Howard E. Hinnant
2016-07-04

`islamic`

Contents

Introduction

This is an Islamic calendar in the style of date.h. Everthing is the same here as for date.h, except that the calendrical arithmetic implements a proleptic Tabular Islamic calendar.

The `islamic` calendar can interoperate with date.h and tz.h just by paying attention to the namespace. For example to convert the *civil* date `2016_y/jul/4` to an islamic date, just do:

```
#include "islamic.h"
#include <iostream>

int
main()
{
    using namespace date::literals;
    std::cout << islamic::year_month_day{2016_y/jul/4} << '\n';
}
```

This outputs:

```
1437-09-28
```

And here is the reverse conversion:

```
#include "islamic.h"
#include <iostream>

int
main()
{
    using namespace islamic::literals;
    std::cout << date::year_month_day{1437_y/9/28} << '\n';
}
```

Which outputs:

```
2016-07-04
```

You can even convert directly to the ISO-week-based calendar:

```
#include "islamic.h"
#include "iso_week.h"
#include <iostream>
```

```
int
main()
{
    using namespace islamic::literals;
    std::cout << iso_week::year_weeknum_weekday{2016_y/jun/13} << '\n';
}
```

Which outputs:

```
2016-W27-Mon
```

You can find the current local islamic date and time with:

```
#include "islamic.h"
#include "tz.h"
#include <iostream>

int
main()
{
    auto zt = date::make_zoned(date::current_zone(), std::chrono::system_clock::now());
    auto ld = date::floor<date::days>(zt.get_local_time());
    islamic::year_month_day ymd{ld};
    auto time = date::make_time(zt.get_local_time() - ld);
    std::cout << ymd << ' ' << time << '\n';
}
```

Example output:

```
1437-09-28 16:24:56.578240
```

This calendar assumes that the Islamic day starts at midnight, like the civil calendar. This is because the only thing that is customized to the Islamic calendar is the calendar itself (days precision time keeping). For time-keeping finer than days, the `<chrono>` library is still in use. The Islamic calendar simply converts to and from `sys_days` (a days-precision `std::chrono::time_point`) like every other calendar.