

DateParser

DOI [10.5281/zenodo.803399](https://doi.org/10.5281/zenodo.803399)

[Last release](#) -- [Master source code](#)

[Main page](#) ([versión en español](#)) -- [Code analysis](#) -- [NuGet Package](#) -- [Video](#)

Introduction

After adding a reference to the FlexibleParser namespace, it is possible to start using DateParser right away. The main functionalities of this library can be divided in the following two groups:

- *DateTime* type (C#) enhancements: better string parsing, easier linkage to time zone information and easier modification of constituent elements.
- Time zones: relevant amount of additional information and more user-friendly usage.

All the public classes of DateParser have some common features meant to maximise their usability and compatibility. For example, implicit conversions to multiple types or custom *ToString()* versions outputting the most adequate information.

```
//dateP.Value is identical to DateTime.Parse("01-01-2001").
DateP dateP = new DateP("01-01-2001");
dateP = "01-01-2001";

//dateP.Value has the same date than DateTime.ParseExact("02-01-2001", "dd-MM-yyyy",
//CultureInfo.CurrentCulture) and the current time.
dateP = new DateP("02-01-2001", new CustomDateTimeFormat("day-month-year"));

//timeZoneWindows.TimeZoneInfo is identical to
//TimeZoneInfo.FindSystemTimeZoneById("E. Europe Standard Time").
//Note that TimeZoneInfo.FindSystemTimeZoneById("E Europe Standard Time") triggers an
//exception.
TimeZoneWindows timeZoneWindows = new
TimeZoneWindows(TimeZoneWindowsEnum.E_Europe_Standard_Time);
timeZoneWindows = TimeZoneWindowsEnum.E_Europe_Standard_Time;
```

DateP

DateP is the main class of DateParser and improves *DateTime* in the following ways:

- It enables new alternatives to extract the date/time information from strings via *CustomDateTimeFormat*.
- It allows real-time modifications of the *DateTime* constituent parts, including the offset of the associated time zone.

```
string[] inputs = new string[]
```

```

{
    "04-05-2017", "4-may-2017", "04/5-2017", "04 05 2017"
};

foreach (string input in inputs)
{
    DateP dateP = new DateP
    (
        input, new CustomDateTimeFormat
        (
            new DateTimeParts[]
            {
                DateTimeParts.Day, DateTimeParts.Month, DateTimeParts.Year
            }
        ),
        0m
    );

    //dateP.Value has always the same date than DateTime.ParseExact("04-05-2017", "dd-
    //MM-yyyy", CultureInfo.CurrentCulture) and the current time.
    //Note that DateTime.ParseExact requires specific arguments to deal with each input
    //string.
}

//The date of dateP.Value has become 05/05/2017, the first Friday after 04/05/2017.
dateP.Week = DayOfWeek.Friday;

//The time of dateP.Value is now 3 hours later, the lag between the new offset and the
//original one.
dateP.TimeZoneOffset = 3m;

```

Time Zones

DateParser supports 6 different types of time zones, each of them is defined by a main class and an enum:

- [TimeZoneOfficial/TimeZoneOfficialEnum](#).
- [TimeZoneIANA/TimeZoneIANAEnum](#).
- [TimeZoneConventional/TimeZoneConventionalEnum](#).
- [TimeZoneUTC/TimeZoneUTCEnum](#).
- [TimeZoneWindows/TimeZoneWindowsEnum](#).
- [TimeZoneMilitary/TimeZoneMilitaryEnum](#).

There are also two other classes dealing with various time zones at the same time:

- [TimeZones](#).
- [TimeZonesCountry](#).

```

//All the information about the CET time zone.
TimeZoneOfficial timeZoneOfficial = "CET";

//Time zones of all the types having something in common with the CET time zone.
TimeZones timezones = new TimeZones(timeZoneOfficial);

//List with all the pairs of official standard/daylight time zones used in Ponferrada's
//country (i.e., Spain).
TimeZonesCountry timeZonesCountry = new TimeZonesCountry("Ponferrada");

```

Further Code Samples

The [test application](#) includes a relevant number of descriptive code samples.

Authorship & Copyright

I, Alvaro Carballo Garcia (varocarbas), am the sole author of each single bit of this code.

Equivalently to what happens with all my other online contributions, this code can be considered public domain. For more information about my copyright/authorship attribution ideas, visit the corresponding pages of my sites:

- <https://customsolvers.com/copyright/>
ES: https://customsolvers.com/copyright_es/
- <https://varocarbas.com/copyright/>
ES: https://varocarbas.com/copyright_es/