

Writing An RFC-Date Parser With Spirit

Peter Simons

teisor v v

simons@computer.org

What this is all about ...

The Spirit Parser Framework is an incredibly flexible and powerful tool for writing any kind of text processor, parser, or converter. It is

```
| "CST" | "CDT"  
| "MST" | "MDT"  
| "PST" | "PDT"  
| 1ALPHA  
| ( ( "+" | "-" ) 4DIGIT )
```

At rst

```
int tm_min;           /* minutes */
int tm_hour;          /* hours */
int tm_mday;          /* day of the month */
int tm_mon;           /* month */
int tm_year;          /* year */
int tm_wday;          /* day of the week */
int tm_yday;          /* day in the year */
int tm_isdst;         /* daylight saving time */
};
```

Using `mktime(3)`, for instance, this structure can be converted into a `time_t`, using `strftime(3)`, it can be formatted into an (almost) arbitrary text format. So this is a good way to represent a date --- and it is easy to pull out by the parser. There is only one problem: The `tm` structure, as defined by the POSIX standard, does not contain a structure, good for 1860

After he told me'er

```
                [
                    uint_p
                ]
            ),
    time      = uint_p
               >> ':'
               >> uint_p
               >> !(
                    ':'
                    >> uint_p
                ),
    zone      = ch_p('+') >> uint4_p
               | ch_p('-') >> uint4_p
               | lexeme_d
               [
                   nocase_d
                   [
                       timezone_p
                   ]
               ]
    );
}
const
```

