iCalendar parser

Ondrej Mosnáček, Lenka Kuníková and Ľubomír Obrátil

11. novembra 2015

Format

We implemented a parser for the iCalendar format, as specified in RFC 5545 (https://tools.ietf.org/html/rfc5545).

Limitations

- The specification defines default encoding for iCalendar files to be UTF-8 (section 3.1.4). Our parser validates UTF-8 encoding of the file, however all strings in the resulting in-memory structure are left UTF-8 encoded.
- The parser does not validate URI references (section 3.3.13) and calendar user addresses (section 3.3.3).
- In many places, the specification allows the occurence of extension components/properties/property parameters. Our parser rejects these entities (otherwise there wouldn't be much to validate about the format).
- The format type tag (section 3.2.8) is validated only for correct format (type-name/subtype-name) but not for valid names.
- The language tag (section 3.2.10) is not validated.
- The value of Method property (section 3.7.2) is not checked, only the general IANA token syntax is validated.
- When parsing time-related properties we do not check the consistency of types of different properties (e. g. DTSTART vs DTEND/RRULE/...) nor the consistency of UTC/local time format. Also, we do not check if date-time values are one after the other where the specification demands it.
- It is not verified that a VTIMEZONE (section 3.6.5) component exists for each timezone identifier reffered to in the document.
- The caret-escaping mechanism specified in RFC 6868 is not implemented.
- The parser works in two stages, where in the first stage the file/stream is initially parsed into a generic tree structure, which is then further processed and validated into the final object representation. Therefore it may load the entire file/stream into memory before rejecting the invalid syntax.

Result

- Language: C++11, no libraries (except of standard library).
- Parser: recursive descent; two stages:
  1. General tree structure is parsed.
  2. The tree is processed into the final object representation.
- About 10 000 lines of code.
- The format turned out to be more complicated than expected.
- We didn't have much time for testing/documentation :(
- Teamwork is difficult when you have only < 3 weeks...