

# THE FOLK DATE PATTERN

Justin Seliger and Degory Valentine  
Brigham Young University

Relevant topics: human-computer interfaces, date authorities

## ABSTRACT

Gathering dates and date ranges from genealogical records can be difficult, especially when trying to preserve the context in which the date is found in the record. Specialized formats and varying degrees of precision complicate date entry and searching. This presentation will introduce a new pattern for handling dates that aims at unifying people's interaction with date and time entry while preserving the context of the record.

## INTRODUCTION

Current patterns for abstracting dates in digitized records transform the user's intention into a precise date by constraining the input (e.g. popup calendars) or by constraining the precision (e.g. only accepting a year). However, our genealogical records contain dates expressed in a variety of unconstrained ways. Once these dates have been transformed, any information contained in how the date was expressed is lost.

This presentation introduces a new pattern for handling dates so the original expression of the date is preserved. It also describes a possible method of handling user interaction with the pattern, a comparison to common date patterns in use today, and some discussion of obvious challenges to implementing this pattern.

For convenience of this presentation, the SQL term "datetime" represents a date with high precision in a sortable format such as "1900-01-01 00:00:00", although the actual storage method and format is irrelevant to this presentation. The term "widget" refers to something users interact with to input data that is not coupled with the storage, e.g. popup calendars, slide bars, or radio buttons.

## OVERVIEW OF PATTERN AND WIDGET

The primary idea of the pattern is to store the raw user input along with the datetimes they represent. Every time the date is displayed, the raw input is used. Essentially, the widget will be interpreting two sets of dates, one that users read and use and a corresponding value that the computer understands. Every time the date needs to be searched or sorted, the datetimes are used. Users never need to worry about datetimes, and the script never needs to worry about the folksy way users describe dates (the "folk dates").

Since people describe dates with varying levels of precision, each folk date should be parsed into two datetimes. Additionally, people specify dates with varying formats for different reasons. Folk dates may include local or period-specific formats. This is where the expression of the date becomes especially meaningful. The following two

tables show folk-date-to-datetime conversions. Table 1 contains examples of simple folk dates that only reference a date or date range. Table 2 shows more expressive examples.

Folk Date	Starting Datetime	Ending Datetime
throughout January	2011-01-01 00:00:00	2011-01-31 23:59:59
8 Aug to 12 Aug 2008	2008-08-08 00:00:00	2008-08-12 23:59:59
August through October	2010-08-01 00:00:00	2010-10-31 23:59:59
8 Aug 08 8:30pm	2008-08-08 08:30:00	2008-08-08 08:30:59

Table 1: simple folk dates

Folk Date	Starting Datetime	Ending Datetime
Christmas 2009	2009-12-25 00:00:00	2009-12-25 23:59:59
2 Richard III	1484-01-01 00:00:00	1484-12-11 23:59:59
1732 – c. 1799	1732-01-01 00:00:00	1799-12-31 23:59:59
Summer of 69	1969-06-01 00:00:00	1969-08-31 23:59:59

Table 2: expressive folk dates

The primary idea of the widget is to allow users to express the date without constraints. Datetimes should be parsed from the unconstrained input. The most available technology in which this can be implemented is the basic HTML text-input element (`<input type="text" />`), so this presentation will describe the widget from that platform.

A user types the date as they wish it to appear. As they type, their input is sent asynchronously by JavaScript to a web service that parses the input and returns applicable datetimes. The parsed datetimes are displayed to the user so they can understand how their date is interpreted. If the parsed dates are incorrect or the input cannot be recognized, the user should be able to manually specify start and end dates. The widget might look like Figure 1. In Figure 1.1 the user begins typing a date range. Since only a single day is parsed, the widget displays the same date twice (starting and ending on August 8<sup>th</sup>). In Figure 1.2 the user has completed typing the date range and both the starting and ending dates are recognized by the widget. In Figure 1.3 the user interacts with the starting date to demonstrate that the dates can be manually adjusted (possibly using some more conventional widget).

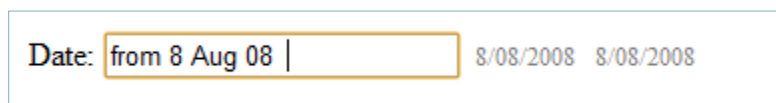


Figure 1.1: folk date widget with single date

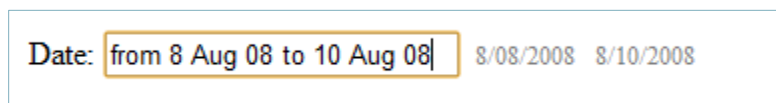


Figure 1.2: folk date widget with date range

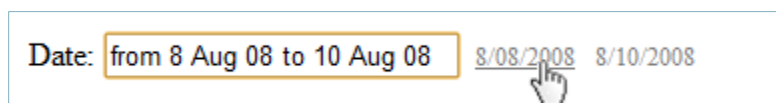


Figure 1.3: users can manually change datetimes as needed

## COMPARISON TO OTHER PATTERNS AND WIDGETS

The appearance of the widget as described above doesn't look very radical, but the actual user interaction and conceptual model is significantly different. Consequently, a comparison to other widgets as a group is meaningful.

Advantages:

- Text is a familiar and standard input type.
- Dates are formatted exactly as the user wants.
- Dates can be displayed as they appear in records.
- Specialized and localized formats or mixed formats become sortable and searchable.
- The widget doesn't restrict user input format.
- More information can be captured using expressive date formats.

Disadvantages:

- Dates may have inconsistent presentations - i.e. "from Jan to Feb" and "between Jan and Feb".
- Relative dates will lose context - i.e. "from last Friday to this Friday" when viewed next month.
- The widget may confuse first-time users by not restricting the format of their input. They might ask, "How do I *need* to write the date?"
- The widget doesn't offer any assistance to users such as an interactive popup calendar.

## CHALLENGES TO IMPLEMENTATION

People can express dates ambiguously, but computers can only understand dates deterministically. Sometime a date cannot be understood without additional context. For example, middle endian and little endian date forms can't be distinguished between the 1st and the 12th of each month. The date expressed as "6/10/2010" is June 10th in middle endian (America), but October 6th in little endian (Europe). A possible solution to this challenge is to allow the widget to make intelligent guesses (like using locale for the previous example). Another solution is to use defaults set by administrators for ambiguous cases. Alternatively, formats can be restricted where possible, e.g. if all users will be in America, only support middle endian form.

People can express dates imprecisely. Some ways to express date ranges cannot have precise datetimes, e.g. "circa 1914". This challenge might be solved by intelligent guessing or through allowing users to manually specify dates.

Accepting every date format may not be practical, but explaining every accepted date format is certainly not. Implementing the folk date parser with a plugin architecture may solve this problem. Each plugin could be responsible for handling a related set of formats. The plugin would be responsible for validating and parsing the folk date, providing an explanation of what formats it handles, and requesting and storing any required configuration data for use in parsing folk dates.

## OTHER USES

Although this pattern has been described using dates, it can apply more generally to any scalar value with a variety of units. Using this pattern, distances can be accepted as feet, meters, inches, or kilometers and still be sorted without losing their common unit expression.

Additionally, the pattern can apply to any numerical range that may benefit from additional descriptors. Some examples include "children ages 8 to 18" and "longboards over 45 centimeters".

The idea and model of The Folk Date Pattern can be applied to a large variety of problems of a similar nature. The concept of separate data for the users and the computer is the next logical step in user interface. The intuitive nature of this type of widget can be used to replace checkbox or radio-button driven user given data. For example, marital status and 'how you heard about us' fields. Finding a balance between intuitive user interfaces and precision is a challenge; the Folk Date Pattern Widget is a step forward.

## RELATED WORK

At the 2009 Family History Technology Workshop, the paper "Implementing a Surname-Study Website with Drupal" [1] described a solution to managing many genealogical record abstracts using Druapl, a content-management system. The paper focused on the efforts of one surname-study website in particular, Kendall Ancestry [2]. Continuing work on this project led to the development of the Folk Date pattern. Many records with dates in regnal format [3] (e.g. 2 Richard III) needed to be abstracted and sortable. Using Drupal's CCK module and the "Computed Field" plugin, record abstracts in Kendall Ancestry implement the Folk Date pattern simply. The record abstracts can accept a "display date" as well as two datetimes. When the record is sorted, the datetimes are always compared, but when the record is displayed the datetimes are hidden unless no "display date" was provided. This implementation lacks the abstraction of having a single input field, but maintains the power of preserved date expressions.

Currently there is no implementation of the widget or plugin architecture as described in this paper. I encourage any interested group or individual to make this pattern a reality. Drupal is a great platform to begin with – a Folk Date field would work very well as a module of the Content Construction Kit.

## CONCLUSION

With the Folk Date pattern, people's naturally expressive way of describing dates is preserved without sacrificing the power of computable precision.

---

## REFERENCES

[1] Valentine, Degory, Christophe Girraud-Carrier and Denise Mortorff. (2009). Implementing a Surname-Study Website with Drupal. In *Proceedings of the Ninth Family History Technology Workshop*.

[2] Kendall Ancestry. Online at <http://kendallancestry.com>.

[3] For more information on Regnal years, see [http://en.wikipedia.org/wiki/Regnal\\_year](http://en.wikipedia.org/wiki/Regnal_year).