

Summary of Citation Style Language version 1.0

This is an informal summary of the specification found here:

<http://citationstyles.org/downloads/specification.html>

CSL <http://citationstyles.org/> is a complex XML based language for definition of citation styles, where the most important ingredient is what in BG has been called citation templates. CSL is used by the citation program Zotero, www.zotero.org and others. Public domain software is available for generation of the output: <http://citationstyles.org/citation-style-language/csl-processors/>

The reason for looking at CSL is to understand some of the issues that might arise when designing technical solutions for citations in BetterGEDCOM. A theoretical possibility is to use CSL (possibly extended) for BG, but CSL is certainly too complex for definition of styles by ordinary users, (but users might perhaps be able to extend an existing style with source types and elements?).

The sequence of the text in this document follows that in the specification.

Some observations:

- CSL contains some functions that might reduce the number of source types, e.g. functions for handling one or more authors.
- CSL handles translation (localization) of styles so you don't have to translate a style, you just define what is specific for a locale – independent of all styles.
- CSL defines 59 general variables (citation elements in BG terminology, see appendix 1) and 34 general types (BG: source types, see appendix 2).
- Use of CSL in BG would most likely require an extension to CSL (??for additional citation types??) additional “variables” and source types, and maybe “qualifiers” (see my separate document on technical solutions for style guides in BG – to be published). Also, the public domain software for generation of CSL based citations must be extended.
- It is so far unclear to me how CSL specifies the variables that can be recorded for a type (=source type).
- CSL has a structure for personal names with some features that may be of interest to BG.
- Personal comment: Being a guy that are not very familiar with the art of citation, I got the impression that this is very complex technology to produce something I don't care too much about how it looks, as long as it makes it possible to find the cited text or whatever. But others most likely have more strict requirements.

Example style specifications: All current styles: <http://www.zotero.org/styles>

Chicago author date style: <http://www.zotero.org/styles/chicago-author-date>

Locales (language adaptations): <https://github.com/citation-style-language/locales>

Many are just copies of the one for English, but see e.g. nb-NO or de-DE.

Summary

1. Style. (Root)

(The spec has no clause numbers as does this document.)

A style can choose between inline citations or “notes” – footnotes and endnotes, and default *locale*. A style contains:

- Info – metadata
- Citation – describes format of citations (inline or *note* = footnote/endnote)
- Bibliography – describes the format of it
- Macros – for reuse of formatting info
- Locale – overrides defaults depending on locale

1.1 Info contains

- Style author or contributor
- Category – 1) format of inline citation, 4 variants, or indicates note 2) A discipline, e.g. medicine – alternatives are predefined
- Style id = URI
- Issn/issn-l – for journal styles
- Link – URI for automatic style updating, or URI for parent style (ref dependent styles)
- Publication timestamp
- Rights – license – e.g. creative commons
- Summary – incl language of summary
- Title – incl language of
- Updated – timestamp - for auto update

1.2 Citation

Describes formatting of citation, one or more sources per citation, - inline or note. Citation contains:

- Sort – how to sort multiple references
- Layout – see layout below

Has attributes: citation specific formatting options, inheritable name options

1.3 Bibliography

Same attributes as Citation. Contains Layout as above, and Sort (how to sort entries in bibliographies)

1.4 Macro

For reuse of formatting instructions. Has a name and contains rendering elements. One macro can be used in several Citations, Bibliographies and other macros. Macros should be used as much as possible. For an example where all output (layout) for a bibliography is defined using macros, see the end of the “Chicago author date” style linked to in the beginning of this document.

1.5 Locale

CSL supports localization of terms, date formats and formatting options. Default Locales can be stored in separate files (see link in the beginning of this document), and/or can be contained in styles. Locale is identified by language, eg en-US. A hierarchy of locales can be used, eg English and en-US, where the latter overrules some of the former.

Dependent styles = alias name for a generic style. A style can be defined, perhaps without name?, and be given several names – eg per journal so it appears as if each journal has its own style.

1.5.1 Locale files

Contains translation of, or special localized notation for, **terms**. An option for a locale is punctuation-in-quotes.

1.5.2 Terms

Terms are localized strings. Terms are identified by its name attribute. Eg term named “no-date” printed as “n. d.”, or the one named “and” printed in German as “und”. Appendix III in CSL defines a long list of terms. A term can be translated into its singular or plural form depending on some condition. Eg. “short page” translates to p. or pp.

Some terms have **forms**: For eg. the term “editor”, the following forms are defined: Long (editor, editors), short (ed eds), verb (edited by), verb-short (ed). A form is also symbol, § for the term section.

1.5.2 Localized dates

Dates are recorded in three elements, day month year. Date formats (sequence/order of elements, separators, prefix, suffix) can be specified for text and numeric dates.

2. Rendering elements (Citation templates in BG terms)

Rendering elements are used to specify which, and in what order, bibliographic data should be included in citations or bibliography entries, and may partly control the formatting.

The following rendering elements are defined (they are described below):

- Layout
- Text
- Date
- Number
- Names
- Label
- Group
- Choose

2.1 Layout

Layout is the container for all rendering elements in citations or bibliographies (it, and its content, contains all the layout rules), and is itself a rendering element. It can have a prefix, suffix and a delimiter used when the content (called text) repeats. Simple example: “(1,2)”, where (=pref,)=suff, comma is delimiter. 1 and 2 are very simple rendering elements, can be much more complex.

Note: It is not 100% clear to me if there must be only one Layout in a citation or if there can be several, one for each rendering element. Most likely only one, it simply specifies that what is in the Layout specifies the layout of the citation or bibliography entry.

2.2 Text

Text are the major building blocks of layouts. Text can be one of (it is a general name for any of these):

- **Variable** that contains the actual information about the source (eg. title, author) or other info in information that is registered for inclusion in a biography or note. It corresponds to BG’s “citation element types”. Variables can have a form, short or long. Standard variables are defined in appendix 1. Seems like there can be only one occurrence of a variable per citation. (Variables are also used in macros.) Special variables are Date and (person) Name variables, for which special rules apply (similar to data types).
- **Macro** The text output by a macro identified by its name.
- **Term** The text of a localized term. The plurals attribute selects either plural or singular. The form attribute selects long, short, verb, verb-short or symbol (see above), with rules for what happens when the form does not exist for the term.
- **Value** Simply a text string, eg “some text”

Text can have attributes, controlling its appearance. Attribute types are: affixes, display, formatting, quotes, strip-periods and text-case. See below.

2.3 Date

Date is used to output localized or non-localized dates, identified by the name of a variable. Dates can have numeric or text form. A date-partS attributes selects which parts of the date to output, year-month-date, year-month or year (not necessarily in the order mentioned here). The date-part description (no S, see below) can be used to override settings in a locale file, so a date can be output as specified for this particular date (non-localized) rather than as specified by a locale.

Date can have attributes controlling its appearance. Attribute types are: affixes, display and formatting See below.

2.3.1 Date part

Date parts are: day (forms: numeric, numeric leading zero, ordinal eg 1st), month which are localized and punctuation (Mar.) can be striped (forms. Long, short, numeric and numeric-leading-zero) and year (forms: long and short). Date part may have attributes controlling its appearance: affixes, formatting, text-case, and a (date-)range-delimiter, see Date range.

2.3.2 Date Ranges

Default separator dash, eg “May-July 1876” Special notation for date ranges can be specified, eg May 2008/June 2009.

2.3.3 AD – BC

Localized terms for AC and BC are added to positive respective negative years.

2.3.4 Seasons

The 4 seasons are localized

2.3.5 Uncertain dates

Uncertain dates can output special format and “ca”

2.4 Number

Numbers can be used to output the variables edition, volume, issue, number and number-of-volumes. Forms can be numeric, ordinal, long-ordinal or roman. There are special rules for variables mixing numbers and letters. Numbers can have formatting defined by the attributes affixes, display, formatting and text-case.

2.5 Names

Names are stored in a set of predefined variables, eg author, editor etc (see appendix I) Each one can contain several names. One Names rendering element can render several variables (eg both author and editor). Special case: If editor and translator contains the same name, the editor and translator terms are replaced with the editor-translator term. A delimiter can be defined to delimit names from different variables. Names can have the following child elements: name, et-al, substitute and label, see below. Formatting are defined by attributes affixes, display and formatting.

2.5.1 Name

Name describes the formatting and separation of names within a variable holding names. The attributes of name are:

- And – specifies the delimiter between the two last names, text (and) or symbol (&)
- Delimiter – between individual names in the variable
- Delimiter-precedes-last – various conditions for using delimiter rather than “and” between last two names
- Et-al-min/et-al-use-first – how many names can be rendered before truncating list of names with et. al. and where to place et. al. Applies to all names.
- Et-al-subsequent-min/et-al-subsequent-last – applies to subsequent names, when citing earlier cited items.

Names must contain family name (without prefix or suffix) and may contain given name, suffix (Jr), non-propping-particles (shown when surname shown, not sorted on, eg “de” in dutch “de Koning”)

and dropping-particles (NOT shown when surname shown, not sorted on, eg “von” in “von Bethoven”)

Additional attributes controlling rendering of names:

- Form – long (whole name), short (family name and non-dropping particle). Special value “count” used for advanced sorting, see spec.
- Initialize-with – Show only initial for given name with specified suffix (eg. J.J. Doe, suffix=.) initialize-with-hyphen controls display of compound given names, eg. Jean-Luc – see below.
- Name-as-sort-order – swap given and surname (Doe, John) in first or all names in variable. Sort order and display order may be different - only for Latin and Cyrillic char sets, Asian always display family name first.
- Sort-separator – alternative to default comma in “Doe, John”

The appearance of name, and individual parts of names, may be controlled by affixes and formatting attributes.

Many examples showing various ways to display names, depending on combinations of the above parameters, character set etc, are given.

2.5.2 Et-al

Choose between “et-al” and “and others” terms (will be localized). Affixes and formatting can be controlled.

2.5.3 Substitute (child element of names)

Controls what should be rendered when a variable containing name(s) is empty, for example some text or the value of other variables in a prioritized order (if some of them are also empty), eg. replace author with editor and if no editor with title (this rule would be specified by a macro shown in the spec.

2.5.4 Label for names

Labels (eg (editor)) for names may render in singular or plural form depending on the number of names in a variable. See general use of label below.

2.6 Label

Labels are the name of variables, and can be printed for variables Page, Location and the Name variables. (It is not clear to me if other variables has a label.) They have a form, same forms as defined for variables (long, verb etc). The plural attribute have values: Contextual (the number of names or pages in the variable controls output), always plural and never plural. Attributes controlling appearance ate affixes, formatting, text-case and strip-periods.

2.7 Group

Group may contain one or more rendering elements (except Layout). Group has attributes: delimiter (delimits the elements in the group if defined), affixes (applies to the whole group) and formatting (applies to the elements – maybe there are extra rules here). Important: If at least one element loads

the value of a variable (directly or via a macro), and if all variables of the group are empty, the group is not rendered. The group thus also has a conditional” functions (if/then). Groups can for example be used to render a variable with a label, eg “(Published by: Company X)”.

2.8 Choose

Choose is used to control conditional rendering of rendering elements. It is an if, else-if, else structure (as in programming languages). The if and elseif “keywords” are followed by a “condition” that is tested (true or false) (similar to an expression in a programming language). Depending on the result of the test, a number of rendering elements that are specified (I will call them the output), are rendered. The types of conditions are:

- Disambiguate – output if the output disambiguates otherwise identical citations – see below.
- Numeric – tests if variable contains numeric data
- Is-uncertain-date – self explanatory
- Locator – tests if locator is of a specified type – see locators.
- Position – position of one cite (citation) within a multi cite citation. I don’t understand all of this, it has to do with “ibid” and locators.
- Type – tests if the source type is a specified type
- Variable – tests if a specified variable is empty

All conditions, except Disambiguate, can test against multiple specified values. Several conditions can be specified for an if/else-if, and the value of a “match” attribute specifies if all, any or none conditions are required for the test to return true.

3. Style behavior

3.1 Options

Options can be set for Citations only, Bibliographies only, or both – the whole Style. “Inheritable name options” for the same three mentioned, and Locale options for Locale.

3.1.1 Citation specific options

3.1.1.1 Disambiguation

The purpose of Disambiguation is to include enough information in 1) a cite to make it unambiguous within the document (ie. [Doe] and [Doe] are ambiguous, but [Doe 2007] and [Doe 2009] are not) or 2) to make names unambiguous within the document. Ambiguous info can be caused by “et al” truncation, missing given names, initialized given names, authors with the same surname (add publication year to disambiguate, [Doe 2007] rather than [Doe]) and other things (=?) that test ambiguous in the choose (see above) construct. Various algorithms for disambiguation and attributes to control them are described.

3.1.1.2 Citation collapsing

Collapse citation numbers and/or years. Variants are 1) citation number ([1, 2, 3, 5] becomes [1-3, 5]) 2) year ((Doe 2000, Doe 2001) becomes (Doe 2000, 2001)) 3) year-suffix ((Doe 2000a, 2000b) becomes (Doe 2000a, b)) 4) year-suffix-ranged ((Doe 2000a, b, c, e) becomes (Doe 2000a-c, e)). Year-

suffix-delimiter set to e.g. semicolon produces (Smith 1999a,b; 2000; Jones 2000). After-collapse delimiter set to semicolon produces (Smith 1999a,b, 2000; Jones 2000, Brown 2007).

3.1.1.3 Note distance

The near-note position (see Choose) tests true if 1) a preceding note referring to the same item exists and 2) the distance (measured in number of foot/endnotes) from the current item is less than the value of near-note-distance (default 5) What is this used for????

3.1.2 Bibliography specific options

3.1.2.1 Whitespace

Attributes controlling 1) hanging indent 2) alignment of second line to start of second field in bibliography entry 3) Line spacing 4) Entry spacing

3.1.2.2 Reference grouping

Attribute subsequent-author-substitute specifies some text (eg "---") that replaces the author name in biographies entries with the same author, except in the first entry, eg:

Doe, Doebook1, 2000.
---, Doebook2,2001.
---, Doebook3, 2002.

3.1.3 Global options

- Hyphenation of Initialized names: Eg for Jean-Luc it produces J.-L. (default) rather than J. L.
- Page ranges – controls appearance of page ranges 1) expanded 321-328 2) minimal 321-8 3) Chicago 321-28 (and more detailed rules in appendix 4)
- More explanation of name particles are described, including the attribute demote-non-dropping-particle, controlling sorting and rendering

3.1.4 Inheritable name options

Certain name related attributes can be set for the style, all citations, all biographies rather than for each place where a name can occur, but if set for that place it will override the setting for style, citation or biography.

3.1.5 Locale options

Punctuation-in-quote – inside or outside quote

3.2 Sorting

Controls sorting of citations and biography entries, defaults to soring in order entered. Sorting is done on specified variables or macros, for each in descending or ascending order, and the sorting precedence (in what order) is specified. Additional control is available. (Biographies are often sorted on author.)

3.2.1 Sorting variable

Specifies special rules for how names, dates, numeric (and other variables) are presented to the sorting function, i.e. reformats them before sorting takes place.

3.2.2 Sorting macros

Macros can do special processing on variables before the result is input to the sorting function. Eg editor substituted for missing author. Several situations and rules are given.

3.3 Formatting

Various formatting, and the elements they can apply to, are specified. Eg, italic, oblique, small caps, bold, light, underline, super/sub-script where many of these can be combined (?).

3.4 Affixes

Specifies a large list of elements that can have specified prefixes and/or suffixes inserted/appended, eg (and).

3.5 Delimiter

Specifies the elements that can have a delimiter, as described above.

3.6 Display

A style can be assigned (in the output) to various rendering elements for use in bibliographies. These styles can be recognized and changed in word processing software after output. The styles are: bloc, left-margin, right-inline, inline. Various examples are given.

3.7 Quotes

Places a rendering element in quotes.

3.7 Strip periods

Periods in the specified rendered text are removed

3.8 Text case

Transforms the specified element to Lower case, uppercase, capitalize-first, all-caps, title case (cf Chicago), sentence case

Appendixes

1. Variables (Standard variables, date variables, name variables)
2. Types – source types
3. Terms (categories (application area, eg biology, law), Locators (eg chapter, page), Months, Ordinals, Quotation marks, Roles (eg author, translator), Seasons, miscellaneous (eg accessed, circa, ibid, et-al, edition))
4. Page abbreviation rules