* 1. **Description**

ISO 8601 is an international standard for representing dates, times, and time intervals in a structured and unambiguous format.

It defines various formats for date and time representation and provides guidelines for their usage.

1. basic
2. extended

ISO 8601 helps ensure consistency and interoperability in date and time-related data exchange across different systems and applications.

Applications where ISO 8601 is used:

1. Software Applications
2. Internet protocol
3. Business and industry
4. Aviation, flight traffic control system
5. Military services

*This document specifically addresses the validation of date and time strings in the extended format as outlined below.*

// Extended Format

// Date and Time in UTC YYYY-MM-DDTHH:MM:DDZ

// Date and Time in with the offset YYYY-MM-DDThh:mm:ss±hh:mm

* 1. **Assumptions and Challenges**

1. ISO 8601 offers several date and time formats. Understanding and testing each combination thoroughly is essential. For now, I am focusing on validating the extended format described earlier, leaving other formats for future consideration.

/\* TODO: Add functionality to include

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Date formats:

\* Basic format Extended format Explanation

\* YYYYMMDD YYYY-MM-DD Complete calendar date

\* ±YYYYYYDDD ±YYYYYY-DDD Expanded ordinal date with two digits added

\* YYYYWww YYYY-Www Week date with precision reduced to week

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Time of day formats:

\* Basic format Extended format Explanation

\* hhmmss hh:mm:ss Complete local time - DONE

\* hhmm,mZ hh:mm,mZ Reduced precision UTC of day with one digit decimal fraction for minutes

\* hhmm±hhmm hh:mm±hh:mm Local time and the difference from UTC — reduced accuracy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Date and time of day formats:

\* Basic format Extended format Explanation

\* YYYYDDDThhmm YYYY-DDDThh-mm Complete ordinal date — reduced precision time of day

\* YYYYMMDDhhmm,m YYYY-MM-DDhh:mm,m Complete calendar date — reduced precision time of day with one digit decimal fraction for minute — no time designator

\* YYYYWwwDThh,hhZ YYYY-Www-DThh,hhZ Complete week date — reduced precision UTC of day with two digit decimal fraction for the hour

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

bool is\_valid\_expanded\_iso8601\_datetime(const char\* dateTime);

// TODO: API used to validate basic format of ISO8601 date time.

// Extended Format

// Date and Time in UTC YYYYMMDDTHHMMDDZ

// Date and Time in with the offset YYYYMMDDThhmmss±hhmm

bool is\_valid\_basic\_iso8601\_datetime\_format(const char\* dateTime);

// TODO: API to implement time interval representation.

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Basic format Extended format

\* YYYYMMDDThhmm,mZ YYYY-MM-DDThh:mm,mZ

\* YYYYMMDDThhmm,m

\* YYYY-MM-DDThh:mm,m

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Basic format Extended format

\* PnnnD N/A

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Basic format Extended format

\* PTnn,nnH N/A

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

bool is\_valid\_iso8601\_datetime\_interval(const char\* dateTime);

// TODO: API to implement Recurring time interval representation.

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Basic format Extended format

\* Rn/YYYYWww/YYYYWww Rn/YYYY-Www/YYYY-Www

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Basic format Extended format

\* Rn/YYYYDDD/PnnDRn/YYYY-DDD/PnnD Rn/YYYY-DDD/PnnD

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

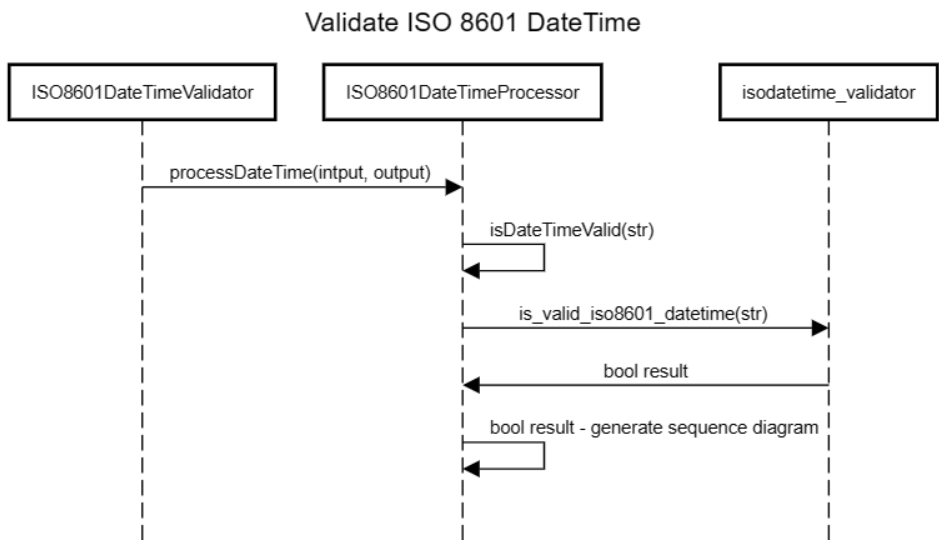
bool is\_valid\_iso8601\_recurring\_time\_interval(const char\* dateTime);

1. Testing performed on a file containing maximum of 10000 records. Validating them all is challenging.
2. If time permit can enhance to implement better logging system.
   1. **Functional description**

List of examples:

|  |  |
| --- | --- |
| Valid Format | Invalid Format |
| 2024-04-21T12:34:56Z  2021-10-21T12:34:56+00:00  1984-11-21T12:34:56-00:00  1994-07-21T23:59:59Z  2004-08-21T23:59:59+19:00  9999-12-21T23:59:59-23:59 | 2024-04-41T12:34:56-05:00  0000-00-00T00:00:00-00:00  9999-99-99T99:99:99-99:99  2024-04-21T12:34:56+99:99 |
|  | Invalid Length  2024-04-21T12:34:56Z000  2024-04-21T12:34:Z  2024-04-21T  2024-04  2024-04-21Z  2024  Invalid TZD  2024-04-21T12:34:56±0500  2024-04-21T12:34:56+05:00Z  2024-04-21T12:34:56Z05:00  InvalidSeparator  2024/04/21T12:34:56Z  2024-04-21T12-34-56Z  InvalidYearFormat  2-04-21T12:34:56Z  24-04-21T12:34:56Z  20244-04-21T12:34:56Z  InvalidMonth  2024-13-21T12:34:56Z  2024-00-21T12:34:56Z  InvalidDay  2024-04-32T12:34:56Z  2024-02-30T12:34:56Z  InvalidHour  2024-04-21T25:34:56Z  2024-04-21T12:60:56Z  2024-04-21T24:07:09Z  InvalidMinute  2024-04-21T12:34:60Z  2024-04-21T12:34:Z  2024-04-21T23:60:09Z  InvalidSecond  2024-04-21T12:34:56±05:00  2024-04-21T12:34:56:Z  2024-04-21T23:07:60Z  InvalidHourOffset  2024-04-21T25:34:56+24:03  2024-04-21T12:60:56-91:23  2024-04-21T24:07:09Z-100:43  InvalidMinuteOffset  2024-04-21T12:34:60+12:60  2024-04-21T12:34:12-11:123  2024-04-21T23:60:09+24:60 |
|  |
|  |

* 1. **Detailed design**



* 1. **Test plan**

**Environment**: debug

**Framework**: google test

**Test cases:**

1. ValidISO8601DateTime
2. InvalidISO8601DateTime
3. InvalidSeparator
4. InvalidYearFormat
5. InvalidMonth
6. InvalidDay
7. InvalidHour
8. InvalidMinute
9. InvalidSecond
10. InvalidHourOffset
11. InvalidMinuteOffset
12. InvalidTimezoneDesignator
13. InvalidLength
14. ReadValidDateTimeSetContaining10000Records
15. ReadInValidDateTimeSetContaining2901Records

**Test Results:**

