIATION_STRUCTURE:=

```
n ASSO @<XREF:INDI>@                          {1:1}    p.25
   +1 RELA <RELATION_IS_DESCRIPTOR>           {1:1}    p.60
   +1 <<SOURCE_CITATION>>                     {0:M}    p.39
   +1 <<NOTE_STRUCTURE>>                      {0:M}    p.37
```

The association pointer only associates ~~INDIvidual~~ INDI (individual) records to ~~INDIvidual~~ INDI (individual) records.

ASSO Usage

Usage of the ASSO record isn't explained here, but in the definition of <<INDIVIDUAL_RECORD>>. The description there briefly describes usage of the INDI.FAMC and the INDI.FAMS records, and then states:

> Other associations or relationships are represented by the ~~ASSOciation~~ ASSO (association) tag. The person's relation or association is the person being pointed to. The association or relationship is stated by the value on the subordinate RELA line. For example:
>
> 0 @I1@ INDI
> —1 NAME Fred/Jones/
> —1 ASSO @I2@
> ——2 RELA Godfather

ASSO.RELA <RELATION_IS_DESCRIPTOR>

The ASSO record is a little-used feature of the FamilySearch GEDCOM specification. One reason is that this feature seems hardly needed; non-familial relationships can be described in notes, and adding an actual association between records adds limited value to that. However, it can be handy to explicitly link individuals together, as the FamilySearch example indicates.

The <RELATION_IS_DESCRIPTOR> does not provide a list of standard values. The ASSO.RELA line value is free-form text that has meaning to the user, not the application.

best practice

- Applications should not assume any particular ASSO.RELA value has any particular meaning.
- Applications should issue a warning when an ASSO.RELA duplicates a familial relationship already in the tree (e.g. grandfather)

GEDCOM-L ASSO.RELA

The ASSO.RELA value should be assumed to be nothing but user-provided text, but it is worth noting that German developers working together through the GEDCOM-L mailing list have agreed to use the following values where appropriate:

- `Godparent`
- `Witness_of_Birth`
- `Witness_of_Death`

GEDCOM-L _ASSO.RELA

They also use the _ASSO record for relationship from an individual to a family, with the following agreed-upon _ASSO.RELA values:

- `Witness_of_Marriage`
- `Witness_of_Civil_Marriage`
- `Witness_of_Religious_Marriage`

discommended

Although earlier versions of GEDCOM featured support for witnesses, the FamilySearch GEDCOM 5.5.1 specification lacks support for witnesses. Various vendors solve that limitation through GEDCOM extensions. This ASSO and _ASSO records approach is strongly discommended.

Witnesses must be associated with the event they witnessed. This _ASSO record approach fails to associate witnesses with an event and because of that, cannot even tell you which marriage someone witnessed.
It is best to use some vendor-defined _WITN record on the event itself.

references

GenWiki: GEDCOM/ASSO-Tag

ASSO for ALIA

The ASSO record should be used for all relationships that aren't direct family relationships. FamilySearch GEDCOM 5.0 and later define the ALIA record for linking INDI records that are possible duplicates. There are two issues with that definition.
One issue is that FamilySearch GEDCOM 3.0 and 4.0 define the ALIA record as the way to document alternate names, and that reuse of the same tag for something else is problematic. In practice, vendors continued to use the ALIA record for alternate names, and no product ever used the ALIA record for linking one INDI record to another.
The other is that the 5.0+ ALIA record is superfluous, because the specification already provides the ASSO record.

best practice

The ALIA record (both uses) should be considered deprecated.
Possibly duplicate INDI records can be associated with each other using the ASSO record; `possible-duplicate` is a good ASSO.RELA value for that usage.

GEDCOM 5.5 ASSO.TYPE

The GEDCOM 5.5 specification of the <<ASSOCIATION_STRUCTURE>> includes a TYPE line, but that is an error, corrected through the GEDCOM 5.5 errata sheet.

CHANGE_DATE:=

n **CHAN**	{1:1}	
~~+1 **DATE** <CHANGE_DATE>~~	{1:1}	p.44
+1 **DATE** <DATE_EXACT>	{1:1}	p.45
+2 TIME <TIME_VALUE>	{0:1}	p.63
+1 <<NOTE_STRUCTURE>>	{0:M}	p.37

The change date is intended to only record the last change to a record. Some systems may want to manage the change process with more detail, but it is sufficient for GEDCOM purposes to indicate the last time that a record was modified.

<<CHANGE_DATE>> versus <CHANGE_DATE>

The <<CHANGE_DATE>> includes a reference to <CHANGE_DATE>, but this is *not* a recursive reference to this same definition (page 31).
The second <CHANGE_DATE> links to another <CHANGE_DATE> definition on another page.

All that second <CHANGE_DATE> definition on page 44 states is that <CHANGE_DATE> is a <DATE_EXACT> (defined on page 45) which must be either 10 or 11 characters long:

CHANGE_DATE:= {Size=1:11}

<DATE_EXACT>
The date that this data was changed.

specification error

Having two different definitions of the same thing is a specification error. Giving the same name to two different things is a specification error too. Having both a <<CHANGE_DATE>> and a <CHANGE_DATE> definition is a specification error.

The correction applied here, not using the second <CHANGE_DATE> definition, but simply specifying <DATE_EXACT> directly instead, resolves the specification error.

CHILD_TO_FAMILY_LINK:=

n **FAMC** @<XREF:FAM>@	{1:1}	p.24

31

32

+1 PEDI <PEDIGREE_LINKAGE_TYPE>	{0:1}	p.57
+1 STAT <CHILD_LINKAGE_STATUS>	{0:1}	p.44
+1 <<NOTE_STRUCTURE>>	{0:M}	p.37

GEDCOM 5.5 PEDI multiplicity

The GEDCOM 5.5 specification erroneously states that the multiplicity of the <PEDI> subrecord is {0:M} instead of {0:1}.
That error was corrected through the GEDCOM 5.5 errata sheet.

EVENT_DETAIL:=

n TYPE <EVENT_OR_FACT_CLASSIFICATION>	{0:1}	p.49
n DATE <DATE_VALUE>	{0:1}	p.47, 46
n <<PLACE_STRUCTURE>>	{0:1}	p.38
n <<ADDRESS_STRUCTURE>>	{0:1}	p.31
n AGNC <RESPONSIBLE_AGENCY>	{0:1}	p.60
n RELI <RELIGIOUS_AFFILIATION>	{0:1}	p.60
n CAUS <CAUSE_OF_EVENT>	{0:1}	p.43
n RESN <RESTRICTION_NOTICE>	{0:1}	p.60
n <<NOTE_STRUCTURE>>	{0:M}	p.37
n <<SOURCE_CITATION>>	{0:M}	p.39
n <<MULTIMEDIA_LINK>>	{0:M}	p.37, 26

EVENT.AGE

GEDCOM 5.5 allows an EVENT.AGE. There is no such thing as the age of an event during the event, so GEDCOM 5.5.1 no longer allows this.
To be more precise, the AGE record has been moved, from <EVENT_DETAIL> to <NDIVIDUAL_EVENT_DETAIL>.

erroneous GEDCOM 5.5 erratum

The GEDCOM 5.5 errata sheet specifies that two lines should be added to the <EVENT_DETAIL> definition, below the <<SOURCE_CITATION>> line, like this:

```
+ 1 <<SOURCE_CITATION>>
    + 2 <<NOTE_STRUCTURE>>
    + 2 <<MULTIMEDIA_LINK>>
```

These additional lines are not present in GEDCOM 5.5.1 specification, because *the GEDCOM 5.5 errata sheet is wrong*!
The <<SOURCE_CITATION>> definition already includes the <<NOTE_STRUCTURE>> and <<MULTIMEDIA_LINK>>, so there is no need (and it would even be wrong) to include them here as well.

FAMILY_EVENT_DETAIL:=

```
n HUSB                                              {0:1}
   +1 AGE <AGE_AT_EVENT>                            {1:1}     p.42
n WIFE                                              {0:1}
   +1 AGE <AGE_AT_EVENT>                            {1:1}     p.42
n <<EVENT_DETAIL>>                                  {0:1}     p.32
```

FAMILY_EVENT_STRUCTURE:=

```
[
n [ ANUL | CENS | DIV | DIVF ]                      {1:1}
   +1 <<FAMILY_EVENT_DETAIL>>                       {0:1}     p.32
|
n [ ENGA | MARB | MARC ]                            {1:1}
+1 <<FAMILY_EVENT_DETAIL>>                          {0:1}     p.32
|
n MARR [Y|<NULL>]                                   {1:1}
   +1 <<FAMILY_EVENT_DETAIL>>                       {0:1}     p.32
|
n [ MARL | MARS ]                                   {1:1}
+1 <<FAMILY_EVENT_DETAIL>>                          {0:1}     p.32
|
n RESI
   +1 <<FAMILY_EVENT_DETAIL>>                       {0:1}     p.32
|
n EVEN [<EVENT_DESCRIPTOR> | <NULL>]                {1:1}     p.48
   +1 <<FAMILY_EVENT_DETAIL>>                       {0:1}     p.32
]
```

32

33

residence isn't an event but an attribute

This <FAMILY_EVENT_STRUCTURE> definition includes RESI (residence) as a possible event. However, the residences of a family aren't events, they are attributes. A building has an age, but a residence has a period. Moving in and moving out are events, typically happening on a single day, while having residence is an attribute, typical valid for period of many years.

The miscategorisation of RESI as an event is a hold-over from earlier GEDCOM version. The FACT record was only introduced in GEDCOM 5.5.1. Prior to GEDCOM 5.5.1, FamilySearch did not really distinguish between events and attributes.

INDIVIDUAL_ATTRIBUTE_STRUCTURE:=

```
[
n CAST <CASTE_NAME>                                              {1:1}     p.43
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n DSCR <PHYSICAL_DESCRIPTION>                                    {1:1}     p.58
   +1 [CONC | CONT ] <PHYSICAL_DESCRIPTION>                      {0:M}     p.58
+1 <<INDIVIDUAL_EVENT_DETAIL>>                                   {0:1}*    p.34
|
n EDUC <SCHOLASTIC_ACHIEVEMENT>                                  {1:1}     p.61
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n IDNO <NATIONAL_ID_NUMBER>                                      {1:1}     p.56
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n NATI <NATIONAL_OR_TRIBAL_ORIGIN>                               {1:1}     p.56
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n NCHI <COUNT_OF_CHILDREN>                                       {1:1}     p.44
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n NMR <COUNT_OF_MARRIAGES>                                       {1:1}     p.44
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n OCCU <OCCUPATION>                                              {1:1}     p.57
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n PROP <POSSESSIONS>                                             {1:1}     p.59
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n RELI <RELIGIOUS_AFFILIATION>                                   {1:1}     p.60
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n RESI/* Resides at */                                           {1:1}
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n SSN <SOCIAL_SECURITY_NUMBER>                                   {1:1}     p.61
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n TITL <NOBILITY_TYPE_TITLE>                                     {1:1}     p.57
```

```
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
|
n FACT <ATTRIBUTE_DESCRIPTOR>                                    {1:1}     p.43
   +1 <<INDIVIDUAL_EVENT_DETAIL>>                                {0:1}*    p.34
]
```

* *Note*: The usage of IDNO or the FACT tag require that a subordinate TYPE tag be used to define what kind of identification number or fact classification is being defined. The TYPE tag can be used with each of the above tags used in this structure.

INDIVIDUAL_EVENT_DETAIL:=

```
    n <<EVENT_DETAIL>>                                        {1:1}      p.32
    n AGE <AGE_AT_EVENT>                                      {0:1}      p.42
```

attributes do not have age

This FamilySearch GEDCOM 5.5.1 specification uses <INDIVIDUAL_EVENT_DETAIL> for both attributes and events. The inclusion of the AGE subrecord makes sense for events, but not for attributes. Some attributes (for example residence and occupation) have *periods* associated with them.
The use of <INDIVIDUAL_EVENT_DETAIL> for attributes is a specification error, attributes should use a separate <INDIVIDUAL_ATTRIBUTE_DETAIL>.

INDIVIDUAL_EVENT_STRUCTURE:=

```
    [
    n [ BIRT | CHR ] [Y|<NULL>]                               {1:1}
       +1 <<INDIVIDUAL_EVENT_DETAIL>>                         {0:1}*     p.34
       +1 FAMC @<XREF:FAM>@                                   {0:1}      p.24
    |
    n DEAT [Y|<NULL>]                                         {1:1}
       +1 <<INDIVIDUAL_EVENT_DETAIL>>                         {0:1}*     p.34
    |
    n [ BURI | CREM ]                                         {1:1}
    +1 <<INDIVIDUAL_EVENT_DETAIL>>                            {0:1}*     p.34
    |
    n ADOP                                                    {1:1}
       +1 <<INDIVIDUAL_EVENT_DETAIL>>                         {0:1}*     p.34
       +1 FAMC @<XREF:FAM>@                                   {0:1}      p.24
          +2 ADOP <ADOPTED_BY_WHICH_PARENT>                   {0:1}      p.42
    |
    n [ BAPM | BARM | BASM | BLES ]                           {1:1}
    +1 <<INDIVIDUAL_EVENT_DETAIL>>                            {0:1}*     p.34
    |
    n [ CHRA | CONF | FCOM | ORDN ]                           {1:1}
       +1 <<INDIVIDUAL_EVENT_DETAIL>>                         {0:1}*     p.34
    |
    n [ NATU | EMIG | IMMI ]                                  {1:1}
       +1 <<INDIVIDUAL_EVENT_DETAIL>>                         {0:1}*     p.34
    |
    n [ CENS | PROB | WILL]                                   {1:1}
       +1 <<INDIVIDUAL_EVENT_DETAIL>>                         {0:1}*     p.34
    |
```

34

35

```
    n [ GRAD | RETI ]                                         {1:1}
       +1 <<INDIVIDUAL_EVENT_DETAIL>>                         {0:1}*     p.34
    |
    ~~n EVEN~~                                              {1:1}
       ~~+1 <<INDIVIDUAL_EVENT_DETAIL>>~~                  {0:1}*    p.34

    n EVEN [<EVENT_DESCRIPTOR> | <NULL>]                    {1:1}     p.48
    +1 <<INDIVIDUAL_EVENT_DETAIL>>                          {0:1}*    p.34
```

]

As a general rule, events are things that happen on a specific date. Use the date form ‘BET date AND date ’ to indicate that an event took place at some time between two dates. Resist the temptation to use a ‘FROM date TO date ’ form in an event structure. If the subject of your recording occurred over a period of time, then it is probably not an event, but rather an attribute or fact.

The EVEN tag in this structure is for recording general events that are not shown in the above <<INDIVIDUAL_EVENT_STRUCTURE>>. The event indicated by this general EVEN tag is defined by the value of the subordinate TYPE tag. For example, a person that signed a lease for land dated October 2, 1837 and a lease for equipment dated November 4, 1837 would be written in GEDCOM as:~~:~~

```
1 EVEN
  2 TYPE Land Lease
  2 DATE 2 OCT 1837
1 EVEN
  2 TYPE Equipment Lease
  2 DATE 4 NOV 1837
```

The TYPE tag can be optionally used to modify the basic understanding of its superior event or attribute. For example:

```
1 GRAD
  2 TYPE College
```

The occurrence of an event is asserted by the presence of either a DATE tag and value or a ~~PLACe~~ PLAC (place) tag and value in the event structure. When neither the date value nor the place value are known then a ~~Y(es)~~ Y (yes) value on the parent event tag line is required to assert that the event happened. For example each of the following GEDCOM structures assert that a death happened:

```
1 DEAT Y
1 DEAT
  2 DATE 2 OCT 1937
1 DEAT
2 PLAC Cove, Cache, Utah
```

Using this convention, as opposed to the just the presence of the tag, protects GEDCOM processors which removes (prunes) lines which have neither a value nor any subordinate line. It also allows a note or source to be attached to an event context without implying that the event occurred.

It is not proper GEDCOM form to use a ~~N(o)~~ N (no) value with an event tag to infer that it did not happen. A convention to handle events which never happened may be defined in the future.

INDIVIDUAL_EVENT_STRUCTURE

The FamilySearch GEDCOM 5.5.1 definition of <<INDIVIDUAL_EVENT_STRUCTURE>> is wrong; to wit, it is missing the <EVENT_DESCRIPTOR> value.
The optional EVENT_DESCRIPTOR value *is* present in the <<FAMILY_EVENT_STRUCTURE>>.
The omission of the <EVENT_DESCRIPTOR> value from the <<INDIVIDUAL_EVENT_STRUCTURE>> is an editing mistake made upon the introduction of FACT and FACT.TYPE in GEDCOM 5.5.1.
This is most clearly evidenced by the example given for <EVENT_DESCRIPTOR>, clearly an INDI.EVEN example, which *does* include an <EVENT_DESCRIPTOR> value:

```
 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
```

The FamilySearch GEDCOM 5.5.1 definition is:

```
INDIVIDUAL_EVENT_DETAIL:=

   ...
   |
   n EVEN                                       {1:1}
     +1 <<INDIVIDUAL_EVENT_DETAIL>>             {0:1}*     p.34
   ]
```

but should be:

```
INDIVIDUAL_EVENT_DETAIL:=

   ...
   |
   n EVEN [<EVENT_DESCRIPTOR> | <NULL>]         {1:1}      p.48
     +1 <<INDIVIDUAL_EVENT_DETAIL>>             {0:1}      p.34
   ]
```

This specification error was first noted by Keith Riggle and mentioned in several software reviews. It was documented by Tim Forsythe in his 22 Dec 2015 GigaTrees blog post *A New GEDCOM 5.5.1 Wrinkle*, but that blog post is no longer available. It is now documented in Keith Riggle's own blog post *The Event Structure in GEDCOM Files*.

references

GenealogyTools: The Event Structure in GEDCOM Files

BIRT Y, CHR Y, DEAT Y

The BIRT (birth), CHR (christening), and DEAT (death) dates may be unknown. The line value of these records may be empty (indicated by `<NULL>` in the definition), but, as stated in chapter 1, records must have value; the line value of most records may only be empty if it has subrecords. If there are no subrecords, the line value `Y` must be used.

The line `1 CHR Y` means *christened* (but do not know where or when), and the line `1 DEAT Y` means *died* (but do not know where or when).
That the lineage-linked form also allows the line `1 BIRT Y` is a bit puzzling; it means *born* (but do not know where or when), and thus conveys exactly the same information as when that line isn't present.

The GEDCOM specification should not allow `1 BIRT Y`. Consider this deprecated.

LDS_INDIVIDUAL_ORDINANCE:=

```
[
n [ BAPL | CONL ]                                       {1:1}
   +1 DATE <DATE_LDS_ORD>                               {0:1}     p.46
   +1 TEMP <TEMPLE_CODE>                                {0:1}     p.63
   +1 PLAC <PLACE_LIVING_ORDINANCE>                     {0:1}     p.58
   +1 STAT <LDS_BAPTISM_DATE_STATUS>                    {0:1}     p.51
~~     +2 DATE <CHANGE_DATE>~~                          {1:1}     p.44
     +2 DATE <DATE_EXACT>                               {1:1}     p.45
   +1 <<NOTE_STRUCTURE>>                                {0:M}     p.37
  +1 <<SOURCE_CITATION>>                                {0:M}     p.39
|
n ENDL                                                  {1:1}
   +1 DATE <DATE_LDS_ORD>                               {0:1}     p.46
   +1 TEMP <TEMPLE_CODE>                                {0:1}     p.63
   +1 PLAC <PLACE_LIVING_ORDINANCE>                     {0:1}     p.58
   +1 STAT <LDS_ENDOWMENT_DATE_STATUS>                  {0:1}     p.52
~~     +2 DATE <CHANGE_DATE>~~                          {1:1}     p.44
     +2 DATE <DATE_EXACT>                               {1:1}     p.45
   +1 <<NOTE_STRUCTURE>>                                {0:M}     p.37
   +1 <<SOURCE_CITATION>>                               {0:M}     p.39
|
n SLGC                                                  {1:1}
   +1 DATE <DATE_LDS_ORD>                               {0:1}     p.46
   +1 TEMP <TEMPLE_CODE>                                {0:1}     p.63
   +1 PLAC <PLACE_LIVING_ORDINANCE>                     {0:1}     p.58
   +1 FAMC @<XREF:FAM>@                                 {1:1}     p.24
   +1 STAT <LDS_CHILD_SEALING_DATE_STATUS>              {0:1}     p.51
~~     +2 DATE <CHANGE_DATE>~~                          {1:1}     p.44
     +2 DATE <DATE_EXACT>                               {1:1}     p.45
   +1 <<NOTE_STRUCTURE>>                                {0:M}     p.37
   +1 <<SOURCE_CITATION>>                               {0:M}     p.39
]
```

<<LDS_INDIVIDUAL_ORDINANCE>> <CHANGE_DATE>

There are two conflicting <CHANGE_DATE> in GEDCOM. The first <<CHANGE_DATE>> definition defines the CHAN record and its subrecords. The second <CHANGE_DATE> definition resolves to <DATE_EXACT>.

This <<LDS_INDIVIDUAL_ORDINANCE>> definition and the <<LDS_SPOUSE_SEALING>> definition directly below use the second <CHANGE_DATE> definition.
This *Annotated Edition* avoids confusion by obsoleting that second definition. The references to the <CHANGE_DATE> have been replaced with direct references to <DATE_EXACT>.

LDS_SPOUSE_SEALING:=

```
n SLGS                                                  {1:1}
   +1 DATE <DATE_LDS_ORD>                               {0:1}     p.46
   +1 TEMP <TEMPLE_CODE>                                {0:1}     p.63
   +1 PLAC <PLACE_LIVING_ORDINANCE>                     {0:1}     p.58
```

```
  +1 STAT <LDS_SPOUSE_SEALING_DATE_STATUS>          {0:1}   p.52
~~    +2 DATE <CHANGE_DATE>~~                           {1:1}   p.44
    +2 DATE <DATE_EXACT>                            {1:1}   p.45
  +1 <<NOTE_STRUCTURE>>                             {0:M}   p.37
  +1 <<SOURCE_CITATION>>                            {0:M}   p.39
```

LDS_SPOUSE_SEALING CHANGE_DATE

See the *<<LDS_INDIVIDUAL_ORDINANCE>> <CHANGE_DATE>* annotation above.

MULTIMEDIA_LINK:=

```
[
n OBJE @<XREF:OBJE>@                                {1:1}   p.26
|
n OBJE
~~  +1 FILE <MULTIMEDIA_FILE_REFN>~~                    {1:M}   p.54
  +1 FILE <MULTIMEDIA_FILE_REFERENCE>               {1:M}   p.54
    +2 FORM <MULTIMEDIA_FORMAT>                     {1:1}   p.54
      +3 MEDI <SOURCE_MEDIA_TYPE>                   {0:1}   p.62
  +1 TITL <DESCRIPTIVE_TITLE>                       {0:1}   p.48
]
```

Note: some systems may have output the following 5.5 structure. The new context above was introduced in order to allow a grouping of related multimedia files to a particular context.

```
n OBJE
  +1 FILE <MULTIMEDIA_FILE_REFERENCE>
  +1 FORM <MULTIMEDIA_FORMAT>
~~    +2 MEDI <SOURCE_MEDIA_TYPE>~~
  +1 TITLE <DESCRIPTIVE_TITLE>
  +1 <<NOTE_STRUCTURE>>
```

non-existent <MULTIMEDIA_FILE_REFN>

The <<MULTIMEDIA_LINK>> syntax references <MULTIMEDIA_FILE_REFN>, but the GEDCOM lineage-linked form doesn't define <MULTIMEDIA_FILE_REFN>, it defines <MULTIMEDIA_FILE_REFERENCE>.

GEDCOM 5.5 <MULTIMEDIA_LINK> definition

Here's the GEDCOM 5.5 <MULTIMEDIA_LINK> definition for comparison:

MULTIMEDIA_LINK:=

```
[           /* embedded form */
n OBJE <XREF:OBJE>@                             {1:1}
|           /* linked form */
n OBJE
  +1 FORM <MULTIMEDIA_FORMAT>                   {1:1}
  +1 TITL <DESCRIPTIVE_TITLE>                   {0:1}
  +1 FILE <MULTIMEDIA_FILE_REFERENCE>           {1:M}
  +1 <<NOTE_STRUCTURE>>                         {0:M}
]
```

> This structure provides two options in handling the GEDCOM multimedia interface. The first alternative (embedded) includes all of the data, including the multimedia object, within the transmission file. The embedded method includes pointers to GEDCOM records that contain encoded image or sound objects. Each record represents a multimedia object or object fragment. An object fragment is created by breaking the multimedia files into several multimedia object records of 32K or less. These fragments are tied together by chaining from one multimedia object fragment to the next in sequence. This procedure will help manage the size of a multimedia GEDCOM record so that existing systems which are not expecting large multimedia records may discard the records without crashing due to the size of the record. Systems which handle embedded multimedia can reconstitute the multimedia fragments by decoding the object fragments and concatenating them to the assigned multimedia file.
>
> The second method allows the GEDCOM context to be connected to an external multimedia file. This process is only managed by GEDCOM in the sense that the appropriate file name is included in the GEDCOM file in context, but the maintenance and transfer of the multimedia files are external to GEDCOM.

two <<MULTIMEDIA_LINK>> forms

The GEDCOM 5.5.1 <<MULTIMEDIA_LINK>> definition allows two alternative forms, the same two forms that the GEDCOM 5.5 definition allow. The first form, called the *embedded form*, uses an @<XREF:OBJE>@, a link to a <<MULTIMEDIA_RECORD>>, while the second form, called the *linked form*, contains details for an external multimedia file.

The GEDCOM 5.5 <<MULTIMEDIA_LINK>> definition makes it clear that the first form is for embedded multimedia objects only, and GEDCOM 5.5.1 does not support embedded multimedia objects, so you might think that this form is obsolete now, but it is not.
On the contrary, it is the preferred form.

The *linked form* of the GEDCOM 5.5.1 <<MULTIMEDIA_LINK>> record offers a subset of the GEDCOM 5.5.1 <<MULTIMEDIA_RECORD>> features. That makes it superfluous.

best practice

It is recommended that all multimedia files be recorded in GEDCOM 5.5.1 using the top-level OBJE record, and that all instances of the <MULTIMEDIA_LINK> record are @<XREF:OBJE>@ links to these <<MULTIMEDIA_RECORD>> records.
Consider the so-called *linked form* of the <<MULTIMEDIA_LINK>> record deprecated.

GEDCOM 5.5.1 misrepresents GEDCOM 5.5 <<MULTIMEDIA_LINK>>

This GEDCOM 5.5.1 <<MULTIMEDIA_LINK>> definition claims that the (linked form of the) GEDCOM 5.5 <<MULTIMEDIA_LINK>> definition (after the indicated corrections) looks like this:

```
MULTIMEDIA_LINK:=

    n OBJE
      +1 FILE <MULTIMEDIA_FILE_REFERENCE>              {1:M}
      +1 FORM <MULTIMEDIA_FORMAT>                      {1:1}
        +2 MEDI <SOURCE_MEDIA_TYPE>                    {0:1}
      +1 TITL <DESCRIPTIVE_TITLE>                      {0:1}
```

However, that is not correct. The linked form of the GEDCOM 5.5 <<MULTIMEDIA_LINK>> definition looks like this:

```
MULTIMEDIA_LINK:=
  n OBJE
     +1 FORM <MULTIMEDIA_FORMAT>                         {1:1}
     +1 TITL <DESCRIPTIVE_TITLE>                         {0:1}
     +1 FILE <MULTIMEDIA_FILE_REFERENCE>                 {1:M}
     +1 <<NOTE_STRUCTURE>>                               {0:M}
```

The GEDCOM 5.5 definition does not include a MEDI subrecord at all, neither as direct subrecord of the OBJE record, nor as a subrecord of the OBJE.FORM record. The GEDCOM 5.5 definition does allow notes, while the GEDCOM 5.5.1 definition does not.

The misrepresentation of the GEDCOM 5.5 syntax seems an unintentional editing error. The two differences between the GEDCOM 5.5 and 5.5.1 definitions that the author probably wanted to draw the readers attention to, are:

- The GEDCOM 5.5 record allows just one file, while the GEDCOM 5.5.1 record allows a group of files
- The GEDCOM 5.5 record has the FORM record as a direct subrecord of the OBJE record, while the GEDCOM 5.5.1 record has the FORM record as a subrecord of the OBJE.FILE record.

syntax exception

The real point of the remark is to demand a syntax exception for “some systems” that use GEDCOM 5.5.1, but still create MULTIMEDIA_LINK records in GEDCOM 5.5 format. The implied demand is that GEDCOM 5.5.1 readers should accept GEDCOM 5.5 MULTIMEDIA_LINK records, even though they are not part of GEDCOM 5.5.1.
See the *GEDCOM 5.5 MULTIMEDIA_LINK clairvoyance* annotation below for further observations and best practices.

GEDCOM 5.5 <<MULTIMEDIA_LINK>> clairvoyance

The *FamilySearch GEDCOM 5.5.1 specification* states that some systems “may have output” (past tense) the GEDCOM 5.5 structure within GEDCOM 5.5.1 files. That sounds like a helpful annotation, but is also a rather remarkable one.
On the day that FamilySearch introduced GEDCOM 5.5.1, FamilySearch already knew that some applications would use GEDCOM 5.5.1, yet continue to use the GEDCOM 5.5 <MULTIMEDIA_LINK> structure instead of GEDCOM 5.5.1 <MULTIMEDIA_LINK> structure?
Unless the FamilySearch GEDCOM editors own a time machine, something is wrong with their claim, or at least their presentation of it.

alternative format

What it really says here, without being open and honest about it, is that third parties are expected to accommodate FamilySearch's unwillingness to follow their own specification in their own product!
We now know that FamilySearch's Personal Ancestral File 5.x uses GEDCOM 5.5.1, but still uses the GEDCOM 5.5 <MULTIMEDIA_LINK> structure. So, this FamilySearch statement tells us that, when they were still writing the GEDCOM 5.5.1 specification, they had already decided that PAF 5 would use GEDCOM 5.5.1, but keep using the GEDCOM 5.5 <MULTIMEDIA_LINK> structure, and then put this note about “some systems” in to get third parties to support PAF's behaviour.

GEDCOM writer best practice

- Use GEDCOM 5.5.1 structures within GEDCOM 5.5.1 files.
- Do not use structures from other GEDCOM versions within GEDCOM 5.5.1 (unless they are identical).

GEDCOM reader best practice

Because PAF will never be updated again, and still has significant market share, third parties should indeed accept the GEDCOM 5.5 <MULTIMEDIA_LINK> structure within GEDCOM 5.5.1 files - but still issue an error for each occurrence.

NOTE_STRUCTURE:=

```
[
n NOTE @<XREF:NOTE>@                                    {1:1}    p.27
|
n NOTE [<SUBMITTER_TEXT> | <NULL>]                      {1:1}    p.63
   +1 [CONC|CONT] <SUBMITTER_TEXT>                      {0:M}
]
```

Note: There are special considerations required when using the CONC tag. The usage is to provide a note string that can be concatenated together so that the display program can do its own word wrapping according to its display window size. The requirement for usage is to either break the text line in the middle of a word, or if at the end of a word, to add a space to the first of the next CONC line. Otherwise most operating systems will strip off the trailing space and the space is lost in the reconstitution of the note.

NOTE.SOUR.NOTE.SOUR…

The GEDCOM 5.5 errata sheet states that an optional reference to a SOUR record should be added to the NOTE_STRUCTURE definition, like this:

```
[
n NOTE @<XREF:NOTE>@ {1:1}
    +1 SOUR @<XREF:SOUR>@
|
n NOTE [<SUBMITTER_TEXT> | <NULL>] {1:1}
    +1 [CONC|CONT] <SUBMITTER_TEXT> {0:M}
    +1 SOUR @<XREF:SOUR>@
]
```

The GEDCOM 5.5.1 specification does *not* include the additional lines.

loop

FamilySearch probably realised that this addition was a bad idea, because it allowed notes to have sources, which may have notes, which may sources, and so on, *ad infinitum*, and thus includes the possibility of a loop.

GEDCOM 5.5.1 allows source citations on top-level NOTE records, and allows notes on source records, but does not allow source citation on those SOUR.NOTE note records.

PERSONAL_NAME_PIECES:=

```
n NPFX <NAME_PIECE_PREFIX>                              {0:1}    p.55
n GIVN <NAME_PIECE_GIVEN>                               {0:1}    p.55
n NICK <NAME_PIECE_NICKNAME>                            {0:1}    p.55
n SPFX <NAME_PIECE_SURNAME_PREFIX                       {0:1}    p.56
```

n SURN <NAME_PIECE_SURNAME>	{0:1}	p.55
n NSFX <NAME_PIECE_SUFFIX>	{0:1}	p.55
n <<NOTE_STRUCTURE>>	{0:M}	p.37
n <<SOURCE_CITATION>>	{0:M}	p.39

erroneous GEDCOM 5.5 erratum

The GEDCOM 5.5 errata sheet specifies that two lines should be added to the <<PERSONAL_NAME_STRUCTURE>> definition, below the <<SOURCE_CITATION>> line (now moved into the <<PERSONAL_NAME_PIECES>> definition), like this:

```
+ 1 <<SOURCE_CITATION>>
    + 2 <<NOTE_STRUCTURE>>
   + 2 <<MULTIMEDIA_LINK>>
```

These additional lines are not present in GEDCOM 5.5.1 specification, because *the GEDCOM 5.5 errata sheet is wrong*!
The <<SOURCE_CITATION>> definition already includes the <<NOTE_STRUCTURE>> and <<MULTIMEDIA_LINK>>, so there is no need (and indeed, it would be wrong) to include them here.

Rufnamen

.

It is odd that the even version 5.5.1 of the GEDCOM specification does not support call names. Many Americans have German ancestors and in Germany, people do not just have call names, their rufnamen (call names) are *official*, recorded in official documents. German genealogy software uses the _RUFNAME record to record these call names.

references

GenWiki: GEDCOM/NAME-Tag

PERSONAL_NAME_STRUCTURE:=

n **NAME** <NAME_PERSONAL>	{1:1}	p.54
+1 TYPE <NAME_TYPE>	{0:1}	p.56
+1 <<PERSONAL_NAME_PIECES>>	{0:1}	p.37
+1 FONE <NAME_PHONETIC_VARIATION>	{0:M}	p.55
+2 TYPE <PHONETIC_TYPE>	{1:1}	p.57
+2 <<PERSONAL_NAME_PIECES>>	{0:1}	p.37
+1 ROMN <NAME_ROMANIZED_VARIATION>	{0:M}	p.56
+2 TYPE <ROMANIZED_TYPE>	{1:1}	p.61
+2 <<PERSONAL_NAME_PIECES>>	{0:1}	p.37

The name value is formed in the manner the name is normally spoken, with the given name and family name (surname) separated by slashes (/). (See <NAME_PERSONAL>, page 54.) Based on the dynamic nature or unknown compositions of naming conventions, it is difficult to provide more detailed name piece structure to handle every case. The NPFX, GIVN, NICK, SPFX, SURN, and NSFX tags are provided optionally for systems that cannot operate effectively with less structured information. 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.

A <NAME_TYPE> is used to specify the particular variation that this name is. For example; if the name type is subordinate to the <NAME_PERSONAL> it could indicate that this name is a name taken at immigration or that it could be an ‘also known as’ name (see page 56.)

Future GEDCOM releases (6.0 or later) will likely apply a very different strategy to resolve this problem, possibly using a sophisticated parser and a name-knowledge database.

PLACE_STRUCTURE:=

n **PLAC** <PLACE_NAME>	{1:1}	p.58
+1 FORM <PLACE_HIERARCHY>	{0:1}	p.58

38

39

+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

place record design error

The FamilySearch GEDCOM 5.5.1 specification added the place coordinates, the LATI and LONG records, as subrecords of the PLAC record. That sounds reasonable and logical, there is no place (pun intended) to put these. However, the PLAC record is itself a subrecord, and a typical GEDCOM file contains many place names over and over again.

The GEDCOM 5.5.1 design seems to expect that GEDCOM writers, when place coordinates are known, include those coordinates with each occurrence of each place name, thus bloating GEDCOM files even more than frequently repeated place names already do.
Worse than that, GEDCOM readers that come across the same place name more than once, may come across conflicting coordinates, and cannot resolve that by keeping all the different ones; this PLAC record allows just one set of coordinates.

top-level place record

Genealogy applications developers have long desired the introduction of a top-level place record, and the introduction of place coordinates in GEDCOM 5.5.1 made a top-level place record a practical necessity, yet it was still not included.

top-level _PLAC record

Several developers have addressed this design error by introducing a top-level PLAC record, and do not record place coordinates within PLAC records, but only within _PLAC records. Different vendors have different top-level PLAC records, but match PLAC record with those _PLAC records the same way: a PLAC and _PLAC record match each other if they contain exactly the same place name string.

top-level _LOC record

Several German developers created a collective set of extensions called GEDCOM EL (Extended Locations). The GEDCOM EL extension were originally created for GEDCOM 5.5, but are also used with GEDCOM 5.5.1.
GEDCOM EL uses a top-level _LOC record.

references

- Gaenovium Presentations: Louis Kessler: Reading Wrong GEDCOM Right
- Behold blog 2011 Dec 24: The Place Record in GEDCOM
- Detecting GEDCOM EL
- GenWiki: GEDCOM 5.5EL

SOURCE_CITATION:=

```
[ /* pointer to source record (preferred)*/
n SOUR @<XREF:SOUR>@                                   {1:1}   p.27
   +1 PAGE <WHERE_WITHIN_SOURCE>                       {0:1}   p.64
   +1 EVEN <EVENT_TYPE_CITED_FROM>                     {0:1}   p.49
      +2 ROLE <ROLE_IN_EVENT>                          {0:1}   p.61
   +1 DATA                                             {0:1}
      +2 DATE <ENTRY_RECORDING_DATE>                   {0:1}   p.48
      +2 TEXT <TEXT_FROM_SOURCE>                       {0:M}   p.63
         +3 [CONC|CONT] <TEXT_FROM_SOURCE>             {0:M}
   +1 <<MULTIMEDIA_LINK>>                              {0:M}   p.37, 26
   +1 <<NOTE_STRUCTURE>>                               {0:M}   p.37
   +1 QUAY <CERTAINTY_ASSESSMENT>                      {0:1}   p.43

| /* Systems not using source records */
n SOUR <SOURCE_DESCRIPTION>                            {1:1}   p.61
   +1 [CONC|CONT] <SOURCE_DESCRIPTION>                 {0:M}
   +1 TEXT <TEXT_FROM_SOURCE>                          {0:M}   p.63
      +2 [CONC|CONT] <TEXT_FROM_SOURCE>                {0:M}
   +1 <<MULTIMEDIA_LINK>>                              {0:M}   p.37, 26
   +1 <<NOTE_STRUCTURE>>                               {0:M}   p.37
   +1 QUAY <CERTAINTY_ASSESSMENT>                      {0:1}   p.43
]
```

The data provided in the <<SOURCE_CITATION>> structure is source-related information specific to the data being cited. (See GEDCOM examples starting on page 74.) Systems that do not use a (<<SOURCE_RECORD>>) must use the non-preferred second ~~SOURce~~ SOUR (source) citation structure option. When systems that support the zero level source record format encounters a source citation that does not contain pointers to source records, then that system needs to create a <<SOURCE_RECORD>> format and store the source description information found in the non-structured source citation in the title area for the new source record.

The information intended to be placed in the citation structure includes:

! The pointer to the <<SOURCE_RECORD>>, which contains a more general description of the source used for the fact being cited.

! Information, such as a page number, to help the user find the cited data within the referenced source. This is stored in the ".SOUR.PAGE" tag context.

! Actual text from the source that was used in making assertions, for example a date phrase as actually recorded in the source, or significant notes written by the recorder, or an applicable sentence from a letter. This is stored in the ".SOUR.DATA.TEXT" tag context.

! Data that allows an assessment of the relative value of one source over another for making the recorded assertions (primary or secondary source, etc.). Data needed for this assessment is data that would help determine how much time from the date of the asserted fact and when the source was actually recorded, what type of event was cited, and what type of role did this person have in the cited source.

- Date when the entry was recorded in source document is stored in the ".SOUR.DATA.TEXT" tag context.
- The type of event that initiated the recording is stored in the “.SOUR.EVEN” tag context. The value used is the event code taken from the table of choices shown in the <EVENT_TYPE_CITED_FROM> primitive on page 49.
- The role of this person in the event is stored in the ".SOUR.EVEN.ROLE" context.

SOURCE_REPOSITORY_CITATION:=

n **REPO** [@<XREF:REPO>@ | <NULL>]	{1:1}	p.27
+1 <<NOTE_STRUCTURE>>	{0:M}	p.37
+1 CALN <SOURCE_CALL_NUMBER>	{0:M}	p.61
+2 MEDI <SOURCE_MEDIA_TYPE>	{0:1}	p.62

This structure is used within a source record to point to a name and address record of the holder of the source document. Formal and informal repository name and addresses are stored in the <<REPOSITORY_RECORD>>. Informal repositories include owner's of an unpublished work or of a rare published source, or a keeper of personal collections. An example would be the owner of a family bible containing unpublished family genealogical entries. More formal repositories, such as the Family History Library, should show a call number of the source at that repository. The call number of that source should be recorded using a subordinate CALN tag. Systems which do not use repository name and address record, should describe where the information cited is stored in the <<NOTE_STRUCTURE>> of the ~~REPOsitory~~ REPO (repository) source citation structure.

<<SOURCE_REPOSITORY_CITATION>> may be <NULL>

The <<SOURCE_REPOSITORY_CITATION>> definition is the only one in the entire FamilySearch GEDCOM specification that allows a pointer value to be absent. This may seem a design error, but it isn't necessary to identify a repository for a widely available published work.

best practice

Applications should encourage users to include a repository with every citation, and offer a “widely available work” checkbox, that stops the application from asking again.

SPOUSE_TO_FAMILY_LINK:=

n **FAMS** @<XREF:FAM>@	{1:1}	p.24
+1 <<NOTE_STRUCTURE>>	{0:M}	p.37

Primitive Elements of the Lineage-Linked Form

The field sizes show the minimum recommended field length within a database that is constrained to fixed length fields. The field sizes are in addition to the GEDCOM level and tag overhead. GEDCOM lines are limited to 255 characters. However, the ~~CONCatenation~~ CONC (concatenation) or ~~CONTinuation~~ CONT (continuation) tags can be used to expand a field beyond this limit. CONT line implies that a new line should appear to preserve formatting. CONC implies concatenation to the previous line without a new line. This is used so that a text note or description can be processed (word wrapped) in a text window without fixed carriage returns. The CONT and CONC tags are being used to extend specified textual values.

ADDRESS_CITY:= {Size=1:60}

The name of the city used in the address. Isolated for sorting or indexing.

ADDRESS_COUNTRY:= {Size=1:60}

The name of the country that pertains to the associated address. Isolated by some systems for sorting or indexing. Used in most cases to facilitate automatic sorting of mail.

ADDRESS_EMAIL:= {Size=5:120}

An electronic address that can be used for contact such as an email address.

email address: double at-sign

Email addresses contain a single at-sign (@), but GEDCOM lines may not contain a single at-sign.
In the GEDCOM grammar, a single at-sign indicates the starts or end of an escape sequence.
The GEDCOM grammar states that a single at-sign must be encoded as a double at sign.

A GEDCOM writer must export the single at-sign as two consecutive at-signs, and a GEDCOM reader must import that double at-sign as a single at-sign.

ADDRESS_FAX:= {Size=5:60}

A FAX telephone number appropriate for sending data facsimiles.

ADDRESS_LINE:= {Size=1:60}

Typically used to define a mailing address of an individual when used subordinate to a RESIdent tag. When it is used subordinate to an event tag it is the address of the place where the event took place. The address lines usually contain the addressee 's name and other street and city information so that it forms an address that meets mailing requirements.

<ADDRESS_LINE> is old-style

<ADDRESS_LINE> is used for old-style addresses; a single line value containing line breaks.
You should use new-style structured addresses; separate subrecord for each address part.

Consider old-style addresses deprecated since 1995 CE. Use structured addresses exclusive.

ADDRESS_LINE1:= {Size=1:60}

The first line of the address used for indexing. This is the value of the line corresponding to the ADDR tag line in the address structure.

ADDRESS_LINE2:= {Size=1:60}

The second line of the address used for indexing. This is the value of the first CONT line subordinate to the ADDR tag in the address structure.

ADDRESS_LINE3:= {Size=1:60}

The third line of the address used for indexing. This is the value of the second CONT line subordinate to the ADDR tag in the address structure.

ADDRESS_POSTAL_CODE:= {Size=1:10}

The ZIP or postal code used by the various localities in handling of mail. Isolated for sorting or indexing.

ADDRESS_STATE:= {Size=1:60}

The name of the ~~state~~ province, state or similar country subdivision used in the address. Isolated for sorting or

41

42

indexing.

ADDRESS_WEB_PAGE:= {Size=5:120}

The world wide web page address.

ADOPTED_BY_WHICH_PARENT:= {Size=1:4}

[HUSB | WIFE | BOTH]
A code which shows which parent in the associated family record adopted this person.
Where:

HUSB = The ~~HUSBand~~ HUSB (husband) in the associated family adopted this person.
WIFE = The WIFE (wife) in the associated family adopted this person.
BOTH = Both ~~HUSBand~~ HUSB (husband) and WIFE (wife) adopted this person.

AGE_AT_EVENT:= {Size=1:12}

[< | > | <NULL>]
[YYy MMm DDDd | YYy | MMm | DDDd |
YYy MMm | YYy DDDd | MMm DDDd |
CHILD | INFANT | STILLBORN]
Where:

> = greater than indicated age
< = less than indicated age
y = a label indicating years
m = a label indicating months
d = a label indicating days
YY = number of full years
MM = number of months
DDD = number of days
CHILD = age < 8 years
INFANT = age < 1 year
STILLBORN = died just prior, at, or near birth, 0 years

A number that indicates the age in years, months, and days that the principal was at the time of the associated event. Any labels must come after their corresponding number, for example; 4y 8m 10d.

ANCESTRAL_FILE_NUMBER:= {Size=1:12}

A unique permanent record number of an individual record contained in the Family History Department's Ancestral File.

<ANCESTRAL_FILE_NUMBER>

The so-called <ANCESTRAL_FILE_NUMBER> is an identifier within the Ancestral File

system. The definition states that this identifier is unique, which merely means that the Ancestral File.system issued each identifier just once.
This definition is easily misunderstood as saying that each individual within Ancestral File has one unique identifier. However, while each identifier corresponds to exactly one individual, individuals are not limited to a single identifier; individuals may have multiple identifiers.

AFN is superfluous

The AFN record is just as superfluous as the SSN record; it was never necessary to define a separate record type, because the IDNO record and it's <NATIONAL_ID_NUMBER> line value already offers the desired functionality.

references

GEDCOM SSN

APPROVED_SYSTEM_ID:= {Size=1:20}

42

43

A system identification name which was obtained through the GEDCOM registration process. This name must be unique, different from any other product. Spaces within the name must be substituted with a 0x5F (underscore _) so as to create one word.

no GEDCOM registration process

There is no GEDCOM registration process. FamilySearch silently abandoned GEDCOM around 2000 CE.
FamilySearch does not provide a list of system identifiers registered before 2000 CE, and most current genealogy applications were created after 2000 CE.

A fairly large but far from extensive list of collected system identifiers is provided by *GEDCOM System Identifiers*.

references

GEDCOM System Identifiers

maximum system identifier length contradicted in Chapter 4

Chapter 4 GEDCOM Product Registration contradicts the <APPROVED_SYSTEM_ID> definition provided here by stating that system identifiers may be up to 40 characters long. The actual lineage-linked form definition, provided here, takes precedence over Chapter 4.
The maximum length of a system identifier is 20 characters (code units).

system identifier case sensitivity

The FamilySearch GEDCOM specification fails to specify the case-sensitivity of system identifiers.
Existing practice is to treat system identifiers as case-insensitive.

spaces in system identifiers

Here, the FamilySearch GEDCOM specification explicitly states that you should not use spaces in system identifiers, but use underscores instead. That restriction is stated, but not motivated. Meanwhile, FamilySearch themselves uses spaces in systems identifiers: `PAF 2.0`, `PAF 2.1`, `PAF 4.0`, even `PAF 5.0`. Quite a few products from other vendors use one or more spaces in their system identifiers too, and that is no problem.
Consider this restriction to never have applied in the past, and to be obsolete now.

System Identifer Best Practice

GEDCOM writer best practice

- Use just one system identifier per product
- Pick a unique system identifier
- Avoid spaces in system identifiers
- Avoid non-ASCII characters
- Respect the maximum length of 20 characters (code units)
- Do not include any version number or other version indication, you have HEAD.SOUR.VERS for that
- You are allowed to use lower case. Take advantage of that for readability.

GEDCOM reader best practice

- Accept spaces in system identifiers
- Be ready to accept non-ASCII characters (i.e. figure out the character encoding first)
- Treat system identifiers as case-insensitive

GEDCOM validator best practice

- Warn that spaces are officially illegal
- Warn against usage of non-ASCII characters
- Issue an error when the system identifier is too long
- Upon encountering an ALL-CAPS system identifier, suggest the use of lower case for readability
- Try to detect version numbers in system identifiers, and issue a strong warning against it

references

GEDCOM System Identifiers

ATTRIBUTE_DESCRIPTOR:= {Size=1:90}

Text describing a particular characteristic or attribute assigned to an individual. This attribute value is assigned to the FACT tag. The classification of this specific attribute or fact is specified by the value of the subordinate TYPE tag selected from the <EVENT_DETAIL> structure. For example if you were classifying the skills a person had obtained;

1 FACT Woodworking
2 TYPE Skills

ATTRIBUTE_TYPE:= {Size=1:4}

[CAST | EDUC | NATI | OCCU | PROP | RELI | RESI | TITL | FACT]
An attribute which may have caused name, addresses, phone numbers, family listings to be recorded. Its application is in helping to classify sources used for information.

AUTOMATED_RECORD_ID:= {Size=1:12}

A unique record identification number assigned to the record by the source system. This number is intended to serve as a more sure means of identification of a record for reconciling differences in data between two interfacing systems.

CASTE_NAME:= {Size=1:90}

A name assigned to a particular group that this person was associated with, such as a particular racial group, religious group, or a group with an inherited status.

CAUSE_OF_EVENT:= {Size=1:90}

Used in special cases to record the reasons which precipitated an event. Normally this will be used subordinate to a death event to show cause of death, such as might be listed on a death certificate.

CERTAINTY_ASSESSMENT:= {Size=1:1}

[0 | 1 | 2 | 3]
The QUAY tag's value conveys the submitter's quantitative evaluation of the credibility of a piece of information, based upon its supporting evidence. Some systems use this feature to rank multiple conflicting opinions for display of most likely information first. It is not intended to eliminate the receiver's need to evaluate the evidence for themselves.

0 = Unreliable evidence or estimated data
1 = Questionable reliability of evidence (interviews, census, oral genealogies, or potential for bias for example, an autobiography)
2 = Secondary evidence, data officially recorded sometime after event
3 = Direct and primary evidence used, or by dominance of the evidence

CERTAINTY_ASSESSMENT

This simple four-value certainty assessment is arguably too simple and too subjective for serious use. It is not used much.
Several modern genealogy applications use a multi-value quality assessment, which is recorded as a GEDCOM extension.

CHANGE_DATE:= ~~{Size=1:11}~~{Size=10:11}

<DATE_EXACT>
The date that this data was changed.

<CHANGE_DATE> size

This <CHANGE_DATE> definition resolves to <DATE_EXACT>. The size of <DATE_EXACT> is {10:11}, therefore the size of <CHANGE_DATE> is {10:11} too.

second <CHANGE_DATE> definition

This is a second <CHANGE_DATE> definition; there already is a <<CHANGE_DATE>> definition. That is a specification error.

The *<<CHANGE_DATE> versus <CHANGE_DATE>>* annotation on the first <<CHANGE_DATE>> definition explains that the definition has been corrected to use the <DATE_EXACT> definition directly, making this second <CHANGE_DATE> definition superfluous.

The only other record definitions to use this second <CHANGE_DATE> definition are the obsolete <<LDS_INDIVIDUAL_ORDINANCE>> definition and <<LDS_SPOUSE_SEALING>> definition.
Both definitions have also been corrected to use the <DATE_EXACT> definition directly.

With these changes, this second <CHANGE_DATE> definition is obsolete.

CHARACTER_SET:= ~~{Size=1:8}~~ {Size=3:12}

[ANSEL |UTF-8 | UNICODE | ASCII]
A code value that represents the character set to be used to interpret this data. Currently, the preferred character set is `ANSEL`, which includes `ASCII` as a subset. `UNICODE` is not widely supported by most operating systems; therefore, GEDCOM produced using the `UNICODE` character set will be limited in its interchangeability for a while but should eventually provide the international flexibility that is desired. See Chapter 3, starting on page 77.

Note:The `IBMPC` *character set is not allowed.* This character set cannot be interpreted properly without knowing which code page the sender was using.

CHAR size

The CHARACTER_SET definition gives the length of the CHAR line value as {1:8}. That is wrong.
Three of the four values are five characters long, and one is seven characters long, so the length specification should be {5|7}.

The provided correction states that the length is {3:12}, and that is not an error. GEDCOM writers should not use anything but the four stated values, but GEDCOM readers have to deal with the legacy of existing not-quite-GEDCOM files.
Illegal character sets that a vendor may want a GEDCOM reader to support include `IBMPC`, `MSDOS`, `ANSI` (4 characters) and `MACINTOSH` (9 characters). GEDCOM readers may also encounter `IBM` or `OEM` (both 3 characters) and `IBM-PC` instead of `IBMPC`, `IBM DOS` and `MS-DOS` instead of `MSDOS`, and `WINDOWS`, `IBM WINDOWS` (11 characters), or `WINDOWS 1252` (12 characters) instead of `ANSI`.

references

GEDCOM Character Encodings

Unicode Support

The CHARACTER_SET definition claims that “UNICODE is not widely supported by most operating systems”. That statement was already wrong on the day that the FamilySearch GEDCOM 5.5.1 specification was published (1999 CE). It is obsolete now.

This is one of the few places in the FamilySearch GEDCOM 5.5.1 specification where “UNICODE” must be read as actually meaning Unicode (version 1.0, UCS-2), instead of meaning UTF-16.
Most current (2018 CE) genealogy applications are Unicode-based. Most code-page genealogy

applications have either been upgraded or retired. However, while most genealogy applications support UTF-8, few support UTF-16.

code pages not allowed

This CHARACTER_SET definition not only states that the `IBMPC` character set is not allowed, but *also explains why not*: there is more than one `IBMPC` code page, and that's problematic.

The CHARACTER_SET definitions fails to mention that MS-DOS, Windows ANSI and MacRoman character sets are illegal too, for the same reason. That the Windows ANSI character set is illegal *is* mentioned within the specification, namely in the Introduction of Chapter 3 Using Character Sets in GEDCOM:

> Systems using code pages to support diacritical characters, such as the windows ANSI 1252 code page, must convert all characters above character code 0x7F to its ANSEL representation for that code page.

CHAR UTF-8

FamilySearch GEDCOM 5.5.1 is the first GEDCOM specification to allow the UTF-8 encoding.
That 5.5.1 allows UTF-8 while 5.5 does not was a major reason for vendors to upgrade from 5.5 to 5.5.1.

The UTF-8 encoding plays an important role in detection of GEDCOM 5.5.1 files. Because of a mispractice started by FamilySearch's Personal Ancestral File), many GEDCOM 5.5.1 files claim to be GEDCOM 5.5 files. Any GEDCOM file that claims to be a GEDCOM 5.5 file but uses the UTF-8 encoding must be recognised as a GEDCOM 5.5.1 file with a truncated version number.

references

GEDCOM Version detection

character encoding advice

Nowadays, many genealogy applications default to UTF-8 as the encoding for their GEDCOM files, and some, such as RootsMagic, do not provide the user any other choice. There is a good reason behind that limitation. It is a bad idea for Unicode applications to allow export to ASCII, ANSEL or any of the illegal code pages, such as Windows ANSI, because conversion to these older character sets will almost certainly result in loss of information: all the characters those older character sets do not support will be replaced by questions marks or gibberish.
To prevent this information loss, Unicode applications should provide export to Unicode (or a future superset of Unicode) only. Unicode applications should provide export to UTF-8 and UTF-16 (`UNICODE` in the above list) only.

GEDCOM writer best practice

- Use legal encodings only
- do not offer ASCII (all the other choices are supersets of ASCII)
- default to UTF-8

code page applications

- offer ANSEL and UTF-8
- do *not* offer Windows ANSI, MacRoman or any other code page

Unicode applications

- offer UTF-8

- optionally offer UTF-16 (`UNICODE`)
- do not offer anything else, not even ANSEL

CHAR BINARY

The GEDCOM 5.0 specification (1991 CE) allows the value `BINARY`, which is defined as "Binary data (future use)".
The GEDCOM 5.3 specification and later do not mention it. It has never been used, and it is not clear how it could ever be used in a text-based format.

CHAR EBCDIC

The GEDCOM 3.0 specification (1987 CE) illustrates the use of the `CHAR` record with an example that has the line value `EBCDIC`. This suggests that FamilySearch may have been using EBCDIC-encoded GEDCOM files internally, in violation of their own specification. No known version of the GEDCOM specification allows `EBCDIC`. In fact, `EBCDIC` is illegal for the same reason that `IBMPC`, `ANSI`, and `MacRoman` are illegal: all these names are ambiguous, as they do not identify a single character set, but a bunch of code pages.

ASCII range

The GEDCOM 5.5 specification contains the erroneous sentence:

> ASCII contains the character set from 0x0 to 0xFF.

The GEDCOM 5.5 errata sheet improved that to read:

> ASCII contains the character set from 0x00 to 0x7F.

which still gets the range wrong, as it should be:

> ASCII contains the character set from 0x00 through 0x7F.

The sentence was removed from the GEDCOM 5.5.1 specification.

CHILD_LINKAGE_STATUS:= {Size=1:15}

[challenged | disproven | proven]
A status code that allows passing on the users opinion of the status of a child to family link.

`challenged` = Linking this child to this family is suspect, but the linkage has been neither proven nor disproven.
`disproven` = There has been a claim by some that this child belongs to this family, but the linkage has been disproven.
`proven` = There has been a claim by some that this child does not belongs to this family, but the linkage has been proven.

lineage status

The <CHILD_LINKAGE_STATUS> feature is arguably bad design; an application that fails to support this feature will simply interpret a child-family link as a child-family link, even when it is "disproven".

"proven" and "disproven"

The usage of the words “proven” and “disproven” for official and legal genealogy is seriously bad practice, sadly still actively promoted by some individuals and organisations. It is best to reserve these words for actual proof, like DNA tests for biological genealogy, and not use them carelessly in a way that dilutes their meaning this much.

COPYRIGHT_GEDCOM_FILE:= {Size=1:90}

A copyright statement needed to protect the copyrights of the submitter of this GEDCOM file.

copyrighted GEDCOM file

The copyright-ability of GEDCOM files is a tricky topic. A typical GEDCOM is a selection taken from compilation of facts, interpretations, and notes, complete with mistakes. Fact aren't copyrightable, but original notes are, and you really need a lawyer if you really care about this topic.

COPYRIGHT_SOURCE_DATA:= {Size=1:90}

A copyright statement required by the owner of data from which this information was downloaded. For example, when a GEDCOM download is requested from ~~the~~ Ancestral File, this would be the copyright statement to indicate that the data came from a copyrighted source.

COUNT_OF_CHILDREN:= {Size=1:3}

The known number of children of this individual from all marriages or, if subordinate to a family record, the reported number of children known to belong to this family group, regardless of whether the associated children are represented in the corresponding structure. This is not necessarily the count of children listed in a family structure.

COUNT_OF_MARRIAGES:= {Size=1:3}

The number of different families that this person was known to have been a member of as a spouse or parent, regardless of whether the associated families are represented in the GEDCOM file.

COUNT_OF_MARRIAGES

<COUNT_OF_MARRIAGES> is really <COUNT_OF_RELATIONSHIPS>: the count of MARR records we would have if all relationships were included.

DATE:= {Size=4:35}

[<DATE_CALENDAR_ESCAPE> | <NULL>]
<DATE_CALENDAR>

date calendar escape sequence

It is odd that FamilySearch's lineage-linked form uses a GEDCOM escape sequence to specify the calendar. The calendar could easily be expressed with say a DATE.TYPE record.
In fact, GEDCOM 5.6 gets rid of the escape sequence by doing just that; it introduces the optional DATE.CLNDR record.

changes

FamilySearch kept changing its mind about the calendar escape sequence. FamilySearch GEDCOM 4.0 introduced the calendar escape sequences, and GEDCOM 5.0 still featured them. FamilySearch GEDCOM 5.3 features the calendar escapes sequences, but GEDCOM 5.4 does not, and states that "The Lineage-Linked GEDCOM Form is restricted to Gregorian calendar forms.". FamilySearch GEDCOM 5.5 and 5.5.1 feature the calendar escape sequences, and GEDCOM 5.6 replaces it with a subrecord.

DATE_APPROXIMATED:= {Size=4:35}

[
ABT <DATE> |
CAL <DATE> |
EST <DATE>
]

Where:

ABT = About, meaning the date is not exact.
CAL = Calculated mathematically, for example, from an event date and age.
EST = Estimated based on an algorithm using some other event date.

About Abt

The FamilySearch GEDCOM specification provides three different date modifiers for approximated dates, with three slightly different definitions. The implied demand to use different date modifiers for slightly different situations is not placed on genealogy software, but on users. In practice, users almost always use ABT. Few users even know that usage of CAL or EST is possible.

About Family Tree Maker Abt

The annotation Date Modifiers in Family Tree Maker provides a compatibility note about date modifiers in Family Tree Maker.

DATE_CALENDAR:= {Size=4:35}

[<DATE_GREG> | <DATE_JULN> | <DATE_HEBR> | <DATE_FREN> ~~| <DATE_FUTURE> ~~]
The selection is based on the <DATE_CALENDAR_ESCAPE> that precedes the <DATE_CALENDAR> value immediately to the left. If <DATE_CALENDAR_ESCAPE> doesn't appear at this point, then @#DGREGORIAN@ is assumed. No future calendar types will use words (e.g., month names) from this list: FROM, TO, BEF, AFT, BET, AND, ABT, EST, CAL, or INT. When only a day and month appears as a DATE value it is considered a date phrase and not a valid date form.

Date Escape	Syntax Selected
@#DGREGORIAN@	<DATE_GREG>
@#DJULIAN@	<DATE_JULN>
@#DHEBREW@	<DATE_HEBR>
@#DFRENCH R@	<DATE_FREN>
@#DROMAN@	for future definition
@#DUNKNOWN@	calendar not known

Swedish Calendar

It is surprising that even GEDCOM version 5.5.1 still doesn't support the Swedish Calendar.
All it takes is adding the escape `@#DSWEDISH@` to the specification.
Conversion between the Swedish, Julian and Gregorian Calendars is trivially easy.

Sweden is the only country to not simply switch from the Julian to the Gregorian Calendar on a single day. Instead of simply skipping 11 calendar days as other countries had done, Sweden decided to gradually switch from the Julian to the Gregorian Calendar by omitting all the leap days in the period 1700 through 1740.
In 1700, Feb 29 was omitted from the Calendar. So, from that day forward, there was as Swedish Calendar that was one day behind the Julian Calendar, and ten days ahead of the Gregorian Calendar. The plan to omit the leap days from 1704 and 1708 wasn't executed: the difference with the Julian Calendar remained one day.
In 1711, Sweden decided to return to the Julian Calendar by adding an extra leap day to 1712, hence the existence of 30 Feb 1712. 30 Feb 1712 in the Swedish Calendar corresponds to 29 Feb 1712 in the Julian Calendar.
From 1 March 1712 forward, Sweden was using the Julian Calendar again, to eventually adopt the Gregorian Calendar in 1753.

<DATE_FUTURE> does not exist

This <DATE_CALENDAR> definition references several other definitions, such as <DATE_GREG> and <DATE_JULN>. The list of referenced calendar definitions is closed by <DATE_FUTURE>, but the GEDCOM specification does not define a <DATE_FUTURE>, nor <FUTURE_DATE>.
The definition is in error.

erroneous intent

Perhaps the intention was to state that the DATE tag may contain calendars not defined in this specification, but that would be an error too; a future version of GEDCOM could add yet another calendar, but support for that calendar would require that GEDCOM version, and would not change the calendars supported by this version of GEDCOM.

date without year

The <DATE_CALENDAR> definition states that “When only a day and month appears as a DATE value it is considered a date phrase and not a valid date form.”. Not allowing dates without a year is an unfortunate decision, and one that remains unmotivated.
The reason *is not* that a value like `23 Aug` is ambiguous, and could be understood to mean either the 23rd of August or August in the year 23; it cannot be interpreted as latter, because GEDCOM demands that years are at least 3 digits longs.

It is quite common for people to know birthdays without knowing the birth year, or remember a death date without the death year.
Knowing a date is so valuable in search for documents that could contain the year, that applications should support dates without a year. GEDCOM readers should accept such dates, but issue a warning that the year is missing.

DATE_CALENDAR_ESCAPE:= {Size=4:15}

[@#DHEBREW@ | @#DROMAN@ | @#DFRENCH R@ | @#DGREGORIAN@ | @#DJULIAN@ | @#DUNKNOWN@]
The date escape determines the date interpretation by signifying which <DATE_CALENDAR> to use. The default calendar is the Gregorian calendar.

supported calendars

The selection of supported calendars reveals a Western bias.
There are many calendars the GEDCOM specification does not support, including the Hijiri, Buddhist, Chinese, Japanese and Tibetan calendars.

space in calendar escape

The French Calendar escape, `@#DFRENCH R@`, contains a space. This looks and feels like a mistake, but the GEDCOM grammar allows it, and it *is* what genealogy applications that support the French Revolutionary calendar use and expect.

known error

At least one developer assumed the space to be a mistake. Alsyd Parentèle, later MindScape Parentèle, now discontinued, exports `@#DFRENCHR@` (no space) instead of `@#DFRENCH R@`. The GEDCOM lineage-linked form does not define a date escape `@#DFRENCHR@`. It is okay for a GEDCOM reader to issue a fatal error and abort upon encountering calendar escape `@#DFRENCHR@`, but it is suggested that GEDCOM readers issue a non-fatal error for each individual occurrence of `@#DFRENCHR@`, interpret it as `@#DFRENCH R@`, and continue processing.

the Gregorian Calendar escape is superfluous

The Gregorian Calendar is the default calendar, so the Gregorian Calendar escape (`@#DGREGORIAN@`) is superfluous.

GEDCOM writer best practice

- do not use the Gregorian Calendar escape (`@#DGREGORIAN@`)

GEDCOM reader best practice

- recognise and discard the Gregorian Calendar escape (`@#DGREGORIAN@`)

DATE_EXACT:= {Size=10:11}

<DAY> <MONTH> <YEAR_GREG>

Date Validation

The FamilySearch GEDCOM specification says nothing about date validation. Genealogy applications should make sure that dates are valid, and that includes knowing the rules for leap years. Getting that right is a significant amount of work, so most developers will want to rely on existing date processing libraries.

references

Behold Blog: Out on Bad Date

DATE_FREN:= {Size=4:35}

[<YEAR> [B.C.] | <MONTH_FREN> <YEAR> |

<DAY> <MONTH_FREN> <YEAR>]
See <MONTH_FREN> page 53.

BC versus B.C.

The GEDCOM specification says to use `B.C.` (with two full stops), but many genealogy applications, including FammilySearch's own PAF use `BC` (without the full stops).

best practice

- GEDCOM writers should use `B.C.` (including the two full stops), as that's what's specified.
- GEDCOM readers should accept `B.C.`, `BC` and `BCE`.

space before B.C.

The FamilySearch GEDCOM 5.5.1 specification lacks a space between <YEAR> and `B.C.`, not only in this <DATE_FREN> definition, but also in the DATE_GREG, DATE_HEBR (!) and DATE_JULN definitions below. The omission of a space is an error, which has been corrected in this *Annotated Edition*.

best practice

- GEDCOM writers should use a space between the year and `B.C.`

DATE_GREG:= {Size=4:35}

[<YEAR_GREG> [B.C.] | <MONTH> <YEAR_GREG> |
<DAY> <MONTH> <YEAR_GREG>]
See <YEAR_GREG> page 65.

Gregorian B.C.

The optional “B.C.” in the definition of <DATE_GREG> is a doubtful feature. Not only is “B.C.” (Before Christ) a christian expression, used instead of the neutral BCE, but an optional “BCE” would still be a doubtful feature.

proleptic Gregorian Calendar

The Gregorian Calender, introduced in 1582; does not have dates before 15 Oct 1582. All Gregorian-looking dates in documents from before 15 Oct 1582 are dates in the Julian Calendar.
The Gregorian Calendar, extended backwards creates the so-called *proleptic Gregorian Calendar*. There are no proleptic Gregorian dates in original documents. The only way to get a Gregorian BCE date is by converting a date from another calendar.

BC versus B.C.

See the *BC versus B.C.* annotation above.

DATE_HEBR:= {Size=4:35}

[<YEAR>~~[B.C.]~~ | <MONTH_HEBR> <YEAR> |
<DAY> <MONTH_HEBR> <YEAR>]
See <MONTH_HEBR> page 53.

Hebrew B.C.

The inclusion of the optional "B.C." in the definition of <DATE_HEBR> is most definitely an error.
Not only is "B.C." (Before Christ) a christian expression, used instead of the neutral BCE, but an optional "BCE" would still be wrong. BCE is only used with proleptic Julian and Gregorian Calendars, it is never used with a Hebrew calendar date.

Anno Mundi

A Hebrew date may be *preceded* by "AM" (without quotes), which is an abbreviation of *Anno Mundi*, Latin for "Year of the World". Users who enter Hebrew dates (complete with the `@#HEBREW@` escape) directly into the date field are not unlikely to use this.
There is no proleptic Hebrew Calendar; the Hebrew Calendar is never used for dates before AM 1.

calendar conversion

In the Hebrew calendar, days start at sunset. When faced with a date without time information, conversion between the Hebrew and Julian or Gregorian calendars should calculate the so-called tabular date, i.e. the corresponding date with the same daylight period.

best practice

- GEDCOM writers should not write "AM", as most GEDCOM readers won't recognise this
- GEDCOM readers should accept "AM", as users may have entered this directly

DATE_JULN:= {Size=4:35}

[<YEAR> [B.C.] | <MONTH> <YEAR> | <DAY> <MONTH> <YEAR>]

Julian B.C.

The inclusion of the optional "B.C." in the definition of <DATE_JULN> is a doubtful feature.
Not only is "B.C." (Before Christ) a christian expression, used instead of the neutral BCE, but an optional "BCE" would still be a doubtful feature.

sources

The mere inclusion of negative Gregorian Calendar and Julian Calendar dates should raise eyebrows already. It was only after the Renaissance that churches in Europe started keeping regular baptism, burial and marriage records. The few genealogies that trace European aristocracy back to Carolus Magnus (Charlemagne) are tenuous at best, and the genealogy of Gothic kings back into the sixth century isn't any better.
A genealogy standard simply *has no need* for Julian BCE dates.

BC versus B.C.

See the *BC versus B.C.* annotation above.

DATE_LDS_ORD:= {Size=4:35}

<DATE_VALUE>
LDS ordinance dates use only the Gregorian date and most often use the form of day, month, and year. Only in rare instances is there a partial date. The temple tag and code should always accompany temple ordinance dates. Sometimes the <LDS_(ordinance)_DATE_STATUS> is used to indicate that an ordinance date and temple code is not required, such as when BIC is used. (See <LDS_(ordinance)_DATE_STATUS> definitions beginning on page 51.)

There is no <LDS_(ordinance)_DATE_STATUS>. It is unexplained shorthand for <LDS_BAPTISM_DATE_STATUS>,<LDS_CHILD_SEALING_DATE_STATUS>, <LDS_ENDOWMENT_DATE_STATUS> and

<LDS_SPOUSE_SEALING_DATE_STATUS>. These four items occur together in the list of definitions, so the hyperlink is to the first one, <LDS_BAPTISM_DATE_STATUS>.

DATE_PERIOD:= {Size=7:35}

```
[
FROM <DATE> |
TO <DATE> |
FROM <DATE> TO <DATE>
]
```

Where:

FROM = Indicates the beginning of a happening or state.
TO = Indicates the ending of a happening or state.

Examples:

FROM 1904 **TO** 1915

= The state of some attribute existed from 1904 to 1915 inclusive.

FROM 1904

= The state of the attribute began in 1904 but the end date is unknown.

TO 1915

= The state ended in 1915 but the begin date is unknown.

46

47

DATE_PHRASE:= {Size=1:35}

<TEXT>
Any statement offered as a date when the year is not recognizable to a date parser, but which gives information about when an event occurred.

date phrase

The <DATE_PHRASE> is a bad idea because it enables *anything* in a DATE field.
Unrecognisable dates should be in a NOTE record, not inside the DATE field itself.
It is strongly recommended that genealogy software keeps DATE records clean, by relegating date phrases to a NOTE.

references

GEDCOM date phrases

"enclosed in matching parentheses"

This GEDCOM 5.5.1 <DATE_PHRASE> definition is subtly different from the GEDCOM 5.5 definition. This is the GEDCOM 5.5 <DATE_PHRASE>definition:

DATE_PHRASE:= {Size=1:35}

<TEXT>
Any statement offered as a date when the year is not recognizable to a date parser, but which gives information about when an event occurred. The date phrase is enclosed in matching parentheses.

It may seem that GEDCOM 5.5 demands that date phrases are enclosed in parentheses, but GEDCOM 5.5.1 does not - but that is not the case.
Both GEDCOM 5.5 and GEDCOM 5.5.1 demand that date phrases are enclosed in parentheses, and both make that demand through the syntax of the <DATE_VALUE> definition. The extra sentence in the GEDCOM 5.5 specification seemed to demand a second set of parentheses, and that is why it was removed.

DATE_RANGE:= {Size=8:35}

[
BEF <DATE> |
AFT <DATE> |
BET <DATE> AND <DATE>
]

Where:

AFT	= Event happened after the given date.
BEF	= Event happened before the given date.
BET	= Event happened some time between date 1 AND date 2

For example, bet 1904 and 1915
indicates that the event state (perhaps a single day) existed somewhere between 1904 and 1915 inclusive.
The date range differs from the date period in that the date range is an estimate that an event happened on a single date somewhere in the date range specified.
The following are equivalent and interchangeable:

Short form	Long Form
1852	BET 1 JAN 1852 AND 31 DEC 1852
1852	BET 1 JAN 1852 AND DEC 1852
1852	BET JAN 1852 AND 31 DEC 1852
1852	BET JAN 1852 AND DEC 1852
JAN 1920	BET 1 JAN 1920 AND 31 JAN 1920

date modifiers in Family Tree Maker

The date modifiers ABT, BEF & AFT are three-letter abbreviations. Several old versions of Family Tree Maker use the illegal four-letter abbreviations ABT., BEF. & AFT. instead; Family Tree Maker adds a fourth character, a full stop, to the abbreviation while it should not.

best practice

A GEDCOM reader need not accept these erroneous dates, but it is easy to support. Even a GEDCOM reader that supports it should still issue a non-fatal error for each occurrence.

date range in FamilySearch PAF

The ability to specify a <DATE_RANGE> is an important feature. Oddly, FamilySearch PAF does not support it, but pops up a message box stating “The date is not standard” when you use it. You can still use date ranges; PAF will (erroneously) treat the <DATE_RANGE> as a <DATE_PHRASE>, and export it exactly the way you entered it (without the enclosing parentheses mandatory for date phrases), so genealogy applications that do support date ranges will import it just fine.
Incidentally, PAF *does* support <DATE_PERIOD>.

DATE_VALUE:= {Size=1:35}

```
[
<DATE> |
<DATE_PERIOD> |
<DATE_RANGE>|
<DATE_APPROXIMATED> |
INT <DATE> (<DATE_PHRASE>) |
(<DATE_PHRASE>)
]
```

The <DATE_VALUE> represents the date of an activity, attribute, or event where:

INT = Interpreted from knowledge about the associated date phrase included in parentheses.

47

48

An acceptable alternative to the date phrase choice is to use one of the other choices such as <DATE_APPROXIMATED> choice as the DATE line value and then include the date phrase value as a NOTE value subordinate to the DATE line tag.

The date value can take on the date form of just a date, an approximated date, between a date and another date, and from one date to another date. The preferred form of showing date imprecision, is to show, for example, MAY 1890 rather than ABT 12 MAY 1890. This is because limits have not been assigned to the precision of the prefixes such as ABT or EST.

Box Around <DATE_RANGE>

There is a box around <DATE_RANGE>. It is not clear why.
Perhaps because FamilySearch PAF does not support BET...AND date ranges.

ABT 12 May 1890

The <DATE_VALUE> does not only allow approximated dates such as ABT MAY 1890, it also allows the almost contradictory value ABT 12 May 1890; ABT in front of an exact date. Practically all cases of ABT in front of an exact date should be either BEF (before) or AFT (after) in front of that date.

date phrases in parentheses

Note the parentheses around <DATE_PHRASE>; the <DATE_VALUE> definition demands that all date phrases be enclosed in parentheses.

date phrases in PAF

FamilySearch PAF 5 includes <DATE_PHRASE> support, but PAF exports date phrases to GEDCOM exactly the way users enter them, without the mandatory enclosing parentheses.

This is wrong.

GEDCOM writer best practice

- Always use parentheses around date phrases as demanded by the GEDCOM specification

GEDCOM reader best practice

- Make sure you reocgnise dates all the supported calendars
- Treat every date value you do not recognise as a date phrase
- Issue an error for date phrase not enclosed in parentheses

references

Behold blog: Out on a Bad Date

FamilySearch PAF DATE record abusage

FamilySearch's own PAF does not respect the <DATE_VALUE> definition, but abuses date fields in ways not intended or allowed by the FamilySearch GEDCOM specification:

- FamilySearch PAF bluntly assumes that all parents are married, and the only way to tell PAF that they are not married is to enter “Not married” in the marriage date field.
- FamilySearch PAF additionally supports “Child”, “Stillborn” and “Infant” in the death date field.
  PAF even allows the use of the three-letter abbreviations “Chi”, “Sti” and “Inf”, but these are immediately expanded to “Child”, “Stillborn” and “Infant”.

PAF exports these values in DATE records as `NOT MARRIED`, `CHILD`, `STILLBORN` and `INFANT`. This DATE field abusage seems to be a holdover from earlier PAF versions.

Genealogy applications that do not know about and therefore do not recognise these values, are likely to treat each such value as a <DATE_PHRASE>, without issuing an error or warning. It is suggested that GEDCOM readers issue a non-fatal error for each individual occurrence of any of these four values, clearly state what action was taken for the value, and continue processing.

Ancestral Quest does not handle this GEDCOM any better than PAF does; even Ancestral Quest 15 will still export `NOT MARRIED` in the marriage date field.

DAY:= {Size=1:2}

dd
Day of the month, where dd is a ~~numeric digit whose value is~~ numeric value within the valid range of the days for the associated calendar month.

“numeric digit”

The description states that dd is a “numeric digit”.
That is incorrect; dd isn't a *digit*, dd is a *numeric value*. That value consists of either one or two digits.

“Leading Zeroes in dd”

FamilySearch does not specify whether dd may start with a zero.

best practice

As they do not state it is illegal, it has to be considered legal by a GEDCOM reader. As they do not state it is legal, a GEDCOM writer should not produce leading zeroes.

legal day value

The description states that the day value must be legal; i.e. the value must be within the valid range of dates for the month.
This demand implies that *genealogy software is expected to validate dates* upon both user input and GEDCOM import.
When validating dates, applications should make sure that the day not only fits the month, but - for leap days - the year and calendar as well.

30 Feb 1712

The date 30 Feb 1712 was a real date in Sweden. There are records with that date. 30 Feb 1712 on the Swedish Calendar corresponds with 29 Feb 1712 on the Julian Calendar. 30 Feb 1712 isn't a date on either the Julian or Gregorian calendar, but only on the Swedish Calendar, which even GEDCOM version 5.5.1 still fails to support.

DESCRIPTIVE_TITLE:= {Size=1:248}

The title of a work, record, item, or object.

DIGIT:= {Size=1:1}

A single digit (0-9).

ENTRY_RECORDING_DATE:= {Size=1:90}

<DATE_VALUE>
The date that this event data was entered into the original source document.

EVENT_ATTRIBUTE_TYPE:= {Size=1:15}

[<EVENT_TYPE_INDIVIDUAL> |
<EVENT_TYPE_FAMILY> |
<ATTRIBUTE_TYPE>]

A code that classifies the principal event or happening that caused the source record entry to be created. If the event or attribute doesn't translate to one of these tag codes, then a user supplied value is expected and will be generally classified in the category of other.

EVENT_DESCRIPTOR:= {Size=1:90}

Text describing a particular event pertaining to the individual or family. This event value is usually assigned to the EVEN tag. The classification as to the difference between this specific event and other occurrences of the ~~EVENt~~ EVEN (event) tag is indicated by the use of a subordinate TYPE tag selected from the <EVENT_DETAIL> structure. For example;

```
—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
```

incomplete place name

The PLAC line value sets a bad example by omitting the country. Place names should be fully specified, like this:

```
2 PLAC Cove, Cache, Utah, United States of America
```

EVENT_OR_FACT_CLASSIFICATION:= {Size=1:90}

A descriptive word or phrase used to further classify the parent event or attribute tag. This should be used whenever either of the generic **EVEN** or **FACT** tags are used. The value of this ~~primative~~ primitive is responsible for classifying the generic event or fact being cited. For example, if the attribute being defined was one of the persons skills, such as woodworking, the **FACT** tag would have the value of `Woodworking', followed by a subordinaxte TYPE tag with the value `Skills~~.~~'.

1 FACT Woodworking
2 TYPE Skills

This groups the fact into a generic skills attribute, and in particular this entry records the fact that this individual possessed the skill of woodworking. Using the subordinate TYPE tag classification method with any of the other defined event tags provides a further classification of the parent tag but does not change the basic meaning of the parent tag. For example, a **MARR** tag could be subordinated with a **TYPE** tag with an ~~EVENT_DESCRIPTOR~~ EVENT_OR_FACT_CLASSIFICATION value of `Common Law.'

1 MARR
2 TYPE Common Law

This classifies the entry as a common law marriage but the event is still a marriage event. Other descriptor values might include, for example,`stillborn' as a qualifier to ~~BIRTh~~ BIRT (birth) or `Tribal Custom' as a qualifier to ~~MARRiage~~ MARR (marriage).

TYPE tag correction

The FamilySearch GEDCOM 5.5.1 description for <EVENT_OR_FACT_CLASSIFICATION> mentions an <EVENT_DESCRIPTOR> value:

> For example, a MARR tag could be subordinated with a **TYPE** tag with an <EVENT_DESCRIPTOR> value of `Common Law.'

This is an editing error, made upon the introduction of FACT and FACT.TYPE in GEDCOM 5.5.1. The text does not discuss an <EVENT_DESCRIPTOR> value, but an <EVENT_OR_FACT_CLASSIFICATION>. The text should read:

> For example, a MARR tag could be subordinated with a TYPE tag with an <EVENT_OR_FACT_CLASSIFICATION> value of `Common Law.'

This specification error was first noted by Keith Riggle and mentioned in several software reviews. It was documented by Tim Forsythe in his 22 Dec 2015 GigaTrees blog post *A New GEDCOM 5.5.1 Wrinkle*, but that blog post is no longer available. It is now documented in Keith Riggle's own blog post *The Event Structure in GEDCOM Files*.

references

GenealogyTools: The Event Structure in GEDCOM Files

<EVENT_OR_FACT_CLASSIFICATION> values

This <EVENT_OR_FACT_CLASSIFICATION> is an open-ended feature. There is no list of pre-defined, accepted or suggested descriptor values.

FamilySearch versus civil marriage

FamilySearch's remark that a common law marriage “is still a marriage event” is odd to say the least, and the implied assumption that it might be reasonable to think of a civil marriage as not really a marriage is too wrong for words.

EVENT_TYPE_CITED_FROM:= {Size=1:15}

[<EVENT_ATTRIBUTE_TYPE>]
A code that indicates the type of event which was responsible for the source entry being recorded. For example, if the entry was created to record a birth of a child, then the type would be BIRT regardless of the assertions made from that record, such as the mother's name or mother's birth date. This will allow a prioritized best view choice and a determination of the certainty associated with the source used in asserting the cited fact.

EVENT_TYPE_FAMILY:= {Size=3:4}

[ANUL | CENS | DIV | DIVF | ENGA | MARR |
MARB | MARC | MARL | MARS | EVEN]

A code used to indicate the type of family event. The definition is the same as the corresponding event tag defined in Appendix A. (See Appendix A, starting on page 83).

EVENT_TYPE_INDIVIDUAL:= {Size=3:4}

[ADOP | BIRT | BAPM | BARM | BASM |
BLES | BURI | CENS | CHR | CHRA |
CONF | CREM | DEAT | EMIG | FCOM |
GRAD | IMMI | NATU | ORDN |
RETI | PROB | WILL | EVEN]

A code used to indicate the type of family event. The definition is the same as the corresponding event tag defined in Appendix A. (See Appendix A, starting on page 83).

EVENTS_RECORDED:= {Size=1:90}

[<EVENT_ATTRIBUTE_TYPE> |
<EVENTS_RECORDED>, <EVENT_ATTRIBUTE_TYPE>]
An enumeration of the different kinds of events that were recorded in a particular source. Each enumeration is separated by a comma. Such as a parish register of births, deaths, and marriages would be BIRT, DEAT, MARR.

FILE_NAME:= {Size=1:90}

The name of the GEDCOM ~~transmission~~ file. ~~If the file name includes a file extension it must be shown~~

~~in the form (filename.ext).~~
A GEDCOM file name should use the format *basename.ext*, and use the file extension `.GED` (or `.ged`)

GEDCOM file name

<FILE_NAME> is the file name specified as the HEAD.FILE line value.
The GEDCOM specification defines a separate <MULTIMEDIA_FILE_REFERENCE> for multimedia files.

file name

The specification clearly states that the line value should be the name of the GEDCOM file. Thus, the HEAD.FILE line value should be the name of the GEDCOM file itself. When a GEDCOM file is renamed by a user, the HEAD.FILE line value does not change. Thus, the line value need not match the current file name, it should match the original file name.

file name only

The FamilySearch GEDCOM specification fails to mention whether the GEDCOM file name should include just the file name, or the entire file path. The limited maximum length of only 90 characters makes it impossible to include any arbitrary file path, but that's probably only because the FamilySearch authors underestimated how many characters a full path might need; the maximum length for <MULTIMEDIA_FILE_REFERENCE> is only 30 characters.
It is strongly suggested that GEDCOM writers include only the file name.

case-preserving

The FamilySearch GEDCOM specification does not state whether file names are case sensitive.
Whether file names are case-sensitive is platform-dependant. Make sure your file name code is case-preserving.

The demand that a file name that includes an extension “must be shown in the form (filename.ext).” is needlessly confusing; it seems to mandate that parentheses should be used around the name. *Do* include the file extension. *Do not* use parentheses around the name.

GEDCOM writer best practice:

- The line value should be the name of the file itself.
- Include only the file name, not the entire path.
- File name handling code should be case-preserving.
- *Do* include the file extension.
- Default the file extension to `.GED`
- *Do not* use parentheses around the name.

GEDCOM_CONTENT_DESCRIPTION:= {Size=1:248}

A note that a user enters to describe the contents of the lineage-linked file in terms of "ancestors or descendants of" so that the person receiving the data knows what genealogical information the ~~transmission~~ file contains.

GEDCOM_FORM:= {Size=14:20}

[LINEAGE-LINKED]
The GEDCOM form used to construct this ~~transmission~~ GEDCOM file. There maybe other forms used such as CommSoft's "`EVENT_LINEAGE_LINKED`" but these specifications define only the `LINEAGE-LINKED` Form. Systems will use this value to specify GEDCOM compatible with these specifications.

GEDCOM form

one value

Note that there is just a single allowed and therefore mandatory value, `LINEAGE-LINKED`. The GEDCOM 5.4 draft additionally allows `EVENT_LINEAGE_LINKED` (but does so in error), and the GEDCOM 5.6 draft defines `LINEAGE-LINKED-XML`, but does not allow its usage; it was probably meant for use in GEDXML.

known errors

- The mandatory value is `LINEAGE-LINKED` (dash), not `LINEAGE_LINKED` (underscore). Millennia's Legacy Family Tree (now owned by MyHeritage) is perhaps the best known but not the only genealogy application to get this wrong. This error occurs in GEDCOM files produced by EasyTree, FamilyMatters, GEDitCOM, Généamania, GenoPro, Secere Family Tree and WebSSG.
- GenoPro GEDCOM files may also contain `LINAGE-LINKED` instead of `LINEAGE-LINKED`; the GenoPro value is missing the first E.
- The French application Genealogic by Infoduc and the French site Genealogie.com both create GEDCOM files that add spaces before and after the dash; `Lineage - Linked` instead of `LINEAGE-LINKED`.

Event GEDCOM

Event GEDCOM, originally known as InterGED, was created back in 1994 by CommSoft, the publishers of the Roots series of genealogy applications. Event GEDCOM was first supported by ROOTS IV. Although Event GEDCOM is arguably superior to LINEAGE-LINKED GEDCOM, Event GEDCOM never caught on. Event GEDCOM isn't widely supported, it is practically forgotten. It was only ever supported by Roots IV, Roots V, Family Gathering and Ultimate Family Tree.
Event GEDCOM files are rare, and backward compatible with lineage-linked GEDCOM files.

FamilySearch misinformation

FamilySearch's statement about CommSoft's Event GEDCOM is incorrect. Event GEDCOM *does not* use the line value "`EVENT_LINEAGE_LINKED`", neither with nor without quotes. Event GEDCOM files are backward compatible with lineage-linked GEDCOM files, and therefore use the line value `LINEAGE-LINKED`, *as instructed by the FamilySearch GEDCOM specification*: “Systems will use this value to specify GEDCOM compatible with these specifications.”.

actual Event GEDCOM

Event GEDCOM 1.0 files contain a additional subrecord, HEAD.FORM.FORM record with the value `EVENT` (illegal in LINEAGE-LINKED GEDCOM), and, although based on GEDCOM 5.3, claims to be GEDCOM version 1.0.

```
0 HEAD
...
1 SOUR FAMGATH
2 VERS 1.0
2 NAME Family Gathering
2 CORP Palladium Interactive, Inc.
...
1 GEDC
2 VERS 1.0
2 FORM LINEAGE-LINKED
3 FORM EVENT
1 DEST FAMGATH
...
```

Family Gathering 1.0 Event GEDCOM header.

references

GEDCOM Form
Event GEDCOM Detection

GENERATIONS_OF_ANCESTORS:= {Size=1:4}

The number of generations of ancestors included in this ~~transmission~~ GEDCOM file. This value is usually provided when FamilySearch programs build a GEDCOM file for a patron requesting a download of ancestors.

Generations of Ancestors

The SUBN.ANCE record lists the number of ancestral generations. FamilySearch allows a number of four digits, and does not provide any further restrictions, thus allowing 9999 generations.

The SUBMISSION_RECORD is a FamilySearch-specific GEDCOM record, even a product-specific record, used with the now obsolete FamilySearch TempleReady product. As the description notes, the value is usually provided by FamilySearch.

large number of generations

The number of generations for the largest genealogies is still well below hundred, and thus needs only two digits. The family tree of Confucius documents about 83 generations.

GENERATIONS_OF_DESCENDANTS:= {Size=1:4}

The number of generations of descendants included in this ~~transmission~~ GEDCOM file. This value is usually provided when FamilySearch programs build a GEDCOM file for a patron requesting a download of descendants.

Generations of Descendants

The SUBN.DESC record lists the number of descendent generations. FamilySearch allows a number of up to four digits, without any further restrictions, thus allowing 9999 generations.

LANGUAGE_ID:= {Size=1:15}

A table of valid Latin language identification codes.

[Afrikaans | Albanian | Anglo-Saxon | Catalan | Catalan_Spn | Czech | Danish | Dutch | English | Esperanto | Estonian | Faroese | Finnish | French | German | Hawaiian | Hungarian | Icelandic | Indonesian | Italian | Latvian | Lithuanian | Navaho | Norwegian | Polish | Portuguese | Romanian | Serbo_Croa | Slovak | Slovene | Spanish | Swedish | Turkish | Wendic]

Other languages not supported until UNICODE:

[Amharic | Arabic | Armenian | Assamese | Belorusian | Bengali | Braj | Bulgarian | Burmese | Cantonese | Church-Slavic | Dogri | Georgian | Greek | Gujarati | Hebrew | Hindi | Japanese | Kannada | Khmer | Konkani | Korean | Lahnda | Lao | Macedonian | Maithili | Malayalam | ~~Mandrin~~

Mandarin | Manipuri | Marathi | Mewari | Nepali | Oriya | Pahari | Pali | Panjabi | Persian | Prakrit | Pusto | Rajasthani | Russian | Sanskrit | Serb | Tagalog | Tamil | Telugu | Thai | Tibetan | Ukrainian | Urdu | Vietnamese | Yiddish]

LANGUAGE_OF_TEXT:= {Size=1:15}

[<LANGUAGE_ID>]
The human language in which the data in the ~~transmission~~ GEDCOM file is normally read or written. It is used primarily by programs to select language-specific sorting sequences and phonetic name matching algorithms.

LANGUAGE_PREFERENCE:= {Size=1:90}

[<LANGUAGE_ID>]
The language in which a person prefers to communicate. Multiple language preference is shown by using multiple occurrences in order of priority.

LDS_BAPTISM_DATE_STATUS:= {Size=5:10}

[CHILD | COMPLETED | EXCLUDED | PRE-1970 | STILLBORN | SUBMITTED | UNCLEARED]

A code indicating the status of an LDS baptism and confirmation date where:

CHILD = Died before becoming eight years old, baptism not required.
COMPLETED = Completed but the date is not known.
EXCLUDED = Patron excluded this ordinance from being cleared in this submission.
PRE-1970 = Ordinance is likely completed, another ordinance for this person was converted from temple records of work completed before 1970, therefore this ordinance is assumed to be complete until all records are converted.
STILLBORN = Stillborn, baptism not required.
SUBMITTED = Ordinance was previously submitted.
UNCLEARED = Data for clearing ordinance request was insufficient.

<LDS_BAPTISM_DATE_STATUS> values

The EXCLUDED value is new in GEDCOM 5.5.1.
The GEDCOM 5.5 specification lists three additional values: CLEARED, INFANT and QUALIFIED.

The following explanations of these values are taken from the GEDCOM 5.5. specification:

CLEARED = Baptism has been cleared for temple ordinance.
INFANT = Died before less than one year old, baptism not required.
QUALIFIED = Ordinance request qualified by authorized criteria.

LDS_CHILD_SEALING_DATE_STATUS:= {Size=5:10}

[BIC | COMPLETED | EXCLUDED | DNS | PRE-1970 | STILLBORN | SUBMITTED | UNCLEARED]

BIC = Born in the covenant receiving blessing of child to parent sealing.
EXCLUDED = Patron excluded this ordinance from being cleared in this submission.
PRE-1970 = (See pre-1970 under <LDS_BAPTISM_DATE_STATUS> on page 51.)
STILLBORN = Stillborn, not required.

= Ordinance was previously submitted.
= Data for clearing ordinance request was insufficient.

SUBMITTED
UNCLEARED

<LDS_CHILD_SEALING_DATE_STATUS> values

The EXCLUDED value is new in GEDCOM 5.5.1.
The GEDCOM 5.5 specification lists two additional values: CLEARED and QUALIFIED.

The <LDS_CHILD_SEALING_DATE_STATUS> values COMPLETED and DNS are missing from the explanatory table.
The following explanations are taken from the GEDCOM 5.5. specification:

CLEARED = Baptism has been cleared for temple ordinance.
COMPLETED = Completed but the date is not known.
DNS = This record is not being submitted for this temple ordinances.
QUALIFIED = Ordinance request qualified by authorized criteria.

LDS_ENDOWMENT_DATE_STATUS:= {Size=5:10}

[CHILD | COMPLETED | EXCLUDED | PRE-1970 | STILLBORN | SUBMITTED | UNCLEARED]

A code indicating the status of an LDS endowment ordinance where:

CHILD = Died before eight years old.
COMPLETED = Completed but the date is not known.
EXCLUDED = Patron excluded this ordinance from being cleared in this submission.
INFANT = Died before less than one year old, baptism or endowment not required.
PRE-1970 = (See pre-1970 under <LDS_BAPTISM_DATE_STATUS> on page 51.)
STILLBORN = Stillborn, ordinance not required.
SUBMITTED = Ordinance was previously submitted.
UNCLEARED = Data for clearing ordinance request was insufficient.

<LDS_ENDOWMENT_DATE_STATUS> values

The EXCLUDED value is new in GEDCOM 5.5.1.
The GEDCOM 5.5 specification lists three additional values: CLEARED, INFANT and QUALIFIED.

The following explanations of these values are taken from the GEDCOM 5.5. specification:

CLEARED = Baptism has been cleared for temple ordinance.
INFANT = Died before less than one year old, baptism not required.
QUALIFIED = Ordinance request qualified by authorized criteria.

LDS_SPOUSE_SEALING_DATE_STATUS:= {Size=3:10}

[CANCELED | COMPLETED | DNS | EXCLUDED | DNS/CAN | PRE-1970 | SUBMITTED | UNCLEARED]

CANCELED = Canceled and considered invalid.
COMPLETED = Completed but the date is not known.
EXCLUDED = Patron excluded this ordinance from being cleared in this submission.
DNS = This ordinance is not authorized.
DNS/CAN = This ordinance is not authorized, previous sealing cancelled.
PRE-1970 = (See pre-1970 under <LDS_BAPTISM_DATE_STATUS> on page 51.)
SUBMITTED = Ordinance was previously submitted.
UNCLEARED = Data for clearing ordinance request was insufficient.

<LDS_SPOUSE_SEALING_DATE_STATUS> values

The EXCLUDED value is new in GEDCOM 5.5.1.
The GEDCOM 5.5 specification lists two additional values: CLEARED and QUALIFIED.

The following explanations of these values are taken from the GEDCOM 5.5. specification:

CLEARED = Baptism has been cleared for temple ordinance.
QUALIFIED

= Ordinance request qualified by authorized criteria.

Cancelled

CANCELED is the American spelling of CANCELLED. It is recommended that GEDCOM readers that support the obsolete <LDS_SPOUSE_SEALING_DATE_STATUS> accept both the English spelling and the American spelling.

MONTH:= {Size=3}

[JAN | FEB | MAR | APR | MAY | JUN | JUL | AUG | SEP | OCT | NOV | DEC]

Where:

JAN	= January
FEB	= February
MAR	= March
APR	= April
MAY	= May
JUN	= June
JUL	= July
AUG	= August
SEP	= September
OCT	= October
NOV	= November
DEC	= December

MONTH_FREN:= {Size=4}

[VEND | BRUM | FRIM | NIVO | PLUV | VENT | GERM | FLOR | PRAI | MESS | THER | FRUC | COMP]

Where:

VEND	= VENDEMIAIRE
BRUM	= BRUMAIRE
FRIM	= FRIMAIRE
NIVO	= NIVOSE
PLUV	= PLUVIOSE
VENT	= VENTOSE
GERM	= GERMINAL
FLOR	= FLOREAL
PRAI	= PRAIRIAL
MESS	= MESSIDOR
THER	= THERMIDOR
FRUC	= FRUCTIDOR
COMP	= JOUR_COMPLEMENTAIRS

MONTH_HEBR:= {Size=3}

[TSH | CSH | KSL | TVT | SHV | ADR | ADS |

NSN | IYR | SVN | TMZ | AAV | ELL]

Where:

TSH	= Tishri
CSH	= Cheshvan
KSL	= Kislev
TVT	= Tevet
SHV	= Shevat
ADR	= Adar
ADS	= Adar Sheni
NSN	= Nisan
IYR	= Iyar
SVN	= Sivan
TMZ	= Tammuz
AAV	= Av
ELL	= Elul

53

54

MULTIMEDIA_FILE_REFERENCE:= ~~{Size=1:30}~~ {Size=1:259}

A complete local or remote file reference to the auxiliary data to be linked to the GEDCOM context. Remote reference would include a network address where the multimedia data may be obtained.

multimedia file

The <MULTIMEDIA_FILE_REFERENCE> definition is for multimedia files only. The GEDCOM specification has a separate <FILE_NAME> definition for the HEAD.FILE line value.

maximum path length

The <FILE_NAME> fails to mention whether the file should include just the file name, or the entire file path. The <MULTIMEDIA_FILE_REFERENCE> specification is clear: it must be a “complete local or remote file reference”, i.e. a full path.

maximum length

Arguably, the GEDCOM specification should not impose a maximum length on file paths, but instead merely demand that the file path is legal and - at least in the context of the source system - valid.

The lineage-linked specification for <MULTIMEDIA_FILE_REFERENCE> states that the file name should be a full path, yet allows only 30 characters. That is not a realistic maximum length for a full path. This is a mistake, plain and simple.

It is strongly suggested that GEDCOM readers, writers and validators treat the maximum length as 259 characters.
The length of 259 characters corresponds to the maximum path length for Windows APIs

without long path support; `MAX_PATH -1`, the minus 1 is because `MAX_PATH` includes space for a terminating null character.
This is a legacy limitation; the NTFS file system and Windows APIs with long path support allow paths up to 32K.

CONC and CONT

The maximum <MULTIMEDIA_FILE_REFERENCE> length that will fit on a single CR/LF-terminated GEDCOM line is 247 characters. The GEDCOM grammar supports longer record values through the CONC tag. However, because of the misleading explicit inclusion of CONC and CONT tags in the lineage-linked specification, some GEDCOM readers still feature limited support for CONC.

A GEDCOM reader that supports CONC correctly, is able to accept file paths of any length. Vendors whose products do not fully support CONC and CONT yet, are strongly encouraged to fix their GEDCOM readers.

case-preserving

The specification does not state whether file names are case sensitive.
Whether file names are case-sensitive is platform-dependant. Make sure your file name code is case-preserving.

references

MSDN: Naming Files, Paths and Namespaces

MULTIMEDIA_FORMAT:= {Size=3:4}

[bmp | gif | jpg | ole | pcx | tif | wav]
Indicates the format of the multimedia data associated with the specific GEDCOM context. This allows processors to determine whether they can process the data object. Any linked files should contain the data required, in the indicated format, to process the file data.

uppercase abbreviations

Although the GEDCOM specification states that a so-called “controlled line_value” (enumeration) is case-insensitive, the use of lower case for these enumeration values is arguably inappropriate. These values are abbreviations that should be in upper case.

multimedia formats

The description states that <MULTIMEDIA_FORMAT> indicates the format of the multimedia data, but the meaning of the values remains undefined, and few developers will immediately recognise all of them.
The description should have included a helpful table like this:

`BMP`	= BitMaP (Windows)
`GIF`	= Graphics Interchange Format (CompuServe)
`JPG`	= Joint Photographic Experts Group
`OLE`	= Object Linking & Embedding (Windows)
`PCX`	= Personal Computer eXchange (PaintBrush)
`TIF`	= Tagged Image File Format
`WAV`	= WAVeform audio file format (Windows)

more multimedia formats

The FamilySearch GEDCOM 5.5.1 specification dates from 1999, but the list of values shown here is, apart from the abbreviation change, identical to that in the GEDCOM 5.5 specification from 1995. Some formats are seriously dated, while many major multimedia formats are missing. Some of the missing formats include:

`AAC`	= Advanced Audio Codec
`AVI`	= Audio Video Interleaved (Windows)
`ePUB`	= Electronic Publication (ebook)
`FLAC`	= Free Lossless Audio Codec
`PDF`	= Portable Document Format
`PNG`	= Portable Network Graphics
`MKV`	= Matroska Video Container
`mobi`	= MobiPocket (ebook)
`MP3`	= MPEG-2 Audio Layer III

<MULTIMEDIA_FORMAT> length

Oddly, although all the values are three characters long, the length of <MULTIMEDIA_FORMAT> is specified as {Size=3:4}; either 3 or 4 characters.
The key to this inconsistency are the `JPG` and `TIF` values. The common abbreviation for Joint Photographic Experts Group is JPEG, and the common abbreviation for Tagged Image File Format is TIFF. The three-letter file extensions for JPEG and TIFF are JPG and TIF, and as a result, those abbreviations sometimes occur elsewhere.
The GEDCOM 5.5 specification includes the four-letter values `JPEG` and `TIFF`, while the GEDCOM 5.5.1 specification includes the three-letter values `JPG` and `TIF`. Some FamilySearch editor changed `JPEG` and `TIFF` to `JPG` and `TIF`, without changing {Size=3:4} to {Size=3}.

JPG versus JPEG, TIF versus TIFF

The odd situation created by this edit is that GEDCOM 5.5 has the four-letter values `JPEG` and `TIFF`, but GEDCOM 5.5.1 has the three-letter values `JPG` and `TIF`. Those two three-letter values are invalid in GEDCOM 5.5 while the four-letter values are invalid in GEDCOM 5.5.1.

best practice

It is strongly suggested that GEDCOM readers support both the four-letter values `JPEG` and `TIFF`, and the three-letter values `JPG` and `TIF`. Upon encountering the wrong abbreviation, say encountering `TIFF` instead of `TIF`, a GEDCOM reader should issue a non-fatal error, and then continue processing as if it had encountered the correct one.

NAME_OF_BUSINESS:= {Size=1:90}

Name of the business, corporation, or person that produced or commissioned the product.

NAME_OF_FAMILY_FILE:= {Size=1:120}

Name under which family names for ordinances are stored in the temple's family file.

NAME_OF_PRODUCT:= {Size=1:90}

The name of the software product that produced this ~~transmission~~ GEDCOM file.

NAME_OF_REPOSITORY:= {Size=1:90}

The official name of the archive in which the stated source material is stored.

NAME_OF_SOURCE_DATA:= {Size=1:90}

The name of the electronic data source that was used to obtain the data in this ~~transmission~~ GEDCOM file. For example, the data may have been obtained from a CD-ROM disc that was named "U.S. 1880 CENSUS CD-ROM vol. 13."

NAME_PERSONAL:= {Size=1:120}

[<NAME_TEXT> |
/<NAME_TEXT>/ |
<NAME_TEXT> /<NAME_TEXT>/ |
/<NAME_TEXT>/ <NAME_TEXT> |
<NAME_TEXT> /<NAME_TEXT>/ <NAME_TEXT>
]

The surname of an individual, if known, is enclosed between two slash (/) characters. The order of the name parts should be the order that the person would, by custom of their culture, have used when giving it to a recorder. Early versions of Personal Ancestral File ® and other products did not use the trailing slash when the surname was the last element of the name. If part of a name is illegible, that part is indicated by an ellipsis (...). Capitalize the name of a person or place in the conventional manner — capitalize the first letter of each part and lowercase the other letters, unless conventional usage is otherwise. For example: McMurray.

54

55

Examples :
William Lee (given name only or surname not known)
/Parry/ (surname only)
William Lee /Parry/
William Lee /Mac Parry/ (both parts (Mac and Parry) are surname parts
William /Lee/ Parry (surname ~~i~~eembedded in the name string)
William Lee /Pa.../

name guidelines

The FamilySearch GEDCOM specification for <NAME_PERSONAL> includes a few quick guidelines on how to record names. The key guideline provided here is to capitalise names in the conventional manner (so not in ALL-CAPS).
The article *Genealogy Name Basics* provides more *do's and don'ts*.

references

Genealogy Name Basics
Five Freaky Features Your Genealogy Software should *not* have

NAME_PHONETIC_VARIATION:= {Size=1:120}

The phonetic variation of the name is written in the same form as the was the name used in the superior <NAME_PERSONAL> primitive, but phonetically written using the method indicated by the subordinate <PHONETIC_TYPE> value, for example if hiragana was used to provide a ~~reading~~ phonetic transcription of a name written in kanji, then the <PHONETIC_TYPE> value would indicate ‘kana’. See page 57.

NAME_PIECE:= {Size=1:90}

The piece of the name pertaining to the name part of interest. The surname part, the given name part, the name prefix part, or the name suffix part.

NAME_PIECE_GIVEN:= {Size=1:120}

[<NAME_PIECE> | <NAME_PIECE_GIVEN>, <NAME_PIECE>] Given name or earned name. Different given names are separated by a comma.

NAME_PIECE_NICKNAME:= {Size=1:30}

[<NAME_PIECE> | <NAME_PIECE_NICKNAME>, <NAME_PIECE>] A descriptive or familiar name used in connection with one's proper name.

NAME_PIECE_PREFIX:= {Size=1:30}

[<NAME_PIECE> | <NAME_PIECE_PREFIX>, <NAME_PIECE>]
Non indexing name piece that appears preceding the given name and surname parts. Different name prefix parts are separated by a comma.
For example:
Lt. Cmndr. Joseph /Allen/ jr.
In this example **Lt. Cmndr.** is considered as the name prefix portion.

nobility titles

The <NAME_PIECE_PREFIX> should not be used for recording nobility titles. Nobility titles must be recorded as the <NOBILITY_TYPE_TITLE> line value of an INDI.TITL record.

NAME_PIECE_SUFFIX:= {Size=1:30}

55

56

[<NAME_PIECE> | <NAME_PIECE_SUFFIX>, <NAME_PIECE>]
Non-indexing name piece that appears after the given name and surname parts. Different name suffix parts are separated by a comma.
For example:
Lt. Cmndr. Joseph /Allen/ **jr.**
In this example **jr.** is considered as the name suffix portion.

NAME_PIECE_SURNAME:= {Size=1:120}

[<NAME_PIECE> | <NAME_PIECE_SURNAME>, <NAME_PIECE>] Surname or family name. Different surnames are separated by a comma.

NAME_PIECE_SURNAME_PREFIX:= {Size=1:30}

[<NAME_PIECE> | <NAME_PIECE_SURNAME_PREFIX>, <NAME_PIECE>] Surname prefix or article used in a family name. ~~Different surname articles are separated by a comma, for example in the name "de la Cruz", this value would be "de, la".~~

do not mess up prefixes

FamilySearch's instruction to insert commas in prefixes is not just unmotivated, it is wrong. You should never modify names. As far as I know, there is no genealogy application that complies with this erroneous instruction.

best practice

Record the prefix *as is* to ensure it will transfer unmodified.

Modifying the prefix by inserting one or more commas is almost sure to create problems. Most genealogy applications do not expect commas in any name part.

NAME_ROMANIZED_VARIATION:= {Size=1:120}

The romanized variation of the name is written in the same form prescribed for the name used in the superior <NAME_PERSONAL> context. The method used to romanize the name is indicated by the line_value of the subordinate <ROMANIZED_TYPE>, for example if romaji was used to provide a ~~reading~~ transliteration of a name written in kanji, then the <ROMANIZED_TYPE> subordinate to the ROMN tag would indicate romaji. See page 61.

NAME_TEXT:= {Size=1:120}

<TEXT> excluding commas, numbers, special characters not considered diacritics.

NAME_TYPE:= {Size=5:30}

[aka | birth | immigrant | maiden | married | <user defined>]

~~_~~

Indicates the name type, for example the name issued or assumed as an immigrant.

aka	= also known as, alias, etc.
birth	= name given on birth certificate.
immigrant	= name assumed at the time of immigration.
maiden	= maiden name, name before first marriage.
married	= name was persons previous married name.
user~~_~~ defined	= other text name that defines the name type.

user defined

The FamilySearch GEDCOM 5.5.1 syntax for <NAME_TYPE> has “user defined” (two words separated by a space), while the description has “user_defined” (two words connected by an underscore). The description has been corrected to use a space, to be consistent with all other occurrences of “user defined”.

NATIONAL_ID_NUMBER:= {Size=1:30}

A nationally-controlled number assigned to an individual. ~~Commonly known national numbers should be assigned their own tag, such as SSN for U.S. Social Security Number.~~ This IDNO line value must be used for all third party numbers. The use of the IDNO tag requires a subordinate TYPE tag to identify what kind of number is being stored.

For example:

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

1 IDNO 43-456-1899
2 TYPE Canadian Health Registration

GEDCOM fragment fixed

The purported GEDCOM fragment isn't GEDCOM, but a mix of GEDCOM and lineage-linked syntax (the line numbers). It has been fixed to be true GEDCOM fragment.

not just national numbers

There is no reason to restrict the IDNO record to national numbers. It should be used for international numbers as well. In fact, IDNO should be used for all third-party numbers, and REFN for user-defined numbers.

IDNO versus SSN

The <NATIONAL_ID_NUMBER> is the line value of the IDNO record.

The FamilySearch definition here states that “Commonly known national numbers should be assigned their own tag, such as SSN for U.S. Social Security Number”. That is a very bad idea. Creation of separate tags for each national number leads to an unmanageable tag explosion, and that is highly undesirable.
GEDCOM does have an SSN tag, and that is a mistake.

NATIONAL_OR_TRIBAL_ORIGIN:= {Size=1:120}

The person's division of national origin or other folk, house, kindred, lineage, or tribal interest. Examples: Irish, Swede, Egyptian Coptic, Sioux Dakota Rosebud, Apache Chiricawa, Navajo Bitter Water, Eastern Cherokee Taliwa Wolf, and so forth.

NEW_TAG:= {Size=1:15}

A user-defined tag that is contained in the ~~GEDCOM current transmission~~ current GEDCOM file. This tag must begin with

56

57

an underscore (_) ~~and should only be interpreted in the context of the sending system.~~ The interpretation of vendor-defined tags depends on the GEDCOM dialect (as specified by HEAD.DEST). The interpretation of truly *user*-defined tags is anyone's guess.

user-defined tags

There are vendor-defined tags and user-defined tags. The FamilySearch GEDCOM specification does not distinguish between those two categories. When FamilySearch GEDCOM mentions user-defined tags, they really means vendor- and product-specific tags, but it is not uncommon for genealogy applications to allow users to define their own tags.

reading third-party extensions

On first reading, the text “should only be interpreted in the context of the sending system” seems to say, using a somewhat odd expression, that only the application that created the GEDCOM extension is allowed to interpret that extension. In other words: that third-party applications should not interpret product-specific tags. That is a bad rule.

Although it is perfectly legal for any genealogy application to ignore so-called user-defined tags, that generally isn't what users want or expect. In practice, many vendors try to support the most important GEDCOM extensions of other vendors on GEDCOM import, as that provides a better user experience than ignoring them. Vendors with products that understand GEDCOM extensions used by other products often tout the product's ability to import GEDCOM files created by those products as a selling point.

not the source, but the destination

FamilySearch probably meant to say something different, something sensible that should be obvious to developers without stating it: the interpretation of vendor- or product-specific

extensions depends on the product, and that product is identified by the HEAD.SOUR line value.
If so, FamilySearch did not think it through, because, however reasonable that statement may sound, it is wrong. The HEAD.SOUR line value identifies the product that created the GEDCOM file - and that is all it identifies. The intended interpretation of GEDCOM extensions is indicated by the HEAD.DEST line value.

references

GEDCOM SOUR and DEST
GEDCOM System Identifiers
Behold blog 2011-11-21: A Plethora of Extra GEDCOM Tags

NOBILITY_TYPE_TITLE:= {Size=1:120}

The title given to or used by a person, especially of royalty or other noble class within a locality.

NULL:= {Size=0:0}

A convention that indicates the absence of any ~~8-bit ASCII~~ character in the value including the null character (0x00) which is prohibited.

NUMBER:= {Size=3:4}

[<DIGIT> | <NUMBER>+<DIGIT>]

<NUMBER> Size

The <NUMBER> definition lacks a size specification; FamilySearch failed to provide a minimum or maximum length.

<NUMBER> only occurs in the definition of <YEAR_GREG>. As a year is at least three digits long, and <YEAR_GREG> is at most 7 characters long (in the case of dual-style dates, which always adds exactly three characters), <NUMBER> is at most 4 digits long.
That maximum length makes sense; 4 digits is plenty for the next 8 millennia: GEDCOM 5.5.1 is from 1999 CE and will surely have been replaced before 10.000 CE.

OCCUPATION:= {Size=1:90}

The kind of activity that an individual does for a job, profession, or principal activity.

ORDINANCE_PROCESS_FLAG:= {Size=2:3}

[yes | no]
A flag that indicates whether submission should be processed for clearing temple ordinances.

d

ordinance process flag

It is oddly inconsistent that the lines values `yes` and `no` are used here, while other GEDCOM records use `Y` and `N` as line values.

PEDIGREE_LINKAGE_TYPE:= {Size=5:7}

[adopted | birth | foster ~~| sealing~~]

A code used to indicate the child to family relationship for pedigree navigation purposes.
Where:

adopted	= indicates adoptive parents.
birth	= indicates birth parents.
foster	= indicates child was included in a foster or guardian family.
~~sealing~~	= ~~indicates child was sealed to parents other than birth parents.~~

sealing isn't a pedigree linkage type

The FamilySearch GEDCOM specification is conceptually confused here. “Sealing” is an LDS rite; that is an event, not a pedigree linkage type. A sealing is a registration of imagined family relationship after death. A so-called sealing of a child to parents does not affect the biological tree, official tree or legal tree; it does not affect the genealogy at all.

A “sealing” need not be registered in a genealogy, but if you choose to register it, you should register it as an event. The <PEDIGREE_LINKAGE_TYPE> `sealing` value is a conceptual error and should never be used.

PERMANENT_RECORD_FILE_NUMBER:= {Size=1:90}

<REGISTERED_RESOURCE_IDENTIFIER>:<RECORD_IDENTIFIER>
The record number that uniquely identifies this record within a registered network resource. The number will be usable as a cross-reference pointer. The use of the colon (:) is reserved to indicate the separation of the "registered resource identifier" (which precedes the colon) and the unique "record identifier"within that resource (which follows the colon). If the colon is used, implementations that check pointers should not expect to find a matching cross-reference identifier in the ~~transmission~~ GEDCOM file but would find it in the indicated database within a network. Making resource files available to a public network is a future implementation.

permanent record file number

The <PERMANENT_RECORD_FILE_NUMBER> is a “future implementation” idea that never happened.
The FamilySearch GEDCOM specification reserved the colon syntax for a future feature that never came to be, in which a cross-reference consists of a <REGISTERED_RESOURCE_IDENTIFIER> and <RECORD_IDENTIFIER> within that resource, separated by a colon.

This is an obsolete future feature.

PHONE_NUMBER:= {Size=1:25}

A phone number.

PHONETIC_TYPE:= {Size=5:30}

[<user defined> | hangul | kana]

Indicates the method used in transforming the text to the phonetic variation.

<user defined> = record method used to arrive at the phonetic variation of the name.
hangul = Phonetic method for ~~sounding~~ transcribing Korean ~~glifs~~ glyphs.
kana = Hiragana and/or Katakana characters were used in ~~sounding~~ transcribing the

kanji character used by Japanese

PHYSICAL_DESCRIPTION:= {Size=1:248}

An unstructured list of the attributes that describe the physical characteristics of a person, place, or object. Commas separate each attribute.
Example:

```
1 DSCR Hair Brown, Eyes Brown, Height 5 ft 8 in
2 DATE 23 JUL 1935
```

PLACE_HIERARCHY:= {Size=1:120}

This shows the jurisdictional entities that are named in a sequence from the lowest to the highest jurisdiction. The jurisdictions are separated by commas, and any jurisdiction's name that is missing is still accounted for by a comma. When a PLAC.FORM structure is included in the <<HEADER>> of a GEDCOM ~~transmission~~ file, it implies that all place names follow this jurisdictional format and each jurisdiction is accounted for by a comma, whether the name is known or not. When the PLAC.FORM is subordinate to an event, it temporarily overrides the implications made by the PLAC.FORM structure stated in the <<HEADER>>. This usage is not common and, therefore, not encouraged. It should only be used when a system has over-structured its place-names.

<PLACE_HIERARCHY>

Specifying the structure of place name isn't a bad idea at all, but FamilySearch's <PLACE_HIERARCHY> fails to do that. The FamilySearch GEDCOM specification fails to specify what values a GEDCOM writer can use within a PLAC.FORM line value, or how a GEDCOM reader should interpret that line value.
Thus, it is not really possible to communicate anything about the place name structure from one genealogy application to another.

The text states that the format specified by HEAD.PLAC.FORM applies to all place names in the entire GEDCOM file, unless explicitly overridden by another PLAC.FORM - but that you really should not override it, because it is not common to do so. That is, when taken literally, nonsense; that something isn't common is no reason not to do it. Surely, when it is the right to do, you should do it.
What the FamilySearch author is failing to state is that support for PLAC.FORM is practically non-existent.

three problems

These are the three problems with the <PLACE_HIERARCHY> definition. Firstly, all the place names in the GEDCOM file would have to follow the specified format exactly, which is too restrictive in practice. Secondly, the <PLACE_HIERARCHY> fails to actually specify anything, because it is insufficiently defined, and therefore impossible to implement. So, thirdly, almost no application even tried to implement it.
The simple, practical truth is that attempts at PLAC.FORM support are practically non-existent because it is impossible to support; the HEAD.PLAC.FORM line value literally has no meaning.

Consider <PLACE_HIERARCHY> strongly deprecated.

PLACE_LATITUDE:= ~~{Size=5:8}~~{Size=2:10}

The value specifying the latitudinal coordinate of the place name. The latitude coordinate is the direction North or South from the equator in degrees and fraction of degrees carried out to give the desired accuracy. For example: 18 degrees, 9 minutes, and 3.4 seconds North would be formatted as N18.150944. Minutes and seconds are converted by dividing the minutes value by 60 and the seconds value by 3600 and adding the results together. This sum becomes the fractional part of the degree’s value.

<PLACE_LATITUDE> size

The <PLACE_LATITUDE> size is specified as {Size=5:8}. The example given in the description, “N18.150944” is 10 code units long.

The size specification for <PLACE_LONGITUDE> is {1:60}, which is an impossibly small minimum combined with a needlessly large maximum.

The correct size specification is {2:10}. The shortest possible value is “N0”, the longest possible value, with 6 decimal precision is “N89.123456”.

PLACE_LIVING_ORDINANCE:= {Size=1:120}

<PLACE_NAME>
The locality of the place where a living LDS ordinance took place. Typically, a living LDS baptism place would be recorded in this field.

PLACE_LONGITUDE:= ~~{Size=1:60}~~{Size=2:11}

The value specifying the longitudinal coordinate of the place name. The longitude coordinate is Degrees and fraction of degrees east or west of the zero or base meridian coordinate. For example: 168 degrees, 9 minutes, and 3.4 seconds East would be formatted as E168.150944.

<PLACE_LONGITUDE> size

The <PLACE_LATITUDE> size is specified as {Size=1:60}. That is an impossibly small minimum, and needlessly large maximum length. The GEDCOM 5.6 specification changes the size specification to {5:8}, which makes it identical to the size specification for <PLACE_LATITUDE>, but that is not correct either. The example given in the description, “E168.15094” is 10 code units long.

The correct size specification is {2:11}. The shortest possible value is “E0”, the longest possible value, with 6 decimal precision is “E179.123456”.

latitude and longitude standard

The FamilySearch GEDCOM specification provides a brief specification of how to record latitude and longitude in the <PLACE_LATITUDE> and <PLACE_LONGITUDE> definitions.

Standard formats for geographical coordinates are defined ISO 6709 *Standard representation of geographic point location by coordinates*.
This format always uses a full stop (.), never a comma (,), as a decimal stop.
Use of the characters N and S for North and South, and E and W for East and West as sign characters is defined in Annex H.

Several applications use the plus (+) and minus (-) sign as sign characters. GEDCOM readers should accept both styles.

PLACE_NAME:= {Size=1:120}

[

<PLACE_TEXT> |
<PLACE_TEXT>, <PLACE_NAME>
]

The jurisdictional name of the place where the event took place. Jurisdictions are separated by commas, for example, "Cove, Cache, Utah, USA~~.~~". If the actual jurisdictional names of these places have been identified, they can be shown using a PLAC.FORM structure either in the HEADER or in the event structure. (See <PLACE_HIERARCHY>, page 58.)

place name: recursive definition

That the last item in the <PLACE_NAME> definition is <PLACE_NAME> again instead of <PLACE_TEXT> isn't an error. This is a recursive definition. What it says is that a place name exists of multiple place parts, separated by commas.

The definition is still wrong, or at least less than crystal clear. The intended separator isn't a comma, but a comma followed by a space.
Place names are not terminated by a full stop or anything else.

bad place name example

The <PLACE_NAME> example, "Cove, Cache, Utah, USA.", is bad genealogy practice. The example abbreviates “United States of America” to “USA”; no part of a place name should be abbreviated.

PLACE_PHONETIC_VARIATION:= {Size=1:120}

The phonetic variation of the place name is written in the same form as was the place name used in the superior <PLACE_NAME> primitive, but phonetically written using the method indicated by the subordinate <PHONETIC_TYPE> value, for example if hiragana was used to provide a ~~reading~~ phonetic transcription of a name written in kanji, then the <PHONETIC_TYPE> value would indicate kana. (See PHONETIC_TYPE page 57.)

PLACE_ROMANIZED_VARIATION:= {Size=1:120}

The romanized variation of the place name is written in the same form prescribed for the place name used in the superior <PLACE_NAME> context. The method used to romanize the name is indicated by the line_value of the subordinate <ROMANIZED_TYPE>, for example if romaji was used to provide a ~~reading~~ transliteration of a place name written in kanji, then the <ROMANIZED_TYPE> subordinate to the ROMN tag would indicate ‘romaji’ (See ROMANIZED_TYPE page 61.)

PLACE_TEXT:= {Size=1:120}

<TEXT> excluding the comma(s).

place text

The odd description “<TEXT> excluding the comma(s)” is probably not immediately clear. The name <PLACE_TEXT> is not very helpful either, PLACE_PART_TEXT would have been a better name.

place parts

A fully specified place name exists of several parts, from place name up to country, separated by commas, <PLACE_TEXT> is one such place name part.

commas

As place name parts are separated by commas, including any commas within place names would create confusion.
The description “<TEXT> excluding the comma(s).” means exactly what it says: if the place name part contains commas, leave them out.

FamilySearch GEDCOM cannot support place names that include commas correctly. The FamilySearch GEDCOM creators probably thought such place names do not exist, and later discovered that such names do exist. You should include such place names without their comma, just as the FamilySearch GEDCOM specification demands.

example

An example of a place name that includes a comma is “De Mijl, Krabbe en Nadort” (without quotes), which existed in the province of Noord-Brabant in the Netherlands. That place should be specified as “De Mijl Krabbe en Nadort, Noord-Brabant, Netherlands” (without quotes).

place name guidelines

The FamilySearch GEDCOM specification does not provide any place name guidelines, just the rule about commas in place name parts. *The* basic guideline is to enter place name parts *as-is*, and that means that you should always use full names, never abbreviations, and always use just the actual name, never adorn it (like append “county”) to it.
The article *Place Name Standardisation Basics* provides a more extensive discussion.

references

Place Name Standardisation Basics

POSSESSIONS:= {Size=1:248}

A list of possessions (real estate or other property) belonging to this individual.

PUBLICATION_DATE:= {Size=10:11}

<DATE_EXACT>
The date this source was published or created.

RECEIVING_SYSTEM_NAME:= {Size=1:20}

The name of the system expected to process the GEDCOM-compatible ~~transmission~~ file.

The registered <RECEIVING_SYSTEM_NAME> for all GEDCOM submissions to the Family History Department **must** be one of the following names:

! "ANSTFILE" when submitting to Ancestral File.
! "TempleReady" when submitting for temple ordinance clearance.

receiving system identifier

The FamilySearch GEDCOM specification only mentions what the HEAD.DEST value should be when the GEDCOM file is intended for Ancestral File or TempleReady, two LDS-specific and obsolete systems. Although the HEAD.DEST record is mandatory, FamilySearch completely fails to specify what the HEAD.DEST value should be in all other cases. The few GEDCOM examples within the specification all use `ANSTFILE`. This strongly suggests that FamilySearch created the HEAD.DEST record as another LDS-specific extension, and never considered its meaning outside of their own systems.

Because HEAD.DEST is mandatory, but the FamilySearch GEDCOM specification fails to specify what a proper value would be, vendors have come with nonsense values for HEAD.DEST. Nonsense values used include `ANY`, `GEDCOM`, `GEDCOM55`, and `Other`. FamilySearch's own PAF uses `Other`.
Use of these nonsense values is wrong, as these are not system-identifiers for any known system.

best practice

Here is the simple rule you should follow: In general, the HEAD.DEST line value should be identical to the HEAD.SOUR line value. Applications should always specify their own system identifier as the HEAD.DEST line value when they use their own GEDCOM extensions. After all, it isn't the HEAD.SOUR line value but the HEAD.DEST line value that governs the interpretation of GEDCOM extensions. Applications should only specify the system identifier of another product as the HEAD.DEST line value when explicitly creating a GEDCOM file for interpretation by that other product.

Known nonsense values should result in a non-fatal error, and then be interpreted as specifying the HEAD.SOUR line value. To avoid complications, known nonsense values should not be used as system identifiers for any new system.

references

- GEDCOM SOUR and DEST
- GEDCOM System Identifiers

RECORD_IDENTIFIER:= {Size=1:18}

An identification number assigned to each record within a specific database. The database to which the <RECORD_IDENTIFIER> pertains is indicated by the <REGISTERED_RESOURCE_NUMBER> which

59

60

precedes the colon (:). If the <RECORD_IDENTIFIER> is not preceded by a colon, it is a reference to a record within the current GEDCOM ~~transmission~~ file.

<REGISTERED_RESOURCE_NUMBER>

This <RECORD_IDENTIFIER> definition mentions a <REGISTERED_RESOURCE_NUMBER>, but the lineage-linked form doesn't define that. The <REGISTERED_RESOURCE_NUMBER> reference should be a reference to <REGISTERED_RESOURCE_IDENTIFIER>.

record identifier colon syntax

The FamilySearch GEDCOM specification reserved the cross-reference colon syntax for a future feature that never came to be, in which a cross-reference consists of a <REGISTERED_RESOURCE_IDENTIFIER> and <RECORD_IDENTIFIER> within that resource, separated by a colon.

In practice, any <RECORD_IDENTIFIER> containing a colon should be reported as an error.

REGISTERED_RESOURCE_IDENTIFIER:= {Size=1:25}

This is an identifier assigned to a resource database that is available through access to a network. This is for future GEDCOM releases.

registered resource identifier

The FamilySearch GEDCOM specification reserved the cross-reference colon syntax for a future feature that never came to be, in which a cross-reference consists of a <REGISTERED_RESOURCE_IDENTIFIER> and <RECORD_IDENTIFIER> within that resource, separated by a colon.

There are no registered resource identifiers. This future feature is obsolete.

RELATION_IS_DESCRIPTOR:= {Size=1:25}

A word or phrase that states object 1's **relation** is object 2. For example you would read the following as "Joe Jacob's ~~great grandson is the submitter pointed to by the @XREF:SUBM@~~ high-school geography teacher is the individual identified by @I551@":

~~0 INDI~~
~~1 NAME Joe /Jacob/~~
~~1 ASSO @<XREF:SUBM>@~~
~~2 RELA great grandson~~

```
0 INDI
1 NAME Joe /Jacob/
1 ASSO @I551@
2 RELA high-school geography teacher
```

ASSO example is quadruply illegal

The ASSO example provided by FamilySearch is quadruply illegal.

- GEDCOM lines may not start with leading white space.
- Within a GEDCOM file, the ASSO line value must be an actual line value, not lineage-linked syntax.
- The <<ASSOCIATION_STRUCTURE>> (ASSO record) demands that the ASSO line value be an <XREF:INDI>, a pointer to an INDI record. It may not be a pointer to the SUBM record.
- Great grandson is a relationship already expressed through the family tree; ASSO may not be used for familial connections already expressed by the family tree.

<RELATION_IS_DESCRIPTOR>

The <RELATION_IS_DESCRIPTOR> is the ASSO.RELA line value, which provides a description of the association created by the <<ASSOCIATION_STRUCTURE>> (ASSO record) it is part of.
Notice that the <RELATION_IS_DESCRIPTOR> *does not* provide any predefined values. It is a free-form text field, by the user for the user.

Applications cannot and should not try to rely on the user or other applications using particular values, or any particular value having any particular meaning.
That said, some common values are: `godfather`, `godmother`, `attendant`, `informant`, `witness` and `Other`, but you may encounter many different values, such as `godparent`, `officiating priest`, including illegal values such as `brother`, or `Grand pere maternel` (maternal grandfather in French).

RELIGIOUS_AFFILIATION:= {Size=1:90}

A name of the religion with which this person, event, or record was affiliated.

RESPONSIBLE_AGENCY:= {Size=1:120}

The organization, institution, corporation, person, or other entity that has responsibility for the associated context. For example, an employer of a person of an associated occupation, or a church that administered rites or events, or an organization responsible for creating and/or archiving records.

RESTRICTION_NOTICE:= {Size=6:7}

[confidential | locked | privacy]

The restriction notice is defined for Ancestral File usage. Ancestral File download GEDCOM files may contain this data.

Where:

confidential = This data was marked as confidential by the user. In some systems data marked as confidential will be treated differently, for example, there might be an option that would stop confidential data from appearing on printed reports or would prevent that information from being exported.

locked = Some records in Ancestral File have been satisfactorily proven by evidence, but because of source conflicts or incorrect traditions, there are repeated attempts to change this record. By arrangement, the Ancestral File custodian can lock a record so that it cannot be changed without an agreement from the person assigned as the steward of such a record. The assigned steward is either the submitter listed for the record or ~~Family History~~ FamilySearch support when no submitter is listed.

privacy = Indicate that information concerning this record is not present due to rights of or an approved request for privacy. For example, data from requested downloads of the

Ancestral File may have individuals marked with ‘privacy’ if they are assumed living, that is they were born within the last 110 years and there isn’t a death date. In certain cases family records may also be marked with the RESN tag of privacy if either individual acting in the role of HUSB or WIFE is assumed living.

obsolete GEDCOM extension

The <RESTRICTION_NOTICE> is not merely an LDS-specific extension but even a product-specific extension, that should never have been included in the GEDCOM specification proper. Ancestral File is an LDS system that was never publicly available. The Ancestral File system and all LDS-extensions are obsolete now.

ROLE_DESCRIPTOR:= {Size=1:25}

A word or phrase that identifies a person's role in an event being described. This should be the same word or phrase, and in the same language, that the recorder used to define the role in the actual record.

ROLE_IN_EVENT:= ~~{Size=1:15}~~ {Size=3:27}

[CHIL | HUSB | WIFE | MOTH | FATH | SPOU | (<ROLE_DESCRIPTOR>)]
Indicates what role this person played in the event that is being cited in this context. For example, if you cite a child's birth record as the source of the mother's name, the value for this field is "MOTH." If you describe the groom of a marriage, the role is "HUSB."~~.~~ If the role is something different than one of the six relationship role tags listed above then enclose the role name within matching parentheses.

<ROLE_IN_EVENT> size

<ROLE_EVENT> allows <ROLE_DESCRIPTOR> within parentheses. The size of <ROLE_DESCRIPTOR> is {1:25}, the enclosing parentheses add 2 to both the minimum and maximum length, so the size of <ROLE_EVENT> must be {3:27}.

<ROLE_IN_EVENT> parentheses

Notice that the six predefined values may not be within parentheses, but that additional, system-defined or user-defined values must be. It is an error to put any of the predefined values within parentheses, or not use parentheses for other values.

ROMANIZED_TYPE:= {Size=5:30}

[<user defined> | pinyin | romaji | wadegiles]
Indicates the method used in transforming the text to a romanized variation.

SCHOLASTIC_ACHIEVEMENT:= {Size=1:248}

A description of a scholastic or educational achievement or pursuit.

SEX_VALUE:= ~~{Size=1:7}~~{Size=1:1}

A code that indicates the sex of the individual:

M	= Male
F	= Female
U	= Undetermined from available records and quite sure that it can ’t be.

<SEX_VALUE>

An additional sex value has come to be used since this specification was created. Genealogy applications must allow users to enter, and their GEDCOM readers must accept one additional sex value; `X` for intersex.

<SEX_VALUE> length

The syntax clearly states that <SEX_VALUE> record must be one of three single-letter values, either `M`, `F` or `U`, yet it also states that the maximum length of this value is 7 characters. The values `M`, `F` and `U` are abbreviations of `MALE`, `FEMALE` or `UNKNOWN`, and that last word is the longest one, exactly 7 seven characters.

The stated length leaves room for the full words, and that may suggest that these are acceptable values too, but *they are not acceptable* in GEDCOM 5.5.1, and weren't acceptable in previous versions of GEDCOM either. A likely explanations for the discrepancy between the actual length of the three possible values and the maximum length is sloppy editing, but it is not impossible that this discrepancy exists because FamilySearch actually had a system that used the full words instead of the one-letter abbreviations.

The FamilySearch GEDCOM 5.5.1 syntax is in error. The minimum and maximum length of <SEX_VALUE> is 1 character.

SOCIAL_SECURITY_NUMBER:= {Size=9:11}

A number assigned to a person in the United States of America for identification purposes.

SSN is USA-specific

There's a minor and a major issue with this definition; the minor issue is that it says “United States”, and the major issue is that it says “United States”.

The minor issue is that “United States” should really be “United States of America”, to remove any possible ambiguity and to set a good example; place name parts should not be abbreviated.

The major issue is that this definition is country-specific. Other countries have social security numbers, but the definition of the SSN record is explicitly country-specific, so it may only be used for American social security numbers.

SSN is ill-conceived and superfluous

ill-conceived

The existence of the SSN record is explained by FamilySearch's <NATIONAL_ID_NUMBER> definition. That definition states that “Commonly known national numbers should be assigned their own tag, such as SSN for U.S. Social Security Number”. That is a seriously bad idea; it explicitly invites the creation of many more country-specific tags, and that would lead to an unmanageable explosion of tags.
The SSN record is an ill-conceived record type.

superfluous

The SSN record is oddly specific, especially when you consider that while a social security number may be handy to have, it isn't a genealogical fact, but merely an identifier in a third-party system. It would be best to have a general record in support of all third-party identifiers - and GEDCOM already offers exactly that; the IDNO record allows users to specify any <NATIONAL_ID_NUMBER>.

The existence of the IDNO record type makes the SSN record superfluous. The obvious IDNO.TYPE line value is `SSN`.

references

GEDCOM SSN

SOURCE_CALL_NUMBER:= {Size=1:120}

An identification or reference description used to file and retrieve items from the holdings of a repository.

SOURCE_DESCRIPTION:= {Size=1:248}

A free form text block used to describe the source from which information was obtained. This text block is used by those systems which cannot use a pointer to a source record. It must contain a descriptive title, who created the work, where and when it was created, and where the source data stored. The developer should encourage users to use an appropriate style for forming this free form bibliographic reference. Developers are encouraged to support the <<SOURCE_RECORD>> method of

61

62

reporting bibliographic reference descriptions.

source description deprecated back in 1995, obsolete now

<SOURCE_DESCRIPTION> was deprecated in 1995 already. The GEDCOM 5.5.1

description clearly states that this free-form text block is only to be used by "those systems which cannot use a pointer to a source record.". Moreover, that condition was already present in GEDCOM 5.5, published back in 1995 CE.

Systems that do not support <<SOURCE_RECORD>> yet are obsolete. There is no real need to continue to support this free-form text field. GEDCOM writers should not produce it, and GEDCOM readers should issue a warning upon encountering it.

SOURCE_DESCRIPTIVE_TITLE:= {Size=1:248}

The title of the work, record, or item and, when appropriate, the title of the larger work or series of which it is a part.

For a *published* work, a book for example, might have a title plus the title of the series of which the book is a part. A magazine article would have a title plus the title of the magazine that published the article.

For An *unpublished* work, such as:

! A letter might include the date, the sender, and the receiver.
! A transaction between a buyer and seller might have their names and the transaction date.
! A family Bible containing genealogical information might have past and present owners and a physical description of the book.
! A personal interview would cite the informant and interviewer.

SOURCE_FILED_BY_ENTRY:= {Size=1:60}

This entry is to provide a short title used for sorting, filing, and retrieving source records.

SOURCE_JURISDICTION_PLACE:= {Size=1:120}

<PLACE_NAME>
The name of the lowest jurisdiction that encompasses all lower-level places named in this source. For example, "Oneida, Idaho" would be used as a source jurisdiction place for events occurring in the various towns within Oneida County. "Idaho" would be the source jurisdiction place if the events recorded took place in other counties as well as Oneida County.

SOURCE_MEDIA_TYPE:= {Size=1:15}

[**audio** | book | card | electronic | fiche | **film** | magazine |
manuscript | map | newspaper | **photo**| tombstone | **video**]
A code, selected from one of the media classifications choices above, that indicates the type of material in which the referenced source is stored.

random bold

There seems to be no discernible rhyme or reason as to why some items are bold and others are not.

SOURCE_ORIGINATOR:= {Size=1:248}

The person, agency, or entity who created the record. For a published work, this could be the author, compiler, transcriber, abstractor, or editor. For an unpublished source, this may be an individual, a government agency, church organization, or private organization, etc.

SOURCE_PUBLICATION_FACTS:= ~~{Size=248}~~ {Size=1:248}

When and where the record was created. For published works, this includes information such as the city of publication, name of the publisher, and year of publication.

For an unpublished work, it includes the date the record was created and the place where it was created. For example, the county and state of residence of a person making a declaration for a pension or the city and state of residence of the writer of a letter.

<SOURCE_PUBLICATION_FACTS> length

The GEDCOM 5.5.1 specification states that the length of <SOURCE_PUBLICATION_FACTS> value *must be* 248 characters (code units): {Size=248}. This is obviously an edit error, already present in GEDCOM 5.5, that was never corrected. The length may be anything from 1 up to and including 248 characters (code units): {Size=1:248}.

SUBMITTER_NAME:= {Size=1:60}

The name of the submitter formatted for display and address generation.

SUBMITTER_REGISTERED_RFN:= {Size=1:30}

A registered number of a submitter of Ancestral File data. This number is used in subsequent submissions or inquiries by the submitter for identification purposes.

<SUBMITTER_REGISTERED_RFN> obsolete

The <SUBMITTER_REGISTERED_RFN> record should never have been part of the GEDCOM specification, but been documented separately, as an LDS-extension. The SUBM.RFN record is not only LDS-specific, but even product-specific; it is a user identifier for Ancestral File. Ancestral File is obsolete now, so <SUBMITTER_REGISTERED_RFN> is obsolete too.

SUBMITTER_TEXT:= {Size=1:248}

Comments or opinions from the submitter.

<SUBMITTER_TEXT>

<SUBMITTER_TEXT> should have been called <NOTE_TEXT>. It has no relation to the submitter record, other than that the submitter record may contain a note record.

TEMPLE_CODE:= {Size=4:5}

An abbreviation of the temple in which LDS temple ordinances were performed. (See Appendix B, page 96.)

TEXT:= {Size=1:248}

A string composed of any valid character from the GEDCOM character set.

TEXT_FROM_SOURCE:= {Size=1:248}

<TEXT>
A verbatim copy of any description contained within the source. This indicates notes or text that are actually contained in the source document, not the submitter's opinion about the source. This should be, from the evidence point of view, "what the original record keeper said" as opposed to the researcher's interpretation. The word TEXT, in this case, means from the text which appeared in the source record including labels.

TIME_VALUE:= {Size=1:12}

[hh:mm:ss.fs]
The time of a specific event, usually a computer-timed event, where:

hh = hours on a 24-hour clock
mm = minutes
ss = seconds (optional)
fs = decimal fraction of a second (optional)

TRANSMISSION_DATE:= {Size=10:11}

<DATE_EXACT>
The date that this ~~transmission~~ GEDCOM file was created.

file creation date

FamilySearch erroneously refers to GEDCOM files as “transmissions”. The <TRANSMISSION_DATE> should be called <FILE_CREATION_DATE> date, because it is the file creation date. The inclusion of the file creation date may seem superfluous, as it should be identical to the file's date stamp, but date stamps tend to change when a file is uploaded or downloaded.

USER_REFERENCE_NUMBER:= {Size=1:20}

A user-defined number or text that the submitter uses to identify this record. For instance, it may be a record number within the submitter's automated or manual system, or it may be a page and position number on a pedigree chart.

user reference number

The <USER_REFERENCE_NUMBER> is the line value of the optional REFN record. The optional REFN occurs in the FAM, INDI, OBJE, NOTE, REPO and SOUR records; so, in all the top-level record types, except the header, trailer, submitter and submission records.

The <USER_REFERENCE_NUMBER> allows users to associate any kind of number with these records. The enumerated top-level record types allow any number of REFN subrecords, so users can associate as many numbers with a record as they like.

INDO versus REFN

The IDNO record is for third-party numbers.
The REFN record is for user-defined numbers.

not limited to digits

While the name of this line value strongly suggests that it should be a number, the line value is not limited to digits. It is free-form text that may include any character. It should really have been called <USER_REFERENCE_IDENTIFIER> instead of <USER_REFERENCE_NUMBER>.

need not be unique

While the examples in the FamilySearch GEDCOM definition strongly suggest that the number should be unique, and while the numbers will often be unique, uniqueness is *not* a requirement. Users can take advantage of the optional REFN records as a tagging mechanism.

USER_REFERENCE_TYPE:= {Size=1:40}

A user-defined definition of the <USER_REFERENCE_NUMBER>.

user reference type

This value is free-form text, but meant to be a value from a relatively short user-defined list.
The <USER_REFERENCE_TYPE> is the line value of the optional REFN.TYPE record; the REFN line value specifies a user reference number, the REFN.TYPE specifies that number's type.
Thus, this allows users to classify the reference numbers they use in any way they see fit.

VERSION_NUMBER:= {Size=1:15}

An identifier that represents the version level assigned to the associated product. It is defined and changed by the creators of the product.

version number

<VERSION_NUMBER> is only used within the <<HEADER>> definition. It used for both the HEAD.SOUR.VERS and HEAD.GEDC.VERSION line values.
It is also used for the ill-conceived optional HEAD.CHAR.VERS record.

The lack of definition provided here allows all version numbers to be arbitrary strings, and that is a rather bad idea. The lack of a fixed format makes it needlessly hard to interpret the version number.
The suggestion that the GEDCOM version number is an arbitrary string is not even debatable, it is plain wrong. The GEDCOM version number definitely isn't an arbitrary string. It is a proper version number.

The FamilySearch GEDCOM specification does not explain why <VERSION_NUMBER> is at most 15 characters long. It's probably an arbitrary choice, but it happens to allow four 3-digit numbers separated by three dots. Thus, it is just the right size for a full *major.minor.revision.build* number.

A typical version number consists of numbers separated by full stops. The full stops used in version numbers are known as dots, but pronounced as “point”. For example, in the version number 2.1, the three character are two, dot and one, while the version number as a whole is pronounced two-point-one.
The dot is only used to separate version number parts from each other; a version number neither begins nor ends with a dot, it always begins and ends with a digit.

GEDCOM version best practice

- Simply use the GEDCOM version number
- Use the full version number; do *not* truncate it

The HEAD.GEDC.VERS annotation provides more extensive discussion and best practices..

product version best practice

- HEAD.SOUR.VERS is the product version record, HEAD.SOUR (the system identifier) is not.
  - Do not change the system identifier between versions
  - Use a single system identifier for all versions of your product
- HEAD.SOUR.VERS is the product version record; HEAD.SOUR.NAME (the product name) is not.
  - Do not put the version number in the product name field
  - It *is* okay to use a *marketing version* in the product name field; e.g. in “Family Tree Maker 2017”, “2017” is the marketing version
- Do not treat the product version number as free text, but as a proper version number.
  - It is *only* the version number. Do not include the product name.
  - It is *just* the version number; the version number already implies a particular Alpha, Beta or General Availability release.
    You can include an Alpha or Beta indicator by appending it to the product name.
- Use the *major.minor.revision.build* format
  - Use up to four numbers separated by dots, e.g. 1.0, 5.2.18, 7.5.5.0
  - Use at most three digits per number
  - Microsoft uses the words build and revision differently; if you are using Microsoft .NET, do not worry about it, simply use *major.minor.build.revision* as defined within .NET.

character set (encoding) version best practice

- Do not use the HEAD.CHAR.VERS record

The HEAD.CHAR.VERS annotation provide more extensive discussion and best practices.

references

Truncated GEDCOM Version
GEDCOM Version Detection

WHERE_WITHIN_SOURCE:= {Size=1:248}

Specific location with in the information referenced. For a published work, this could include the volume of a multi-volume work and the page number(s). For a periodical, it could include volume, issue, and page numbers. For a newspaper, it could include a column number and page number. For an unpublished source or microfilmed works, this could be a film or sheet number, page number, frame number, etc. A census record might have an enumerating district, page number, line number, dwelling number, and family number. The data in this field should be in the form of a label and value pair, such as Label1: value, Label2: value, with each pair being separated by a comma. For example, Film: 1234567, Frame: 344, Line: 28.

XREF:= ~~{Size=1:22}~~ {Size=1:20}

Either a pointer or an unique cross-reference identifier. If this element appears before the tag in a GEDCOM line, then it is a cross-reference identifier. If it appears after the tag in a GEDCOM line, then it is a pointer. The method of delimiting a pointer or cross-reference identifier is to enclose the pointer or cross-reference identifier within at signs (@), for example, @I123@. A <XREF> may not begin with a number sign (#). This is to avoid confusion with an escape sequence prefix (@#). The use of a colon (:) in the <XREF> is reserved for creating future network cross-references and the use of an exclamation (!) is reserved for intra-record pointers. Uniqueness of the cross-reference identifier is required within the ~~transmission file~~ GEDCOM file.

cross-reference character limitations

A cross-reference identifier may not start with a number sign (#). The colon (:) and exclamation mark (!) are reserved for non-existent features; the colon is reserved for the cross-reference colon syntax and the exclamation mark is reserved for the cross-reference exclamation mark syntax.
Both are obsolete non-features, so actual GEDCOM cross-references should not contain any colons or exclamation marks.

cross-reference size

The stated size, {Size=1:22}, is incorrect. A GEDCOM record identifier must be at least 1, and may be at most 20 (not 22) characters long. Within a GEDCOM file, the identifier is enclosed within an opening and closing at-sign (@); thus, an enclosed cross-reference is at least 3 (not 1) and at most 22 characters long.

The correct size specification for cross-reference *without* the enclosing at-signs is {Size=1:20}, while the correct size specification for cross-reference *with* the enclosing at-signs is {Size=3:22}. It is never {Size=1:22}.

Examination of the lineage-linked syntax shows that XREF is the cross-reference identifier without the enclosing at-signs, so the correct size specification is {1:20}.

references

GEDCOM Identifiers: Length

XREF:FAM:= ~~{Size=1:22}~~{Size=1:20}

A pointer to, or a cross-reference identifier of, a ~~fam_record~~ <<FAM_RECORD>>.

<XREF:FAM>

<XREF:FAM> is a cross-reference identifier that points to a <<FAM_RECORD>> (FAM record).

The *cross-reference size* annotation on <XREF> explains why the size of <XREF:FAM> isn't {1:22} but {1:20}.

XREF:INDI:= ~~{Size=1:22}~~{Size=1:20}

A pointer to, or a cross-reference identifier of, an individual record.

<XREF:INDI>

<XREF:INDI> is a cross-reference identifier that points to a <<INDIVIDUAL_RECORD>> (INDI record).

The *cross-reference size* annotation on <XREF> explains why the size of <XREF:INDI> isn't {1:22} but {1:20}.

XREF:NOTE:= ~~{Size=1:60}~~{Size=1:20}

A pointer to, or a cross-reference identifier of, a note record.

<XREF:NOTE>

<XREF:NOTE> is a cross-reference identifier that points to a <<NOTE_RECORD>> (NOTE record).

The *cross-reference size* annotation on <XREF> explains why the size of <XREF:NOTE> isn't {1:22} but {1:20}.
That the originally specified size is {1:60}, instead of {1:22} like it is for most cross-reference types, is another edit error.

XREF:OBJE:= ~~{Size=1:22}~~{Size=1:20}

A pointer to, or a cross-reference identifier of, a multimedia object.

<XREF:OBJE>

<XREF:OBJE> is a cross-reference identifier that points to a <<MULTIMEDIA_RECORD>> (OBJE record).

The *cross-reference size* annotation on <XREF> explains why the size of <XREF:OBJE> isn't {1:22} but {1:20}.

XREF:REPO:= ~~{Size=1:22}~~ {Size=1:20}

A pointer to, or a cross-reference identifier of, a repository record.

<XREF:REPO>

The <XREF:REPO> is cross-reference identifier that points to a <<REPOSITORY_RECORD>> (REPO record).

The *cross-reference size* annotation on <XREF> explains why the size of <XREF:REPO> isn't {1:22} but {1:20}.

XREF:SOUR:= ~~{Size=1:22}~~ {Size=1:20}

A pointer to, or a cross-reference identifier of, a ~~SOURce~~ SOUR (source) record.

<XREF:SOUR>

<XREF:SOUR> is a cross-reference identifier that points to a <<SOURCE_RECORD>> (SOUR record).

The *cross-reference size* annotation on <XREF> explains why the size of <XREF:SOUR> isn't {1:22} but {1:20}.

XREF:SUBM:= ~~{Size=1:22}~~ {Size=1:20}

A pointer to, or a cross-reference identifier of, a ~~SUBMitter~~ SUBM (submitter) record.

<XREF:SUBM>

XREF:SUBM is a cross-reference identifier that points to a <<SUBMITTER_RECORD>> (SUBM record).

The *cross-reference size* annotation on <XREF> explains why the size of <XREF:SUBM>

isn't {1:22} but {1:20}.

XREF:SUBN:= ~~{Size=1:60}~~{Size=1:20}

A pointer to, or a cross-reference identifier of, a ~~SUBmissioN~~ SUBN (submission) record.

<XREF:SUBN>

<XREF:SUBN< is a cross-reference identifier that points to a <<SUBMISSION_RECORD>> (SUBN record).

The *cross-reference size* annotation on <XREF> explains why the size of <XREF:SUBN> isn't {1:22} but {1:20}.
That the originally specified size is {1:60}, instead of {1:22} like it is for most cross-reference types, is another edit error.

YEAR:= {Size=3:4}

A numeric representation of the calendar year in which an event occurred.
Years less than hundred must be padded to at least 3 digits by using leading zeroes.

one- and two-digit years illegal

Notice that a year is a number that is at least 3 characters long; the FamilySearch GEDCOM specification does not allow years of less than 3 digits. One-digit and two-digit years are illegal.

3 or 4 digits

GEDCOM 3.0 and 4.0 demand that all years be 4 digits long. It was only with GEDCOM 5.0 that FamilySearch started to allow 3-digit dates, even while its *How to record dates* section still instructs the reader to enter a 4-digit date:

> Type the day (one or two characters) first, then the month (capitalize the first three letters of the English name of the month; do not use a period at the end), and then the year (four-character numeric year). The day and month may be omitted if unknown.

The demand that year be at least 3 digits long remains present in all subsequent version of GEDCOM, including GEDCOM 5.6.

no year less than 100

There is no year zero in the Julian or Gregorian Calendar, the year 1 CE is preceded by the year 1 BCE.
The demand that years must have at least three digits can be interpreted as the FamilySearch GEDCOM specification not allowing the years 1 CE through 99 CE, nor 99 BCE through 1 BCE.
This interpretation is certainly encouraged by FamilySearch's *PAF 2.1 Family Records Data Structure Description* (23 Jun 1988), which contains the following definition for an 11-bit year field within a 3-byte date field:

```
YEAR (bits 0-11) - A number between 100 and 2047
```

That's a remarkable definition; it explicitly excludes the years 1 through 99, without giving a reason for doing so.

PAF is based on Ancestral Quest 3.0, and one of the changes listed for Ancestral Quest 3.0 on its version history page is “Allowed for dates to preceed 100 AD”.

treat date phrase

The GEDCOM 5.0 specification has this definition of <YEAR>:

> **YEAR**:= {Size=3:4, Type=NUMBER}
>
> A numeric representation of the calendar year in which an event occurred. Years less than 3 digits long will be treated as a number in a phrase.

The second sentence of this definition, “Years less than 3 digits long will be treated as a number in a phrase.”, only occurs in GEDCOM 5.0. It's gone in GEDCOM 5.3, which suggests that FamilySearch reconsidered, and did not think that particular instruction a good idea.

Still, this definition makes it crystal clear that FamilySearch does not like years of two digits or less, that the minimum length of 3 characters is quite deliberate. FamilySearch does not explain this limitation, nor what to do with years smaller than 100.

This restriction does not exist to combat the tendency of people to abbreviate dates, by leaving off the century. Such a genealogy would be so confusing to read, that the author would quickly start using full years. Moreover, genealogy consistency checks will quickly alert users to such a mistake.
The actual reason is a practical one, revealed by PAF's behaviour on entering short years.

Personal Ancestral File

FamilySearch PAF 5.2.18 from 2002 CE is the closest thing we have to a reference implementation. When you enter a year less than 1000 in PAF, PAF will pad it out to 4 digits by using leading zeroes on screen, and export those leading zeroes to GEDCOM.

Additional PAF behaviour reveals *why* FamilySearch does not like years of less than 3 digits. If you enter a year (without leading zeroes) of 31 or less, PAF does not add leading zeroes, but presents a pop-up message-box instead, with the complaint that “The date you typed is not standard”. That's a rather generic message, that PAF even pops up for things that GEDCOM allows, but PAF does not support.

PAF only does this when the year is 31 or less, and that reveals the reason why. Not only are GEDCOM date formats are quite flexible, users from different locals may also format dates differently. PAF simply does not know whether you mean 30 Mar 15 when you type 15 Mar 30. The demand to use at least three digits for years is to avoid ambiguity between years and days.
Years must be at least three digits long to easily distinguish between days and years.

references

- FamilySearch GEDCOM Specifications
- Ancestral Question Version History: Version 3.0 Features

YEAR_GREG:= {Size=3:7}

[<NUMBER> | <NUMBER>/<DIGIT><DIGIT>]
The slash "/" <DIGIT><DIGIT> is a year modifier which shows the possible date alternatives for pre-1752 dates brought about by ~~a~~ changing the beginning of the year from MAR to JAN in the English calendar change of 1752, for example, 15 APR 1699/00. ~~A (B.C.) appended to the <YEAR> indicates a date before the birth of Christ.~~

one- and two-digit years illegal

This <YEAR_GREG> definition demands, just like the <YEAR> definition, that the year is at least 3 digits long.

This definition does not disallow the years 1 through 99, but implies that leading zeroes must be used, to pad the year out to at least three digits.

See the *one- and two digit years illegal* annotation for <YEAR>, directly above, for a brief discussion.

<YEAR_GREG> BC

Not only is the usage of the christian expression “B.C.” (Before Christ) instead of the neutral BCE out of place in a standard already, inclusion of the phrase “before the birth of Christ” makes it even less neutral.

Even mention of then neutral “BCE” would be out of place here, in the definition of <YEAR_GREG>, as “B.C.” is an optional part of <DATE_GREG> and *does not occur* in <YEAR_GREG>.

dual style dates

The sentence about the slash is as clear as mud. Addition of the word “is” and a few other minor corrections helps, but not much. It is not just the slash, but *the slash and the two digits that follows it, taken together*, that the description refers to as a “year modifier”.

What the syntax and the sentence try to describe is an ill-conceived practice, widely known by the even iller conceived misnomers *dual dating* and *double dating*. These names suggests that there are two dates, and that is exactly why they are misnomers; there is just one date.
It is one single date shown in two calendars, the Old Style (OS) calendar and the New Style (NS) calendar. A better name for this awful syntax is would be *dual calendering*, but that already means something else, so I call this *dual style dates*.

bad example

It does not help that the one example used to illustrate the dual style dates, “15 APR 1699/00” is the worst example possible. It is both a pathological boundary case (involving the years 1699 and 1700, not 1699 and 1600) as well as an erroneous use of dual styling; Old Style or New Style, there simply is no confusion what year April 15 is in.

Old Style versus New Style

The Old Style calendar starts the year on March 25, while the New Style starts the year on January 1. The change from Old Style to New Style created some ambiguity; the 21st of February might be in the year 1750 according to the Old Style calendar, but in the year 1751 according to the New Style calendar. I could write “21 Feb 1751” hoping that you assume it is New Style, but I might assume wrongly, and if you believe it to be an Old Style date, you might misinterpret it as 21 Feb 1752 New Style. Writing it “21 Feb 1750/51” disambiguates the date, by explicitly stating that it is 21 Feb 1750 Old Style and 21 Feb 1751 New Style.

usage

Dual styling may be needed for dates in January, February and most of March (up to, but not including March 25). For the rest March, and all of April through December, the Old Style calendar and the New Style calendar agree on the year, so those dates should not and may not be dual-dated.
That is why the example “15 APR 1699/00” is plain wrong, and arguably not a date at all.

The second year, indicated by the two digits that follow the slash, is always the next year, and is always indicated by exactly two digits. So “21 Feb 1750/52” is illegal, “21 Feb 1759/60” involves the years 1759 and 1760, and “21 Feb 1759/0” is illegal.

Julian/Gregorian

There is also a dual-calendar practice used because of the change from the Julian Calendar (JC) to the Gregorian Calendar (GC, usually indicated with CE). For example, “2/12 May 1608” is 2 May 1608 JC and 12 May 1608 CE.
This dual-calendaring practice is not supported by GEDCOM; you must pick a calendar. Within GEDCOM, the Gregorian Calendar is the default calendar, but several other calendars can be specified through the <DATE_CALENDAR_ESCAPE>.

dual style restrictions

The FamilySearch GEDCOM specification fails to specify restrictions on dual style dates.

best practice

Genealogy applications should ensure correct usage by enforcing appropriate restrictions.

- Dual style dates may only be used with the Julian and Gregorian Calendar
- The use of two years is indicated by a slash (never a hyphen or any other character)
- The slash must always be followed by two digits, never more or less
- The year indicated by those digits must be the next year
- Dual style may only be used for otherwise ambiguous dates, so only for Jan 1 through March 24
- Dual style should only be used for the year 1582 and later; earlier dates should simply specify the Julian Calendar
- Dual style may only be used for the year 1923 and earlier

65

66

Compatibility with Other GEDCOM Versions

GEDCOM compatibility is measured on a per tag basis, and depends on how similar the data models are for the two different communicating products and on how consistently they understood and complied with the GEDCOM Standard. A few inconsistencies in the use of specific tags also crept into different releases of the standard itself, due to lack of foresight or inadvertent errors. Within these limits, GEDCOM compatible products can exchange data based on GEDCOM 2.0, 3.0, 4.0, and 5.x. Of course, newer GEDCOM releases significantly extend the data model for which the newer tag contexts will not be supported by older products. Some products have introduced their own variations into their GEDCOM form. This will likely provide unique compatibility problems.

The following are areas in which incompatibilities may arise:

! **Source Structure**:
The ~~SOURce~~ SOUR (source) structure was not supported by GEDCOM in versions before 5.x. However, some products, prior to GEDCOM 5.x, developed a ~~SOURce~~ SOUR (source) structure for citing sources. These structures varied from product to product, which affects how source citations are interpreted. Products based on 5.x GEDCOM, prior to GEDCOM 5.4, may have used the more detailed source structure suggested by the previous 5.x versions. Older systems already handling sources will need to be modified. GEDCOM 5.x draft products are encouraged to update their programs to *The GEDCOM Standard 5.5* as soon as possible.

obsolete GEDCOM 5.5 recommendation

Legacy applications should not be updated to GEDCOM 5.5, but to GEDCOM 5.5.1.

! **FAMC Pointer**:
The INDI.FAMC structure has been modified a lot since GEDCOM 4.0. In previous GEDCOM 5.x versions the FAMC structure may contain subordinate adoption events and/or LDS sealing to parent events. See the compatibility implications concerning the LDS sealing to parent event treated in the

"LDS Ordinances Dates" in the next paragraph.

! **LDS Ordinance Dates**:
The structure for LDS sealing of child to parent was changed in previous GEDCOM 5.x draft versions from the FAM.CHIL.SLGC structure to the INDI.FAMC.SLGC structure. This was to allow access to child sealing information without having to follow a pointer to the family record. Personal Ancestral File 2.31 writes the sealing date in the INDI.FAMC.SLGC structure but reads this information from either format. A new major release of PPersonal Ancestral File will change to the newer approach.

GEDCOM 5.4 places the sealing of child to parent event at the same level as all of the other events that are subordinate to the ~~INDIvidual~~ INDI (individual) tag. If a system is keeping track of which family the individual is sealed to, then a FAMC pointer is additionally inserted subordinate to the SLGC event tag that points to the sealed-to-family.

To accommodate previous GEDCOM imports, systems handling the LDS ordinance events should look for the child sealing information in either INDI.SLGC (see <<SLDS_INDIVIDUAL_ORDINANCE>> page 35, 36, INDI.FAMC.SLGC or FAM.CHIL.SLGC structures. Ancestral File exports did not separate the temple code from the ordinance date. Ordinance dates downloaded from Ancestral File may contain an ordinance date followed by a two digit temple code rather than a separate temple code line.

67

68

GEDCOM 4.x systems used certain key words as part of the ordinance dates. GEDCOM 5.x separated these codes from the dates and specified that they should be values of a subordinate ~~STATus~~ STAT (status) tag. Previous GEDCOM 5.x implementations may have implemented this feature using a TYPE tag instead of the ~~STATus~~ STAT (status) tag. (See <LDS_(ordinance)_DATE_STATUS>, page 51, 52.)

! **Adoption Events**:
In GEDCOM 5.x, the ~~ADOPtion~~ ADOP (adoption) event was moved from the FAM.CHIL structure to the INDI.FAMC.ADOP structure, it also appears in the INDI.ADOP structure. In GEDCOM 5.4 the ADOPtion event appears only as an individual event which optionally contains a FAMC pointer to the adoptive family. Subordinate to this pointer is another ~~ADOPtion~~ ADOP (adoption) tag which indicates whether the HUSB or WIFE in the pointed at family was the adoptive parent (see ~~ADOPTED_BY_WHICH_PARENTS~~ <ADOPTED_BY_WHICH_PARENT> primitive on page 42). Pedigree navigation is provided only by <<CHILD_TO_FAMILY_LINK>> structure found on page 31.

<ADOPTED_BY_WHICH_PARENTS> does not exist

The text mentions the "<ADOPTED_BY_WHICH_PARENTS> primitive on page 42", but there is no <ADOPTED_BY_WHICH_PARENTS> primitive.
There is an <ADOPTED_BY_WHICH_PARENT> (singular, not plural) primitive.

! **Codes in Event Date**:

Some applications, such as Personal Ancestral File, pass key words as part of certain event dates. Some of these key words were *INFANT, CHILD, STILLBORN,* etc. These have to do with being an approximate age at an event.

In this version of GEDCOM, the information has been removed from the date value and specified by an <AGE_AT_EVENT> key word value which indicates a descriptive age value at the time of the enclosing event. (See <AGE_AT_EVENT>, page 42.) For example:

```
1 DEAT
  2 DATE 13 MAY 1984
  2 AGE STILLBORN
```

meaning this person died at age approximately 0 days old.

```
1 DEAT
  2 DATE 13 MAY 1984
  2 AGE INFANT
```

meaning this person died at age less than 1 year old.

! **Multiple Names**:
GEDCOM 5.x requires listing different names in different NAME structures, with the preferred instance first, followed by less preferred names. However, Personal Ancestral File and other products that only handle one name may use only the last instance of a name from a GEDCOM ~~transmission~~ file. This causes the preferred name to be dropped when more than one name is present. The same thing often happens with other multiple-instance tags when only one instance was expected by the receiving system.

multiple names order

GEDCOM 5.4 and earlier stated that, in the case of multiple names, the preferred name should be listed first, to make sure that products that support just one name will pick the preferred one. That requirement was dropped in GEDCOM 5.5, apparently *because* PAF does not pick the first name but the last name.
The unfortunate result of that change is that GEDCOM writers are officially unable to indicate the preferred name. It remains a good idea to maximise compatibility with other genealogy applications by exporting the preferred name first and, on GEDCOM import, treat the first name as the preferred name.

PAF's behaviour is hardly an issue in practice, as PAF is a legacy application on its way out; PAF users mostly export their data from PAF into a GEDCOM file, and hardly anyone imports GEDCOM files into PAF.

GEDCOM writer best practice

- support multiple names
- export the preferred name first

GEDCOM reader best practice

- support multiple names
- treat the first name as the preferred name

PAF _AKA

It *is* an issue that FamilySearch PAF does support multiple names, but only through an _AKA record instead of multiple NAME records.
It is recommend to support import of _AKA records, so that PAF users switching to your product do not lose their data.

references

Also Known As

! **Alias Names**:
One or two systems used the ~~ALIAs~~ ALIA (alias) tag for representing multiple names. This form is not supported in the GEDCOM Standard version 5.5.

"one or two systems"

FamilySearch is decidedly less than honest when it claims that "one or two systems" used the ALIA record for recording multiple names. The truth is that FamilySearch prescribed that usage in previous versions of GEDCOM.
See the *alias ALIA* annotation.

! **Event Structure**:
The address structure, as part of the place structure, provided more detail than desired for the ~~PLACe~~ PLAC (place) structure. Therefore it was removed from beneath the place structure and added to the <<EVENT_DETAIL>> structure at the same level as ~~PLACe~~ PLAC (place). The SITE tag was also eliminated from the ~~PLACe~~ PLAC (place) structure since the site of an event is really part of an address, such as Primary Children's Hospital, 100N Medical Drive, Salt Lake City, Utah.

! **Supplemental Attributes or Facts**:
Sometimes other attributes or facts are used to describe an individual's actions, physical description, employment, education, places of residence, etc. These are not generally thought of as events. However, they are often described like events because they were observed at a particular time and/or place. GEDCOM 5.x lists these attributes under the <<INDIVIDUAL_ATTRIBUTE_STRUCTURE>> on page 33 and allows them to be recorded in the same way as events. The attribute definition allows a value on the same line as the attribute tag. In addition, it allows a subordinate date period, place and/or address, etc. to be transmitted, just as the events are. Previous versions, which handled just a tag and value, can be read as usual by handling the subordinate attribute detail as an exception.

Modifications in Version 5.5 as a result of the 5.4 (draft) review

! Added tags for storing detailed **address pieces** under the **address** structure.

! Added **nickname** and **surname prefix** name pieces to the personal name structure. Removed the convention of specifying a nickname in double quotes. This convention was introduced in GEDCOM 5.4 (draft).

! Added subordinate source citation to the note structure.

! Added encoding rules for including embedded multimedia objects. (Removed in 5.5.1)

! Added a RIN tag to the record structures. The RIN tag is a **record identification** assigned to the record by the source software. Its intended use is to allow for automated access to that record upon receipt of return transactions or other reconciliation processes.

! The meaning of a GEDCOM tag without a value on its line depends on its subordinate context for any assertions intended by the researcher. For example, In an event structure, a subordinate DATE and/or ~~PLACe~~ PLAC (place) value imply that an event happened. However, a subordinate NOTE or ~~SOURce~~ SOUR (source) context by themselves do not imply that the event took place. For a researcher to indicate that an event took place without knowing a date or a place requires that a ~~Y(es)~~ Y (yes) value be added to the event tag line.

Using this convention protects GEDCOM processors which may remove (prune) lines that have no value and also no subordinate lines. A ~~N(o)~~ N (no) value must not be used on an event tag line to assert that the event never happened. This requires the definition of a different tag.

! Returned the calendar escape sequence to support alternate calendars.

! The definition of the date value was refined to include many of the potential ways in which a person may define an imprecise date in a free form text field. Systems which guide users through a date statement should not result in such a precise way of stating an imprecise date. For example, if software was to estimate a marriage date based on an algorithm involving the birth date of the couple's first child, hardly needs to say "EST ABT 1881" but rather "EST 1881."

EST ABT

This text seems confused in more than one way, but we need not worry too much about that, as it isn't the standard proper.

The example value used, “EST ABT 1881” is certainly odd, because it is illegal; the <DATE_APPROXIMATED> definition makes it very clear that an approximated date can contain either `EST` or `ABT`, but not both.

! The following tags were added:
ADR1, ADR2, CITY, NICK, POST, SPFX

ADR1 and ADR2, not ADR3

GEDCOM 5.5 introduced the ADR1 and ADR2 record, the ADR3 record was only introduced in GEDCOM 5.5.

Changes Introduced or Modified in Draft Version 5.4

Some changes introduced in GEDCOM draft version 5.4 are not compatible with earlier 5.x draft forms. Some concepts have been removed with the intent to address them in a future release of GEDCOM. The following features are either new or different:

! The use of the SCHEMA **has been eliminated**. 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.

! The <<EVENT_RECORD>> context **has been eliminated**. This context was intended to support the evidence record concept in the Lineage-Linked GEDCOM Form, which ended up being more complicated than first supposed. Understanding the difference between the role of a source record and the role of a so-called evidence record requires further study.

<<EVENT_RECORD>>

The <<EVENT_RECORD>> is a top-level record defined in the GEDCOM 5.3 draft only. The GEDCOM 5.3 draft was used by Family Tree Maker for Windows version 1.0 through 3.40 and by Family Origins version 1 through 3, but neither product ever created these records.

! **Non-standard tags** (see <NEW_TAG>, page 56) can be used within a GEDCOM ~~transmission~~ file, provided that the first character is an underscore (for example _NUTAG). Non-standard tags should be used only when structured information cannot be represented using existing context. Using a Note field is a more universal way of transmitting genealogical data that does not fit into the standard GEDCOM structure.

! The <<SOURCE_RECORD>> structure **was simplified** into five basic sections: data or classification, author, title, publication facts, and repository. The data or classification section contains facts about the data represented by this source and is used to analyze the collection of sources that the researcher used. The author, title, publication facts, and repository sections provide free-form text blocks that inform subsequent researchers how to access the source data that the original researcher used.

70

71

! **Added** a <<SOURCE_CITATION>> structure subordinate to the fact being cited. It is generally best if the source citation contains only information specific to the fact being cited and then points to the more general description of the source, defined in a <<SOURCE_RECORD>>. This reduces redundancy, provides a way of controlling the GEDCOM record size, and more closely represents the normalized data model.

! Systems that describe sources using the ~~AUTHor~~ AUTH (author), ~~TITLe~~ TITL (title), ~~PUBLication~~ PUBL (publication), and ~~REPOsitory~~ REPO (repository) fields *can and* ***should always pass*** *this information in GEDCOM using a ~~SOURce~~ SOUR (source) record* pointed to by the <<SOURCE_CITATION>>. Systems that only allow free form source notes should encourage forming the source information so that it include text about these categories:

! TITL: A descriptive title of the source
! AUTH: Who created the work
! PUBL: When and where was it created
! REPO: Where can it be obtained or viewed

When possible provide the tag for these categories within the text so that a receiving system could parse them to fit the recommended source/citation structure.~~.~~

structured free-form citations

Note that this item, presented as one of several changes and modification, provides a guideline for presenting structured citations in the line value of a free-form citation record: provide the tags “so that a receiving system could parse them to fit the recommended source/citation structure”.

Thus, what used to be a truly free-form field in which the user could format citations as they saw fit, and is still available as such, is considered a legacy field, a bottleneck for transferring structured citations through.
However, this guideline does not stop users from creating free-form citations as they did

before, and an example of just how the free-form field should be formatted is missing anyway, so systems have to continue to support free-form citations as before.

! Some attributes of individuals such as their EDUC (education), ~~OCCUpation~~ OCCU (occupation), ~~RESIdence~~ RESI (residence), or nobility ~~TITLe~~ TITL (title) need to be described using a date and place. Therefore, the structure to describe these attributes was formatted to be the same as for describing events. That is, these attributes are further defined using a date, place, and other attributes used to describe events. (See <<EVENT_DETAIL>>, page 32 and ~~<<INDIVIDUAL_ATTRIBUTE_STRUCTURE, PAGE 33>>~~ <<INDIVIDUAL_ATTRIBUTE_STRUCTURE>>, page 33.)

! The LDS ordinance structure was extended to include the place of a living LDS ordinance. The TYPE tag line was changed to a **~~STATus~~ STAT (status)** line. This allows statements such as *BIC, canceled, Infant,* and so forth to be **removed from the date** line and be added here under the ~~STATus~~ STAT (status) tag. (See <LDS_*(ordinance)*_DATE_STATUS>, page 51, 52) where *(ordinance)* represents any of the following: BAPTISM, ENDOWMENT, CHILD_SEALING, or SPOUSE_SEALING.

LDS_(ordinance) is shorthand

In other words, LDS_*(ordinance)*_DATE_STATUS> is shorthand for LDS_BAPTISM_DATE_STATUS>, LDS_ENDOWMENT_DATE_STATUS>, LDS_CHILD_SEALING_DATE_STATUS> and LDS_SPOUSE_SEALING_DATE_STATUS>.

! Previous GEDCOM 5.x versions overloaded the FAMC pointer structure with subordinate events which connected individual events and an associated family. An **adoption event**, for example, was shown subordinate to the FAMC pointer to indicate which was the adoptive family. The **sealing of child to parent** event (SLGC) was also shown in this manner. GEDCOM 5.4 recognizes that these are events and should be at the same level as the other individual events. To show the associated family, a subordinate FAMC pointer is placed subordinate to the appropriate event. (See <<INDIVIDUAL_EVENT_STRUCTURE>> page 34 and LDS_INDIVIDUAL_ORDINANCE at page 35, 36.)

sealing is an event

Sealing is an LDS rite. Here, in a list of changes made for GEDCOM 5.4 (1995 CE), FamilySearch correctly notes that the sealing rite isn't a relationship, but an event. Yet, there are several annotations in this GEDCOM 5.5.1 specification (1999 CE) that still have to make exactly that point, because FamilySearch GEDCOM continued to treat sealing as a relationship anyway.

! The date modifier (**int**) was added to the date format to indicate that the associated date phrase has been interpreted and the interpretation follows the **int** prefix in the date field. The date phrase is also included in the date value enclosed in parentheses. (See <DATE_APPROXIMATED>, page 45.)

! The <AGE_AT_EVENT> primitive definition now includes the key words `STILLBORN, INFANT,` and *CHILD*. These words should be interpreted as being an **approximate age** at an event. (See <AGE_AT_EVENT>, page 42.)

! The family event context in the ~~FAMily~~ FAM (family) record now allows the **ages of both** the husband and wife at the time of the event to be shown. (See <<FAMILY_RECORD>>, page 24)

<<FAMILY_RECORD>> does not exist

This enumeration of GEDCOM 5.4 changes mentions a <<FAMILY_RECORD>>, but the specification does not define a <<FAMILY_RECORD>>, it defines a <<FAM_RECORD>>.

! The <<PERSONAL_NAME_STRUCTURE>> structure now allows name pieces to be specifically identified as subordinate parts of the name line. Most products will not use subordinate name pieces. A nickname can now be included on the name line by enclosing it in double quotation marks (changed in 5.5.) *Note:* Systems using the **subordinate name parts must still provide the name structure formed in the same way** specified for <NAME_PERSONAL> (see page 54.)

! A submission record was added to GEDCOM to enable the sending system to transmit information which will enable the receiving system to more appropriately process the GEDCOM data. The format currently designed for the submission record was created specifically for TempleReady™ system and for GEDCOM files being downloaded from Ancestral File™. (See SUBMISSION_RECORD, page 28.)

! A RESTRICTION (**RESN**) tag and a <RESTRICTION_NOTICE> primitive were added to the <<INDIVIDUAL_RECORD>> context. (Also added to the family record in 5.5.1.) This allows some records in Ancestral File to be marked for privacy (indicating some personal information is not included) and some records to be marked as locked (indicating that Ancestral File will not make changes to the record without authorization from an assigned record steward).

! The following tags are no longer used in the Lineage-Linked Form:
ARVL, BROT, BUYR, CEME, CNTC, CPLR, DEFM, DPRT, EDTR, FIDE, FILM, GODP, HDOH, HEIR, HFAT, HMOT, INFT, INDX, INTV, ISA, ISSU, ITEM, LABL, LCCN, LGTE, MBR, NAMS, NAMR, OFFI, ORIG, OWNR, PERI, PORT, PWIF, PUBR, RECO, SELR, SEQU, SERS, SIBL, SIGN, SIST, SITE, TXPY, XLTR, WFAT, WITN, WMOT, AUDIO, IMAGE, PHOTO, SCHEMA, VIDEO

! The following tags were added:
BLOB, CTRY, CREM, FCOM, GIVN, NPFX, NSFX, OBJE, PEDI, RELA, RESI, RESN, SUBN, SURN, STAT

Changes Introduced in Draft Version 5.3

Version 5.3 introduced the following changes to the GEDCOM standard:

! An address structure was defined.

! A new tag for marital status (MSTA) at the time of an event was added to the event structure. (This was removed in version 5.4.)

! A mechanism for creating user-defined tags was added. These were defined in a <<SCHEMA>>

definition in the header record of 5.3. (<<SCHEMA>> was removed in version 5.4.)

! The Unicode standard (ISO 10646) was introduced as an additional character set. (This was reduced to potential character set in version 5.4. See Chapter 3, page 77.)

! A <<MULTIMEDIA_LINK>> structure was introduced to provide linking and embedding of digitized photo, video, and sound files. (This was modified in version 5.4). (See <<MULTIMEDIA_LINK>> page 37 and <<MULTIMEDIA_RECORD>> page 26)

! The source structure NAME tag, meaning the name of the source in the <<SOURCE_CITATION>>, was changed back to the ~~TITLe~~ TITL (title) tag and is used to show the title of a book, article, or descriptive title of non-titled sources.

! The <<SOURCE_CITATION>> was changed. Usage of CPLR, XLTR, and INFT tags in source substructures was discontinued.

! The FORM {FORMAT} tag was added subordinate to the ~~PLACe~~ PLAC (place) and the ~~GEDCom~~ GEDC (GEDCOM) tags in the <<HEADER>> record and also subordinate to the ~~PLACe~~ PLAC (place) tag in the <<PLACE_STRUCTURE>>. The PLAC.FORM line in the header record indicates that all of the locality names are specified in a consistent hierarchical sequence as specified by the value of the FORM. For example: 2 FORM City, County, State. GEDCOM 5.2 used the TYPE tag, subordinate to the PLAC tag instead of the FORM tag, for this purpose. This provision is for products which have overly structured the place value.

Packaging the GEDCOM ~~Transmission~~ File

The GEDCOM ~~transmission~~ file is normally created on a DOS or Macintosh® compatible diskette. The DOS filename extension is (.GED). Macintosh filenames do not use file extensions.

When the GEDCOM file is too large to fit on a single diskette, the file is divided after any whole-line (last character is the **terminator**), and the DOS filename extension becomes (G##) where (##) is (00) for the second disk, (01) for the third, and so forth. For Macintosh filenames, append the two digits to the subsequent filenames in parentheses. (See the example below.) This allows the receiving software to ensure that disks are read in the correct sequence.

Given that the user-supplied portion of the file name is SMITH, then the complete filenames for a threedisk ~~transmission~~ GEDCOM file would be:

Disk	DOS Filename	Macintosh Filename
1	SMITH.GED	SMITH
2	SMITH.G00	SMITH(00)
3	SMITH.G01	SMITH(01)

The required GEDCOM HEADer record appears only on the first disk and the required TRLR (trailer) record appears only on the last disk and **must be followed by the terminator**.

multi-volume GEDCOM files

The Multi-volume GEDCOM support described here complicates GEDCOM readers needlessly. Diskettes and multi-volume files are a thing of the past now, but there was never a need for a GEDCOM-specific method anyway, as you can simply use a multi-volume ZIP file.

Few products support multi-volume GEDCOM files.
Do not bother supporting multi-volume GEDCOM files.
Consider multi-volume GEDCOM files strongly deprecated.

references

Multi-volume GEDCOM files

user-defined tags
We do not encourage the use of user-defined GEDCOM tags. Applications requiring the use of nonstandard tags should define them with a leading underscore so that they will not conflict with future GEDCOM standard tags. Systems that read user-defined tags must consider that they have meaning only with respect to a system contained in the ~~HEAD.SOUR~~ HEAD.DEST context.

user-defined tags context

FamilySearch's statement that the meaning of user-defined tags is defined by the context of the system in HEAD.SOUR is wrong.
The meaning of user-defined tags is defined by the context of the system in HEAD.DEST.

The system in HEAD.DEST is often, but not always, the same system as that in HEAD.SOUR; applications can and do generate GEDCOM files intended for particular third-party systems.

Escape Sequence Format for the Lineage-Linked Form
Very few Lineage-Linked GEDCOM compatible systems uses the **escape** sequence feature provided in the GEDCOM grammar.

Sample Lineage-Linked GEDCOM ~~Transmission~~ File

The example below is a sample ~~transmission~~ GEDCOM file of genealogical information about three individuals who are members of the same family— father, mother, and child. In the example, "Joe/Williams/" is the value specified by the tag NAME under the INDI tag for the record (@3@). Other values in other lines, such as the birth date and place, provide additional information about Joe Williams. The value (@4@) specified by the FAMC tag is a pointer to the <<FAM_RECORD>> (@4@) of which Joe Williams is a child. Included also in this ~~transmission~~ GEDCOM file example are three other record types: a source record, a submitter record, and a repository record. These records are pointed to from within other records in the ~~transmission~~ GEDCOM file. This shows how pointer values can be used in creating Lineage-Linked GEDCOM Form.

Example: (**Indentation and bolding arebolding added for readability only**.)

0 HEAD
—1 SOUR **PAF**
——2 VERS **2.1**
—1 DEST **ANSTFILE**
—1 SUBM **@5@**
—1 SUBN **@8@**
—1 GEDC
——2 VERS **5.4**
——2 FORM **Lineage-Linked**
—1 CHAR **ANSEL**
0 @1@ INDI
—1 NAME **Robert Eugene/Williams/**
—1 SEX **M**
—1 BIRT
——2 DATE **02 OCT 1822**
——2 PLAC **Weston, Madison, Connecticut**
——2 SOUR **@6@**
———3 PAGE **Sec. 2, p. 45**

```
———3 EVEN BIRT
————4 ROLE CHIL
```

```
—1 DEAT
——2 DATE 14 APR 1905
——2 PLAC Stamford, Fairfield, CT
—1 BURI
——2 PLAC Spring Hill Cem., Stamford, CT
—1 RESI
——2 ADDR 73 North Ashley
———3 CONT Spencer, Utah UT84991
——2 DATE from 1900 to 1905
—1 FAMS @4@
—1 FAMS @9@
0 @2@ INDI
—1 NAME Mary Ann/Wilson/
—1 SEX F
—1 BIRT
——2 DATE BEF 1828
——2 PLAC Connecticut
—1 FAMS @4@
0 @3@ INDI
—1 NAME Joe/Williams/
—1 SEX M
—1 BIRT
——2 DATE 11 JUN 1861
——2 PLAC Idaho Falls, Bonneville, Idaho
—1 SLGC
——2 FAMC @4@                                      /* note: this ptr is not required but is allowed in 5.5 */
—1 FAMC @4@
—1 FAMC @9@
——2 PEDI Adopted
—1 ADOP
——2 FAMC @9@
——2 DATE 16 MAR 1864
—1 SLGC
——2 FAMC @9@
——2 DATE 2 OCT 1987
——2 TEMP SLAKE
0 @4@ FAM
—1 MARR
——2 DATE DEC 1859
——2 PLAC Rapid City, South Dakota
—1 SLGS
——2 DATE 14 JUN 1975
——2 TEMP SLAKE
—1 HUSB @1@
```

```
—1 WIFE @2@
—1 CHIL @3@
0 @5@ SUBM
—1 NAME Reldon /Poulson/
—1 ADDR 1900 43rd Street West
——2 CONT Billings, MT 68051
—1 PHON (406) 555-1232
0 @6@ SOUR
```

75

76

```
—1 DATA
——2 EVEN BIRT, DEAT, MARR
———3 DATE FROM Jan 1820 TO DEC 1825
———3 PLAC Madison, Connecticut
——2 AGNC Madison County Court, State of Connecticut
—1 TITL Madison County Birth, Death, and Marriage Records
—1 ABBR VITAL RECORDS
—1 REPO @7@
——2 CALN 13B-1234.01
———3 MEDI Microfilm
0 @7@ REP
—1 NAME Family History Library
—1 ADDR 35 N West Temple Street
——2 CONT Salt Lake City, Utah
——2 CONT UT 84150
0 @8@ SUBN
—1 SUBM @5@
—1 FAMF Reg Poulson Family
—1 TEMP SLAKE
0 @9@ FAM
—1 HUSB @1@
—1 CHIL @3@
0 TRLR
```

invalid GEDCOM: SUBN placement error

This example is invalid. The SUBN record is obsolete, but the very first thing that Chapter 2, Record Structures of the Lineage-Linked Form states, is that the optional submission record, when included, must be included directly after the header record.

invalid GEDCOM: C style comment in GEDCOM

This example is invalid. This GEDCOM file contains a C style comment. GEDCOM files do not allow comments. The presence of that comment makes the FAMC line value invalid.

correction of GEDCOM 5.5 error

This example GEDCOM file is not identical to the corresponding example in GEDCOM 5.5.
An error present in the GEDCOM 5.5 example has been corrected in GEDCOM 5.5.1.

The GEDCOM 5.5 example presents the MEDI record as a subrecord of REPO record:

```
0 @6@ SOUR
—....
—1 REPO @7@
——2 CALN 13B-1234.01
——2 MEDI Microfilm
```

The example in the GEDCOM 5.5.1 specification correctly shows the MEDI record as a subrecord of the CALN record:

```
0 @6@ SOUR
—....
—1 REPO @7@
——2 CALN 13B-1234.01
———3 MEDI Microfilm
```

bad GEDCOM practice: old style addresses

The example contains a bad GEDCOM practice.
It uses old style unstructured addresses (a single value split over multiple lines by CONT records), instead of the new style structured addresses (separate subrecords for separate address parts).

references

GEDCOM ADDR

bad GEDCOM practice: numbers as identifiers

The example uses unadorned numbers as identifiers.
That is not illegal, but it is bad practice; it becomes too easy to accidentally use the same identifier for two different record types.
It is common practice to use a single letter indicating the type of record followed by a number.

references

Common GEDCOM Identifier Naming Convention

bad genealogy practice: abbreviated place names

The example contains bad genealogy practice.

- All the place names are incomplete; the country (United States of America) is missing.
- Object Names are abbreviated: “Spring Hill Cem.” instead of “Spring Hill Cemetery”
- A state name is abbreviated: “CT” instead of “Connecticut”

references

Place Name Standardisation Basics

The following is an example of a <<SOURCE_CITATION>> subordinate to the birth event being cited that does not contain a pointer to a <<SOURCE_RECORD>>. (This is not encouraged.)

```
0 INDI
—1 NAME Fred /Jones/
—1 BIRT
——2 DATE 14 MAY 1812
——2 PLAC Tonbridge, Kent, England
——2 SOUR Waters, Henry F., Genealogical Gleanings in England: Abstracts of W
———3 CONC ills Relating to Early American Families. 2 vols., reprint 1901, 190
———3 CONC 7. Baltimore: Genealogical Publishing Co., 1981.
———3 CONT Stored in Family History Library book 942 D2wh; films 481,057-58 Vol 2, pa
———3 CONC ge 388.
```

deprecated source citation style

This example has been highlight as deprecated, because the specification says this old style of including sources is deprecated.

By the way, the indentation is illegal, but the CONC and CONT usage is correct.

Chapter 3 Using Character Sets in GEDCOM

Introduction

GEDCOM needs to accommodate different character sets to facilitate the sharing of genealogical data in different languages. To minimize the number of differing standards, we have chosen to have each system convert its usage to ANSEL, and eventually to UNICODE.

Currently, ANSEL is to be used to represent Latin-based characters. Unicode should be used.

ANSEL or Unicode

What it says here is not that systems should only be using ANSEL and Unicode, and nothing else. What it says here is that while systems can use any character set and encoding they like, they must, on export, convert their data to either ANSEL or Unicode. Another, more specific rule presented below, states that code-page applications must always convert to ANSEL.
These rules are obsolete. The "eventually" mentioned above has come and gone.
All genealogy applications should be Unicode-based and always export to Unicode.

best practice

- genealogy application should be Unicode-based
- GEDCOM export should default to UTF-8

export to UTF-8 only

Unicode genealogy applications should only export to Unicode.
The GEDCOM export should default to using the UTF-8 encoding. It may offer UTF-16 as another output option, it may also be UTF-8 only.
It is not necessary to offer UTF-16 when UTF-8 is offered already, and not offering users a choice keeps the export simple.

import versus export

Real world applications have to deal with multiple legal and illegal character sets on import of legacy GEDCOM files, but Unicode applications should not export to anything but Unicode (or its successor).
Offering users the choice to export to anything less than Unicode does not make sense, as doing so will almost certainly lose characters not supported by the other character set.

real world

RootsMagic has offered nothing but UTF-8 GEDCOM only since it became Unicode-based with RootsMagic 4 released in 2008 CE. Software MacKiev Family Tree Maker changed to UTF-8 only export starting with Family Tree Maker 2017, and Ahnenblatt will change to UTF-8 only export with Ahnenblatt 3.0, expected in 2018 CE.

The GEDCOM Standard does not address the implementation methods for multilingual processing, such as keyboard arrangements, sorting sequences, or character and graphic representations (font styles, proportional spacing, and so forth) on the CRT or printers. However, the Unicode standard has defined formatting characters that will indicate the direction of the text presentation and other text formatting character codes.

Systems using code pages to support diacritical characters, such as the windows ANSI 1252 code page, must convert all characters above character code 0x7F to its ANSEL representation ~~for that code page~~. System that use code pages should no longer export to ANSEL, but to UTF-8.
Most of the genealogy systems developed so far use ASCII, ANSEL, or both. ANSEL accommodates the set of Latin-based languages, as explained below.

convert code pages to ANSEL

What it says here is, simply put, that code-page applications, such as 16-Windows applications using Windows ANSI (code page 1252), must always convert all characters to ANSEL.
The phrase "for that code page" is an error.

optimisation

The phrase "all characters above character code 0x7F" may seem to impose some condition or make the conversion a partial conversion, but it does not. Remember that ANSEL, the Windows ANSI code page and the Mac System code pages are all extensions of ASCII; their first 128 code points are identical. The phrase merely hints at an implementation optimisation; the ASCII part of the table need not be converted.

code-page applications must export using ANSEL

The text does not state *when* the applications must convert, but that should be fairly obvious: when exporting data to GEDCOM. So, what it says here that is that code-page applications must always use ANSEL for their GEDCOM files.

dated rule: code-page applications must export using ANSEL

This paragraph states that code-page application must use the ANSEL encoding when exporting to GEDCOM. That instruction was fine for GEDCOM 5.5, but should already have been modified for GEDCOM 5.5.1, as GEDCOM 5.5.1 added support for UTF-8.

The real point of the this instruction is that code page applications should not use their system-specific code page, but the cross-platform ANSEL standard instead.

As written, the text demands ANSEL and thus forbids using anything but ANSEL, and that is a mistake; the cross-platform UTF-8 encoding is an excellent choice too.

modern rule: all applications must export UTF-8

Further down in this chapter, the GEDCOM specification states:

> UNICODE character set or the 8-bit UTF-8 form should be used for multi-language support as soon as operating systems begin providing adequate storage and display support.

Unicode support has not only been more than adequate since the previous millennium, Unicode is the current standard: every major operating system is Unicode-based.
All modern applications are Unicode-based, and the very few code-page genealogy applications still being used all support UTF-8 export. Thus, the dated rule that code-page applications should export to ANSEL can and should be replaced with the simpler rule that **all genealogy applications should export to UTF-8**.

All code-page applications should have been retired in the previous millennium already, but those that are still around should offer the UTF-8 encoding, and default to UTF-8.

~~8-Bit~~ ANSEL

“8-bit ANSEL”

ANSEL is an 8-bit character set, so “8-bit ANSEL” is a misleading pleonasm. This phrase suggests that there is not only an 8-bit ANSEL, but another, perhaps a 16-bit, ANSEL as well.
There is just one ANSEL standard, and it is a 8-bit character set.

The ~~8-Bit~~ ANSEL (American National Standard for Extended Latin Alphabet Coded Character Set for Bibliographic Use, Z39.47-1985 copyright) is currently the preferred character set for GEDCOM. However this will shortly change to UNICODE and UTF-8 as the support for these latter character forms becomes fully supported by the computer industry.

preferred character set

Back in 1999, only four years after the release of GEDCOM 5.5, it still made some sense to have ANSEL as the preferred character set, and state that Unicode and UTF-8 will shortly become the preferred character set and encoding.
Nowadays (2018 CE), Unicode is the preferred character set for all operating systems and applications. Genealogy applications should use Unicode internally, and GEDCOM export should default to using UTF-8.

Historically, ANSEL is the default character set. The HEAD.CHAR record isn't mandatory in older GEDCOM standards. For newer versions of GEDCOM, the absence of the HEAD.CHAR is a fatal error. For older versions, it merely means that the GEDCOM reader has to assume the GEDCOM file is ANSEL-encoded.

“character forms”

Unicode and UTF-8 aren't “character forms”. That is non-standard FamilySearch terminology.
Unicode is a character set.
UTF-8 is a Unicode encoding.

ANSEL character set makes it possible to preserve the integrity of most Latin based languages by providing a method of using the standard ASCII character set and supplementing it with both nonspacing character modifiers (diacritic) as well as some useful spacing special characters.

Note: Non-spacing means that the diacritic is printed without advancing the device's print position. The character being modified is then printed in the same position, resulting in a combined image of both the character and the diacritic(s).

Storing ANSEL requires storing the non-spacing graphic character(s) preceding the ASCII character that the diacritic is to modify. The ANSEL standard specifies an extended 8-bit configuration ~~(above 128)~~ (above 127) to represent the spacing and non-spacing graphic characters that make up most of the Latin based languages. ANSEL is a super-set of ASCII. The standard ASCII characters including the control characters are preserved.

Above 127, not above 128

ASCII is a 7-bit character set, with 128 code points numbered 0x00 through 0x7F (0 through 127). Extension of ASCII to an 8-bit character set like ANSEL does not start above 128, but above 127.

“extended 8-bit configuration”

There is no such thing as an “extended 8-bit configuration” (of ASCII). That is sloppy, made-up terminology without a widely understood meaning, or in fact any meaning at all.

ANSEL is an 8-bit character set. ANSEL is an extension of ASCII. Thus, ANSEL is an *8-bit extension of ASCII*.

ANSEL is known by two other names:

! ANSI Z39.47-1985
! American Library Association character set, used in library systems worldwide, including the MARC (Machine-Readable Catalog) format.

The codes used for the ANSEL character set is described in Appendix C. The full definition may be purchased from:

American National Standards Institute 1430 Broadway
New York, N.Y. 10018

Character set codes 0x0 through 0x7F are the same for ~~8-Bit ANSEL~~ ANSEL and ~~8-Bit ASCII (USA version — ANSI 8-Bit)~~ ASCII, and ASCII-based code pages. Character set codes 0x80 through 0xFF are unique to the ANSEL character set.

“USA version” of ASCII

There is no such thing as “8-bit ASCII”. The phrase “8-bit ASCII” is a contradiction in terms, as ASCII is a 7-bit character set.
There is no such thing as an “USA version” of ASCII either. ASCII *is* American; ASCII is an abbreviation of *American* Standard Code for Information Interchange.

“ANSI 8-bit”

The phrase “ANSI 8-Bit” is meaningless. FamilySearch probably means Windows code page 1252, informally known as Windows ANSI.
However, the mere mention of Windows ANSI is odd already, as this specification explicitly disallows Windows ANSI, in one of the first few sentences of this chapter.

The way FamilySearch uses it: “8-Bit ASCII (USA version— ANSI 8-Bit)”, *as if* “USA version-ANSI 8-bit” means something, is seriously confused.

ASCII ~~(USA Version)~~

When a Western language does not *need* diacritic characters or other special characters, and if you are not transmitting binary data, you will find it convenient to use ASCII ~~(8-bit USA version)~~ if your computer already supports it. This is a standard of the American National Standards Institute (ANSI). Most of the basic printable characters of ANSEL and ASCII ~~(USA version — ANSI 8-Bit)~~ are identical.

“ASCII (8-bit)”, “ASCII (USA version)”

There is no such thing as “ASCII (8-bit USA version)” or “ASCII (USA version— ANSI 8-Bit)”. ASCII is a 7-bit character set, so “ASCII (8-bit ...)” is a contradiction in terms.

There is no "USA version" of ASCII. ASCII *is* American. ASCII is an abbreviation of *American Standard Code for Information Interchange*.
There are no country-specific versions of ASCII, there is just ASCII, which is American.

"ANSI 8-bit"

The phrase "ANSI 8-Bit" is meaningless too.
The apparent mention of some 8-bit "ANSI" character set here is odd, as GEDCOM explicitly disallows the "Windows ANSI" character set.

UNICODE

The Unicode standard is a character code designed to encode text for storage in computer files. It is being developed in close relationship to the ISO 10646 standard. The design of the Unicode standard is based on the simplicity and consistency of today's prevalent character code set, extended ASCII code set, but goes far beyond ASCII's limited ability to encode only the Latin alphabet: the Unicode encoding provides the capacity to encode most all of the characters used for written languages throughout the world. In order to accommodate the many thousands of characters used in the international text, the Unicode standard uses a 16-bit code set instead of extended ASCII's 8-bit code set. This expansion provides codes for approximately 65,000 characters. The text representation of the Unicode 16-bit numbers is U+0041 which is assigned to the letter A, 65 decimal. The Unicode standard includes characters for the Latin alphabet used for English, the Cyrillic alphabet used for Russian, the Greek, Hebrew, and Arabic alphabets. Other alphabets used in countries across Europe, Africa, the Indian subcontinent, and Asia, such as Japanese Kana, Korean Hangul, and Chinese Bopomofo are included. (See "The Unicode standard version 2.0", published by Addison-Wesley Publishing, for character code standards.)

Unicode is not 16-bit

Unicode isn't a 16-bit character set, but a 21-bit character set.
Unicode 1.0 was a 16-bit character set, but from Unicode 2.0 onwards, it has been a 21-bit character set.

Unicode 2.0 was released in July of 1996, several years before FamilySearch published this GEDCOM 5.5.1 specification. The FamilySearch GEDCOM specification specifically refers to *The Unicode standard version 2.0* published by Addison-Wesley, yet still erroneously claims that "the Unicode standard is a 16-bit character set".

Unicode size

The text refers to Unicode as "This expansion" and claims that it "provides codes for approximately 65,000 characters".
Unicode is bigger than that, a lot bigger than that.

There is a difference between the number code points and the number of characters encoded by those code points. For example, ANSEL is an 8-bit character set, which uses only uses 191 out of 256 code points, yet encodes more than 256 different characters. FamilySearch knows this, which is why they choose ANSEL as GEDCOM's preferred character set.

A 16-bit character set can accommodate 65.536 code points, but Unicode isn't a 16-bit character set, it is a 17-planes character set, which amounts to a 21-bit code space. The Unicode character set has more than one million code points, 1.114.112 to be precise (17 planes of 65.536 code points each). Of these, 1.111.998 code points are available for characters (1.114.112 code points minus 2048 surrogates and 66 non-characters).

The exact number of assigned coded has increased with every new version of Unicode. Back in 1999, the number of assigned code points was already well above one hundred thousand. Nowadays (2018 CE), the number of assigned code points is more than a quarter million.

"Chinese Bopomofo"

The phrase "Chinese Bopomofo" is, depending on viewpoint, either a pleonasm or a contradiction in terms. It is wrong either way.

Bopomofo is a phonetic script for transcribing Chinese, created by a Chinese commission, and first published in 1913. It is known as Zhùyīn in China, where it was replaced by Pinyin. It remains in use in Taiwan, where it is known as Bopomofo.

If the Unicode environment is used to produce a GEDCOM ~~transmission~~ file, the header record would also be in Unicode, requiring receiving systems to determine whether the ~~transmission~~ GEDCOM file is Unicode or ASCII before they could interpret the GEDCOM header.

contradicted by ANSEL demand

The above sentence is sloppy in more ways than one (see other annotation below), but still clearly communicates that a GEDCOM file encoded in UTF-16 has a GEDCOM header encoded in UTF-16.
This statement is contradicted by the ANSEL demand made in chapter 2 for the GEDCOM header: “All character codes greater than 0x7F **must be converted to ANSEL**.”.

The ANSEL demand in chapter 2 is wrong, this statement is correct; an UTF-16 encoded GEDCOM file has an UTF-16 encoded header.

8-bit versus 16-bit

The above sentence is tremendously confused, but does try to express an important point, which should be made elsewhere.
Not only does GEDCOM support more character sets than just ASCII and Unicode, the dichotomy isn't between Unicode and ASCII at all. The dichotomy is between UTF-16, a 16-bit encoding of Unicode, and 8-bit encodings, including UTF-8, which *is* Unicode. The dichotomy is between 8-bit and 16-bit encodings.

A GEDCOM reader establishes the character set used by reading and interpreting the line value of the HEAD.CHAR record. However, code designed for pre-GEDCOM 5.5 8-bit files won't work, because a UTF-16 encoded GEDCOM file has an UTF-16 encoded header.
A GEDCOM reader must distinguish between 8-bit and 16-bit encodings before it can try to read the GEDCOM header.

detecting character encoding

This one remark about ~~UNICODE versus ASCII~~ 16-bit versus 8-bit encoding is all the FamilySearch GEDCOM 5.5.1 says detecting GEDCOM files and the encoding used by GEDCOM files.
This is inadequate.

references

0 HEAD Value
GEDCOM & FTW TEXT Magic
GEDCOM Magic Values
GEDCOM Character Encodings

UTF-8

UTF-8 is a transformation format (encoding) that allows Unicode characters to be ~~transformed into an 8-bit form~~ encoded using 8-bit bytes. The transformation scheme allows ~~existing ASCII~~ Unicode ~~characters~~ code points ~~in the range of 0x0 to 0x7F~~ in the range of 0x0 through 0x7F to be ~~represented~~ encoded using ~~the normal 8-bit character code~~ a single byte for each code point. ~~Characters from the Unicode character set~~ Unicode code points ~~from~~ in the range ~~U+0080 to U+07FF~~ U+0080 through U+07FF ~~can be represented by~~ are encoded in two 8-bit ~~codes~~ bytes , and ~~characters~~ code points ~~from~~ in the range ~~U+0800 to U+FFFF~~ U+0800 through U+FFFF are ~~represented by~~ encoded in three 8-bit ~~codes~~ bytes, while code points in the range U+10000 through U+1FFFFF are encoded in four 8-bit bytes, code points in the range U+200000 through U+3FFFFFF are encoded in five 8-bit bytes, and code points in the range U+4000000 through U+7FFFFFFF are encoded in six 8-bit bytes. This ~~method~~ encoding is used by many systems handling mostly Latin characters to save a significant amount of space.
~~The UTF-8 transformation method allows for efficient transformation to and from the Unicode text. Basically, this 8-bit form uses the high order bits to determine how to decode each of the 8-bit forms back to the Unicode representation.~~ Conversion from the UTF-8 encoding to UTF-32 encoding is fairly simple. The high-order bits of each byte indicate how to convert it. Conversion between UTF-8 and UTF-16 is somewhat tricky and needs to be done carefully..

UTF-8 Corrections

Unicode and UTF-16 are not the same thing

The many small corrections in the above paragraph, corrections of fact as well as corrections of erroneous and confused expressions, are embarrassing.

Throughout the text, FamilySearch confuses UTF-16 and Unicode. FamilySearch writes *as if* Unicode is 16-bit character set (it is not, it is a 21-bit character set), *as if* UTF-16 equals Unicode (it does not, it is just one of the encodings) and *as if* UTF-8 isn't Unicode (UTF-8 is Unicode, it is one of the possible Unicode encodings).

UTF-8 Conversion

The original statement, after textual corrections, that the UTF-8 encoding is easily (“efficiently”) transformed into UTF-16 is misleading at best.
UTF-8 is easily transformed into UTF-32 and back. UTF-16 is easily transformed into UTF-32 and back.
However, conversion from UTF-8 to UTF-16 needs some care; implementation can be optimised, but conversion from UTF-16 to UTF-8 should conceptually go through UTF-32 to ensure the result is UTF-8 and not CESU-8. Conversion from UTF-8 to UTF-16 should conceptually go through UTF-32 as well, *and* should include detection of invalid bytes as well as invalid byte sequences.

How to Change Character Sets

The character set for an entire ~~transmission~~ GEDCOM file is specified in the character set line of the header record.

one encoding for the entire file

The above sentence clearly states that the entire GEDCOM file uses just one character encoding, specified in the GEDCOM header.
The definition of the GEDCOM header in Chapter 2 Lineage-Linked ~~Grammar~~ Form contradicts this statement. The GEDCOM header definition contains a demand to use ANSEL for all character with code points larger than 0x7F.

The statement made here is correct. The ANSEL-demand made in the GEDCOM header definition is impossible to comply with.

references

GEDCOM header encoding

The example below shows the specification in the header record:

```
Lvl Tag Value
0 HEAD
—1 SOUR PAF
——2 VERS 2.1
—1 DEST ANSTFILE
—1 CHAR ANSEL
```

~~The character set change remains in effect until the TRLR record is encountered at the end of the transmission .~~

“character set change”

The above sentence about a “character set change” is confused. There are no character set changes in GEDCOM.
Each GEDCOM file uses one encoding for the entire GEDCOM file.

This odd sentence is a hold-over from an unholy idea FamilySeach once had, namely the ability *to change from one character set to another in the middle of a GEDCOM file*. The GEDCOM 5.0 specification describes how that was supposed to work. This strange notion is gone in GEDCOM 5.3, and the offending sentence should have been deleted from GEDCOM 5.3 already, yet remains present in the GEDCOM 5.4, 5.5 & 5.5.1 specifications.

references

FamilySearch GEDCOM Specifications

Best Practice: UTF-8 Byte Order Mark

GEDCOM files are text files, so UTF-8 GEDCOM files are UTF-8 text files. For UTF-8 text files, use of a Byte Order Mark (BOM) is optional. For UTF-8 GEDCOM files, use of a (BOM) should be mandatory.

The FamilySearch GEDCOM 5.5.1 specification says nothing about the Byte Order Mark (BOM), not even for UTF-16 GEDCOM files. However, FamilySearch PAF has always given the right example; all PAF UTF-8 GEDCOM files start with a BOM.

Starting UTF-8 GEDCOM files with a BOM is best practice. Including the BOM makes sure that text editors will recognise the GEDCOM file as UTF-8 encoded, so that they do not misrecognise the encoding and mess up the file.
It also makes sure that (really old) applications that ignore the encoding specified in the header, and do not understand UTF-8 will not mangle your data by importing the file as say ANSI- or ANSEL-encoded; the BOM prevents them from recognising the file as a GEDCOM file, thus exposing their limitation.

UNICODE character set or the 8-bit UTF-8 form should be used for multi-language support as soon as operating systems begin providing adequate storage and display support.

Unicode should be used

The phrase “UNICODE character set or the 8-bit UTF-8 form” is terminologically confused, yet the statement remains clear: Unicode and UTF-8 should be used as soon as operating systems support is adequate. FamilySearch does not state what they consider adequate, so this is a judgement call

left to developers implementing GEDCOM. Arguably, operating system support for Unicode was adequate already when FamilySearch published the GEDCOM 5.5.1 specification on 2 Oct 1999. It certainly has been more than adequate for close to two decades now.
So, what the FamilySearch GEDCOM 5.5.1 specification states here, quite unequivocally, is that **genealogy applications should use Unicode and UTF-8 now**.

This specifically means that

- the rule, stated earlier in this chapter, that code-page genealogy applications should export using ANSEL, is dated now
- the modern rule is that all genealogy applications, even the code-paged based one, should export using UTF-8

For more information about character sets, see the following:

! Extended Latin Alphabet Coded Character Set for Bibliographic Use. American National Standards (ANSEL), Z39.47, 1985. This is an out of date version, but is the version established as the GEDCOM ANSEL standard.

! "The Unicode standard", version 2.0, published by Addison-Wesley Publishing.

Unicode books

The Unicode Standard has seen several editions since the publication of GEDCOM 5.5.1 back in 1999. It is a reference, not very suitable for learning about Unicode.
The Unicode Consortium web site offers many helpful resources, including an introduction to Unicode and a FAQ.

Two useful books are:

- *Unicode Demystified; A Practical Programmer's Guide to the Encoding Standard* by Richard Gillam (2003, Addison-Wesley, pp. 853, ISBN 0-201-70052-2)
- *Developing International Software, Second Edition* by Dr. International (2003, Microsoft Press, pp. 1060, ISBN 0-6536-1583-7)
  This is the revised edition of *Developing International Software for Windows 95 and Windows NT* by Nadine Kano (1995, Microsoft Press, pp. 743, ISBN 1-55615-840-8).

Chapter 4 GEDCOM Product Registration

Registering GEDCOM Products

Developers of GEDCOM compatible products should register their product with the Family History Department GEDCOM coordinator.

Registration means that:

- The registered GEDCOM product is listed in Family History Department publications as being GEDCOM 5.5 compatible.
- The product's sample output file has been reviewed according to the established standard, and exceptions have been reported.
- The developer will be informed of future GEDCOM issues.
- Submissions to Family History Department resource files will be accepted from users of registered products.

To register a GEDCOM product, a developer must send the following information to the GEDCOM coordinator:

- A file containing a small sample of the product's GEDCOM output for evaluation. All of the fields that the product manages must be included in this data so that it can be tested for compatibility with other developers' products.

- A proposed unique ~~SOURce~~ SOUR (source) name that identifies the *product,* not the company. This identifier should be included in the GEDCOM header record as a value to the SOUR tag. This name can have up to 40 characters, can have mixed upper and lower case, and cannot have embedded spaces. Use either an underscore (_) to connect multiple words or else a combination of upper and lower case letters (for example, FamilyRecords or Family_Records, **not** Family Records). The Family History Department will ensure uniqueness within the first 10 characters of this name.

- Optionally, but strongly suggested, a copy of the GEDCOM product with installation procedures and a text file containing relevant technical documentation about the product's GEDCOM implementation. The product will be protected and used only for GEDCOM output verification and in providing some support to submitting users of your product.

System Identifier Registration

FamilySearch never published a list of registered system identifiers. FamilySearch abandoned GEDCOM around 2000 CE, and it has not been possible to register system identifier with them since.

That FamilySearch abandoned registration of system identifiers does not imply that system identifiers are obsolete.
System identifiers are still a part of the GEDCOM specification, and are useful for recognising files produced by certain products.
More observations and best practices are provided through annotations on the <APPROVED_SYSTEM_ID> definition.
A fairly extensive list of collected system identifiers is provided by *GEDCOM System Identifiers*.

references

GEDCOM System Identifiers

system identifier not 40 characters

Chapter 2 documents <APPROVED_SYSTEM_ID> as an identifier that is at most 20 characters long. This chapter contradicts the lineage-linked form specification, by stating that a system identifier may be up to 40 characters long.
The definition in Chapter 2 is the actual specification, so chapter 4 is wrong. The maximum length of a system identifier is 20 characters.

Send product registration information to:

EMAIL:
gedcom@gedcom.org

MAIL:
Family History Department
GEDCOM Coordinator— 3T
50 East North Temple Street
Salt Lake City, UT 84150

TELEPHONE (USA):
801-240-4534
801-240-3442

Appendix A Lineage-Linked GEDCOM Tag Definition

Introduction

Appendix A is a glossary of the tags used in the Lineage-Linked GEDCOM Form. These tags are used in a hierarchical structure to describe individuals in terms of their families, names, dates, places, events, roles, sources, relationships. Control information and other kinds of data intended for computer processing is also included. (An example of the tags used in the Lineage-Linked Form begins on page 74.) To ensure all transmitted information in the Lineage-Linked GEDCOM is uniformly identified the standardized tags cannot be placed in any other context than shown in Chapter 2. It is legal to extend the context of the form, but only by using user-defined tags which must begin with an underscore. This will not violate the lineage-linked GEDCOM standard unless the context for the grammar of the Lineage-Linked GEDCOM Form is violated. The use of the underscore in the user tag name is to signal a non-standard construct is being used. This notifies the reading system of a discrepancy and will avoid future conflicts with tags that may be standardized in subsequent GEDCOM releases.

no redefinition of tags

This text (which was last changed for the GEDCOM 5.5 release of 2 Jan 1996), has the right intention, but the wrong instruction. The idea is that the tags of the lineage-linked form may not be redefined, and the lineage-linked form may not be extended by anything but user-defined tags.

The meaning of GEDCOM tags is defined by their context, e.g. HEAD.SOUR and FAM.SOUR are two different kind of sources. For the standard to be a standard, no vendor can redefine the records of the lineage-linked form to mean something else. Thus, vendors cannot assign a new meaning to say HEAD.SOUR or FAM.SOUR. Vendors are also disallowed to extend the lineage-linked form using existing tags, so are not allowed to introduce say OBJE.SOUR, but would have to use OBJE._SOUR instead.

Contrary to what the above text states, vendors *are* allowed to introduce _TAG.SOUR; that does not redefine any existing SOUR tag in any way. A GEDCOM reader *should not* assume that predefined tags occur only in predefined context, but only assume that GEDCOM-defined records are as defined.

illegal tags

The above paragraph states that it's legal to extend the context of the form, but only by using user-defined tags which must begin with an underscore. In other words, user-defined tags that do not start with an underscore are illegal.

GEDCOM readers that come across an illegal tag should issue a fatal error and abort. It may seem harsh to reject a GEDCOM file for one illegal tag, but there simply is no reason for a GEDCOM file to contain illegal tags. It's a trivial effort for the developer of the offending application to use underscores.

The obvious exception to this rule is a new version of the offending application reading an GEDCOM file produced by an earlier version of that application; applications should be able to read files produced by earlier versions of the same application, even if these files should otherwise be rejected.

best practice

- Applications may not use vendor- or product-specific tags that do not start with an underscore

- Applications that allow users to define tags must demand that these start with an underscore
- GEDCOM readers that come across a tag that does not start with an underscore should issue a fatal and abort

FTW TEXT

The list of definitions below gives both the tag name and, within curly brackets, a longer text. Typically, the longer text is a valid English word, or two words connected by an underscore, and the tag an abbreviation of that word or phrase.

GEDCOM uses the tags, most of which are four code units long. Exactly one genealogy application decided to use the text within the curly brackets instead. That application is Family Tree Maker for Windows (FTW), and those files are not GEDCOM files. These file are known as FTW TEXT files.

Back when Family Tree Maker for Windows (Classic) was quite popular, these files used to be quite prevalent. There are few FTW TEXT files around any more. They can converted to GEDCOM files using a freely downloadable copy of Family Tree Maker 2005 Starter Edition. So, there is no need to support FTW TEXT in modern genealogy applications.

FTW TEXT detection

Detecting FTW TEXT files is easy.
While the first line of a GEDCOM file is `0 HEAD`, the first line of an FTW TEXT is `0 HEADER`.

GEDCOM reader best practice

- do *not* add support for FTW TEXT
- **do detect** FTW TEXT, and when detected
  - inform the user that it isn't a GEDCOM file, but an FTW TEXT file
  - advise the user to convert the FTW TEXT to GEDCOM using Family Tree Maker 2005 Starter Edition
- if you *do* support FTW TEXT
  - do not do so silently, do not let the user keep thinking it's a GEDCOM file when it isn't
  - clearly state that you detected FTW TEXT, and are processing their FTW TEXT file
  - make sure you support both GEDCOM and FTWTEXT, but not FGTEWDTCEOXMT (a mixture of both); keep detecting illegal tags

references

FTW TEXT
Family Tree Maker 2005 Starter Edition (direct download link)

Lineage-Linked GEDCOM Tag Definitions

This section provides the definitions of the standardized GEDCOM tags and shows their formal name inside of the {braces}. The formal names are not used in place of the tag. Full understanding must come from the context in which the tag is used.

ABBR {ABBREVIATION}:=
A short name of a title, description, or name.

ADDR {ADDRESS}:=
The contemporary place, usually required for postal purposes, of an individual, a submitter of information, a repository, a business, a school, or a company.

ADR1 {ADDRESS1}:=
The first line of an address.

ADR2 {ADDRESS2}:=
The second line of an address.

ADR3 {ADDRESS3}:=
The third line of an address.

ADOP {ADOPTION}:=

~~Pertaining to creation of a legally approved child-parent relationship that does not exist biologically.~~
Adoption is a legal event that changes a child's legal parents from one set of parents to another set of parents.
While some of the parents involved are likely to be biological or official parents, neither assumption should be made.

adoption

This FamilySearch definition reflects a narrow understanding of family and unwarranted assumptions. Their definition assumes that adoption is about changing the legal parents from the biological parents to parents who aren't biological parents of the child. Everything they assume is wrong.

- Adoption is an event that changes who the legal parents are.
- A child that has been adopted can be adopted again.
- The official parents should not be assumed to be the biological parents.
  That assumption should never be made without evidence.
- A child can *and often is* adopted by a biological or official parent.
- In many jurisdictions, a child is technically always adopted by a couple, even if one of them is already is a legal parent.

AFN ~~{AFN}~~{ANCESTRAL FILE NUMBER}:=

A unique permanent record file number of an individual record stored in Ancestral File.

AFN superfluous

The AFN record is not only obsolete because Ancestral File is obsolete, it was always superfluous, as GEDCOM already defines the IDNO record and it's <NATIONAL_ID_NUMBER> line value.

AGE {AGE}:=

The age of the individual at the time an event occurred, or the age listed in the document.

AGNC {AGENCY}:=

The institution or individual having authority and/or responsibility to manage or govern.

ALIA {ALIAS}:=

An indicator to link different record descriptions of a person who may be the same person.

ANCE {ANCESTORS}:=

Pertaining to forbearers of an individual.

83

84

ANCI {ANCES_INTEREST}:=
Indicates an interest in additional research for ancestors of this individual. (See also DESI, page 86.)

ANUL {ANNULMENT}:=
Declaring a marriage void from the beginning ~~(never existed)~~ (retroactively invalid).

ASSO {ASSOCIATES}:=
An indicator to link friends, neighbors, relatives, or associates of an individual.

AUTH {AUTHOR}:=
The name of the individual who created or compiled information.

BAPL {BAPTISM-LDS}:=
The event of baptism performed at age eight or later by priesthood ~~authority~~ of the LDS ~~Church~~. (See also BAPM, next)

BAPM {BAPTISM}:=
The event of baptism ~~(not LDS)~~, performed in infancy or later. (See also BAPL, above, and CHR, page 85.)

BARM {BAR_MITZVAH}:=
The ~~ceremonial event~~ religious ceremony held when a Jewish boy reaches age 13.

BASM {BAS_MITZVAH}:=
The ~~ceremonial event~~ religious ceremony held when a Jewish girl reaches age 13, also known as "Bat Mitzvah."

BIRT {BIRTH}:=
~~The event of entering into life.~~
The emergence of offspring from their mother as a separate being.

stillbirth

The FamilySearch definition of birth is plain wrong. Birth does not equal entering into life. A stillbirth is still a birth.

BLES {BLESSING}:=
~~A religious event of bestowing divine care or intercession. Sometimes given in connection with a naming ceremony.~~
An LDS rite, in which a priest lays hands and performs a prayer.

BLES

The definition given by FamilySearch is vague, and does not even state that this event is LDS-specific.
Unlike other religious rites, such as baptism, this one has nothing to do with genealogy.

84

85

BURI {BURIAL}:=

~~The event of the proper disposing of the mortal remains of a deceased person.~~
The action of burying a body. That includes burial at sea.
Within GEDCOM, it also includes interment (entombment).

burial

FamilySearch defines burial is defined as “proper disposing”, but cremation as “disposal”. Additionally, the phrase “mortal remains” is inappropriate religious language, which assumes the existence of immortal remains.

The BURI event can and therefore should be interpreted as including all forms of burial, including burial at sea, and as there is no separate event for interment, BURI is used for that too.

CALN {CALL_NUMBER}:=

The number used by a repository to identify the specific items in its collections.

CAST {CASTE}:=

The name of an individual's rank or status in society which is sometimes based on racial or religious differences, or differences in wealth, inherited rank, profession, occupation, etc.

CAUS {CAUSE}:=

A description of the cause of the associated event or fact, such as the cause of death.

CENS {CENSUS}:=

The event of the periodic count of the population for a designated locality, such as a national or state Census.

CHAN {CHANGE}:=

Indicates a change, correction, or modification. Typically used in connection with a DATE to specify when a change in information occurred.

CHAR {CHARACTER}:=

~~An indicator of the~~The character set and encoding used ~~in writing this automated information~~ in this GEDCOM file.

CHAR

FamilySearch's definition is wrong. First of all, the CHAR line value is not merely “an indicator”, it is an unambiguous statement.
Secondly, it does not merely state the character set, but the character set *and* encoding.

CHIL {CHILD}:=

The ~~natural, adopted, or sealed (LDS)~~ biological, official or legal (adopted) child of ~~a father and a mother~~ a parent or parents.

sealing rite is an event

FamilySearch was doubly confused when they added “sealed” to the enumeration of possible parent-child relationships.
First of all, they are confusing an *event*, the sealing rite, with a *relationship*. Secondly, the sealing rite neither is, nor affects any biological, official or legal parent-child relationship.

The so-called sealings are about imagined family relationships after death. A sealing does not affect any actual genealogical parent-child relationship, biological, official or legal. The sealing rite need not be documented in a genealogy, but if documented, should be documented as an event.

CHR {CHRISTENING}:=

The religious event ~~(not LDS)~~ of baptizing and/or naming a child.

CHRA {ADULT_CHRISTENING}:=

The religious event ~~(not LDS)~~ of baptizing and/or naming an adult person.

CITY {CITY}:=

A lower level jurisdictional unit. Normally an incorporated municipal unit.

CONC {CONCATENATION}:=

An indicator that additional data belongs to the superior value. The information from the CONC value is to be connected to the value of the superior preceding line without adding a space and without adding a ~~carriage return and/or~~ new line character. Values that are split for a CONC tag ~~must~~ should always be split at a nonspace. If the value is split on a space the space ~~will~~ may be lost when concatenation takes place. This is because of the treatment that spaces get as a GEDCOM delimiter, many GEDCOM values are trimmed of trailing spaces and some systems look for the first non-space starting after the tag to determine the beginning of the value.

CONC new line

The phrase “without a carriage return and/or new line character” is seriously confused.
In GEDCOM a line terminator (new line character) consists of a carriage return, a line feed or a combination of those two. There may not be both a carriage return and a line terminator. A carriage return may only occur as part of a line terminator. A carriage return occurring outside a line terminator is a fatal error.

leading and trailing white space in CONC line values

A CONC record without a line value is an error; not GEDCOM. GEDCOM readers should treat it as a fatal error, as it indicates serious issues with the system that generated the GEDCOM file. GEDCOM readers that try to import everything should still issue an error message.

A long line of spaces and tabs exceeding the maximum length of line value must be split using CONC records. As trailing white space isn't allowed, this necessitates creation of a CONC line value that starts with white space, and, very rarely, a CONC line value consisting of nothing but white space.

A GEDCOM reader should not trim initial white space from CONC line values, but accept it *as-is* (without any error or warning). A GEDCOM reader *may* trim trailing white space from CONC line values, but should *not* do; it should import the trailing spaces, but also issue a warning. A GEDCOM reader should accept a CONC line value that is nothing but whitespace.

CONF {CONFIRMATION}:=

The religious ~~event~~ rite ~~(not LDS)~~ ~~of conferring the gift of the Holy Ghost and, among protestants, full~~

~~church membership.~~ that confirms membership of a church (*confirms* because previously established by baptism).

CONL {CONFIRMATION_LDS}:=

The religious event by which a person receives membership in the LDS Church.

CONT {CONTINUED}:=

An indicator that additional data belongs to the superior value. The information from the CONT value is to be connected to the value of the superior preceding line ~~with a carriage return and/or~~ after first appending a new line character. Leading spaces could be important to the formatting of the resultant text. When importing values from CONT lines the reader should assume only one delimiter character (the mandatory space) following the CONT tag. Assume that the rest of the leading spaces are to be a part of the value.

CONT New Line

The phrase "carriage return and/or new line character" is seriously confused.
In GEDCOM a line terminator (new line character) consists of a carriage return, a line feed or a combination of those two. There may not be both a carriage return and a line terminator. A carriage return may only occur as part of a line terminator. A carriage return occurring outside a line terminator is a fatal error.

CONT record with line value

A CONT record without a line value is *not* an error, but the GEDCOM encoding of an empty line.

COPR {COPYRIGHT}:=

A statement that accompanies data to protect it from unlawful duplication and distribution.

CORP {CORPORATE}:=

A name of an institution, agency, corporation, or company.

CREM {CREMATION}:=

Disposal of ~~the remains of a person's~~ a body by fire by burning it to ashes.

CREM disposal

FamilySearch describes burial as "proper disposing" but cremation as "disposal".

CTRY {COUNTRY}:=

The name ~~or code~~ of the country.

DATA {DATA}:=

~~Pertaining to stored automated information.~~

Data.

DATE {DATE}:=

The time of an event in a calendar format.

DEAT {DEATH}:=

~~The event when mortal life terminates.~~

The end of a life.

DEAT

FamilySearch's use of the phrase "mortal life" is to imply an immortal life is inappropriate religious language for a genealogy standard.

DESC {DESCENDANTS}:=
Pertaining to offspring of an individual.

DESI {DESCENDANT_INT}:=
Indicates an interest in research to identify additional descendants of this individual. (See also ANCI,

86

87

page 84.)

DEST {DESTINATION}:=
A system receiving data.

DIV {DIVORCE}:=
~~An event of dissolving a marriage through civil action.~~
The legal dissolution of a marriage.

DIVF {DIVORCE_FILED}:=
An event of filing for a divorce by a spouse.

DSCR {PHYSICAL_DESCRIPTION}:=
The physical characteristics of a person, place, or thing.

EDUC {EDUCATION}:=
Indicator of a level of education attained.

EMAIL {EMAIL}:=
An electronic mail address.

EMAIL versus EMAI

Throughout the FamilySearch GEDCOM 5.5.1 specification, the tag used for the new email record is EMAIL (five letters). It is only here, in the list of definitions, that EMAI (four letters) is used. This is probably an edit error by an author who mistakenly believed that tags are limited to four characters.

GEDCOM writer best practice

- A GEDCOM writer must use EMAIL, never EMAI.

GEDCOM reader best practice

- A forgiving GEDCOM reader may treat EMAI as a synonym for EMAIL

EMIG {EMIGRATION}:=
An event of leaving one's homeland with the intent of residing elsewhere.

ENDL {ENDOWMENT}:=
A religious event where an endowment ordinance for an individual was performed by priesthood ~~authority~~ in an LDS temple.

priesthood pleonasm

The LDS priesthood has authority within the LDS. The phrase “priesthood authority” is a pleonasm.

ENGA {ENGAGEMENT}:=

An event of recording or announcing an agreement between two people to become married.

EVEN {EVENT}:=

Pertaining to a noteworthy happening related to an individual, a group, or an organization.
An ~~EVENt~~ EVEN (event) structure is usually qualified or classified by a subordinate use of the TYPE tag.

FACT {FACT}:=

Pertaining to a noteworthy attribute or fact concerning an individual, a group, or an organization.
A FACT structure is usually qualified or classified by a subordinate use of the TYPE tag.

FAM {FAMILY}:=

~~Identifies a legal, common law, or other customary relationship of man and woman and their children, if any, or a family created by virtue of the birth of a child to its biological father and mother.~~
A family group, which is a couple consisting of two partners, either or both of which may be unknown, with or without children.
These partners may or may not be spouses, and may or may not have children, but are biological, official or legal parents to each of any of the children in this family group.

FAMC {FAMILY_CHILD}:=

~~Identifies the family in which an individual appears as a child.~~
Identifies a couple (either of both which may be unknown) to whom this individual is a child.

FAMF {FAMILY_FILE}:=

Pertaining to, or the name of, a family file. Names stored in a file that are assigned to a family for doing temple ordinance work.

FAMS {FAMILY_SPOUSE}:=

Identifies ~~the family~~ a couple in which an individual appears ~~as a spouse~~.

FAMS

The FAM record isn't a family record, but a couple record.
The INDI.FAMS line value links an individual record to a couple record. That individual may or may not be a spouse in that couple; it should not be assumed that the person is a spouse.

FAX ~~{FACIMILIE }~~{FACSIMILE}:=

Electronic ~~facimilie~~ facsimile transmission.

FCOM {FIRST_COMMUNION}:=

~~A religious rite, the first act of sharing in the Lord's supper as part of church worship.~~
Literally the first communion an individual partakes in. Communion is a rite within christian churches, and the first communion is considered a rite of passage.

FILE {FILE}:=

~~An information storage place that is ordered and arranged for preservation and reference.~~
A container for some data, that can be treated as single unit.

FORM {FORMAT}:=

~~An assigned name given to a consistent format in which information can be conveyed.~~
The name of a *GEDCOM form*.

GEDC.FORM

This definitions fits HEAD.GEDC.FORM.
The HEAD.GEDC.FORMline value is a <GEDCOM_FORM>.
The FamilySearch GEDCOM 5.5.1 specification defines just one possible GEDCOM form: `LINEAGE-LINKED`.

PLAC.FORM

Neither FamilySearch's original definition, nor the replacement definition fits HEAD.PLAC.FORM, but that does not really matter.
The HEAD.PLAC.FORM line value is a <PLACE_HIERARCHY>, and HEAD.PLAC.FORM is deprecated because it is literally meaningless.

FONE {PHONETIC}:=

A phonetic variation of a superior text string

GEDC {GEDCOM}:=

Information about the use of GEDCOM in a ~~transmission~~ file.

GIVN {GIVEN_NAME}:=

A given or earned name used for official identification of a person.

GRAD {GRADUATION}:=

An event of awarding educational diplomas or degrees to individuals.

HEAD {HEADER}:=

Identifies information pertaining to an entire GEDCOM ~~transmission~~ file.

HUSB {HUSBAND}:=

~~An individual in the family role of a married man or father.~~
A individual within a family group (FAM), often male, often partner to a woman, and a biological, official or legal parent to the children of this family group.

HUSB may be wife

The name FamilySearch chose for the HUSB record strongly suggests that the line value must identify a husband.
In actual practice, the FAM.HUSB value need not identify a husband, it may identify a woman in a lesbian relationship.

references

- Same-Sex Marriage in GEDCOM

father

The word father has multiple meanings. Not all fathers fulfil a family role; some are merely the biological father.

IDNO {IDENT_NUMBER}:=
~~A number~~ An identifier, often called a number, assigned to identify a person within some significant external system.

IDNO

The FamilySearch definition states that the value is a number, but that is not correct. The IDNO line value need not be a number.
The value typically is an identification "number" containing spaces or dashes, like the example given in the <NATIONAL_ID_NUMBER> definition: `43-456-1899`.

The only usage of IDNO defined in the lineage-linked form is as a subrecord of INDI, and the INDO.IDNO value must be a <NATIONAL_ID_NUMBER>, which is defined as "A nationally-controlled number assigned to an individual". The prime example is a passport number.

IDNO versus REFN

The difference between the INDO record and the REFN record is that the INDO record is for third-party numbers, and the REFN record is for user-defined numbers.

IMMI {IMMIGRATION}:=

An event of entering into a new locality with the intent of residing there.

INDI {INDIVIDUAL}:=
A person.

LANG {LANGUAGE}:=
The name of the language used in a communication or transmission of information.

LATI {LATITUDE}:=
~~A value indicating a coordinate position on a line, plane, or space.~~
Latitude of a position on the globe.

LONG {LONGITUDE}:=
~~A value indicating a coordinate position on a line, plane, or space.~~
Longitude of a position on the globe.

MAP {MAP}:=
Pertains to a representation of measurements usually presented in a graphical form.

MARB {MARRIAGE_BANN}:=
An event of an official public notice given that two people intend to marry.

MARC {MARR_CONTRACT}:=

An event of recording a formal agreement of marriage, including the prenuptial agreement in which marriage partners reach agreement about the property rights of one or both, securing property to their children.

MARL {MARR_LICENSE}:=

An event of obtaining a legal license to marry.

MARR {MARRIAGE}:=

~~A legal, commonlaw, or customary event of creating a family unit of a man and a woman as husband and wife~~.

Marriage is an official and legal event, defined by the applicable law and customs of the land and the time, that creates a couple, possibly with children.

The MARR record is not only used for marriages, but for all couples and relationships types.

marriage record

The name of this record is ill-chosen. The MARR record is not really a marriage record, but a couple record.

Despite its name, the MARR record can and must be used for all relationship types, with a marriage contract merely being the default relationship type for the couple.

references

Marriage in GEDCOM

same-sex marriage

The erroneous definition provided by FamilySearch isn't a definition of marriage, but one of heterosexual marriage.

Despite FamilySearch's definition provided here, the MARR record can and must be used for both heterosexual and homosexual marriages.

references

Same-Sex Marriage in GEDCOM

MARS {MARR_SETTLEMENT}:=

An event of creating an agreement between two people contemplating marriage, at which time they agree to release or modify property rights that would otherwise arise from the marriage.

MEDI {MEDIA}:=

Identifies information about the media or having to do with the medium in which information is stored.

NAME {NAME}:=

A word or combination of words used to help identify an individual, title, or other item. More than one NAME line should be used for people who were known by multiple names.

NATI {NATIONALITY}:=

The ~~national heritage~~ nationality of an individual.

NATU {NATURALIZATION}:=

The event of obtaining citizenship.

NCHI {CHILDREN_COUNT}:=

~~The number of children that this person is known to be the parent of (all marriages) when subordinate to an individual, or that belong to this family when subordinate to a FAM_RECORD.~~
The number of children that this individual (INDI.NCHI) or family group (FAM.NCHI) has.

NICK {NICKNAME}:=

A descriptive or familiar that is used instead of, or in addition to, one's proper name.

NMR {~~MARRIAGE~~RELATIONSHIP_COUNT}:=

~~The number of times this person has participated in a family as a spouse or parent.~~
The number of different couple records (MARR records) this person occurs in.

number of “marriages”

This record really means number of relationships, not number of marriages.

NOTE {NOTE}:=

Additional information provided by the submitter for understanding the enclosing data.

NPFX {NAME_PREFIX}:=

Text which appears on a name line before the given and surname parts of a name.
~~i.e.~~e.g. ~~(~~**Lt. Cmndr.**~~)~~ Joseph /Allen/ jr.
In this example Lt. Cmndr. is considered as the name prefix portion.

NPFX

FamilySearch highlighted the name prefix by using both bold and parentheses.
The parentheses are a bad example; you should *not* include parentheses in a name field.

NSFX {NAME_SUFFIX}:=

Text which appears on a name line after or behind the given and surname parts of a name.
~~i.e.~~e.g. Lt. Cmndr. Joseph /Allen/ ~~(~~**jr.**~~)~~
In this example jr. is considered as the name suffix portion.

NSFX

FamilySearch highlighted the name suffix by using both bold and parentheses..
The parentheses are a bad example; you should *not* include parentheses in a name field.

OBJE {OBJECT}:=

Pertaining to a grouping of attributes used in describing something. Usually referring to the data required to represent a multimedia object, such an audio recording, a photograph of a person, or an image of a document.

OCCU {OCCUPATION}:=

The type of work or profession of an individual.

ORDI {ORDINANCE}:=

Pertaining to a religious ordinance in general.

ORDN {ORDINATION}:=

A religious event of receiving authority to act in religious matters.

ordination

Ordination is promotion to office within a religious organisation. Promotion to some office or rank isn't a genealogical event, but a family history event.

It is remarkable that GEDCOM features a record for promotion to a religious rank, but not for promotion to military rank, something family historians are sure to encounter, and has genealogical relevance; military service is often the cause of death and uncertainty about death.

The ORDN record is so rarely used, that it mainly occurs in GEDCOM test files...
GEDCOM already supports non-genealogical events through the EVEN (event) record, so the ORDN record is superfluous.

```
1 EVEN
2 TYPE Ordination
```

Consider the ORDN record deprecated.

90

91

PAGE {PAGE}:=
A number or description to identify where information can be found in a referenced work.

PEDI {PEDIGREE}:=
Information pertaining to an individual to parent lineage chart.

PHON {PHONE}:=
A unique number assigned to access a specific telephone.

PLAC {PLACE}:=
A jurisdictional name to identify the place or location of an event.

POST {POSTAL_CODE}:=
A code used by a postal service to identify an area to facilitate mail handling.

PROB {PROBATE}:=
An event of judicial determination of the validity of a will. May indicate several related court activities over several dates.

PROP {PROPERTY}:=
Pertaining to possessions such as real estate or other property of interest.

PUBL {PUBLICATION}:=
Refers to when and/or where a work was published or created.

QUAY {QUALITY_OF_DATA}:=
An assessment of the certainty of the evidence to support the conclusion drawn from evidence.

REFN {REFERENCE}:=

A description or number used to identify an item for filing, storage, or other reference purposes.

RELA {RELATIONSHIP}:=
A relationship value between the indicated contexts.

RELI {RELIGION}:=
A religious denomination to which a person is affiliated or for which a record applies.

REPO {REPOSITORY}:=
An institution or person that has the specified item as part of their collection(s).

RESI {RESIDENCE}:=
An address or place of residence that a family or individual resided.

RESN {RESTRICTION}:=
A processing indicator signifying access to information has been denied or otherwise restricted.

RETI {RETIREMENT}:=
An event of exiting an occupational relationship with an employer after a qualifying time period.

RFN {REC_FILE_NUMBER}:=
A permanent number assigned to a record that uniquely identifies it within a known file.

> There are two RFN record types within in the GEDCOM 5.5.1 lineage-linked form; SUBM.RFN and INDI.RFN.
>
> - The SUBM.RFN line value must be a <SUBMITTER_REGISTERED_RFN>, and its definition makes it clear that this record is not only LDS-specific, but even product-specific; it is a user-identifier for Ancestral File. User identifiers for Ancestral File are obsolete as Ancestral File is no longer in use.
> - The INDI.RFN line value must be a <PERMANENT_RECORD_FILE_NUMBER>, and its definition makes it clear that this is something for “future implementation” - which never happened.
>
> Both the SUBM.RFN and INDI.RFN record types are obsolete.

RIN {REC_ID_NUMBER}:=
A number assigned to a record by an originating automated system that can be used by a receiving system to report results pertaining to that record.

ROLE {ROLE}:=
A name given to a role played by an individual in connection with an event.

ROMN {ROMANIZED}:=
A romanized variation of a superior text string.

SEX {SEX}:=
Indicates the sex of an individual--~~male or female.~~male, female or unknown.

male or female

The FamilySearch definition is wrong, GEDCOM allows male, female *and unknown*. Although the FamilySearch GEDCOM 5.5.1 specification does not allow it, GEDCOM readers should also accept intersex, see the <SEX_VALUE> annotation.

SLGC {SEALING_CHILD}:=
A religious event pertaining to the sealing of a child to his or her parents in an LDS temple ceremony.

SLGS {SEALING_SPOUSE}:=
A religious event pertaining to the sealing of a husband and wife in an LDS temple ceremony.

SOUR {SOURCE}:=
The initial or original material from which information was obtained.

SPFX {SURN_PREFIX}:=
A name piece used as a non-indexing pre-part of a surname.

SSN {SOC_SEC_NUMBER}:=
A number assigned by the United States Social Security Administration. Used for tax identification purposes.

STAE {STATE}:=
A geographical division of a larger jurisdictional area (country), ~~such as a State~~ such as a province or state.

STAT {STATUS}:=

An assessment of the state or condition of something.

SUBM {SUBMITTER}:=
An individual or organization who contributes genealogical data to a file or transfers it to someone else.

SUBN {SUBMISSION}:=
Pertains to a collection of data issued for processing.

SURN {SURNAME}:=
A family name passed on or used by members of a family.

TEMP {TEMPLE}:=
The name or code that represents the name of an LDS ~~Church Temple~~ temple.

TEXT {TEXT}:=
The exact wording found in an original source document.

TIME {TIME}:=
A time value in a 24-hour clock format, including hours, minutes, and optional seconds, separated by a colon (:). Fractions of seconds are shown in decimal notation.

TITL {TITLE}:=

A description of a specific writing or other work, such as the title of a book when used in a source context, or a formal designation used by an individual in connection with positions of royalty or other social status, such as Grand Duke.

TRLR {TRAILER}:=
At level 0, specifies the end of a GEDCOM ~~transmission~~ file.

TYPE {TYPE}:=
A further qualification to the meaning of the associated superior tag. The value does not have any computer processing reliability. It is more in the form of a short one or two word note that should be displayed any time the associated data is displayed.

VERS {VERSION}:=
Indicates which version of a product, item, or publication is being used or referenced.

WIFE {WIFE}:=
~~An individual in the role as a mother and/or married woman.~~
An individual within a family group (FAM), often female, often partner to a man, and a biological, official or legal parent to the children of this family group.

WIFE may be husband

The name FamilySearch chose for the WIFE record strongly suggests that the line value must identify a wife.
In actual practice, the FAM.WIFE value need not identify a woman, it may identify a man in a gay relationship.

references

Same-Sex Marriage in GEDCOM

mother

The word mother has multiple meanings. Not all mothers fulfil a family role, some are merely biological mothers.

WILL {WILL}:=
A legal document treated as an event, by which a person disposes of his or her estate, to take effect after death. The event date is the date the will was signed while the person was alive. (See also ~~PROBate~~ PROB (probate), page 91.)

WWW {WEB}:=
World Wide Web home page.

Appendix B LDS Temple Codes

See Temple.txt found at ftp site:
ftp://gedcom.org/pub/genealogy/gedcom

LDS temple codes

The so-called LDS temple codes are abbreviations used for LDS temples. These codes are only used within LDS-specific GEDCOM records, which are all obsolete.

Software that still supports LDS-specific GEDCOM records typically includes an up-to-date list of temple codes.
FamilySearch's Personal Ancestral File 5.2.18 contains a hard-coded list that was up-to-date at the time of release (in 2002 CE).

The GEDCOM 5.5 specification includes an list of temple codes as an appendix. The GEDCOM 5.5 errata sheet added the Toronto temple to that list.
The GEDCOM 5.5.1 specification refers to a list that can be updated separate from the GEDCOM specification, but the URL provided for the file `temple.txt` is not valid.

An up-to-date web page of LDS temple codes can be now be found at `https://ldschurchtemples.org/codes/`, but developers may prefer to take the XML or JSON table provided by FamilySearch's *temple-codes* project on GitHib.

references

- LDS temple codes
- GitHub: FamilySearch: temple-codes

Appendix C ANSEL Character Set

The following tables show the spacing and non-spacing diacritic characters that are contained in the ANSEL set required by the languages supported by GEDCOM 5. This table was added to give help to those receiving the GEDCOM standard on disk. *The graphic characters shown are not always accurate, however the name of the diacritic and the decimal equivalent should agree with the ANSEL standard.* ANSEL is implemented by replacing any code page characters that are greater than hex 7E with its ANSEL equivalent from this table. Sometimes this requires one character found from the spacing character table and sometimes this requires two characters. Most diacritical markings require two characters. In this case the non-spacing diacritical mark is found from the non-spacing character table followed by the base character, for example, an a with an ~~angstrom~~ ångström or little circle above is represented by two characters, hex EA followed by the normal hex code for the letter a, which is hex 61.

! HEX is the hexadecimal equivalent to the column and row of the American National Standard Z39.47-1985 table showing the ANSEL character graphic and its 8 bit binary representation. The hexadecimal equivalent is obtained from converting the C/R column in to a hexadecimal number, for example 14/10 converts to EA hex or 234 dec.

! wpcode column shows the WordPerfect (code page and character number, for example 1,2) chosen as the closest representation of the diacritic as shown in WordPerfect 5.1 Appendix P.

! Dec column shows to the decimal equivalent for that diacritic as is used in the ANSEL character set.

! Name column gives the English name of the diacritic.

! example of use column shows an example of words using this diacritic. For the non-spacing diacritic, this mark appears before the character in which it should be superimposed.

~~ANSEL Non-spacing graphic characters~~

~~HEX~~	~~wpcode~~	~~Dec~~	~~Graphic~~	~~Name~~	~~example of use~~
~~E1~~	~~1,0~~	~~225~~	~~●~~	~~grave accent~~	~~règle~~
~~E2~~	~~1,6~~	~~226~~	~~´~~	~~acute accent~~	~~está~~
~~E3~~	~~1,3~~	~~227~~	~~^~~	~~circumflex accent~~	~~même~~
~~E4~~	~~1,2~~	~~228~~	~~˜~~	~~tilde~~	~~niño~~
~~E5~~	~~1,8~~	~~229~~	~~¯~~	~~macron~~	~~g●j●js~~

97

98

~~C/R~~	~~wpcode~~	~~Dec~~	~~Graphic~~	~~Name~~	~~example of use~~
~~E6~~	~~1,22~~	~~230~~	~~●~~	~~breve~~	~~alt●~~

~~E7~~	~~1,15~~	~~231~~	●	~~dot above~~	~~●aba~~
~~E8~~	~~1,7~~	~~232~~	¨	~~umlaut (diaeresis)~~	~~öppna~~
~~E9~~	~~1,19~~	~~233~~	△	~~hacek~~	~~v●dy~~
~~EA~~	~~1,14~~	~~234~~	●	~~circle above (angstrom)~~	~~hår~~
~~ED~~	~~1,10~~	~~237~~	●	~~high comma, off center~~	~~rozdel●ovac~~
~~EE~~	~~1,16~~	~~238~~	●	~~double acute accent~~	~~id●szaki~~
~~F0~~	~~1,17~~	~~240~~	●	~~cedilla~~	~~ça~~
~~F1~~	~~1,18~~	~~241~~	●	~~right hook, ogonek~~	~~viet●~~
~~F6~~	~~2,7~~	~~246~~	●	~~underscore~~	~~s̲amar~~
~~FE~~	~~1,9~~	~~254~~	●	~~high comma, centered~~	~~g●otermika~~

98

99

~~ANSEL Spacing graphic characters~~

~~C/R~~	~~wpcode~~	~~Dec~~	~~Graphic~~	~~Name~~	~~example of use~~
~~A1~~	~~1,152~~	~~161~~	●	~~slash L uppercase~~	~~●ód●~~
~~A2~~	~~1,80~~	~~162~~	~~Ø~~	~~slash O uppercase~~	~~Øst~~
~~A3~~	~~1,78~~	~~163~~	●	~~slash D uppercase~~	~~●uro~~
~~A4~~	~~1,88~~	~~164~~	~~Þ~~	~~thorn uppercase~~	~~Þann~~
~~A5~~	~~1,36~~	~~165~~	~~Æ~~	~~ligature AE uppercase~~	~~Ægir~~
~~A6~~	~~1,166~~	~~166~~	~~Œ~~	~~ligature OE uppercase~~	~~Œuvre~~
~~A8~~	~~1,1~~	~~168~~	●	~~middle dot~~	~~novel●la~~
~~A9~~	~~5,28~~	~~169~~	●	~~musical flat~~	
~~AA~~	~~4,32~~	~~170~~	~~®~~	~~registered trademark~~	
~~AB~~	~~6,1~~	~~171~~	~~±~~	~~plus or minus~~	
~~AE~~	~~1,11~~	~~174~~	●	~~alif~~	~~Un●yusho~~
~~B0~~	~~2,11~~	~~176~~	●	~~ayn~~	~~fa●il~~
~~B1~~	~~1,153~~	~~177~~	●	~~slash l lowercase~~	~~rozbi●~~
~~B2~~	~~1,81~~	~~178~~	~~ø~~	~~slash o lowercase~~	~~høj~~
~~B3~~	~~1,79~~	~~179~~	●	~~slash d lowercase~~	~~●avola~~
~~B4~~	~~1,89~~	~~180~~	~~þ~~	~~thorn lowercase~~	~~þann~~
~~B5~~	~~1,37~~	~~181~~	~~æ~~	~~ligature ae lowercase~~	~~skæg~~
~~B6~~	~~1,167~~	~~182~~	~~œ~~	~~ligature oe lowercase~~	~~œuvre~~
~~B8~~	~~1,24~~	~~184~~	~~ı~~	~~dotless i lowercase~~	~~masalı~~
~~B9~~	~~4,11~~	~~185~~	~~£~~	~~British pound~~	~~£5.00~~
~~BA~~	~~1,87~~	~~186~~	~~ð~~	~~eth~~	~~verður~~
~~C3~~	~~4,23~~	~~195~~	~~©~~	~~copyright mark~~	~~©1993~~
~~C5~~	~~4,8~~	~~197~~	~~¿~~	~~inverted question mark~~	~~¿Qué~~
~~C6~~	~~4,7~~	~~198~~	~~¡~~	~~inverted exclamation mark~~	~~¡Esta~~

~~CF~~	~~1,23~~	~~207~~	~~ß~~	~~Ess Zed~~	~~Preußen~~

ANSEL Non-spacing graphic characters

HEX	wpcode	Dec	Graphic	Name	example of use
E1	1,0	225	`	grave accent	règle
E2	1,6	226	´	acute accent	está
E3	1,3	227	ˆ	circumflex accent	même
E4	1,2	228	˜	tilde	niño
E5	1,8	229	¯	macron	gājējs
E6	1,22	230	˘	breve	altă
E7	1,15	231	˙	dot above	żaba
E8	1,7	232	¨	umlaut (diaeresis)	öppna
E9	1,19	233	ˇ	hacek	vždy
EA	1,14	234	˚	circle above (angstrom)	hår
ED	1,10	237	̕	high comma, off center	rozdelo̕vac
EE	1,16	238	˝	double acute accent	időszaki
F0	1,17	240	¸	cedilla	ça
F1	1,18	241	˛	right hook, ogonek	vietą
F6	2,7	246	_	underscore	s̲amar
FE	1,9	254	̓	high comma, centered	ge̓otermika

ANSEL Spacing graphic characters

C/R	wpcode	Dec	Graphic	Name	example of use
A1	1,152	161	Ł	slash L — uppercase	Łódź
A2	1,80	162	Ø	slash O — uppercase	Øst
A3	1,78	163	Đ	slash D — uppercase	Đuro
A4	1,88	164	Þ	thorn — uppercase	Þann
A5	1,36	165	Æ	ligature AE — uppercase	Ægir
A6	1,166	166	Œ	ligature OE — uppercase	Œuvre
A8	1,1	168	·	middle dot	novel·la
A9	5,28	169	♭	musical flat	B♭
AA	4,32	170	®	registered trademark	ABC®
AB	6,1	171	±	plus or minus	A±B
AE	1,11	174	◌’	alif	Un’yusho
B0	2,11	176	◌‘	ayn	fa‘il
B1	1,153	177	ł	slash l— lowercase	rozbił
B2	1,81	178	ø	slash o— lowercase	høj
B3	1,79	179	đ	slash d— lowercase	đavola
B4	1,89	180	þ	thorn— lowercase	þann
B5	1,37	181	æ	ligature ae— lowercase	skæg
B6	1,167	182	œ	ligature oe— lowercase	œuvre
B8	1,24	184	ı	dotless i— lowercase	masalı
B9	4,11	185	£	British pound	£5.00
BA	1,87	186	ð	eth	verður
C3	4,23	195	©	copyright mark	©1993
C5	4,8	197	¿	inverted question mark	¿Qué?
C6	4,7	198	¡	inverted exclamation mark	¡Esta!

CF	1,23	207	ß	Ess Zed	Preußen

ANSEL Tables

The FamilySearch GEDCOM 5.5.1 specification ANSEL tables generally do not appear as FamilySearch intended.
Many characters in the graphics column and many a text shown in example of use column are obviously incorrect: thirty-two black circles are shown instead of the intended character, and some examples are missing.

WordPerfect fonts

The black circles are a fall-back character that Adobe Reader displays because the tables demands WordPerfect-specific fonts that most users do not have on their system, to wit, `WP MultinationalA Roman`, `WP MathA` and `WP Phonetic`.
The FamilySearch table shown here does not duplicate the reliance on these WordPerfect fonts, it just shows the black circles. The replacement table does not rely on application-specific fonts, and should display correctly.

Corrected and improved ANSEL Tables

These corrected tables shows all graphics and example of use correctly. The table was repaired by referencing both the original ANSEL specification and the older FamilySearch GEDCOM 5.5 specification.

The table has been further improved by displaying the *actual* combining characters, not some non-combining character that approximates it, in the graphics column. For example, the first character shown is U+0300 Combining Grave Accent, not U+0060 Grave Accent.

These improvements result in the table the FamilySearch GEDCOM 5.5.1 specification meant to show. That, however, is still not the table you should be using.

bonus tables

Different versions of FamilySearch GEDCOM contain different ANSEL tables; the different GEDCOM versions omit different ANSEL characters and add different extensions.

A bonus section provides an extensive ANSEL table, that includes all ANSEL characters and all LDS extensions. It additionally lists the corresponding Unicode code points and character name.

references

GEDCOM 5.5.1 Specification ANSEL Table

end of specification

The end of the FamilySearch GEDCOM 5.5.1 specification.

Bonus ANSEL Tables

Bonus ANSEL Tables

tables combined

The ANSEL specification was originally created by the Standards Committee on Coded Character Sets for Bibliographic Information Interchange of America.
The bonus ANSEL tables provided here are combination of the original tables in the ANSEL standard and the ANSEL tables published in different GEDCOM versions. This table includes all the ANSEL characters that FamilySearch omitted, as well as all the LDS extensions they added in several GEDCOM versions.

left hoof

The ANSEL character 0xF7 left hook appears as "left hoof" in the ANSEL standard. That is an acknowledged typographical error.

Unicode mapping

These tables do not only have more rows for more code points, they also have two extra columns. The two extra columns in these bonus tables list the corresponding Unicode code point and character name. Thus, these tables provide an ANSEL to Unicode mapping, but there is more to conversion from ANSEL to Unicode or back than just these tables. One issue is that in ANSEL, modifying characters come before the base characters, while in Unicode they come after the base character. The bonus chapter *ANSEL / Unicode conversion* provides more information.

combining characters

The tables display the actual combining characters, not some non-combining character that approximates it, in the graphics column. For example, the second character in this complete table is U+0300 Combining Grave Accent, not U+0060 Grave Accent. To show the combining characters correctly and make the relative position of the combining character more obvious, the table follows the convention of using ◌ as a place-holder character.

ANSEL pre-composed characters

Most of the characters in the second table are pre-composed characters, but some are *Spacing Modifier Letters*; they are letters that modify the preceding character, and *do* take up horizontal space.

The existence of pre-composed characters in ANSEL arguably breaks its design philosophy, and is something that requires attention when converting from Unicode to ANSEL.

references

- GEDCOM 5.5.1 Specification ANSEL Table
- LDS ANSEL versus LDS ANSEL
- ANSEL Alif & Ayn
- ANSEL / Unicode Conversion Tables

ANSEL Non-spacing graphic characters

Hex	wpcode	Dec	Graphic	Name	example of use	code point	Name
E0	2,4	224	◌̉	low rising tone mark	củi	U+0309	Combining Hook Above
E1	1,0	225	◌̀	grave accent	règle	U+0300	Combining Grave Accent
E2	1,6	226	◌́	acute accent	está	U+0301	Combining Acute Accent
E3	1,3	227	◌̂	circumflex accent	même	U+0302	Combining Circumflex Accent
E4	1,2	228	◌̃	tilde	niño	U+0303	Combining Tilde
E5	1,8	229	◌̄	macron	gājējs	U+0304	Combining Macron
E6	1,22	230	◌̆	breve	altă	U+0306	Combining Breve
E7	1,15	231	◌̇	dot above	żaba	U+0307	Combining Dot Above
E8	1,7	232	◌̈	umlaut (diaeresis)	öppna	U+0308	Combining Diaeresis
E9	1,19	233	◌̌	hacek	vždy	U+030C	Combining Caron
EA	1,14	234	◌̊	circle above (angstrom)	hår	U+030A	Combining Ring Above
EB	2,11	235	◌︠	ligature, left half	akademii︠a︡	U+FE20	Combining Ligature Left Half
EC	2,12	236	◌︡	ligature, right half	akademii︠a︡	U+FE21	Combining Ligature Right Half
ED	1,10	237	◌̕	high comma, off center	rozdel̕ovac	U+0315	Combining Comma Above Right
EE	1,16	238	◌̋	double acute accent	időszaki	U+030B	Combining Double Acute Accent
EF	2,25	239	◌̐	candrabindu	Ali̐ev	U+0310	Combining Cadrabindu
F0	1,17	240	◌̧	cedilla	ça	U+0327	Combining Cedilla
F1	1,18	241	◌̨	right hook, ogonek	vietą	U+0328	Combining Ogonek
F2	2,0	242	◌̣	dot below	teḍa	U+0323	Combining Dot Below
F3	2,1	243	◌̤	double dot below	k̲h̲ut̤bah	U+0324	Combining Diaeresis Below
F4	2,3	244	◌̥	circle below	Samskr̥ta	U+0325	Combining Ring Below
F5	2,6	245	◌̳	double underscore	G̳hulam	U+0333	Combining Double Low Line
F6	2,7	246	◌̲	underscore	s̲amar	U+0332	Combining Low Line
F7	2,16	247	◌̦	left hook	dārziņa	U+0326	Combining Comma Below
F8	2,14	248	◌̜	right cedilla	kho̜ng	U+031C	Combining Left Half Ring Below
F9	2,9	249	◌̮	half circle below (upadhmaniya)	ḫumantuš	U+032E	Combining Breve Below
FA		250	◌︢	double tilde, left half	n︢g︣alan	U+FE22	Combining Double Tilde Left Half
FB		251	◌︣	double tilde, right half	n︢g︣alan	U+FE23	Combining Double Tilde Right Half
FC	1,5	252	◌/	diacritic slash through char		U+0338	Combining Long Solidus Overlay
FD		253	�	*unused*		U+FFFD	Replacement Character
FE	1,9	254	◌̓	high comma, centered	ge̓otermika	U+0313	Combining Comma Above
FF		255	�	*illegal*		U+FFFD	Replacement Character

ANSEL Spacing graphic characters

Hex	wpcode	Dec	Graphic	Name	example of use	code point	Name
A0		160	�	*unused*		U+FFFD	Replacement Character
A1	1,152	161	Ł	slash L — uppercase	Łódź	U+0141	Latin Capital Letter L with Stroke
A2	1,80	162	Ø	slash O — uppercase	Øst	U+00D8	Latin Capital Letter O with Stroke
A3	1,78	163	Đ	slash D — uppercase	Đuro	U+0110	Latin Capital Letter D with Stroke
A4	1,88	164	Þ	thorn — uppercase	Þann	U+00DE	Latin Capital Letter Thorn
A5	1,36	165	Æ	ligature AE — uppercase	Ægir	U+00C6	Latin Capital Letter AE
A6	1,166	166	Œ	ligature OE — uppercase	Œuvre	U+0152	Latin Capital Ligature OE
A7	1,6	167	◌ʹ	mjagkij znak	fakulʹtet	U+02B9	Modifier Letter Prime
A8	1,1	168	·	middle dot	novel·la	U+00B7	Middle Dot
A9	5,28	169	♭	musical flat	B♭	U+266D	Musical Flat Sign
AA	4,32	170	®	registered trademark	ABC®	U+00AE	Registered Sign
AB	6,1	171	±	plus or minus	A±B	U+00B1	Plus-Minus Sign
AC	1,230	172	Ơ	hook O - uppercase	BƠ	U+01A0	Latin Capital Leter O with Horn
AD	1,232	173	Ư	hook U - uppercase	XƯA	U+01AF	Latin Capital Letter U with Horn
AE	1,11	174	◌’	alif	Un’yusho	U+02BC	Modifier Letter Apostrophe
AF		175	�	*unused*		U+FFFD	Replacement Character
B0	2,11	176	◌ʻ	ayn	faʻil	U+02BB	Modifier Letter Turned Comma
B1	1,153	177	ł	slash l— lowercase	rozbił	U+0142	Latin Small Letter L with Stroke
B2	1,81	178	ø	slash o— lowercase	høj	U+00F8	Latin Small Letter O with Stroke
B3	1,79	179	đ	slash d— lowercase	đavola	U+0111	Latin Small Letter D with Stroke
B4	1,89	180	þ	thorn— lowercase	þann	U+00FE	Latin Small Letter Thorn
B5	1,37	181	æ	ligature ae— lowercase	skæg	U+00E6	Latin Small Letter AE
B6	1,167	182	œ	ligature oe— lowercase	œuvre	U+0153	Latin Small Ligature OE
B7	1,16	183	◌ʺ	hard sign (tvjordyj znak)	obʺi͡avlenie	U+02BA	Modified Letter Double Prime
B8	1,24	184	ı	dotless i— lowercase	masalı	U+0131	Latin Small Letter Dotless I
B9	4,11	185	£	British pound	£5.00	U+00A3	Pound Sign
BA	1,87	186	ð	eth	verður	U+00F0	Latin Small Letter Eth
BB		187	�	*unused*		U+FFFD	Replacement Character
BC	1,231	188	ơ	hook o - lowercase	Sơ	U+01A1	Latin Small O with Horn
BD	1,233	189	ư	hook u - uppercase	Tự Đức	U+01B0	Latin Small U with Horn
BE		190	□	empty box (LDS-extension)		U+25A1	Empty Box
BF		191	■	black box (LDS-extension)		U+25A0	Black Box
C0	6,33	192	°	degree sign	10°C.	U+00B0	Degree Sign
C1	6,49	193	ℓ	script l	25 ℓ.	U+2113	Script Small L
C2	4,71	194	℗	phono copyright mark	Decca℗	U+2117	Sound recording copyright
C3	4,23	195	©	copyright mark	©1993	U+00A9	Copyright Sign
C4	5,27	196	♯	music sharp sign	D♯	U+266F	Music Sharp Sign
C5	4,8	197	¿	inverted question mark	¿Qué?	U+00BF	Inverted Question Mark
C6	4,7	198	¡	inverted exclamation mark	¡Esta!	U+00A1	Inverted Exclamation Mark
C7		199	�	*unused*		U+FFFD	Replacement Character
C8		200	�	*unused*		U+FFFD	Replacement Character
C9		201	�	*unused*		U+FFFD	Replacement Character
CA		202	�	*unused*		U+FFFD	Replacement Character
CB		203	�	*unused*		U+FFFD	Replacement Character
CC		204	�	*unused*		U+FFFD	Replacement Character
CD		205	e	e in middle of line (LDS extension)		U+0065	Latin Small Letter E
CE		206	o	o in middle of line (LDS extension)		U+006F	Latin Small Letter O
CF	1,23	207	ß	Ess Zed	Preußen	U+00DF	Latin Small Letter Sharp S

ANSEL / Unicode conversion

ANSEL to Unicode conversion

There is more to conversion from ANSEL to Unicode than just the ANSEL to Unicode conversion tables. In ANSEL, combining characters (modifiers) come before the base characters, while in Unicode they come after the base character.

After conversion to Unicode, the result should be normalised to the normalisation for appropriate for the platform. On Mac OS X, the string must be converted to Unicode Normal Form D (decomposed characters), while on Windows, it must be converted to Unicode Normal Form C (composed characters). This normalisation step may not be skipped for either platform or normal form, as ANSEL text may contain both composed and decomposed characters.

ANSEL to Unicode conversion algorithm

1. Convert each ANSEL code point into the corresponding Unicode code point
2. mirror the positions of each base character and its modifiers (modifiers after instead of before base characters)
3. Convert to Unicode Normal Form C or Unicode Normal Form D, as appropriate for the platform

Unicode to ANSEL conversion

Conversion from Unicode to ANSEL may seem simple; just perform the ANSEL to Unicode conversion step in reverse order:

1. Make sure the Unicode string is in Unicode Normal Form D
2. mirror the positions of each base characters and its modifiers (modifiers before instead of after base characters)
3. Replace Unicode code points with their ANSEL equivalents

However, Unicode to ANSEL conversion is more complicated than that. There are two issues that require attention:

- unsupported characters
- ANSEL's pre-composed characters

unsupported characters

First of all, ANSEL does not support all the same characters that Unicode supports, and does not feature an Replacement Character like Unicode does. The widespread convention for such cases is to replace the unsupported character by a question mark. When doing so, care much be taken replace an unsupported character and its modifiers by just one single question mark.

ANSEL does not support all same combining characters as Unicode either. Generally speaking, when a modifier isn't supported, the Unicode character isn't supported, and should be replaced by a question mark. It is arguably better, albeit inconsistent, to keep the base character if supported, and simply lose any unsupported modifiers.

ANSEL's pre-composed characters

Special attention must be paid to ANSEL's pre-composed characters. For example, ANSEL does not contain an equivalent for *U+031B Combining Horn*, but it does contain equivalents for the pre-composed characters *U+101A Latin Capital Letter O with Horn* and *U+10AF Latin Capital Letter U with Horn*. If your Unicode to ANSEL conversion does not take the existence of ANSEL's pre-composed characters into account, your conversion will reduce those characters to their base character.

Unicode to ANSEL conversion algorithm

1. Make sure the Unicode string is in Unicode Normal Form D
2. De-normalise for pre-composed characters supported by ANSEL
3. mirror the positions of each base characters and its modifiers (modifiers before instead of after base characters)
4. Replace Unicode code points with their ANSEL equivalents
   - Replace unsupported characters with a question mark
   - Take care to replace an unsupported characters and all its modifiers by a single question mark

    - If a base character is supported, but a modifier is not, keep the base character, but leave of the modifier

The de-normalisation step is essential for taking advantage of ANSEL's pre-composed characters, and not ending up with just their base character instead.

operating system support

It is generally unwise to try and roll your own character set conversion functions, and best to rely on built-in functions for any character set handling and conversions. All major operating systems, operating environments, and several third-party libraries provide character set conversion functions, but no major systems and few libraries support ANSEL. However, all major operating systems provide functions for conversion of strings to Unicode Normal Forms:

- Microsoft Windows provides the `NormalizeString()` and `IsNormalizedString()` functions.
- Apple Mac OS X provides the `NSString()` function.
- Oracle Java provides the `Normalizer2` class, with multiple methods.
- Microsoft .NET provides the `String.Normalize()` method.
- Google Android provides the same `Normalizer2` class.
- Apple iOS provides the same `NSString()` function as Apple Mac OS X.

best practice for genealogy applications

- support import of ANSEL GEDCOM files
- do not export to ANSEL GEDCOM
    - always export to UTF-8 or UTF-16
- use built-in functions for Unicode normalisation

references

ANSEL / Unicode Conversion Algorithms

GEDCOM Validation

Developers must make sure that the GEDCOM files exported by their application are valid GEDCOM files. There are several GEDCOM validators and some other tools available to help with that. These tools are fairly mature, but are themselves applications that may contain errors. When in doubt, try multiple tools, and re-read the specification.

GEDCOM Validators

GedChk

GedChk is a free GEDCOM validator by FamilySearch. GedChk is a free command-line application for MS-DOS, for which only version 0.9 Beta is available. GedChk supports GEDCOM 5.5 and 5.5.1. GedChk is available for FTP from the LDS website. GedChk is not well-behaved enough to run in a Windows DOS Box, but it does run in DOSBox, which is freely available for many operating systems.

VGedX

VGedX is a free GEDCOM validator created by Tim Forsythe, the creator of Gigatrees.
VGedX is a command-line application for Windows, complemented by windowed configuration interface. VGedX supports GEDCOM 5.5, GEDCOM 5.5.1 and GEDCOM 5.6, and the user interface is available in multiple languages. VGedX is available for Windows, and requires 64-bit Windows 7 or later.

Chronoplex GEDCOM Validator

The Chronoplex GEDCOM Validator is a free product of Chronoplex (Andrew Hoyle), creators of Chronoplex Family Tree. The Chronoplex GEDCOM Validator is a Microsoft .NET application for Windows. The Chronoplex GEDCOM Validator supports GEDCOM 5.5, GEDCOM 5.5.1 and GEDCOM 5.6, and the user interface is available in multiple languages. The Chronoplex GEDCOM Validator requires Windows XP SP3 or later, and Microsoft .NET 4.0 or later.

GED-inline

GED-inline is a free on-line GEDCOM validator by Nigel Munro Parker.
GED-inline supports GEDCOM 5.5 and GEDCOM 5.5.1.

additional validation tools

Behold

Behold is a commercial genealogy viewer by Louis Kessler. Behold's forgiving GEDCOM reader provides many error and warning messages, that Behold be thought of as a GEDCOM validator in disguise.
Behold supports every version of GEDCOM, including GEDCOM 1.0. Behold requires Windows NT 4.0 or later.

GedPad

GedPad is free product of Nigel Button Software, makers of The Complete Genealogy Reporter. GedPad isn't a validator, but a GEDCOM editor with some validation features. GedPad's user-interface is a bit unusual and it does not support ASCII or ANSEL files, but it can be a handy tool to have around when some GEDCOM file gives you trouble.

Genealogica Grafica

Genealogica Grafica is a free product by Tom de Neef, and the successor to his previous product, KStableau. Genealogica Grafica is a tool for creation of web charts and reports. On loading a GEDCOM file, it provides a report that includes GEDCOM errors and genealogy consistency checks. Genealogica Grafica support GEDCOM 5.5 and GEDCOM 5.5.1. Genealogica Grafica requires Windows.

references

- GEDCOM Validation
- DOSBox
- ftp://ftp.ldschurch.org/genealogy/GEDCOM/Utility/GedChk/

- VGedX, the Gigatrees GEDCOM Validator
- Chronoplex GEDCOM Validator
- GED-inline
- Behold
- GedPad
- Genealogica Grafica

GEDCOM 5.5.1 Version Detection

GEDCOM 5.5.1 version detection is complicated by the fact that several applications produce GEDCOM 5.5.1 files with a HEAD.GEDC.VERSvalue of `5.5` instead of the correct `5.5.1`.
In fact, nowadays, most GEDCOM files that claim to be GEDCOM 5.5 files are probably GEDCOM 5.5.1 files.

FamilySearch PAF is the best known application to lie about the GEDCOM version used, but not the only one. Other products that use GEDCOM 5.5.1 but lie that they are using GEDCOM 5.5 include Ancestral Quest, MyHeritage Family Tree Builder, Ancestry.com's New Family Tree Maker, Family Tree Heritage, Geni.com, GEDitCOM II, gramps, Heredis 98, Kith and Kin, Legacy Family Tree, Ancestry.com's Online Family Tree, Personal Ancestry Writer II, Reunion, The Next Generation of Genealogy Sitebuilding (TNG), and WikiTree.

Some products that correctly identify their use of GEDCOM 5.5.1 include Ahnenblatt, Family Echo, Ancestry.com's New Family Tree Maker, findmypast Family Tree, gedcom4j, Legacy Family Tree, MacFamilyTree, Picturae's Memorix Maior, phpGedView, RootsMagic, and WebTrees.

Some products appear in both list because some versions of the product get it wrong, while later versions get it right. For the detection algorithm presented here, all that matters is when products started using GEDCOM 5.5.1 yet continued to label their GEDCOM files (incorrectly) as GEDCOM 5.5. It does not matter whether or when the developers fixed their product to starting labelling their GEDCOM 5.5.1 files correctly.

GEDCOM 5.5 versus GEDCOM 5.5.1 file content

It is possible to distinguish between GEDCOM 5.5 and GEDCOM 5.5.1 files based on their content, for example with the following rules:

- if the GEDCOM files contains a OBJE.BLOB record, it is GEDCOM 5.5
- if the GEDCOM files contains a PLAC.MAP record, it is GEDCOM 5.5.1

However, a GEDCOM reader should be able to determine the actual GEDCOM version without looking beyond the GEDCOM header.

GEDCOM 5.5.1 detection algorithm

This is a the algorithm for distinguishing between GEDCOM 5.5 and GEDCOM 5.5.1. Numbered step should be performed in the order shown. As soon as a rule matches and the actual GEDCOM version is known, no further tests should be performed.

- Detect the character encoding first
- read the HEAD.GEDC.VERSline value
- if the value is `5.5.1`, it is GEDCOM 5.5.1
- if the value is `5.5`, it may be either GEDCOM 5.5 or GEDCOM 5.5.1

    1. (Optional): check the system identifier (HEAD.SOUR line value)
       against a list of system identifiers for products discontinued before (or some time after) the introduction of GEDCOM 5.5.1:
       if there is a match, it's definitely GEDCOM 5.5, and not GEDCOM 5.5.1.

    2. Check the GEDCOM encoding
        - if the character encoding is UTF-8, it is GEDCOM 5.5.1

    3. Check for tags introduced in GEDCOM 5.5.1
        - if the header contains a HEAD.SOUR.CORP.ADR3 record, it is GEDCOM 5.5.1
        - if the header contains a HEAD.SOUR.CORP.EMAIL record, it is GEDCOM 5.5.1
        - if the header contains a HEAD.SOUR.CORP.FAX record, it is GEDCOM 5.5.1
        - if the header contains a HEAD.SOUR.CORP.WWW record, it is GEDCOM 5.5.1

    4. Check for known products and versions.

        1. Make sure you have a system identifier.
            - if the system identifier is too long (it is for The Next Generation of Genealogy Sitebuilding), issue a non-fatal error and continue
            - If the (HEAD.SOUR line value is missing (illegal!)

            - try to identify the correct system identifier by other means (see section on MacFamilyTree below)
            - if you failed to identify it, issue a fatal error and abort
            - if you managed to identify it, issue a non-fatal error and continue with that system identifier

    2. Make sure you have a version number.
        - If the (HEAD.VERS line value is missing, issue a fatal error and abort.
        - If the (HEAD.VERS line value isn't a proper version number
            - try to extract the version number (see section on Family Tree Maker below)
            - if you failed to extract it, issue a fatal error and abort
            - if you managed extract it, issue a non-fatal error and continue with that version number

    3. Perform the known product / version number checks.
        - if the system identifier is `PAF`
            - if the version number is less than `5.0`, it is GEDCOM 5.5
            - if the version number is `5.0` or more, it is GEDCOM 5.5.1
        - See the table below for more products and version numbers.

5. Check for user-defined tags being used instead of GEDCOM 5.5.1 tags
    - if the header contains a HEAD.SOUR.CORP._EMAIL record, it must be GEDCOM 5.5
    - if the header contains a HEAD.SOUR.CORP._FAX record, it must be GEDCOM 5.5
    - if the header contains a HEAD.SOUR.CORP._WWW record, it must be GEDCOM 5.5

6. None of the previous rules matched? Assume it is 5.5 as stated.

- if the HEAD.GEDC.VERSis `5.5`, but the actual GEDCOM version is 5.5.1, issue a non-fatal error
- Issue an informative message that the file will be processed as GEDCOM 5.5 or GEDCOM 5.5.1, whichever version was detected

Notice the last two instructions in the algorithm. It is good to let the user know that an ostensible GEDCOM 5.5 file is really a GEDCOM 5.5.1 file. It is good to explicitly inform the user how the file will be processed.

Product table

Version that product started using GEDCOM 5.5.1

application	system identifier	version
Personal Ancestral File	PAF	5.0
Ancestral Quest	AncestQuest	12.0
Family Tree Maker	FTM	21.0.0.466
GenoPro	GenoPro	2.0
gramps	Gramps	2.0
Lifelines	Lifelines	3.0
MacFamilyTree	MacFamilyTree	5.7.8
MagiKey Family Tree	MagiKey Family Tree	*all versions*
Family Tree Builder	MYHERITAGE	5.5
Reunion	Reunion	9.0
RootsMagic	RootsMagic	*all versions*
The Next Generation of Genealogy Sitebuilding	The Next Generation of Genealogy Sitebuilding	7.0
PRO-GEN	PRO-GEN	3.0

This list is not exhaustive.

product information

Family Tree Maker version number

The Family Tree Maker name is used for what are really three different, consecutive products (and then there's also Family Tree Maker for Macintosh):

- the original Family Tree Maker, an MS-DOS application
- Family Tree Maker for Windows a Windows applications
- New Family Tree Maker for Windows, a Microsoft .NET application

GEDCOM system identifiers

Those three different products use only two different GEDCOM system identifiers. Family Tree Maker for MS-DOS uses `FTM` Family Tree Maker for Windows uses `FTW` and New Family Tree Maker for Windows uses `FTM` again.
This is wrong, and may mess up some tests, but does not mess up this test, because the version numbers continued to increase with new products. Family Tree Maker 2012 Service Pack 2 (version 21.0.0.46) is the first version to produce ostensible GEDCOM 5.5 files that are really GEDCOM 5.5.1 files. Family Tree Maker for MS-DOS pre-dates GEDCOM 5.5.1, and all its versions numbers are considerably lower than those of New Family Tree Maker for Windows. So, if the Family Tree Make version number is 21.0.0.466 or later, it is GEDCOM 5.5.1.

FTM version number

When Ancestry.com created New Family Tree Maker for Windows, they messed up the version number record, by practically treating it as product name record. This was only fixed after Software MacKiev became the owner of Family Tree Maker.
Family Tree Maker 2012's version number is `21.0.0.38`, but the HEAD.SOUR.VERS line value isn't `21.0.0.38`, it is `Family Tree Maker (21.0.0.38)` instead. That value is illegal, for two reasons:

- it clearly isn't a proper version number
- this value (29 characters) is longer than the maximum HEAD.SOUR.VERS line value length of 15 code units

A GEDCOM reader should definitely issue an error when encountering this, and it is okay for that error to be an fatal error; it is fine to reject an ostensible GEDCOM file if it does not even have a valid GEDCOM header.
It is not difficult to recognise the `Family Tree Maker (`*`version number`*`)` pattern and extract the version number from it, and this is what a GEDCOM reader has to do before it can perform this test.

32-bit versus 64-bit version number

From Family Tree Maker 2014 onwards, Family Tree Maker comes in both a 32-bit and a 64-bit build, and those two different builds have different version numbers. The version number of the 32-bit Family Tree Maker 2014 is `22.0.0.207`, while the version number of the 64-bit Family Tree Maker 2014 is `22.0.0.1207`.
This does not complicate the test, as the switch to GEDCOM 5.5.1 happened with version Family Tree Maker 2012 Service Pack 2 (version 21.0.0.46), for which there is only a 32-bit build, and therefore just one version number to compare to.

Family Tree Maker for Macintosh

There are two Family Tree Maker for Mac products.

- Family Tree Maker for Mac, based on Family Tree Maker for Windows, released by Broderbund in 1997 CE
- Family Tree Maker for Mac, based on New Family Tree Maker, released by Ancestry.com in 2010

The new Family Tree Maker for Mac uses the same system identifier (`FTM`) as New Family Tree Maker (for Windows), and that is wrong, because it is definitely another product. Family Tree Maker for Mac use the same version numbering as New Family Tree Maker, and Ancestry.com Family Tree Maker for Mac also use the same illegal format HEAD.SOUR.VERS format.
The GEDCOM 5.5.1 detection code for Family Tree Maker does not to distinguish between Family Tree Maker (for Windows) and Family Tree Maker for Mac.

recognising MacFamilyTree GEDCOM files

Recognising MacFamilyTree GEDCOM files is harder than it should be. In many MacFamilyTree GEDCOM files, the HEAD.SOUR value is empty. This is illegal, and a GEDCOM reader should definitely issue an error, and it is okay for that error to be an fatal error, because it's fine to reject an ostensible GEDCOM file if it does not even have a valid GEDCOM header.
It is possible for a GEDCOM reader to recognise the GEDCOM file as MacFamilyTree GEDCOM file; if the HEAD.SOUR line value is empty, but the HEAD.SOUR.NAME line value is `MaFamilyTree`, it is a MacFamilyTree GEDCOM file.

Legacy Family Tree

Legacy Family Tree GEDCOM files may lack the GEDCOM version number! See the *Legacy Family Tree GEDCOM Version* annotation.

table updates

This algorithm can be improved by updating the table with more product-specific information. This bonus chapter is based on and practically identical to the *GEDCOM 5.5.1 Version Detection* article, which is likely to be updated sooner and more frequently than this chapter.

references

- GEDCOM 5.5.1 Version Detection
- GEDCOM Version Detection
- Truncated GEDCOM Version
- GEDCOM System Identifiers
- New Family Tree Maker versions
- Ancestry.com & Software MacKiev Family Tree Maker GEDCOM Header

end of specification & bonus

The end of the FamilySearch GEDCOM 5.5.1 specification and Bonus ANSEL Tables.