



Extended Legacy Format (ELF): Data Model

13 August 2018

Editorial note — This is an **exploratory draft** of the genealogical data model for FHISO's proposed suite of Extended Legacy Format (ELF) standards. This document is not endorsed by the FHISO membership, and may be updated, replaced or obsoleted by other documents at any time.

Comments on this draft should be directed to the tsc-public@fhiso.org mailing list.

FHISO's **Extended Legacy Format** (or **ELF**) is a hierarchical serialisation format and genealogical data model that is fully compatible with GEDCOM, but with the addition of a structured extensibility mechanism. It also clarifies some ambiguities that were present in GEDCOM and documents best current practice.

The **GEDCOM** file format developed by The Church of Jesus Christ of Latter-day Saints is the *de facto* standard for the exchange of genealogical data between applications and data providers. Its most recent version is GEDCOM 5.5.1 which was produced in 1999, but despite many technological advances since then, GEDCOM has remained unchanged.

Note — Strictly, [GEDCOM 5.5] was the last version to be publicly released back in 1996. However a draft dated 2 October 1999 of a proposed [GEDCOM 5.5.1] was made public; it is generally considered to have the status of a standard and has been widely implemented as such.

FHISO are undertaking a program of work to produce a modernised yet backward-compatible reformulation of GEDCOM under the name ELF, the new name having been chosen to avoid confusion with any other updates or extensions to GEDCOM, or any future use of the term by The Church of Jesus Christ of Latter-day Saints. This document is one of two that form the initial suite of ELF standards, known collectively as ELF 1.0.0:

- **ELF: Serialisation Format.** This standard defines a general-purpose serialisation format based on the GEDCOM data format which encodes a *dataset* as a hierarchical series of *lines*, and provides low-level facilities such as escaping and extensibility mechanisms.
- **ELF: Date, Age and Time Microformats.** This standard defines microformats for representing dates, ages and times in arbitrary calendars, together with how they are applied to the Gregorian, Julian, French Republican and Hebrew calendars. These formats are largely identical to those used in GEDCOM, but the framework should serve as a basis for future work on calendars.

- **ELF: Data Model.** This standard defines a data model based on the lineage-linked GEDCOM form, reformulated in terms of the serialisation model described in this document. It is not a major update to the GEDCOM data model, but rather a basis for future extension.

1 Version and IRI

This document describes version “1.0.0” of the ELF data model. The version string uses the semantic versioning tradition outlined in <https://semver.org>: three integers, separated by periods, where an application expecting on version can process any other with the same first integer.

The IRI for this any version of the ELF data model can be created by prepending <https://fhiso.org/TR/elf-data-model/v> to the version; for this document’s version, that IRI is

`https://fhiso.org/TR/elf-data-model/v1.0.0`

Editorial note — This IRI was generated just as a placeholder; the correct IRI is not yet determined.

2 General

2.1 Conventions used

Where this standard gives a specific technical meaning to a word or phrase, that word or phrase is formatted in bold text in its initial definition, and in italics when used elsewhere. The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **NOT RECOMMENDED**, **MAY** and **OPTIONAL** in this standard are to be interpreted as described in [RFC 2119].

An application is **conformant** with this standard if and only if it obeys all the requirements and prohibitions contained in this document, as indicated by use of the words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL** and **SHALL NOT**, and the relevant parts of its normative references. Standards referencing this standard **MUST NOT** loosen any of the requirements and prohibitions made by this standard, nor place additional requirements or prohibitions on the constructs defined herein.

Note — Derived standards are not allowed to add or remove requirements or prohibitions on the facilities defined herein so as to preserve interoperability between applications. Data generated by one *conformant* application must always be acceptable to another *conformant* application, regardless of what additional standards each may conform to.

If a *conformant* application encounters data that does not conform to this standard, it **MAY** issue a warning or error message, and **MAY** terminate processing of the document or data fragment.

This standard depends on FHISO's **Basic Concepts for Genealogical Standards** standard. To be *conformant* with this standard, an application **MUST** also be *conformant* with [Basic Concepts]. Concepts defined in that standard are used here without further definition.

Note — In particular, precise meaning of *string*, *character*, *namespace name*, *prefix notation*, *prefix*, *whitespace* and *term* are given in [Basic Concepts].

Indented text in grey or coloured boxes does not form a normative part of this standard, and is labelled as either an example or a note.

Editorial note — Editorial notes, such as this, are used to record outstanding issues, or points where there is not yet consensus; they will be resolved and removed for the final standard. Examples and notes will be retained in the standard.

The grammar given here uses the form of EBNF notation defined in §6 of [XML], except that no significance is attached to the capitalisation of grammar symbols. *Conforming* applications **MUST NOT** generate data not conforming to the syntax given here, but non-conforming syntax **MAY** be accepted and processed by a *conforming* application in an implementation-defined manner.

This standard uses the *prefix notation*, as defined in §4.3 of [Basic Concepts], when discussing specific *terms*. The following *prefix* binding is assumed in this standard:

```
elf https://terms.fhiso.org/elf/
```

Note — The particular *prefix* assigned above have no relevance outside this standard document as *prefix notation* is not used in the formal data model defined by this standard. This notation is simply a notational convenience to make the standard easier to read.

2.2 Line and block strings

A **line string** is a *string* that **SHALL** be *whitespace-normalised* before being processed.

Note — In a *line string*, the production S defined in [Basic Concepts] collapses to a single space *character* (U+0020).

A **line break** is defined as either a carriage return, a line feed, or a carriage return followed by a line feed. It matches the production LB:

```
LB ::= #xD #xA? | #xA
```

A **padded linebreak** is defined as a *linebreak* preceded by zero or more space *characters* or tabs. It matches the production PLB:

```
PLB ::= (#x20 | #x9)* LB
```

Linebreak normalisation is the process of replacing each *padded linebreak* with a single *linebreak*, where all are replaced by the same *linebreak* variant, and removing any U+0020 (space) and U+0009 (tab) from the end of the string.

Editorial note — GEDCOM 5.5.1 is inconsistent on its definition of line break handling. Pages 10 and 37 state that initial spaces are preserved and trailing are removed (on most systems), which is given in the above rules; however page 85 indicates states that both initial and trailing spaces are removed, albeit obliquely.

A **block string** is a *string* that SHALL be *linebreak-normalised* before being processed.

Note — *linebreak normalisation* is provided as a data model parallel to how lines are encoded with CONT tags in [ELF-Serialisation]. It is possible that a future version of this standard might change this presentation, perhaps redefining *block string* as a list of *strings*, each representing a single conceptual “line”.

2.3 Structure type identifiers

The **structure type identifiers** used in this specification are *terms*.

The *term names* of the *structure type identifiers* defined in this standard all begin `https://terms.fhiso.org/elf/`. It is RECOMMENDED that any *extension structure type identifiers* also use the `https` IRI scheme defined in §2.7.1 of [RFC 7230], and an authority component consisting of just a domain name (or subdomain) under the control of the party defining the *extension structure type identifier*.

3 Microformats

Several microformats are used in payloads of various structures below.

3.1 Comma-separated list

A list of *line-strings* serialized with commas in between. There is no mechanism provided for including commas or leading or trailing spaces within an element of a comma-separated list.

3.2 Personal name format

A full name, presented in the order usually spoken and with the capitalization typical of the culture of the named individual.

The text SHOULD NOT include commas or digits.

Editorial note — GEDCOM 5.5.1 (page 56) *requires* that names not include “commas, numbers, or special characters not considered diacritics”. The above is less strict; should we instead require it, or remove it altogether?

It SHOULD include exactly two U+002F SOLIDUS / characters, one on each side of the family name or surname if present, or adjacent to one another if no family name or surname name is known.

Portions of the name MAY be elided and replaced by three U+002E FULL STOP . . . This SHOULD only be done if part of a name is known to exist but its content is not known.

3.3 Language tag format

Editorial note — TO DO: fill this section using the “either GEDCOM or IANA” concept outlined in `languages.tsv`.

4 ELF Datasets

Every Extended Legacy Format (ELF) dataset is two sets of *structures*. The first, described as the `elf:Document`, is a set of `elf:Records` which, with their *substructures*, provide the principle data of the dataset. The other, described as the `elf:Metadata`, is a set of additional structures which, with the *substructures*, provide metadata about the dataset as a whole.

4.1 Multiple versions of the truth

Note — This entire section is non-normative

Sometimes a researcher encounters a state where they are unsure which of a set of alternatives is true. GEDCOM did not provide guidance on how this state should be recorded, and hence neither does ELF. However, we are aware of several approaches that have been used, and with potential caveats associated with each, which ELF implementers should be aware of.

Some researchers create one copy of each possible truth. While this is sometimes obvious (e.g., if a person is listed with two birth events), it is sometimes very much not obvious (e.g., a person listed with two residences might have lived in both places, or the researcher might be unsure which place is correct). Implementers should avoid suggesting either meaning was intended by the creators of data they import.

Some researchers create multiple instances of a single-value substructure, such as an `elf:Event` with several `elf:PLACE_STRUCTUREs`, one for each possible location of the event. Depending on how you read it, this usage can be seen as prohibited by GEDCOM or permitted as an extension; it is definitely permitted as an extension in ELF. Because of this ambiguity in GEDCOM, some tools are likely to have trouble reading data in this format. Additionally, it is not unambiguously talking about uncertainty either; a researcher might believe that a single event occurred in two locations or the like.

Some researchers include just one version in the data (or none at all) and add `elf:NOTE_STRUCTURES` that describe the alternatives. Assuming clear writing and a shared language, these can be fairly unambiguous but almost never understandable by software.

None of the above is clearly the right solution, nor does any one appear to be the most common in existing data. While a future version of this specification might include extensions to handle uncertainty, ELF 1.0.0 is intended to mirror GEDCOM closely and does not include any such extension.

4.2 Structures

A **structure** consists of the following parts:

Structure type

Each *structure* has a **structure type**, identified by a *term* called its **structure type identifier**.

Each *structure type* has a defined semantic meaning, supertype, and permitted *payload*.

The dataset itself is not a *structure*, but may be treated as one in many ways. Its *structure type identifier* is `[elf:Document]`.

Payload

Each *structure* has at most one of the following **payload** types:

- A **pointer** to another *structure*, which *must* be a *record* within the same *dataset*.
- A *string* or subtype thereof.

Superstructure

Each *structure* has at most exactly one **superstructure**, which is either another *structure* or the dataset itself.

A *structure* is said to be **within** its *superstructure* and also *within* everything its *superstructure* is *within*, recursively. A *structure* *must not* be *within* itself.

Substructures

Each *structure* may contain any number of **substructures**; by definition, X is a *substructure* of Y if and only if Y is the *superstructure* of X.

If a *structure* contains more than one *substructure* with the same *structure type*, those *substructures* are stored in a specific order. However, the order of *substructures* with different *structure types* is not preserved by their *superstructure*. Unless otherwise specified in the definition of a particular *structure*, *substructures* with the same *structure type* shall be interpreted as being in preference order, with the first such *substructure* being most preferred.

Note — The exact meaning of “preferred” is not defined either here nor in any known GEDCOM standard. For example, when seeing multiple names, one is preferred by virtue of being first but implementations SHOULD NOT infer that that one was preferred by the individual in question nor by any particular contributor to the dataset.

When a specific order of *substructures* is suggested or required by the data model (for example, `elf:CHILD_POINTERS` should be in birth order) or when distinct semantics are present

in each *substructure* (for example, `elf:CHILD_TO_FAMILY_LINKs` to both birth and adoptive families) user interfaces are RECOMMENDED to either present all such information, not just the first listed; or to clearly indicate that additional information is being elided.

Example — Given the following data

```
0 @I1@ INDI
1 NAME Henry /Herman/
1 NAME Henry /Harmon/
```

a view using only a single name variant should use the first (Herman, not Harmon) because it comes first and is thus interpreted as being preferred.

Within this standard document, a postfix shorthand notation is used to indicate the expected **cardinality** of *substructures* of a *structure*:

- “*” indicates zero or more.
- “?” indicates either zero or one. If more than one *substructure* of this type is present, all but the first are considered extensions.
- “!” indicates exactly one. If more than one *substructure* of this type is present, all but the first are considered extensions. If zero are present, the superstructure is considered an extension.

4.3 Supertypes and subtypes

Editorial note — Basic concepts defines both the subtype of a datatype and the subclass of a class. Neither applies here: datatypes are simple types without substructures and classes define contexts for terms, not types.

Each *structure type* may have any number of **direct supertypes**, which are also *structure types*. The set of **supertypes** of a *structure type* contains all of its *direct supertypes* and the *supertypes* of all of its *supertypes*, recursively.

By definition, a *structure types* semantics includes the union of the semantics of all of its *supertypes*. This may include permitted payloads and the meaning of substructures. A *structure type* MUST NOT inherit from a set of *supertypes* that contain contradictory semantics.

Editorial note — The above text is overly vague and needs tightening up.

Editorial note — Working notes on inheritance semantics:

A subtype could inherit

- its supertype’s substructures (1)
 - and their tags (1a)
 - with optional substructures made required (1b)

- with new substructures added (1c)
- its supertype's payload (2)
 - with additional constraints (2a)
 - with addition of payload to payload-lacking structures (2b)
- its supertype's superstructures (3)
 - and the supertype's tag within each superstructure (3a)
 - with a provided different tag within each superstructure (3b)

However, not all of these can be provided and maintain consistency. My current belief is we cannot have (3a) and I'm not sure about (3b).

I'm confident (3a) cannot work because tags are the only provided means of determining the type of a substructure.

I think (3b) can work, but it raises a question when a structure has a required substructure and that substructure type has a subtype. GEDCOM expects every 1 EVEN to contain a 2 TYPE; if there is instead a 2 XYZ where XYZ is a subtype of TYPE, GEDCOM parsers may reject the data as ill-formed. My gut is to only permit (3b) if the supertype is abstract. Without either (3a) or (3b), (3) is meaningless.

A few constraints to make inheritance work:

1. If an application finds a subtype it does not understand, but does understand its supertype, it **MUST NOT** create or edit an instance of the subtype using its knowledge of the supertype. Doing so would violate (1b), (1c), (2a), and (2b).
2. No structure type may inherit from two or more supertypes that share a common substructure unless all of the following are met:
 - Each supertype uses the same tag for the substructure. This is required because tags are lost on deserialisation so the association of particular substructures with particular supertypes via tag are not preserved.
 - The semantic meaning of the substructure type is identical in each supertype. This is required because all substructure values will be shared by both supertypes, so any difference will create conflict.

It may be simpler to forbid common-substructure supertypes altogether...

3. As a corollary, if a new structure type defines itself as a substructure of two different superstructures, it **MUST** define itself as having the same tag and same semantics in both (to avoid creating conflicts with other multiple inheritance situations) *unless* no substructure could inherit from both because of one of the other constraints listed here.
4. No structure type may inherit from two or more supertypes that have the same tag for distinct substructure types.

I am not fully convinced that the above limitations are sufficient, but do not have counter-examples for them (yet).

Some *structure types* are **abstract**, meaning they **MUST NOT** be identified as the *structure type* of any *structure*. Their purpose is to provide inherited semantics via being used as *supertypes*.

5 Abstract types

The following abstract types are presented in alphabetical order.

5.1 elf:Agent

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

An elf:Agent structure represents an entity that may be contacted, such as a person, corporation, or archive.

Supertype

elf:Structure

Substructures

elf:ADDRESS ?
elf:ADDRESS_WEB_PAGE *
elf:ADDRESS_FAX *
elf:ADDRESS_EMAIL *
elf:PHONE_NUMBER *

Subtypes

elf:NAME_OF_BUSINESS
elf:REPOSITORY_RECORD
elf:SUBMITTER_RECORD

5.2 elf:Event

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

This supertype was introduced in GEDCOM and encompasses all details about individuals and families to which a time and/or place may be reasonably attached, whether they be events or attributes. However, it is not used for some of the core structural connections between individuals and families used to structure traditional family trees.

Supertype

elf:Structure

Superstructures

None

Substructures

elf:EVENT_OR_FACT_CLASSIFICATION ?
elf:DATE_VALUE ?

```

elf:PLACE_STRUCTURE ?
elf:ADDRESS ?
elf:RESPONSIBLE_AGENCY ?
elf:RELIGIOUS_AFFILIATION ?
elf:CAUSE_OF_EVENT ?
elf:RESTRICTION_NOTICE ?
elf:NOTE_STRUCTURE *
elf:SOURCE_CITATION *
elf:MULTIMEDIA_LINK *

```

Subtypes

```

elf:FamilyEvent
elf:IndividualAttribute
elf:IndividualEvent

```

Note — GEDCOM suggested that `elf:Event` was a subtype of `elf:Agent` and thus could have `elf:ADDRESS_WEB_PAGE`, etc., inside; this appears to be a mistake as almost no historical event has any of that information.

5.3 elf:FamilyEvent

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

This supertype was introduced in GEDCOM and encompasses all details about families to which a time and/or place may be reasonably attached, whether they be events or attributes. However, it is not used for some of the core structural connections between individuals and families used to structure traditional family trees.

Supertype

```
elf:Event
```

Superstructures

```
elf:FAM_RECORD
```

Substructures

```

elf:Parent1Age ?
elf:Parent2Age ?

```

Subtypes

```

elf:ANNULMENT
elf:CENSUS#Family
elf:DIVORCE
elf:DIVORCE_FILED
elf:ENGAGEMENT
elf:MARRIAGE_BANN
elf:MARRIAGE_CONTRACT

```

```

elf:MARRIAGE
elf:MARRIAGE_LICENSE
elf:MARRIAGE_SETTLEMENT
elf:RESIDENCE
elf:EVENT#Family

```

Payload

A *string*, which may be limited by subtypes.

The special value Y indicates an assertion that the event in question did occur, even if it has no subordinate date or place.

5.4 elf:IndividualAttribute

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

This supertype was introduced in GEDCOM and encompasses all non-event details about individuals. However, it is not used for some of the core structural connections between individuals and families used to structure traditional family trees.

Supertype

```
elf:Event
```

Superstructures

```
elf:INDIVIDUAL_RECORD
```

Substructures

inherited

Subtypes

```

elf:CASTE_NAME
elf:PHYSICAL_DESCRIPTION
elf:SCHOLASTIC_ACHIEVEMENT
elf:NATIONAL_ID_NUMBER
elf:NATIONAL_OR_TRIBAL_ORIGIN
elf:COUNT_OF_CHILDREN#Individual
elf:COUNT_OF_MARRIAGES
elf:OCCUPATION
elf:POSSESSIONS
elf:RELIGIOUS_AFFILIATION#Individual
elf:RESIDES_AT
elf:SOCIAL_SECURITY_NUMBER
elf:NOBILITY_TYPE_TITLE
elf:ATTRIBUTE_DESCRIPTOR

```

5.5 `elf:IndividualEvent`

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

This supertype was introduced in GEDCOM and encompasses all events that an individual engaged with. However, it is not used for some of the core structural connections between individuals and families used to structure traditional family trees.

Supertype

`elf:Event`

Superstructures

`elf:INDIVIDUAL_RECORD`

Substructures

`elf:AGE_AT_EVENT`

Subtypes

`elf:BIRTH`
`elf:CHRISTENING`
`elf:DEATH`
`elf:BURIAL`
`elf:CREMATION`
`elf:ADOPTION`
`elf:BAPTISM`
`elf:BAR_MITZVAH`
`elf:BAS_MITZVAH`
`elf:BLESSING`
`elf:ADULT_CHRISTENING`
`elf:CONFIRMATION`
`elf:FIRST_COMMUNION`
`elf:ORDINATION`
`elf:NATURALIZATION`
`elf:EMIGRATION`
`elf:IMMIGRATION`
`elf:CENSUS#Individual`
`elf:PROBATE`
`elf:WILL`
`elf:GRADUATION`
`elf:RETIREMENT`
`elf:EVENT#Individual`

Payload

A *string*, which may be limited by subtypes.

The special value Y indicates an assertion that the event in question did occur, even if it has no subordinate date or place.

5.6 elf:ParentPointer

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

Editorial note — This supertype is added primarily as a place-holder for potential future extensions to support more general models of families.

Supertype

elf:Structure

Subtypes

elf:PARENT1_POINTER

elf:PARENT2_POINTER

Payload

A pointer to an elf:INDIVIDUAL_RECORD

5.7 elf:PersonalName

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

This supertype is a convenience to represent the set of structured name substructures shared by various personal name structures.

Supertype

elf:Structure

Substructures

elf:NAME_PIECE_PREFIX ?

elf:NAME_PIECE_GIVEN ?

elf:NAME_PIECE_NICKNAME ?

elf:NAME_PIECE_SURNAME_PREFIX ?

elf:NAME_PIECE_SURNAME ?

elf:NAME_PIECE_SUFFIX ?

elf:NOTE_STRUCTURE *

elf:SOURCE_CITATION *

Subtypes

elf:PERSONAL_NAME_STRUCTURE

elf:NAME_PHONETIC_VARIATION

elf:NAME_ROMANIZED_VARIATION

5.8 elf:Record

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

A **record** is a core element of the dataset. *Records* MUST NOT be substructures of any *structure*. *Pointers* may only point to *records*.

Supertype

elf:Structure

Superstructures

elf:Document

Substructures

elf:AUTOMATED_RECORD_ID ?

elf:CHANGE_DATE ?

elf:NOTE_STRUCTURE *

elf:USER_REFERENCE_NUMBER *

Subtypes

elf:FAM_RECORD

elf:INDIVIDUAL_RECORD

elf:MULTIMEDIA_RECORD

elf:NOTE_RECORD

elf:REPOSITORY_RECORD

elf:SOURCE_RECORD

elf:SUBMITTER_RECORD

5.9 elf:Structure

This is an abstract datatype and should not be used as the *structure type identifier* of any concrete structure.

This represents the top of the type hierarchy and has no semantics of its own.

6 Concrete data types

The following concrete types are presented in alphabetical order. Metadata types are presented in the following section.

6.1 elf:ADDRESS_CITY

Supertype

elf:Structure

Superstructures

elf:ADDRESS

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 60 characters.

The city, town, or similar name in an address.

Default tag

CITY

6.2 elf:ADDRESS_COUNTRY

Supertype

elf:Structure

Superstructures

elf:ADDRESS

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 60 characters.

The nation, country, or similar name in an address.

Default tag

CTRY

6.3 elf:ADDRESS_EMAIL

Supertype

elf:Structure

Superstructures

elf:Agent

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

This contains an email address. It is *recommended* that this match production `addr-spec` of [RFC 5322].

Default tag

EMAIL

EMAI

6.4 elf:ADDRESS_FAX

Supertype

elf:Structure

Superstructures

elf:Agent

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 60 characters.

This contains a telephone number that will connect to a fax machine. It is *recommended* that this be an international telephone number.

Editorial note — Add appropriate reference to ITU-T T.4 (I think; I'm not up on these standards)

Default tag

FAX

6.5 `elf:ADDRESS_LINE1`

Supertype

`elf:Structure`

Superstructures

`elf:ADDRESS`

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 60 characters.

The first line of the address, preceding the city. . Default tag

ADR1

6.6 `elf:ADDRESS_LINE2`

Supertype

`elf:Structure`

Superstructures

`elf:ADDRESS`

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 60 characters.

The second line of the address, preceding the city.

Default tag

ADR2

6.7 `elf:ADDRESS_LINE3`

Supertype

`elf:Structure`

Superstructures

`elf:ADDRESS`

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 60 characters.

The third line of the address, preceding the city.

Default tag

ADR3

6.8 `elf:ADDRESS_POSTAL_CODE`

Supertype

`elf:Structure`

Superstructures

`elf:ADDRESS`

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 10 characters.

The postal code of this address, as defined and used by the postal system in the area.

Default tag

POST

6.9 `elf:ADDRESS_STATE`

Supertype

`elf:Structure`

Superstructures

`elf:ADDRESS`

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 60 characters.

The state, province, or similar name in the area.

Default tag

STAE

6.10 `elf:ADDRESS_WEB_PAGE`

Supertype

`elf:Structure`

Superstructures

`elf:Agent`

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

This *should* contain a single IRL, as defined in [RFC 1736].

Editorial note — Look up RFC 1736 and make sure the above is correct.

Default tag

WWW

6.11 elf:ADDRESS**Supertype**

elf:Structure

Superstructures

elf:Event

elf:Agent

Substructures

elf:ADDRESS_LINE1

elf:ADDRESS_LINE2

elf:ADDRESS_LINE3

elf:ADDRESS_CITY

elf:ADDRESS_STATE

elf:ADDRESS_POSTAL_CODE

elf:ADDRESS_COUNTRY

Payload

A *block string* of arbitrary length.

The fully-formatted address, as it would appear for shipment labels. This *should not* be omitted even if all of its information is contained in substructures.

Default tag

ADDR

6.12 `elf:ADOPTED_BY_WHICH_PARENT`**Supertype**`elf:Structure`**Superstructures**`elf:ADOPTIVE_FAMILY`**Substructures**

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 4 characters.

Known values include {HUSB, WIFE, BOTH}. HUSB means the adoption was to the individual indicated by the [`elf:PARENT1_POINTER`] of the [`elf:FAM_RECORD`] pointed to by the payload of the containing superstructure; WIFE means the adoption was to the individual indicated by the [`elf:PARENT2_POINTER`] pointed to by the payload of the containing superstructure; and BOTH means both of those individuals were part of the adoption.

Default tag

ADOP

6.13 `elf:ADOPTION`

The creation of a parent-child relationship not associated with birth.

Supertype`elf:IndividualEvent`**Substructures**`elf:ADOPTIVE_FAMILY?`**Default tag**

ADOP

6.14 `elf:ADOPTIVE_FAMILY`**Supertype**`elf:Structure`**Superstructures**`elf:ADOPTION`**Substructures**`elf:ADOPTED_BY_WHICH_PARENT`**Payload**

A pointer to a `elf:FAM_RECORD`.

The pointed-to record describes the family unit into which the individual was adopted.

Default tag

FAMC

6.15 `elf:ADULT_CHRISTENING`

Adult christening, a religious rite in some Christian denominations typically performed when converting to the religion.

Supertype

`elf:IndividualEvent`

Default tag

CHRA

6.16 `elf:AGE_AT_EVENT`

Supertype

`elf:Structure`

Superstructures

`elf:IndividualEvent`

`elf:Parent1Age`

`elf:Parent2Age`

Payload

A line string in the lexical space of the `elf:Age datatype` defined in §6 of [ELF Dates].

Default tag

AGE

6.17 `elf:ALIAS_POINTER`

Supertype

`elf:Structure`

Superstructures

`elf:INDIVIDUAL_RECORD`

Payload

A pointer to an `elf:INDIVIDUAL_RECORD`

Points to a different [`elf:INDIVIDUAL_RECORD`] that may describe the same historical individual as the superstructure.

Default tag

ALIA

6.18 `elf:ANCESTOR_INTEREST_POINTER`**Supertype**`elf:Structure`**Superstructures**`elf:INDIVIDUAL_RECORD`**Payload**A pointer to an `elf:SUBMITTER_RECORD`Indicates that the pointed-to [`elf:SUBMITTER_RECORD`] describes someone interested in the ancestors of the individual described by the superstructure.**Default tag**

ANCI

6.19 `elf:ANNULMENT`

Declaring a marriage to be invalid, as though it had never occurred.

Supertype`elf:FamilyEvent`**Default tag**

ANUL

6.20 `elf:ASSOCIATION_STRUCTURE`**Supertype**`elf:Structure`**Superstructures**`elf:INDIVIDUAL_RECORD`**Substructures**`elf:RELATION_IS_DESCRIPTOR !``elf:SOURCE_CITATION *``elf:NOTE_STRUCTURE *`**Payload**A pointer to a `elf:INDIVIDUAL_RECORD`

Editorial note — While GEDCOM unambiguously stated this was a pointer to an `INDIVIDUAL_RECORD`, it also contained an example (under the definition of `RELATION_IS_DESCRIPTOR`) where it was a pointer to a `SUBMITTER_RECORD` instead.

Default tag

ASSO

6.21 `elf:ATTRIBUTE_DESCRIPTOR`

A generic attribute, the type of which **MUST** be more fully described in a `elf:EVENT_OR_FACT_CLASSIFICATION` substructure.

Supertype

`elf:IndividualAttribute`

Substructures

`elf:EVENT_OR_FACT_CLASSIFICATION!`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

A textual description of the specific attribute; typically more specific than the generic attribute type classification of the [`elf:EVENT_OR_FACT_CLASSIFICATION`] substructure.

Default tag

FACT

6.22 `elf:AUTOMATED_RECORD_ID`

Supertype

`elf:Structure`

Superstructures

`elf:Record`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 12 characters.

A record identifier (numeric or textual) that is supposed to be unique within an originating system. Since there is no indication of which system, nor a mechanism for spanning systems, this has little value when communicating between systems.

Default tag

RIN

6.23 `elf:BAPTISM`

Baptism, a common Christian rite, typically involving water and indicating entry into a particular faith or denomination, performed at different ages in different denominations.

Supertype

`elf:IndividualEvent`

Default tag

BAPM

6.24 `elf:BAR_MITZVAH`

Bar Mitzvah, a Jewish rite (typically for 13-year-old boys).

Supertype

`elf:IndividualEvent`

Default tag

BARM

6.25 `elf:BAS_MITZVAH`

Bas Mitzvah, a Jewish rite (typically for 13-year-old girls).

Supertype

`elf:IndividualEvent`

Default tag

BASM

6.26 `elf:BINARY_OBJECT`

Binary object was in GEDCOM 5.5 but removed from GEDCOM 5.5.1. Implementations SHOULD be able to parse them, but SHOULD NOT generate new binary objects.

Editorial note — The definition of the base-64 encoding used the terminology “byte” when GEDCOM had elsewhere defined its stream as consisting of characters, not bytes. It is unclear to me if it is possible to follow the spec for an encoding that does not permit byte 0xFF as a single character.

Supertype

`elf:Structure`

Superstructures

`elf:MULTIMEDIA_RECORD`

Substructures

None

Payload

A *block string* containing two or more lines of base-64 encoded data, in the custom format described below.

The first line of a blob is always empty. Each subsequent line is between 4 and 72 characters long, encoded in a base-64 format that differs from other base-64 encodings in two ways.

First, it uses byte 0xFF as padding instead of the more common U+003D (EQUALS SIGN =) (how to represent the padding when byte 0xFF is not a legal character in the encoding is not defined by this specification).

Second, it maps six-bit values to code points as follows:

Byte range	Code point mapping
0x00–0x0B	byte + 0x2E
0x0C–0x25	byte + 0x35
0x25–0x3F	byte + 0x3B

See also the discussion under [elf:CONTINUED_BINARY_OBJECT] for how multiple elf: BINARY_OBJECT payloads are combined to represent large binary values.

Default tag

BLOB

6.27 elf:BIRTH

The exiting of the womb.

Supertype

elf: IndividualEvent

Substructures

elf: WITHIN_FAMILY ?

Default tag

BIRT

6.28 elf:BLESSING

A religious rite invoking divine favour on an individual.

Supertype

elf: IndividualEvent

Default tag

BLES

6.29 elf: BURIAL

The depositing of the body (in whole or in part) of the deceased.

Supertype

elf: IndividualEvent

Default tag

BRI

6.30 elf:CASTE_NAME

Supertype

elf:IndividualAttribute

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

The social, religious, or racial caste to which an individual belongs.

Default tag

CAST

6.31 elf:CAUSE_OF_EVENT

Supertype

elf:Structure

Superstructures

elf:Event

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

Introduced to record the cause of death as a substructure to a [elf:DEATH] structure, but permitted under any event in case a cause of the event is known.

Default tag

CAUS

6.32 elf:CENSUS#Family

An inventory of persons or households in a population.

Supertype

elf:FamilyEvent

Default tag

CENS

6.33 elf:CENSUS#Individual

An inventory of persons or households in a population.

Supertype

elf:IndividualEvent

Default tag

CENS

6.34 `elf:CERTAINTY_ASSESSMENT`**Supertype**`elf:Structure`**Superstructures**`elf:SOURCE_CITATION`**Payload**

A *line string* containing a numeric value. Known values include {0, 1, 2, and 3}.

A ranking hint for display: if two `elf:SOURCE_CITATION`s contain `elf:CERTAINTY_ASSESSMENT`s with different payloads, the numerically larger `elf:CERTAINTY_ASSESSMENT` may be displayed as being inside a more reliable `elf:SOURCE_CITATION` than is the numerically smaller payload.

GEDCOM defined the four specific values as having the following meanings

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

Note — It is unclear that GEDCOM's four categories have the relative reliability their ordering suggests, nor that `elf:CERTAINTY_ASSESSMENT`s in extant files contain meaningful information. It is not difficult to find example GEDCOM where all `elf:SOURCE_CITATION`s have a `elf:CERTAINTY_ASSESSMENT` with payload 3 even when some clearly cite sources providing secondary evidence of the facts containing the citation.

Default tag

QUAY

6.35 `elf:CHANGE_DATE_DATE`**Supertype**`elf:Structure`**Superstructures**`elf:CHANGE_DATE`**Substructures**`elf:TIME_VALUE ?`**Payload**

A *line string* in the *lexical space* of the `elf:DateExact datatype` defined in §4.1.1 of [ELF Dates]. Indicates the last change to the containing structure.

Default tag

DATE

6.36 `elf:CHANGE_DATE`**Supertype**`elf:Structure`**Superstructures**`elf:Record`**Substructures**`elf:CHANGE_DATE_DATE !``elf:NOTE_STRUCTURE *`**Payload**

None

Default tag

CHAN

Editorial note — GEDCOM uses the token `CHANGE_DATE` in two ways. Page 31 defines what we call `elf:CHANGE_DATE`, a structure containing a date and an arbitrary number of notes; page 44 defines what we call `elf:CHANGE_DATE_DATE`, a payload-only format structure.

6.37 `elf:CHILD_LINKAGE_STATUS`**Supertype**`elf:Structure`**Superstructures**`elf:CHILD_TO_FAMILY_LINK`**Substructures**

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 15 characters.

Contains a description of the confidence that this relationship exists. Known values include {challenged, disproven, proven}.

No matter the contents, there *should* be a [`elf:NOTE_STRUCTURE`] within this structure's superstructure that describes the proof or challenge.

Default tag

STAT

6.38 elf:CHILD_POINTER

A pointer one of the children in a family.

The preferred order of the children pointers within a family structure is chronological by birth.

Supertype

elf:Structure

Superstructures

elf:FAM_RECORD

Payload

A pointer to an elf:INDIVIDUAL_RECORD

It *must* be the case that the pointed-to [elf:INDIVIDUAL_RECORD] contains a [elf:CHILD_TO_FAMILY_LINK] pointing to the superstructure of this structure.

Default tag

CHIL

6.39 elf:CHILD_TO_FAMILY_LINK

Supertype

elf:Structure

Superstructures

elf:INDIVIDUAL_RECORD

Substructures

elf:PEDIGREE_LINKAGE_TYPE ?

elf:CHILD_LINKAGE_STATUS ?

elf:NOTE_STRUCTURE *

Payload

A *pointer* to a elf:FAM_RECORD

It *must* be the case that the pointed-to [elf:FAM_RECORD] contains a [elf:CHILD_POINTER] pointing to the superstructure of this structure.

Default tag

FAMC

6.40 elf:CHRISTENING

A religious rite occurring at or near birth.

Supertype

elf:IndividualEvent

Substructures

elf:WITHIN_FAMILY?

Default tag

CHR

6.41 elf:CONFIRMATION

Confirmation, a religious rite in some Christian denominations associated with gaining full fellowship in the religion and/or receiving the Holy Ghost.

Supertype

elf:IndividualEvent

Default tag

CONF

6.42 elf:CONTINUED_BINARY_OBJECT

Binary object was in GEDCOM 5.5 but removed from GEDCOM 5.5.1. Implementations SHOULD be able to parse them, but SHOULD NOT generate new binary objects.

Supertype

elf:Structure

Superstructures

elf:MULTIMEDIA_RECORD

Substructures

None

Payload

A pointer to a elf:MULTIMEDIA_RECORD.

Used to split elf:BINAR_Y_OBJECTs across multiple records. Prior to decoding, the payloads of all elf:BINAR_Y_OBJECT in the superstructure should be concatenated in the order in which they appear, and then concatenated with the elf:BINAR_Y_OBJECTs in the pointed-to record and those pointed to by its elf:CONTINUED_BINARY_OBJECT, recursively.

Default tag

OBJE

6.43 `elf:COUNT_OF_CHILDREN#Family`**Supertype**`elf:Structure`**Superstructures**`elf:FAM_RECORD`**Payload**

A *line string* taking the form of a decimal number. It is RECOMMENDED that implementations support payloads of at least 3 characters.

The total number of children this family unit had, either at some (unspecified) point in time or in total its entire existence. This does not need to match the number of children identified through [`elf:CHILD_POINTER`] substructures of the containing superstructure.

Default tag

NCHI

Editorial note — It seems odd to me that `elf:COUNT_OF_CHILDREN#Family` is not a `elf:FamilyEvent` (or `elf:FamilyAttribute`, though no such supertype currently exists) as surely the number of children of a family would need sourcing and an as-of date? Should we leave it as a stand-alone structure, or boost it to event status?

6.44 `elf:COUNT_OF_CHILDREN#Individual`**Supertype**`elf:IndividualAttribute`**Payload**

A *line string* taking the form of a decimal number. It is RECOMMENDED that implementations support payloads of at least 3 characters.

The total number of children this person ever had. This does not need to match the number of children individually identified in the dataset.

Default tag

NCHI

6.45 `elf:COUNT_OF_MARRIAGES`**Supertype**`elf:IndividualAttribute`**Payload**

A *line string* taking the form of a decimal number. It is RECOMMENDED that implementations support payloads of at least 3 characters.

The total number of marriages this person ever had. This does not need to match the number of marriages individually identified in the dataset.

Default tag

NMR

6.46 `elf:CREMATION`

The burning of the body (in whole or in part) of the deceased.

Supertype

`elf:IndividualEvent`

Default tag

CREM

6.47 `elf:DATE_PERIOD`

Supertype

`elf:Structure`

Superstructures

`elf:EVENTS_RECORDED`

Payload

A *line string* in the *lexical space* of the `elf:DatePeriod` datatype defined in §3.4 of [ELF Dates]. Indicates the period during which the source recorded events.

Default tag

DATE

6.48 `elf:DATE_VALUE`

Supertype

`elf:Structure`

Superstructures

`elf:Event`

Payload

A *line string* in the *lexical space* of the `elf:DateValue` datatype defined in §3.3 of [ELF Dates]. Indicates when the event or attribute described by the containing structure occurred or was witnessed.

Default tag

DATE

6.49 elf:DEATH

The end of life.

Supertype

elf:IndividualEvent

Default tag

DEAT

6.50 elf:DESCENDANT_INTEREST_POINTER

Supertype

elf:Structure

Superstructures

elf:INDIVIDUAL_RECORD

Payload

A pointer to an elf:SUBMITTER_RECORD

Indicates that the pointed-to [elf:SUBMITTER_RECORD] describes someone interested in the descendants of the individual described by the superstructure.

Default tag

DESI

6.51 elf:DESCRIPTIVE_TITLE

Supertype

elf:Structure

Superstructures

elf:MULTIMEDIA_RECORD – GEDCOM 5.5

elf:MULTIMEDIA_FILE_REFERENCE – GEDCOM 5.5.1

elf:MULTIMEDIA_LINK

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 248 characters.

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

Default tag

TITL

6.52 elf:DIVORCE_FILED

The legal action expressing intent to divorce.

Supertype

elf:FamilyEvent

Default tag

DIVF

6.53 elf:DIVORCE

The ending of a marriage between still-living individuals.

Supertype

elf:FamilyEvent

Default tag

DIV

6.54 elf:EMIGRATION

The departure from the nation or land in which one has nativity or citizenship.

Supertype

elf:IndividualEvent

Default tag

EMIG

6.55 elf:ENGAGEMENT

The agreement of a couple to enter into a marriage in the future.

Supertype

elf:FamilyEvent

Default tag

ENGA

6.56 `elf:ENTRY_RECORDING_DATE`**Supertype**`elf:Structure`**Superstructures**`elf:SOURCE_CITATION_DATA`**Payload**

A *line string* in the *lexical space* of the `elf:DateValue` datatype defined in §3.3 of [ELF Dates].

Indicates when the portion of the source being cited was entered into the source.

Default tag

DATE

6.57 `elf:EVENT#Family`

A generic event, the type of which SHOULD be more fully described in a `elf:EVENT_OR_FACT_CLASSIFICATION` substructure.

Supertype`elf:FamilyEvent`**Payload**

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

A textual description of the specific event; typically more specific than the generic event type classification of the [`elf:EVENT_OR_FACT_CLASSIFICATION`] substructure.

Unlike other [`elf:FamilyEvent`]s, Y is not a special value. `elf:EVENT#Family` events are always assertions that the event occurred.

Default tag

EVEN

6.58 `elf:EVENT#Individual`

A generic event, the type of which SHOULD be more fully described in a `elf:EVENT_OR_FACT_CLASSIFICATION` substructure.

Supertype`elf:IndividualEvent`**Payload**

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

A textual description of the specific event; typically more specific than the generic event type classification of the [`elf:EVENT_OR_FACT_CLASSIFICATION`] substructure.

Unlike other [elf:FamilyEvent]s, Y is not a special value. elf:EVENT#Family events are always assertions that the event occurred.

Editorial note — GEDCOM does not list this payload for individual EVEN, only family EVEN, but other text in GEDCOM suggests that this was an oversight, not an intentional omission.

Default tag

EVEN

6.59 elf:EVENTS_RECORDED

Supertype

elf:Structure

Superstructures

elf:SOURCE_RECORD_DATA

Substructures

elf:DATE_PERIOD

elf:SOURCE_JURISDICTION_PLACE

Payload

A *comma-separated list*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

A list of valid payload values of [elf:EVENT_TYPE_CITED_FROM]. Indicates that the source includes documentation of these events or attributes.

Default tag

EVEN

6.60 elf:EVENT_OR_FACT_CLASSIFICATION

Supertype

elf:Structure

Superstructures

elf:Event

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

A classification for the superstructure's category, more precise than its type alone provides but generic enough to be anticipated to be re-used.

Default tag

TYPE

6.61 `elf:EVENT_TYPE_CITED_FROM`**Supertype**`elf:Structure`**Superstructures**`elf:SOURCE_CITATION`**Substructures**`elf:ROLE_IN_EVENT`**Payload**

A *line string*. It is RECOMMENDED that implementations support payloads of at least 15 characters.

Known values include {CAST, EDUC, NATI, OCCU, PROP, RELI, RESI, TITL, FACT, ANUL, CENS, DIV, DIVF, ENGA, MARR, MARB, MARC, MARL, MARS, ADOP, BIRT, BAPM, BARM, BASM, BLES, BURI, CENS, CHR, CHRA, CONF, CREM, DEAT, EMIG, FCOM, GRAD, IMMI, NATU, ORDN, RETI, PROB, WILL, EVEN}. Indicates that the cited source was created to document the event or attribute described by the subtype of `elf:Event` whose default tag is the provided value.

Example —

A marriage certificate may document the spouses' birth dates; however, its `elf:EVENT_TYPE_CITED_FROM`'s payload should be MARR, not BIRT.

Default tag

EVEN

6.62 `elf:FAM_RECORD`

Used to record couple and parent/child relationships.

Because of the social context in which GEDCOM was first created and because `elf:FAM_RECORDS` are used in some software applications to present binary ancestry trees and n -ary descendancy trees, each `elf:FAM_RECORD` is limited to having at most one “first-position” parent; at most one “second-position” parent; and any number of ordered children. GEDCOM explicitly stated that the first-position parent was male and the second-position parent was female; that is not always true of how GEDCOM has been used in practice and MUST NOT be assumed by any conformant ELF implementation.

Supertype`elf:Record`**Superstructures**`elf:Document`**Substructures**`elf:RESTRICTION_NOTICE ?``elf:FamilyEvent *`

elf:PARENT1_POINTER ?
elf:PARENT2_POINTER ?
elf:CHILD_POINTER *
elf:COUNT_OF_CHILDREN#Family ?
elf:MULTIMEDIA_LINK *
elf:SOURCE_CITATION *
elf:SUBMITTER_POINTER *

Payload

None

Default tag

FAM

6.63 elf:FIRST_COMMUNION

First communion, a religious rite in many Christian denominations associated with first partaking of the communion of the Lord's Supper.

Supertype

elf:IndividualEvent

Default tag

FCOM

6.64 elf:GRADUATION

The conclusion of formal education.

Supertype

elf:IndividualEvent

Default tag

GRAD

6.65 elf:IMMIGRATION

The entering of a nation or land in which one does not have nativity or citizenship.

Supertype

elf:IndividualEvent

Default tag

IMMI

6.66 `elf:INDIVIDUAL_RECORD`

A representation of a historical individual, together with the facts and events believed to apply to that individual and the sources of those data.

Supertype

`elf:Record`

Superstructures

`elf:Document`

Substructures

`elf:RESTRICTION_NOTICE ?`
`elf:PERSONAL_NAME_STRUCTURE *`
`elf:SEX_VALUE ?`
`elf:IndividualEvent *`
`elf:IndividualAttribute *`
`elf:CHILD_TO_FAMILY_LINK *`
`elf:SPOUSE_TO_FAMILY_LINK *`
`elf:ASSOCIATION_STRUCTURE *`
`elf:ALIAS_POINTER *`
`elf:ANCESTOR_INTEREST_POINTER *`
`elf:DESCENDANT_INTEREST_POINTER *`
`elf:MULTIMEDIA_LINK *`
`elf:SOURCE_CITATION *`
`elf:SUBMITTER_POINTER *`

Note — GEDCOM permitted a `PERMANENT_RECORD_FILE_NUMBER` with tag `RFN`, the value of which was under-defined and not included in this one.

Payload

None

Default tag

`INDI`

6.67 `elf:LANGUAGE_PREFERENCE`

Supertype

`elf:Structure`

Superstructures

`elf:SUBMITTER_RECORD`

Substructures

None

Payload

A line string matching the Language Tag microformat.

Indicates a language in which the person described by the superstructure prefers to communicate.

Default tag

LANG

6.68 `elf:MAP_COORDINATES`

Contains the location of a place in a global coordinate system.

Supertype

`elf:Structure`

Superstructures

`elf:PLACE_STRUCTURE`

Substructures

`elf:PLACE_LATITUDE !`

`elf:PLACE_LONGITUDE !`

Payload

None

Default tag

MAP

6.69 `elf:MARRIAGE_BANN`

A public notice of an intent to marry.

Supertype

`elf:FamilyEvent`

Default tag

MARB

6.70 `elf:MARRIAGE_CONTRACT`

A formal contractual agreement to marry.

Supertype

`elf:FamilyEvent`

Default tag

MARC

6.71 `elf:MARRIAGE_LICENSE`

Obtaining a legal license to marry.

Supertype

`elf:FamilyEvent`

Default tag

MARL

6.72 `elf:MARRIAGE_SETTLEMENT`

A legal arrangement to modify property rights upon marriage.

Supertype

`elf:FamilyEvent`

Default tag

MARS

6.73 `elf:MARRIAGE`

The creation of a family unit (via a legal, religious, customary, common-law, or other form of union).

Supertype

`elf:FamilyEvent`

Default tag

MARR

6.74 `elf:MULTIMEDIA_FILE_REFERENCE`**Supertype**`elf:Structure`**Superstructures**`elf:MULTIMEDIA_RECORD``elf:MULTIMEDIA_LINK`**Substructures**`elf:MULTIMEDIA_FORMAT!``elf:DESCRIPTIVE_TITLE?`**Payload**

A *line string*. It is RECOMMENDED that implementations support payloads of at least 30 characters.

Contains locator information (such as a file path or IRL) for a file containing auxiliary data.

Editorial note — GEDCOM is vague on payload format. Presumably there exist a variety of formats in the wild. Should we perform a survey and see if we can provide guidance on payload format?

Default tag

FILE

6.75 `elf:MULTIMEDIA_FORMAT`**Supertype**`elf:Structure`**Superstructures**`elf:MULTIMEDIA_RECORD – GEDCOM 5.5``elf:MULTIMEDIA_FILE_REFERENCE – GEDCOM 5.5.1``elf:MULTIMEDIA_LINK`**Substructures**`elf:SOURCE_MEDIA_TYPE?`**Payload**

A *line string*. Known values include {bmp, gif, jpg, ole, pcx, tif, wav}.

Indicates the format of the multimedia data associated with the superstructure of the `elf:MULTIMEDIA_FORMAT` structure.

Editorial note — Should we expand this to allow MIME-type as well as the seven known Windows-style file endings?

Default tag

FORM

6.76 `elf:MULTIMEDIA_LINK`**Supertype**`elf:Structure`**Superstructures**

`elf:SOURCE_CITATION`
`elf:FAM_RECORD`
`elf:INDIVIDUAL_RECORD`
`elf:SOURCE_RECORD`
`elf:SUBMITTER_RECORD`
`elf:Event`

Substructures

`elf:MULTIMEDIA_FILE_REFERENCE *`
`elf:DESCRIPTIVE_TITLE ?`
`elf:MULTIMEDIA_FORMAT ? – GEDCOM 5.5`

Payload

Either a *pointer* to a `elf:MULTIMEDIA_RECORD` or none.
 If the payload is a pointer, it *should not* contain substructures.

Default tag`OBJE`

Editorial note — TO DO: review GEDCOM 5.5 to make sure this is right

6.77 `elf:MULTIMEDIA_RECORD`

The form of this record was changed between GEDCOM 5.5 and GEDCOM 5.5.1. Implementations should accept both formats but export only 5.5.1 format.

Supertype`elf:Record`**Superstructures**`elf:Document`**Substructures**

`elf:MULTIMEDIA_FILE_REFERENCE * – GEDCOM 5.5.1`
`elf:MULTIMEDIA_FORMAT ! – GEDCOM 5.5`
`elf:DESCRIPTIVE_TITLE ? – GEDCOM 5.5`
`elf:CONTINUED_BINARY_OBJECT ? – GEDCOM 5.5`
`elf:BINARY_OBJECT ! – GEDCOM 5.5`

Payload

None

Default tag

OBJE

6.78 `elf:NAME_OF_REPOSITORY`

Supertype

`elf:Structure`

Superstructures

`elf:REPOSITORY_RECORD`

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 90 characters.

The name of the repository described by the superstructure.

Default tag

NAME

6.79 `elf:NAME_PHONETIC_VARIATION`

Supertype

`elf:PersonalName`

Superstructures

`elf:PERSONAL_NAME_STRUCTURE`

Substructures

`elf:PHONETIC_TYPE`

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 120 characters.

Contains a phonetic presentation of the same name as its superstructure.

Default tag

FONE

6.80 `elf:NAME_PIECE_GIVEN`

Supertype

`elf:Structure`

Superstructures

`elf:PersonalName`

Substructures

None

Payload

A comma-separated list. It is RECOMMENDED that implementations support payloads of at least 120 characters.

A list of given or earned names.

Default tag

GIVN

6.81 `elf:NAME_PIECE_NICKNAME`

Supertype

`elf:Structure`

Superstructures

`elf:PersonalName`

Substructures

None

Payload

A comma-separated list. It is RECOMMENDED that implementations support payloads of at least 30 characters.

A list of familiar or informal names.

Default tag

NICK

6.82 `elf:NAME_PIECE_PREFIX`

Supertype

`elf:Structure`

Superstructures

`elf:PersonalName`

Substructures

None

Payload

A comma-separated list. It is RECOMMENDED that implementations support payloads of at least 30 characters.

A list of non-name elements traditionally placed before the proper name, such as titles.

Default tag

NPFX

6.83 `elf:NAME_PIECE_SUFFIX`

Supertype

`elf:Structure`

Superstructures

`elf:PersonalName`

Substructures

None

Payload

A comma-separated list. It is RECOMMENDED that implementations support payloads of at least 30 characters.

A list of non-name elements traditionally placed after the proper name, such as generational marks and ordinals.

Default tag

NSFX

6.84 `elf:NAME_PIECE_SURNAME_PREFIX`

Supertype

`elf:Structure`

Superstructures

`elf:PersonalName`

Substructures

None

Payload

A comma-separated list. It is RECOMMENDED that implementations support payloads of at least 120 characters.

A list of non-name elements traditionally attached to and placed before a surname of family name, such as prepositions.

Default tag

SPFX

6.85 `elf:NAME_PIECE_SURNAME`

Supertype

`elf:Structure`

Superstructures

`elf:PersonalName`

Substructures

None

Payload

A *comma-separated list*. It is RECOMMENDED that implementations support payloads of at least 120 characters.
 A list of surnames and family names.

Default tag

SURN

6.86 `elf:NAME_ROMANIZED_VARIATION`

Supertype

`elf:PersonalName`

Superstructures

`elf:PERSONAL_NAME_STRUCTURE`

Substructures

`elf:ROMANIZED_TYPE`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.
 Contains the same name as its superstructure, but presented using ASCII letters.

Default tag

ROMN

6.87 `elf:NAME_TYPE`

Supertype

`elf:Structure`

Superstructures

`elf:PERSONAL_NAME_STRUCTURE`

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 30 characters.
 The kind of name the superstructure contains. Known values include {aka, birth, immigration, maiden, married}; additional values are encouraged as appropriate.

known value	meaning
aka	also known as: an unofficial pseudonym
birth	name given at or near birth
immigrant	name assumed when immigrating

known value	meaning
maiden	name used prior to marriage
married	name assumed at marriage

Note — GEDCOM's definition of the married payload was "name was persons previous married name," suggestion it was only to be used after the married name was no longer used; this nuanced definition does not appear to have been used in practice.

Default tag

TYPE

6.88 elf:NATIONAL_ID_NUMBER**Supertype**

elf:IndividualAttribute

Substructures

elf:EVENT_OR_FACT_CLASSIFICATION!

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 30 characters.

An identifier used by a nation to a particular individual. If an appropriate nation-specific alternative is present, it *should* be used.

Default tag

IDNO

6.89 elf:NATIONAL_OR_TRIBAL_ORIGIN**Supertype**

elf:IndividualAttribute

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

A group to which a person is associated, typically by birth.

Default tag

NATI

6.90 `elf:NATURALIZATION`

Editorial note — GEDCOM used American spelling; should we change it to British?

The gaining of citizenship in a new nation or land.

Supertype

`elf:IndividualEvent`

Default tag

NATU

6.91 `elf:NOBILITY_TYPE_TITLE`**Supertype**

`elf:IndividualAttribute`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

A title given a person associated with a local or national notion of nobility or royalty.

Default tag

TITL

6.92 `elf:NOTE_RECORD`**Supertype**

`elf:Record`

Superstructures

`elf:Document`

Payload

A *block string* of arbitrary length.

Default tag

NOTE

Note — GEDCOM did not `elf:NOTE_STRUCTURE` as a substructure of `elf:NOTE_RECORD`, but they do appear in the wild and have valid semantics (notes about the note itself) so `elf:NOTE_RECORD` inherits the `elf:NOTE_STRUCTURE` substructure from `elf:Record` in this specification.

6.93 elf:NOTE_STRUCTURE

Editorial note — We list here all *superstructures* where a `elf:NOTE_STRUCTURE` is listed in GEDCOM. We have discussed saying instead that a `elf:NOTE_STRUCTURE` can appear in any `elf:Structure` (including other `elf:NOTE_STRUCTURE` as we have use cases for notes about notes) but as that is technically a divergence from GEDCOM, we have refrained from adding it to this draft.

Supertype

`elf:Structure`

Superstructures

`elf:Record`
`elf:SOURCE_RECORD_DATA`
`elf:ASSOCIATION_STRUCTURE`
`elf:CHILD_TO_FAMILY_LINK`
`elf:Event`
`elf:PersonalName`
`elf:SOURCE_CITATION`
`elf:SOURCE_REPOSITORY_CITATION`
`elf:SPOUSE_TO_FAMILY_LINK`
`elf:CHANGE_DATE`
`elf:PLACE_STRUCTURE`

Substructures

None

Payload

Either a *pointer* to a `elf:NOTE_RECORD` or a *block string* of arbitrary length.

Default tag

NOTE

6.94 elf:OCCUPATION

Supertype

`elf:IndividualAttribute`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

What this person does as a livelihood.

Default tag

OCCU

6.95 `elf:ORDINATION`

The bestowal of religious authority or office.

Supertype

`elf:IndividualEvent`

Default tag

ORDN

6.96 `elf:PARENT1_POINTER`

A pointer to the spouse or parent traditionally presented on the left fork of a vertical family tree or on the upper fork of a horizontal family tree. In a heterosexual pair union, this is traditionally the husband or father.

Supertype

`elf:ParentPointer`

Superstructures

`elf:FAM_RECORD`

Payload

A pointer to an `elf:INDIVIDUAL_RECORD`

Default tag

HUSB

6.97 `elf:PARENT2_POINTER`

A pointer to the spouse or parent traditionally presented on the right fork of a vertical family tree or on the bottom fork of a horizontal family tree. In heterosexual pair unions, this is traditionally the wife or mother.

Supertype

`elf:ParentPointer`

Superstructures

`elf:FAM_RECORD`

Payload

A pointer to an `elf:INDIVIDUAL_RECORD`

Default tag

WIFE

6.98 `elf:PEDIGREE_LINKAGE_TYPE`**Supertype**`elf:Structure`**Superstructures**`elf:CHILD_TO_FAMILY_LINK`**Substructures**

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 7 characters.

Contains a description of how this child is related to the superstructure's pointed-to [`elf:FAM_RECORD`]. Known values include {adopted, birth, foster}.

Default tag

PEDI

6.99 `elf:PERSONAL_NAME_STRUCTURE`**Supertype**`elf:PersonalName`**Superstructures**`elf:INDIVIDUAL_RECORD`**Substructures**`elf:NAME_TYPE``elf:NAME_PHONETIC_VARIATION``elf:NAME_ROMANIZED_VARIATION`**Payload**

A *line string* matching the Personal Name microformat. It is RECOMMENDED that implementations support payloads of at least 120 characters.

In the event that this payload disagrees with the substructures of this structure, the payload *should* be taken as more correct.

Default tag

NAME

6.100 `elf:PHONETIC_TYPE`**Supertype**`elf:Structure`**Superstructures**`elf:NAME_PHONETIC_VARIATION``elf:PLACE_PHONETIC_VARIATION`

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 30 characters.

Identifies the phonetic scheme used in the superstructure. Known values include {hangul, kana}.

Editorial note — Should we add others, like ipa?

Default tag

TYPE

6.101 elf:PHONE_NUMBER**Supertype**

elf:Structure

Superstructures

elf:Agent

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 30 characters.

This contains a telephone number. It is *recommended* that this be an international telephone number.

Editorial note — Add appropriate reference to ITU-T E.123 and E.164 (I think; I'm not up on these standards)

Default tag

PHON

6.102 elf:PHYSICAL_DESCRIPTION**Supertype**

elf:IndividualAttribute

Payload

A *block string* of arbitrary length.

Appearance and/or other physical characteristics.

Default tag

DSCR

6.103 `elf:PLACE_HIERARCHY`

Editorial note — This feels like a strange way of serializing an ordered map, and thus perhaps better defined as a pseudo-structure?

Supertype

`elf:Structure`

Superstructures

`elf:PLACE_STRUCTURE`

`elf:DEFAULT_PLACE_FORMAT`

Substructures

None

Payload

A *comma-separated list*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

A list of names of what the specific components of [`elf:PLACE_STRUCTURE`] represent.

Default tag

FORM

6.104 `elf:PLACE_LATITUDE`

Degrees north or south of the equator

Supertype

`elf:Structure`

Superstructures

`elf:MAP_COORDINATES`

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 12 characters.

Decimal degrees from the equator. Either the letter N (for north) or S (for south), followed (without a space) by a decimal number between 0 and 90.

Note — Only decimal degrees supported; degree-minute-second representations MUST NOT appear in this payload.

Default tag

LATI

6.105 `elf:PLACE_LONGITUDE`**Supertype**`elf:Structure`**Superstructures**`elf:MAP_COORDINATES`**Substructures**

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 12 characters.

Decimal degrees from the prime meridian. Either the letter E (for east) or W (for west), followed (without a space) by a decimal number between 0 and 180.

Note — Only decimal degrees supported; degree-minute-second representations MUST NOT appear in this payload.

Default tag

LONG

6.106 `elf:PLACE_PHONETIC_VARIATION`**Supertype**`elf:Structure`**Superstructures**`elf:PLACE_STRUCTURE`**Substructures**`elf:PHONETIC_TYPE`**Payload**

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

Contains a phonetic presentation of the same place, in the same format, as its superstructure.

Default tag

FONE

6.107 `elf:PLACE_ROMANIZED_VARIATION`**Supertype**`elf:Structure`**Superstructures**`elf:PLACE_STRUCTURE`

Substructures

elf:ROMANIZED_TYPE

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

Contains an ASCII letter presentation of the same place, in the same format, as its superstructure.

Default tag

ROMN

6.108 elf:PLACE_STRUCTURE**Supertype**

elf:Structure

Superstructures

elf:Event

Substructures

elf:PLACE_HIERARCHY ?

elf:PLACE_PHONETIC_VARIATION *

elf:PLACE_ROMANIZED_VARIATION *

elf:MAP_COORDINATES ?

elf:NOTE_STRUCTURE *

Payload

A *comma-separated list*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

A list of names of regions, where each element of the list is subsumed within all subsequent elements.

If this structure has a [elf:PLACE_HIERARCHY] substructure or there is a default [elf:PLACE_HIERARCHY] defined for the dataset, then this payload *should* contain one name for each jurisdictional elements in that [elf:PLACE_HIERARCHY], using empty strings in place of any unknown or non-present elements.

Default tag

PLAC

Note — If an individual region name contains a comma, that comma cannot be represented in the place structure format. As there is no escaping mechanism provided, it must either be omitted or replaced with a substitute marking.

6.109 elf:POSSESSIONS

Supertype

elf:IndividualAttribute

Payload

A *line string* of arbitrary length.

A list of objects or land owned by the person.

Default tag

PROP

6.110 elf:PROBATE

The judicial actions associated with the disposition of the estate of the deceased.

Supertype

elf:IndividualEvent

Default tag

PROB

6.111 elf:Parent1Age

An intermediate structure to indicate the age of a spouse or parent at the time of an event.

Supertype

elf:Structure

Superstructures

elf:FamilyEvent

Substructures

elf:AGE_AT_EVENT

Payload

None

Default tag

HUSB

6.112 elf:Parent2Age

An intermediate structure to indicate the age of a spouse or parent at the time of an event.

Supertype

elf:Structure

Superstructures

elf:FamilyEvent

Substructures

elf:AGE_AT_EVENT

Payload

None

Default tag

WIFE

6.113 elf:RELATION_IS_DESCRIPTOR**Supertype**

elf:Structure

Superstructures

elf:ASSOCIATION_STRUCTURE

Substructures

None

Payload

A *line string* of arbitrary length. It is RECOMMENDED that implementations support payloads of at least 25 characters.

Describes the nature of the association described by the superstructure. This is a directed relationship. If the payload text is *R*, the person described by the record pointed to by the payload of the superstructure is *P*, and the person described by the superstructure of the superstructure is *Q* then this payload means “*P* is *Q*’s *R*”.

Example —

The following ELF fragment records that Galahad was employed by Arthur:

```
0 @arthur@ INDI
1 NAME Arthur //
1 ASSO @galahad@
2 RELA employee
0 @galahad@ INDI
1 NAME Galahad //
1 ASSO @arthur@
2 RELA employer
```

Default tag

RELA

6.114 `elf:RELIGIOUS_AFFILIATION#Individual`

Supertype

`elf:IndividualAttribute`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

The name of a religion with which the person affiliates.

Default tag

RELI

6.115 `elf:RELIGIOUS_AFFILIATION`

Supertype

`elf:Structure`

Superstructure

`elf:Event`

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

The name of a religion with which the event was affiliated.

Default tag

RELI

6.116 `elf:REPOSITORY_RECORD`

A representation of where a source or set of sources is located. May be formal, like a library, or informal, like the owner of a family bible.

Supertype

`elf:Record`

`elf:Agent`

Superstructures

`elf:Document`

Substructures

`elf:NAME_OF_REPOSITORY!`

Payload

None

Default tag

REPO

6.117 elf:RESIDENCE

Residence: either the fact of residing at, or the event of moving to, a particular location.

Supertype

elf:FamilyEvent

Default tag

RESI

6.118 elf:RESIDES_AT

Indicates that the person resided at the location indicated by the elf:ADDRESS substructure.

Supertype

elf:IndividualAttribute

Substructures

elf:ADDRESS!

Payload

None

Default tag

RESI

6.119 elf:RESPONSIBLE_AGENCY

Supertype

elf:Structure

Superstructures

elf:SOURCE_RECORD_DATA

elf:Event

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

The group or entity that was responsible for this event or data.

Default tag

AGNC

6.120 `elf:RESTRICTION_NOTICE`**Supertype**`elf:Structure`**Superstructures**`elf:INDIVIDUAL_RECORD``elf:FAM_RECORD``elf:Event`**Substructures**

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 12 characters.

Specifies how the superstructure should be treated. Known values and their meaning are listed in the following table:

Known value	Meaning
confidential	should not be distributed or exported
locked	should not be edited
privacy	has had information omitted to maintain confidentiality

Default tag

RESN

6.121 `elf:RETIREMENT`

The cessation of gainful employment, typically because sufficient wealth has been accumulated to no longer necessitate such.

Supertype`elf:IndividualEvent`**Default tag**

RETI

6.122 `elf:ROLE_IN_EVENT`**Supertype**`elf:Structure`**Superstructures**`elf:EVENT_TYPE_CITED_FROM`**Substructures**

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 25 characters.

When contained within an [elf:INDIVIDUAL_RECORD], indicates what role that individual played in the event described by this structure's superstructure. It has no defined meaning (and thus *should not* be used) outside of that context.

Known values and their meanings are listed in the following table. It is expected that additional values are also used when these are insufficient:

Known value	Role the individual played in the event
CHIL	child
FATH	father
HUSB	husband
MOTH	mother
SPOU	spouse
WIFE	wife

Default tag

ROLE

6.123 elf:ROMANIZED_TYPE**Supertype**

elf:Structure

Superstructures

elf:NAME_ROMANIZED_VARIATION

elf:PLACE_ROMANIZED_VARIATION

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 30 characters.

Identifies the romanization scheme used in the superstructure. Known values include {pinyin, romanji, wadegiles}.

Default tag

TYPE

6.124 elf: SCHOLASTIC_ACHIEVEMENT**Supertype**

elf: IndividualAttribute

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 248 characters.

An educational degree or attainment.

Default tag

EDUC

6.125 elf: SEX_VALUE**Supertype**

elf: Structure

Superstructures

elf: INDIVIDUAL_RECORD

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 7 characters.

The sex of the individual. Known values and their meanings are listed in the following table:

Known value	Sex
F	female
M	male
U	not knowable from available records

Note — GEDCOM was silent on if this was to be interpreted as biological sex or gender identity, and it is likely that data exists with both intended meanings.

Editorial note — A revision of or extension to this structure has been discussed by FHISO and is anticipated in a future release of this standard.

Default tag

SEX

6.126 `elf:SOCIAL_SECURITY_NUMBER`

Supertype

`elf:IndividualAttribute`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 11 characters.

A variant of [`elf:NATIONAL_ID_NUMBER`] assigned by the United States of America.

Default tag

SSN

Editorial note — I have not made this a subtype of IDNO because IDNO has a required TYPE where SSN does not.

6.127 `elf:SOURCE_CALL_NUMBER`

Supertype

`elf:Structure`

Superstructures

`elf:SOURCE_REPOSITORY_CITATION`

Substructures

`elf:SOURCE_MEDIA_TYPE ?`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

An identifier used by the repository to refer to the cited source.

Default tag

CALN

6.128 `elf:SOURCE_CITATION_DATA`

Supertype

`elf:Structure`

Superstructures

`elf:SOURCE_CITATION`

Substructures

`elf:ENTRY_RECORDING_DATE ?`

`elf:TEXT_FROM_SOURCE *`

Payload

None

Default tag

DATA

6.129 `elf:SOURCE_CITATION`**Supertype**`elf:Structure`**Superstructures**

`elf:INDIVIDUAL_RECORD`
`elf:FAM_RECORD`
`elf:Event`
`elf:ASSOCIATION_STRUCTURE`
`elf:PersonalName`

Substructures

`elf:WHERE_WITHIN_SOURCE ?`
`elf:EVENT_TYPE_CITED_FROM ?`
`elf:SOURCE_CITATION_DATA ?`
`elf:TEXT_FROM_SOURCE *`
`elf:NOTE_STRUCTURE *`
`elf:MULTIMEDIA_LINK *`
`elf:CERTAINTY_ASSESSMENT ?`

Payload

Either a *pointer* to a `elf:SOURCE_RECORD` or a *block string* of arbitrary length.

If the payload is a pointer, then the structure *should not* contain any `[elf:TEXT_FROM_SOURCE]` substructures.

If the payload is a block string, then the structure *should not* contain any `[elf:WHERE_WITHIN_SOURCE]`, `[elf:EVENT_TYPE_CITED_FROM]`, or `[elf:SOURCE_CITATION_DATA]` substructures.

It is *recommended* that only the pointer payload version be created.

Default tag

SOUR

Note — The text-payload version has significantly less internal structure than does the pointer version. Also note that the text-payload and pointer-payload versions may both contain `elf:TEXT_FROM_SOURCE`, but while the text-payload version has it as a direct substructure, the pointer-payload version has it both through the pointed-to structure and nested inside its `elf:SOURCE_CITATION_DATA` substructure.

6.130 `elf:SOURCE_DESCRIPTIVE_TITLE`**Supertype**`elf:Structure`**Superstructures**`elf:SOURCE_RECORD`**Substructures**

None

Payload*A block string* of arbitrary length.

A description of the source defined by the superstructure; for example, a periodical article's `elf:SOURCE_DESCRIPTIVE_TITLE` might include the title of the article and the title of the periodical; a family bible's `elf:SOURCE_DESCRIPTIVE_TITLE` might include a list of past and present owners and the book's dimensions and appearance.

Note — Although this tag is called a “title”, it is not (just) the title of the work in the usual sense of the word.

Default tag

TITL

6.131 `elf:SOURCE_FILED_BY_ENTRY`**Supertype**`elf:Structure`**Superstructures**`elf:SOURCE_RECORD`**Substructures**

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 60 characters.

A short title for this source. Intended to be used for sorting and filing.

Default tag

ABBR

6.132 `elf:SOURCE_JURISDICTION_PLACE`**Supertype**`elf:Structure`**Superstructures**`elf:EVENTS_RECORDED`

Substructure

None

Payload

A *comma-separated list*. It is RECOMMENDED that implementations support payloads of at least 120 characters.

A list of names of regions, where each element of the list is subsumed within all subsequent elements. An assertion that all events recorded in this source would have a [elf:PLACE_STRUCTURE] payload ending with this payload.

Note — While similar to the format of a elf:PLACE_STRUCTURE payload, this differs in a few key ways:

- it must use the default elf:DEFAULT_PLACE_FORMAT as it has no elf:PLACE_HIERARCHY substructure.
- it may (and often does) omit the first several elements of the list. Unlike a elf:PLACE_STRUCTURE, the omitted parts are not represented by empty strings, but by removal of their entire entry.
 - a [elf:PLACE_STRUCTURE] for an unknown location in Nevada would be “ , Nevada, USA”
 - a [elf:SOURCE_JURISDICTION_PLACE] the entirety of Nevada “Nevada, USA”

Default tag

PLAC

6.133 elf:SOURCE_MEDIA_TYPE**Supertype**

elf:Structure

Superstructures

elf:MULTIMEDIA_FORMAT

elf:SOURCE_CALL_NUMBER

Substructure

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 15 characters.

The medium of the source. Known values include {audio, book, card, electronic, fiche, film, magazine, manuscript, map, newspaper, photo, tombstone, video}

Default tag

MEDI

6.134 `elf:SOURCE_ORIGINATOR`

Editorial note—GEDCOM has this singular (0 or 1 per source record) and describes it listing only one creator. Should we change it to multiple, or de-describe it as listing all creators?

Supertype

`elf:Structure`

Superstructures

`elf:SOURCE_RECORD`

Substructure

None

Payload

A *block string* of arbitrary length.

The name of the primary creator of the source.

Default tag

AUTH

6.135 `elf:SOURCE_PUBLICATION_FACTS`**Supertype**

`elf:Structure`

Superstructures

`elf:SOURCE_RECORD`

Substructure

None

Payload

A *block string* of arbitrary length.

Full publication information for the source: when, where, and by whom it was created..

Default tag

PUBL

6.136 `elf:SOURCE_RECORD_DATA`**Supertype**

`elf:Structure`

Superstructures

`elf:SOURCE_RECORD`

Substructures

`elf:EVENTS_RECORDED *`

`elf:RESPONSIBLE_AGENCY ?`

`elf:NOTE_STRUCTURE *`

Payload

None

Default tag

DATA

6.137 `elf:SOURCE_RECORD`

Supertype

`elf:Record`

Superstructures

`elf:Document`

Substructures

`elf:SOURCE_RECORD_DATA ?`
`elf:SOURCE_ORIGINATOR ?`
`elf:SOURCE_DESCRIPTIVE_TITLE ?`
`elf:SOURCE_FILED_BY_ENTRY ?`
`elf:SOURCE_PUBLICATION_FACTS *`
`elf:TEXT_FROM_SOURCE ?`
`elf:SOURCE_REPOSITORY_CITATION *`
`elf:MULTIMEDIA_LINK *`

Payload

None

Default tag

SOUR

6.138 `elf:SOURCE_REPOSITORY_CITATION`

Supertype

`elf:Structure`

Superstructures

`elf:SOURCE_RECORD`

Substructures

`elf:SOURCE_CALL_NUMBER *`
`elf:NOTE_STRUCTURE *`

Payload

Either a *pointer* to a `elf:REPOSITORY_RECORD` or none.
If the payload is none, there *should* be a [`elf:NOTE_STRUCTURE`] describing where the information described by the containing structure can be found.

Default tag

REPO

6.139 elf:SPOUSE_TO_FAMILY_LINK**Supertype**

elf:Structure

Superstructures

elf:INDIVIDUAL_RECORD

Substructures

elf:NOTE_STRUCTURE *

Payload*A pointer* to a elf:FAM_RECORD

It *must* be the case that the pointed-to [elf:FAM_RECORD] contains a [elf:ParentPointer] pointing to the superstructure of this structure.

Default tag

FAMS

6.140 elf:SUBMITTER_NAME**Supertype**

elf:Structure

Superstructures

elf:SUBMITTER_RECORD

Substructure

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 60 characters.

The name of the submitter, formatted as it should be displayed.

Default tag

NAME

6.141 elf:SUBMITTER_POINTER

Editorial note — GEDCOM limits these to at most one per HEAD, FAM, and INDI. This seems odd; surely a source, note, etc., can also have a submitter, and there can be more than one contributing submitter per record...

Supertype

elf:Structure

Superstructures

elf:FAM_RECORD

elf:INDIVIDUAL_RECORD

elf:Metadata

Payload

A pointer to an elf:SUBMITTER_RECORD

Indicates that the pointed-to [elf:SUBMITTER_RECORD] describes a contributor of information to the containing structure, or the principle contributor of the entire dataset if in [elf:Metadata].

Default tag

SUBM

6.142 elf:SUBMITTER_RECORD

Note — A elf:SUBMITTER_RECORD describes an individual engaged in genealogical research; an elf:INDIVIDUAL_RECORD describes a subject of that research. Datasets may contain both kind of record describing the same person.

Although never permitted in its normative text, GEDCOM included at least one example (under the definition of RELATION_IS_DESCRIPTOR) where a pointer documented to point to an elf:INDIVIDUAL_RECORD pointed to a elf:SUBMITTER_RECORD instead. Implementations are encouraged to support reading datasets with that behavior, but SHOULD NOT create them.

Supertype

elf:Record

elf:Agent

Superstructures

elf:Document

Substructures

elf:SUBMITTER_NAME !

elf:MULTIMEDIA_LINK *

elf:LANGUAGE_PREFERENCE *

SHOULD NOT contain a elf:USER_REFERENCE_NUMBER even though it is an elf:Record

Payload

None

Default tag

SUBM

Note — GEDCOM permitted a SUBMITTER_REGISTERED_RFN with tag RFN, the value of which needed to be preregistered with Ancestral File, a service that is no longer available. RFN has thus been removed from this specification, making it an extension tag.

6.143 elf:TEXT_FROM_SOURCE

Supertype

elf:Structure

Superstructures

elf:SOURCE_RECORD

elf:SOURCE_CITATION

elf:SOURCE_CITATION_DATA

Substructure

None

Payload

A *block string* of arbitrary length.

An excerpt of contents of the source.

Default tag

TEXT

6.144 elf:TIME_VALUE

Supertype

elf:Structure

Superstructures

elf:CHANGE_DATE_DATE

elf:TRANSMISSION_DATE

Substructures

None

Payload

A *line string* in the *lexical space* of the `elf:Time datatype` defined in §5 of [ELF Dates].

Default tag

TIME

6.145 elf:USER_REFERENCE_NUMBER

Supertype

elf:Structure

Superstructures

elf:Record

Substructures

elf:USER_REFERENCE_TYPE ?

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 30 characters.

A user-defined identifier (textual or numeric) of this record. In GEDCOM, the examples suggests this was to allow brief links to another record keeping system, though its non-multi-values character limits that functionality.

Default tag

REFN

6.146 elf:USER_REFERENCE_TYPE

Supertype

elf:Structure

Superstructures

elf:USER_REFERENCE_NUMBER

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 40 characters.

A user-defined definition of the superstructure.

Default tag

TYPE

6.147 elf:WHERE_WITHIN_SOURCE

Supertype

elf:Structure

Superstructures

elf:SOURCE_CITATION

Substructures

None

Payload

A *block string* of arbitrary length.

Location information expressing what part of the cited source is being cited.

Default tag

PAGE

6.148 `elf:WILL`

The creation of a legal document regarding the disposition of a person's estate upon death.

Supertype

`elf:IndividualEvent`

Default tag

WILL

6.149 `elf:WITHIN_FAMILY`

Supertype

`elf:Structure`

Superstructures

`elf:BIRTH`

`elf:CHRISTENING`

Payload

A pointer to a `elf:FAM_RECORD`.

The pointed-to record describes the family unit associated with the individual event described by the superstructure.

Default tag

FAMC

7 Concrete metadata types

The following concrete types are presented in alphabetical order. Data types are presented in the previous section.

7.1 `elf:COPYRIGHT_GEDCOM_FILE`

Supertype

`elf:Structure`

Superstructures

`elf:Metadata`

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

Contains a copyright statement for the entire dataset.

Default tag

COPR

7.2 `elf:COPYRIGHT_SOURCE_DATA`

Supertype

`elf:Structure`

Superstructures

`elf:NAME_OF_SOURCE_DATA`

Substructures

None

Payload

A *block string* of arbitrary length.

Contains a copyright statement for the source dataset described by the superstructure.

Default tag

COPR

7.3 `elf:DEFAULT_PLACE_FORMAT`

Contains the default `elf:PLACE_HIERARCHY` for the full document stream.

Supertype

`elf:Structure`

Superstructures

`elf:Metadata`

Substructures

`elf:PLACE_HIERARCHY?`

Payload

None

Default tag

PLAC

7.4 elf:DOCUMENT_SOURCE**Supertype**

elf:Structure

Superstructures

elf:Metadata

Substructures

elf:VERSION_NUMBER

elf:NAME_OF_PRODUCT

elf:NAME_OF_BUSINESS

elf:NAME_OF_SOURCE_DATA

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 20 characters.

In early GEDCOM, this was a unique string assigned to each product through a registration process. That process no longer exists.

Editorial note — Do we want to make a new recommendation for the contents of this payload? Perhaps an IRI + date pair? A UUID? A generic “UNREGISTERED_PRODUCT” string or the like?

Default tag

SOUR

7.5 elf:FILE_NAME

Editorial note — What is the purpose of this structure? Clearly it cannot always match the name of the physical file, which can be renamed without editing; nor are there any limitations given on it in GEDCOM besides that it include an extension *if* the file containing it has an extension in its name. Without knowing its purpose, I don’t know how to document this structure.

Supertype

elf:Structure

Superstructures

elf:Metadata

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

The base name (i.e., not a full path) of a file.

Default tag

FILE

7.6 `elf:GEDCOM_CONTENT_DESCRIPTION`

Supertype

`elf:Structure`

Superstructures

`elf:Metadata`

Substructures

None

Payload

A *block string* of arbitrary length.

A description of the intended scope of the contents of the dataset.

Default tag

NOTE

7.7 `elf:GEDCOM_FORM`

A holder for formatting and version information.

Supertype

`elf:Structure`

Superstructures

`elf:GEDCOM_FORMAT`

Substructures

None

Payload

The exact string LINEAGE-LINKED

Default tag

FORM

7.8 `elf:GEDCOM_FORMAT`

A holder for formatting and version information.

Supertype

`elf:Structure`

Superstructures

`elf:Metadata`

Substructures

`elf:GEDCOM_FORM`

`elf:VERSION_NUMBER`

Payload

None

Default tag

GEDC

7.9 `elf:LANGUAGE_OF_TEXT`

Editorial note — Should this really be a pseudo-structure? If we re-work this as having language-tagged strings as payloads, then it is; but if we leave the strings in this document as non-language-tagged then it is data instead.

Supertype

`elf:Structure`

Superstructures

`elf:Metadata`

Substructures

None

Payload

A *line string* matching the Language Tag microformat.

Indicates the default language of the free-text payloads in the dataset.

Default tag

LANG

7.10 `elf:NAME_OF_BUSINESS`

Supertype

`elf:Agent`

Superstructures

`elf:DOCUMENT_SOURCE`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

The name of the entity that produced the product described by the superstructure.

Default tag

CORP

7.11 `elf:NAME_OF_PRODUCT`

Supertype

`elf:Structure`

Superstructures

`elf:DOCUMENT_SOURCE`

Substructures

None

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

The name of the product described by the superstructure.

Default tag

NAME

7.12 `elf:NAME_OF_SOURCE_DATA`

Supertype

`elf:Structure`

Superstructures

`elf:DOCUMENT_SOURCE`

Substructures

`elf:PUBLICATION_DATE`

`elf:COPYRIGHT_SOURCE_DATA`

Payload

A *line string*. It is RECOMMENDED that implementations support payloads of at least 90 characters.

The name of an electronic data source from which this dataset was extracted.

Default tag

DATA

7.13 `elf:PUBLICATION_DATE`

Supertype

`elf:Structure`

Superstructures

`elf:NAME_OF_SOURCE_DATA`

Substructures

None

Payload

A line string in the lexical space of the `elf:DateExact` datatype defined in §4.1.1 of [ELF Dates]. Contains the date the source dataset (described by the superstructure) was published or created.

Default tag

DATE

7.14 `elf:RECEIVING_SYSTEM_NAME`

Supertype

`elf:Structure`

Superstructures

`elf:Metadata`

Substructures

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 20 characters. Identifies the intended recipient software of this dataset.

Default tag

DEST

7.15 `elf:SUBMITTER_POINTER`

Used both in metadata and in data; see `elf:SUBMITTER_POINTER` for a discussion.

7.16 `elf:TRANSMISSION_DATE`

Supertype

`elf:Structure`

Superstructures

`elf:Metadata`

Substructure

elf:TIME_VALUE

Payload

A line string in the *lexical space* of the elf:DateExact datatype defined in §4.1.1 of [ELF Dates].
The date that this dataset was created.

Default tag

DATE

7.17 elf:VERSION_NUMBER**Supertype**

elf:Structure

Superstructures

elf:DOCUMENT_SOURCE

elf:GEDCOM_FORMAT

Substructure

None

Payload

A line string. It is RECOMMENDED that implementations support payloads of at least 15 characters.

A version identifier, with syntax and semantics varying by context. If the superstructure is [elf:GEDCOM_FORMAT], the payload *should* be the exact string “5.5.1”.

Default tag

VERS

8 Pseudo-structures**8.1 elf:Document**

Not a structure at all, elf:Document is a special IRI used as the *structure type identifier* of the *superstructure* of a *structure* that does not have a *superstructure* but instead is directly included in the dataset.

8.2 elf:Metadata

Not a structure at all, elf:Metadata is a special IRI used as the *structure type identifier* of the *superstructure* of a *structure* that does not have a *superstructure* but instead is directly included in the metadata of the dataset.

9 Extensions

There are several reasons why a *structure* may be considered an **unknown extension** to this data model:

- It is not the first *substructure* of its type in its *superstructure*, and the *superstructure* expected no more than one *substructure* of its type.
- It is a *substructure* of a type not expected by its *superstructure*.
- It is missing a required *substructure*, or that *substructure* has some error that makes it an *extension*.
- Its *payload* is not appropriate for its *structure type*: it has a *payload* where none was expected, or a *string* where a *pointer* was expected or vice versa, or a *pointer* to the wrong *structure type* or to an *extension*.
- Its *structure type identifier* does not indicate any known *structure type*.

Unknown extensions are permitted by this data model, and a dataset **MUST NOT** be rejected simply because it contains *unknown extensions*. A *conformant* application **MAY** remove *unknown extension structures*, but doing so is **NOT RECOMMENDED**. Implementations **SHOULD NOT** create, modify, move, or duplicate *structures* it considers to be *unknown extensions*.

If a *structure* is removed as part of removing an *unknown extension*, all of its *substructures* and all *structures* that point to it **MUST** also be removed, recursively.

Note — [ELF-Serialisation] converts most serialisation errors into *unknown extensions* with a *structure type identifier* beginning `elf:Unknown`. FHISO currently does not intend to define semantics for `elf:Unknown`-prefixed *structure type identifiers* in any future ELF standard.

9.1 Extension types

This data model may be extended by creating new **extension types**, which are *structure types* that are not documented in ELF 1.0.0.

Extension types' structure type identifiers **SHOULD** be an IRI with an authority component owned by the extension author, as documented in [Basic Concepts].

Example — Suppose an implementation not understanding the `http://example.com/WEALTH` extension type is processing a dataset containing two `elf:INDIVIDUAL_RECORDS`, one of which has a `http://example.com/WEALTH` substructure with payload “34K/A”.

- The implementation may choose to ignore the `http://example.com/WEALTH` or to display it in some default fashion.
- If the dataset is modified and exported

- If the `elf:INDIVIDUAL_RECORD` with that `http://example.com/WEALTH` still exists in the dataset, the `http://example.com/WEALTH` structure *should* be preserved, but it *may* be omitted.
- No additional `http://example.com/WEALTH` structure *should* be created.
- The payload of the existing `http://example.com/WEALTH` structure *should not* have been modified.

If the implementation discovers the meaning of `http://example.com/WEALTH`, it is welcome to create and modify `http://example.com/WEALTH` structures as it sees fit (subject to any constraints specific to that structure type).

10 References

10.1 Normative references

[Basic Concepts]

FHISO (Family History Information Standards Organisation). *Basic Concepts for Genealogical Standards*. First public draft. (See <https://fhiso.org/TR/basic-concepts>.)

[ELF Dates]

FHISO (Family History Information Standards Organisation). *Extended Legacy Format (ELF): Date, Age and Time Microformats*. Exploratory draft. (See <https://fhiso.org/TR/elf-dates>.)

[RFC 2119]

IETF (Internet Engineering Task Force). *RFC 2119: Key words for use in RFCs to Indicate Requirement Levels*. Scott Bradner, 1997. (See <http://tools.ietf.org/html/rfc2119>.)

[RFC 3987]

IETF (Internet Engineering Task Force). *RFC 3987: Internationalized Resource Identifiers (IRIs)*. Martin Duerst and Michel Suignard, 2005. (See <http://tools.ietf.org/html/rfc3987>.)

[RFC 5322]

IETF (Internet Engineering Task Force). *RFC 5322: Internet Message Format*. P. Resnick, 2008. (See <http://tools.ietf.org/html/rfc5322>.)

[RFC 7230]

IETF (Internet Engineering Task Force). *RFC 7230: Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*. Roy Fielding and Julian Reschke, eds., 2014. (See <http://tools.ietf.org/html/rfc7230>.)

[XML]

W3C (World Wide Web Consortium). *Extensible Markup Language (XML) 1.1*, 2nd edition. Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, and John Cowan eds., 2006. W3C Recommendation. (See <https://www.w3.org/TR/xml11/>.)

10.2 Other references

[GEDCOM 5.5]

The Church of Jesus Christ of Latter-day Saints. *The GEDCOM Standard*, release 5.5. 2 Jan 1996, as amended by the errata sheet dated 10 Jan 1996.

[GEDCOM 5.5.1]

The Church of Jesus Christ of Latter-day Saints. *The GEDCOM Standard*, draft release 5.5.1. 2 Oct 1999.