

CIS*2750 Assignment 1

Module 2

Overview

The purpose of the parser we are building in A1 is to extract the major components from a calendar file, and fill in our data structures, which will be used in later assignments. Make sure that, in addition to the iCalendar specification, you also carefully read the comments in [CalendarParser.h](#) - each data structure includes detailed comments indicating the constraints for various components.

We will not be performing full validation of a file we are reading. Instead, we want to make sure that the file conforms to the iCalendar spec sufficiently for us to fill in our data structures. The data structures reflect the format specification. The required components are given their own variables, while all the optional components are stored in lists.

The errors below reflect the various ways in which a file can fail to comply with the iCalendar specification. Please note that in Assignment 1 we are only validating the overall "shape" on most properties (unless specified otherwise). Therefore, as long as the file does not contain one of the errors listed below, it is considered valid for A1. In A2, we will implement much stricter validation in a separate function.

You can assume that the input file does not contain any components (RFC5545 Sect. 3.6) other than Events and Alarms (i.e. no to-do's, time zones, etc.).

Errors

```
ICalErrorCode createCalendar(char* fileName, Calendar** obj);
```

- **INV_FILE** is returned if there's a problem with the **fileName** argument or the file itself:
 - it is null
 - it is an empty string
 - file doesn't exist or cannot be opened
 - file doesn't have the .ics extension
 - file has invalid line endings
- **INV_CAL** is returned if the file can be opened and read, but the calendar object contained in the file is invalid
 - invalid or missing opening/closing tags
 - missing required properties or components, i.e. version, prodID, or event
 - If a required property or component is present, but is invalid for some reason, you must return one of the errors described below
 - If the error is contained in one of the events in the calendar, you return one of the appropriate errors (**INV_EVENT**, **INV_ALARM**, or **INV_DT** - see below)
- **INV_VER** is returned if the calendar version property is present, but malformed (i.e. property value is missing or is not a number).
- **DUP_VER** is returned if the calendar version property is present and appears more than once
- **INV_PRODID** is returned if the product ID property is present, but malformed (i.e. property value is missing).
- **DUP_PRODID** is returned if the product ID property appears more than once
- **INV_EVENT** is returned if the event component is present, but malformed in some way
 - missing closing tag
 - missing required properties
 - UID is present, but malformed (missing value)

- If one of the required DateTime properties is present but malformed, you must return `INV_DT` (see below)
- If an Alarm is present, but malformed, you must return `INV_ALARM` (see below)
- `INV_DT` is returned if the event creation and start date-time properties are present, but one (or both) of them is malformed in some way - i.e. do not conform to the FORM #1 or FORM #2 in Section 3.3.5 of the RFC5545. For the purposes of our course, we will consider FORM #3 to be invalid.
- `INV_ALARM` is returned if the alarm component is present, but malformed in some way
 - missing or malformed required properties. For the purposes of our assignments, an alarm must have an action and a trigger
 - missing closing tag
- `OTHER_ERROR` is returned if some other, non-calendar error happens (e.g. malloc returns NULL).

Notes:

- Always return the first errors you encounter
- If the name of a property (e.g. Event, Alarm, or DateTime) is misspelled, it counts as missing
- Any property with a missing "description" (parameters/values) is invalid, and makes its nearest enclosing component invalid.
 - E.g. if a file has an event with a malformed UID property (no value), this event is invalid and `INV_EVENT` is returned