fhiso

CFPS 102
(Call for Papers Submission number 102)

Proposal to support dates BC as negative years

Submitted by: Smith, Richard

Created: 2014-01-29

URL: Most recent version: http://fhiso.org/files/cfp/cfps102.pdf
This version: http://fhiso.org/files/cfp/cfps102_v1-0.pdf

Description: This paper presents a case for allowing dates BC to be recorded using the standard Julian and Gregorian calendars, proposes a representation for such dates that is naturally sortable.

Keywords: dates, calendars, proleptic calendars, epoch, BC

Abstract

This paper presents a case for allowing negative years to be encoded in the representations Julian and Gregorian dates to represent dates BC. Several competing formats for dates BC are discussed including those used by ISO 8601 and the subtle variant used by XML Schema. The importance is reiterated of having a naturally sorting representation, which ISO 8601's is not. In the absence of a known existing format with these properties, this paper proposes a novel format which is marked by two-character `-^` prefix. It is proposed that this be allowed in both the Gregorian and Julian calendars currently being proposed, together with any future calendars.

1 Introduction

The proposal for a representation of Gregorian dates in CFPS 17 [1] do not allow for the representation of dates before the start of the year 1 AD. This restriction applies by extension to the proposal for Julian dates in CFPS 44 [2], and is justified in CFPS 17 as follows:

> This proposal does not consider negative years, e.g. for addressing dates BC. Although the ISO standard suggests these can be supported by agreement between two parties, it breaks the goal of the fixed field widths. Also, such dates would not be Gregorian, and possibly not even Julian.

This paper takes the contrary position — that support for dates BC is desirable in several important use cases.

2 Objections to dates BC

2.1 Their representation is not fixed-width

The first objection to dates BC is that they break the goal of having fixed-width fields in the representation. It is the contention of this paper that having fixed-width field is not itself a particularly useful goal; rather, it is several of the properties that often arise from fixed-width fields that are worth pursuing.

Natural sorting is one such goal. Natural sorting was defined in CFPS 43 as when "lexicographical sort of the representation is sufficient to place the dates in chronological order" [3]. When sorted lexicographically (for example, as done by the `strcmp` function in many popular programming languages), `1031-07-20` sorts

before 972-03-27, despite the fact that 1031 is a later year than 972. By fixing the width of the year field at four digits, the latter year is represented as 0972-03-27, and therefore sorts correctly. Having a single *canonical representation* is another desirable property. There should be one way and only one way of representing any give date. Again, this is achieved by requiring a four-digit year.

If a representation for years BC can be found that preserved these properties, then it should not matter whether years BC and years AD have the same field length.

2.2 They are anachronistic

Not every source used by genealogists is a primary source. Sometimes the primary source no longer exists; sometimes they are not accessible or require expert translation or interpretation beyond the capabilities of the genealogist. A secondary source may well express dates differently to primary sources, perhaps using a calendar or year numbering scheme that did not exist at the time. Many history books, for example, give the date of birth of the Emperor Augustus as 23 September 63 BC, even though contemporary documents would not have written the year as 63 BC. Primary sources may have written 691 AUC, the year being counted *ab urbe condita*, since the founding of Rome. However, as was established in CFPS 43, counting years from a different epoch does not constitue a different calendar [3].

In this example, not only is the year BC anachronistic, the date 23 September is not Julian (and certainly not Gregorian). This is because it is not certain how long each year before 5 AD was. In fact, in the *proleptic* Julian calendar — the Julian calendar extended back beyond 5 AD— the birth of Augustus would probably have been in about July. An application wishing to record this date accurately would need to do it in a custom Roman calendar, but other applications might then struggle to handle the date gracefully.

An alternative is to convert the date to the Julian calendar, and represent it with just a year (63 BC), or a month and a year (presumably July 63 BC). Although less precise, the proleptic Julian form has the advantage that it will be much more widely understood by applications, and can be readily compared to dates after the commencement of the Julian calendar. Furthermore, if the year is only known approximately, then the difference between calendars is largely irrelevant and the advantages of giving the approximate year in the proleptic Julian calendar significantly outweigh the advantages of using a poorly-supported Roman calendar.

In practice, an application might choose to use CFPS 14's mechanism for dual dating to store both a Roman and a proleptic Julian date, with a suitable annotation to mark one date as being per the source and the other being a conversion of that [4].

More generally, it seems desirable to provide a mechanism by which applications can provide an alternative representation of a date, in order to assist applications in the handling of obscure calendars that may not receive widespread application support.

2.3 They have no genealogical use

Another possible argument against allowing years BC is that they are irrelevant to genealogy. It is sometimes said that St. Arnulf, Bishop of Metz (*c* 582–640) is the earliest person from whom a documented descent to modern times exists. This, however, is a Eurocentric statement, and there are serious Chinese academic genealogies going back to Confucius (551–479 BC). As an international standards body, the FHISO should certainly permit the recording of such culturally-important lineage.

Even in Western Europe, serious genealogies have been postulated that extend back considerably further and, although they are not generally considered proven, genealogists may still wish to record them. An unproven genealogy might be recorded, for example, to facilitate discussion of the uncertain links. Even palpably false genealogies, such as the 'Jesus bloodline' popularised in *The Da Vinci Code*, can be worth recording, if only so that they can be recognised and dismissed when their bogus parts turn up in other genealogies.

Finally, not all genealogists are critical with what they accept, and major genealogy programs are unlikely to adopt a system that does not cater for the needs of the less serious genealogists. Plenty of genealogies can be found on the Internet extending back to the biblical Adam who is generally given a date of birth of *c* 4000 BC. They may be worthless nonsense, but the importance of accepting them should not be underestimated, and even serious genealogists might keep record of such descents for their private amusement.

3 Technical requirements for dates BC

3.1 Specification of epochs

A specific technical scenario in which it is necessary to express dates BC (and more generally, negative years in arbitrary calendars) is the specification of epochs, as described in CFPS 67 [5].

The Byzantine calendar is the same as the Julian calendar except that years begin on 1 September, and are counted from the supposed date of creation. The arguments given in CFPS 43 suggest that this should not be considered a different

calendar to the Julian calendar, but rather a stylistic variation of it [3]. However, if an application is to handle arbitrary epochs such as this one, it needs a way of representing, in the Julian calendar, the date of the epoch. The Byzantine epoch is 1 September 5509 BC. Even if there is no suggestion that genealogical dates this long ago are relevant, it is sometimes necessary to write such dates for technical reasons: here, to specify the conversion between the Byzantine *anno mundi* and Western *anno domini* epochs.

3.2 Calendar compatility

One possible way of representing dates BC is to have separate calendars for the proleptic versions of the Julian and Gregorian calendars, and to represent years counting from some arbitrary epoch earlier than anything that might sensibly be considered, say 10000 BC. This is fine insofar that such a calendar would be naturally sorting. The problem arises when dates BC and dates AD are to be considered in conjunction. In general, handling dates in separate calendars is difficult, and an application cannot order them or work out how far apart they are without converting everything to a common calendar.

The compound calendars introduced in CFPS 38 only provide a partial solution to this [6]. The example in that paper considered the transition from the Julian to the Gregorian calendar in England in 1752. However the year number didn't change during this, and in England the year 1751 came before 1752 which came before 1753. The (compound) English calendar still therefore naturally sorts. A compound calendar composed from a proleptic calendar with epoch of, say, 10000 BC followed by the usual Julian calendar would not be naturally sorting as the year after 9999 (representing 1 BC) would be 0001 (for 1 AD).

3.3 Confusion with ISO 8601

The ISO 8601 representation fails to provide natural sorting of dates. It uses -0062 as the representation of 63 BC, which fails to naturally sort with other dates BC. Certainly it is not hard to write code that does the comparison correctly by recognising the leading - sign and treats the year as a negative number, but treats the remaining components (the month and day) positively. But that supposes an environment in which a custom comparator is permitted. In many environments, it is not: XSLT is an example. (A workaround is possible with XSLT but it is cumbersome. This is likely to be the case in other environments too.)

Perhaps more seriously, it is counterintuitive because it represents 63 BC with -0062 [7]. Worse, not all standards have followed ISO 8601's representation of negative years, because until the ISO standard's second edition (in 2000), it was

unclear whether 0000 was a valid ISO 8601 year. This has resulted in much confusion. XML Schema's dates are ISO-compliant *except* that it contradicts ISO 8601 by representing 63 BC as -0063 [8, §3.2.7]. This paper therefore recommends against using ISO 8601's date format for years BC. Irrespective of whether the correct ISO interpretation is followed, or whether XML Schema's more intuitive interpretation is used, confusion is very likely to result.

4 A possible representation

As mentioned in §2.1, the ISO 8601 representation of dates BC is not naturally sorting. This paper suggests that the year y BC should be represented using the number $10000 - y$. A prefix would still be needed so that 9937 (for 63 BC) sorts 0014. A '-' sign is a possibility, but that introduces an unnecessary and undesirable incompatibility with ISO 8601 and standards such as XML Schema that follow it. (In fact, XML Schema's handling of dates BC contradicts ISO 8601, in that the latter includes a year 0000 while the former does not [8].)

In order to sort before positive dates, the prefix must begin with an ASCII character between 33 and 47, inclusive: `! " # $ % & ' ( ) * + , - . /`. None of these immediately seem ideal on its own (although `!` seems the best of the bunch), so this paper tentatively proposes a two-character prefix of '`-^`'. The minus sign indicates a negative year number, while the '`^`' disambiguates it from an ISO 8601-like negative year. In this scheme, 23 September 63 BC would be represented `-^9937-09-23`. The year is fixed-width, but the date BC is two characters longer than a corresponding date AD. Dates expressed this way compare correctly under a straightforward lexicographic comparison, and a date still has a single canonical representation.

(A reason for choosing '`^`' is that it often represents the exclusive-or operation in many computing languages. In machines using a ones' complement representation of negative numbers, $-y$ is represented by `0xffff...` $\hat{}\ y = $ `0x10000...` $- y$. What is propopsed here is the decimal analogue of that.)

This paper proposes that negative years should be allowed in every calendar, including the Gregorian calendar of CFPS 17 [1] and the Julian calendar of CFPS 44 [2]. It is tentatively proposed that the `-^` syntax described here is adopted.

4.1 Rejected fixed-width alternatives

If completely fixed-width fields are desired, it would be possible to use `!` in place of `-^` for negative years, and to require a `+` prefix on all positive dates. This may seem like an obvious win, but for the reasons discussed in §2.1, fixed-width fields

do not bring any significant advantages beyond those already achieved. However, the requirement to use a + prefix on positive years has a big drawback: it is incompatible with many ISO-compatible formats that do not allow a leading +. The date types of XML Schema are an important example [8]; the TEI encoding format is another [9]. This paper does not make such a proposal for this reason.

Another way of making the representation fixed-width, while retaining compatibility with XML Schema for dates AD, is to use one fewer digit in the year for dates BC— e.g. to write `!937` for the year 63 BC, but 2014 for the current year. However, some of the use cases discussed here require years before 1000 BC. For example, the Byzantine epoch was in 5509 BC. This paper does not recommend such a strategy for that reason.

References

[1] Tony Proctor, 2013, *Proposal to Accommodate Gregorian Dates using a Modified ISO 8601* (CFPS 17), `http://fhiso.org/files/cfp/cfps17.pdf`

[2] Richard Smith, 2013, *Proposal to support the Julian calendar similarly to CFPS 17* (CFPS 44), `http://fhiso.org/files/cfp/cfps44.pdf`

[3] Richard Smith, 2013, *Proposal to add style to the wholly-numeric representation of dates in CFPS 13* (CFPS 43), `http://fhiso.org/files/cfp/cfps43.pdf`

[4] Tony Proctor, 2013, *Proposal for a Generalised Dual-Date Representation* (CFPS 14), `http://fhiso.org/files/cfp/cfps14.pdf`

[5] Richard Smith, 2013 *Proposal to extend the calendar style mechanism of CFPS 43 into an abstract formatting model* (CFPS 67), `http://fhiso.org/files/cfp/cfps67.pdf`

[6] Richard Smith, 2013, *Proposal for compound calendars to resolve a difficulty with default calendars* (CFPS 38), `http://fhiso.org/files/cfp/cfps38.pdf`

[7] International Organization for Standardization, 2004, *Data elements and interchange formats — Information interchange — Representation of dates and times* (ISO 8601:2004)

[8] World Wide Web Consortium, 2004, *XML Schema Part 2: Datatypes (Second Edition)*, `http://www.w3.org/TR/xmlschema-2/`

[9] Text Encoding Initiative Consortium, 2013, *TEI P5: Guidelines for Electronic Text Encoding and Interchange*, `http://www.tei-c.org/Guidelines/P5/`