

The Name of the Day and other Date Operations with Packages `scrdate`

Markus Kohm

Version v3.50 of 2026-06-08*

The package `scrdate` defines some macros to handle the name of the day and the date in ISO format.

Up to version 3.49.2 it was part of the KOMA-Script collection. Since version 3.50 it is a standalone package.

Contents

1 User Manual	1
2 Implementation	4
Index	10
Change History	11

1 User Manual

Originally, the `scrdate` package could only give the day of the week for the current date. Nowadays, it offers this and more for any date in the Gregorian calendar.

`\CenturyPart`
`\DecadePart` The command `\CenturyPart{⟨year⟩}` returns the value of the century digits—thousands and hundreds—of a `⟨year⟩`. The command `\DecadePart{⟨year⟩}` command, on the other hand, gives the value of the remaining digits, i.e. the tens and the units. The `⟨year⟩` can have any number of digits. You can assign the value directly to a counter or use it for calculations with `\numexpr... \relax` or `\inteval`. To output it as an Arabic number, you should prefix it with `\the`.

Example: You want to calculate and print the century of the current year.

```
The year \the\year\ is year \the\DecadePart{\year} of the
\engordnumber{\numexpr \CenturyPart{\year}+1\relax} century.
```

The result would be:

*Monday.

The year 2026 is year 26 of the 21st century.

This example uses the **engord** package.

Note that the counting method used here treats the year 2000 as year 0 — and therefore the first year — of the 21st century. If necessary, however, you can make a correction with `\numexpr` or `\interval`, as shown for the ordinal number in the example.

`\DayNumber`
`\ISODayNumber`

These two commands return the value of the day-of-the-week number for any date. They differ only in the method of specifying the date. While the `\DayNumber{<year>}{<month>}{<day>}` command requires the `<year>`, `<month>`, and `<day>` as separate parameters, the `\ISODayNumber{<ISO date>}` command expects an `<ISO date>` as a single argument, `<ISO date>`, using the ISO notation `<year>-<month>-<day>`. It does not matter if the `<month>` or `<day>` have one or two digits. You can use the result of both commands to assign directly to a counter or for calculations using `\numexpr... \relax` or `\interval`. To print it as an Arabic number, you should prefix it with `\the`.

Example: You want to know the number of the day of the week of the 1st May 2027.

The 1st~May~2027 has \the\ISODayNumber{2027-5-1}
as the number of the day of the week.

The result will be:

The 1st May 2027 has 6 as the number of the day of the week.

The days of the week are numbered as follows: Sunday = 0, Monday = 1, Tuesday = 2, Wednesday = 3, Thursday = 4, Friday = 5, and Saturday = 6.

`\DayNameByNumber`
`\DayName`
`\ISODayName`

Usually you are less interested in the number of the day of the week than in its name. Therefore, the command `\DayNameByNumber{<number of the day of the week>}` returns the name of the day of the week corresponding to a `<number of the day of the week>`. This number can be the result, for example, of `\DayNumber` or `\ISODayNumber`. The two commands `\DayName{<year>}{<month>}{<day>}` and `\ISODayName{<ISO date>}` directly return the name of the day of the week of a given date.

Example: You want to know the name of the day of the week of the 24th December 2027.

Please pay by \ISODayName{2027-12-24},
24th~December~2027 the amount of \dots.

The result will be:

Please pay by Friday, 24th December 2027 the amount of

Once again, it is particularly worth noting that you can perform calculations, to a certain extent:

Example: You want to know the names of the day of the week 12 days from now and 24 days before the 24th December 2027.

In 12~days, the name of the day of the week will be
`\DayNameByNumber{\DayNumber{\year}{\month}{\day}+12}`, and
24~days before the 24th~December~2027 it will be
`\DayNameByNumber{\ISODayNumber{2027-12-24}-24}`, while two
weeks and three days after a Wednesday will be a
`\DayNameByNumber{3*7+3}`.

The result could be, for example:

In 12 days, the name of the day of the week will be Saturday, and 24 days before the 24th December 2027 it will be Tuesday, while two weeks and three days after a Wednesday will be a Saturday.

`\ISOToday` In the previous examples, the current date was always specified clumsily using the
`\IsoToday` $\TeX$ registers `\year`, `\month`, and `\day`. The `\ISOToday` and `\IsoToday` commands directly
`\todayname` return the current date in ISO-notation. These commands differ only in the fact that
`\todaynumber` `\ISOToday` always outputs a two-digit month and day, while `\IsoToday` outputs single-digit
numbers for values less than 10. The `\todayname` command directly returns the name of
the current day of the week, while `\todaynumber` returns the number of the current day
of the week. You can find more information about using this value in the explanations of
`\DayNumber` and `\ISODayNumber` above.

Example: I want to show you on what day of the week this document was typeset:

`This document was created on a \todayname.`

This will result, for example, in:

`This document was created on a Monday.`

For languages that have a case system for nouns, note that the package cannot decline words. The terms are given in the form appropriate for displaying a date in a letter, which is the nominative singular for the currently supported languages. Given this limitation, the example above will not work correctly if translated directly into some other languages.

The names of the weekdays in `scrdate` all have initial capital letters. If you need the names completely in lower case, for example because that is the convention in the relevant language, simply wrap the command with the $\LaTeX$ `\MakeLowercase` command:

`\MakeLowercase{\todayname}`

This converts the whole argument into lower-case letters. Of course, you can also do this for `\DayNameByNumber`, `\DayName`, and `\ISODayName` commands described above.

`\nameday` Just as you can directly modify the output of `\today` with `\date`, so you can change the output of `\todayname` to `\name` with `\nameday{\name}`.

Example: You change the current date to a fixed value using `\date`. You are not interested in the actual name of the day, but want only to show that it is a workday. So you write:

`\nameday{workday}`

After this, the previous example will result in:

`This document was created on a workday.`

There's no corresponding command to change the result of `\ISOToday` or `\IsoToday`.

`\newdaylanguage` Currently the `scrdate` package recognizes the following languages:

- Croatian (`croatian`),

- Czech (`czech`),
- Danish (`danish`),
- Dutch (`dutch`),
- English (`american`, `australian`, `british`, `canadian`, `english`, `UKenglish`, and `USenglish`),
- Finnish (`finnish`),
- French (`french`),
- German (`austrian`, `german`, `naustrian`, `ngerman`, `nswissgerman`, and `swissgerman`),
- Italian (`italian`),
- Norwegian (`norsk`),
- Polish (`polish`),
- Slovak (`slovak`),
- Spanish (`spanish`),
- Swedish (`swedish`).

You can also configure it for additional languages. To do so, the first argument of `\newdaylanguage{<language>}{<Monday>}{<Tuesday>}{<Wednesday>}{<Thursday>}{<Friday>}{<Saturday>}{<Sunday>}` is the name of the language, and the other arguments are the names of the corresponding days of the week.

In the current implementation, it does not matter whether you load `scrdate` before or after `ngerman`, `babel`, or similar packages. In each case the correct language will be used provided it is supported.

To be more precise, as long as you select a language in a way that is compatible with `babel`, `scrdate` will use the correct language. If this is not the case, you will get (US) English names.

Of course, if you create definitions for a language that was previously unsupported, please mail them to the author of `scrdate` or add a ticket to the issue tracker. There is a good chance that future versions of `scrdate` will add support for that language.

2 Implementation

```
1 <*package>
```

With version 3.05a `scrdate` was rewritten. First step was to make more macros full expandable to provide `\MakeUppercase` and `\MakeLowercase`. Second was to extend the user interface by some new functionality.

Currently we need `\XdivY` and `\XmodY` of package `scrbase`. But in future we will switch over to L3.

```
2 \RequirePackage{scrbase}
```

`\CenturyPart` This is the century part of a year number and so only a shortcut to `\XdivY{...}{100}`, that is defined at `scrbase`.

```
3 \newcommand*{\CenturyPart}[1]{\XdivY{#1}{100}}
```

`\DecadePart` This is the year number without the century part and therefore only a shortcut to `\XmodY{...}{100}`, that is defined at `scrbase`.

```
4 \newcommand*\DecadePart[1]{\XmodY{#1}{100}}
```

`\@GaussYear` (*intern*) At the Gauss calculation of the day of the week January and February relates to the year before. This macro does the correction for any date.

```
5 \newcommand*\@GaussYear[3]{%
6   \ifcase #2
7     \PackageError{scrdate}{month out of range}{%
8       You've asked for the Gauss year of ISO date #1-#2-#3,\MessageBreak
9       this means, that month has invalid value '#2'.}%
10  \or
11    \numexpr #1 - 1\relax
12  \or
13    \numexpr #1 - 1\relax
14  \else
15    #1
16  \fi
17 }
```

`\DayNumber` Returns the numerical value of the day of week. Note, that Sunday is 0, Monday is 1, ..., Saturday is 6. We use the Gauss calculation of the day of the week. First argument is the year, second the month and last the day of the month.

```
18 \newcommand*\DayNumber[3]{%
19   \XmodY{%
20     \numexpr #3
21     + \ifcase #2
22       \PackageError{scrdate}{month out of range}{%
23         You've asked for the dayname of ISO date #1-#2-#3,\MessageBreak
24         this means, that month has invalid value '#2'.}%
25       \or 28 \or 31 \or 2 \or 5 \or 7 \or 10 \or 12 \or 15 \or 18
26       \or 20 \or 23 \or 25
27     \else
28       \PackageError{scrdate}{month out of range}{%
29         You've asked for the dayname of ISO date #1-#2-#3,\MessageBreak
30         this means, that month has invalid value '#2'.}%
31     \fi
32     + \DecadePart{\@GaussYear{#1}{#2}{#3}}
33     + \XdivY{\DecadePart{\@GaussYear{#1}{#2}{#3}}}{4}
34     + \XdivY{\CenturyPart{\@GaussYear{#1}{#2}{#3}}}{4}
35     - 2 * \CenturyPart{\@GaussYear{#1}{#2}{#3}} \relax
36   }{7}%
37 }
```

`\ISODayNumber` The same like `\DayNumber` but with ISO date argument.

```
38 \newcommand*\ISODayNumber[1]{\expandafter\@IsoDayNumber#1\@nil}
```

`\@IsoDayNumber` (*intern*)

```
39 \newcommand*\@IsoDayNumber{}
40 \def\@IsoDayNumber#1-#2-#3\@nil{\DayNumber{#1}{#2}{#3}}
```

`\DayName` Returns the name of the day of the week. Arguments like `\DayNumber`.

```
41 \newcommand*{\DayName}[3]{\@dayname{\DayNumber{#1}{#2}{#3}}}
```

`\ISODayName` The same like `\DayName` but with ISO date argument.

```
42 \newcommand*{\ISODayName}[1]{\@dayname{\ISODayNumber{#1}}}
```

`\DayNameByNumber` Returns the name of the day of the week. The argument is a number that will be transposed to the range 0...6.

```
43 \newcommand*{\DayNameByNumber}[1]{%
44   \@dayname{\XmodY{#1}{7}}%
45 }
```

`\ISOToday` Returns the ISO date.

```
46 \newcommand*{\ISOToday}{%
47   \the\year-\ifnum \month<10 0\fi\the\month-\ifnum \day<10 0\fi\the\day%
48 }
```

`\IsoToday` Returns the ISO date.

```
49 \newcommand*{\IsoToday}{%
50   \the\year-\the\month-\the\day%
51 }
```

`\todaysname` Using `\DayName` this is very, very simple.

```
52 \newcommand*{\todaysname}{\DayName{\year}{\month}{\day}}
```

`\todaysnumber` Using `\DayNumber` this is very, very simple.

```
53 \newcommand*{\todaysnumber}{\DayNumber{\year}{\month}{\day}}
```

`\nameday` We simply have to redefine `\todaysname`.

```
54 \newcommand\nameday[1]{\renewcommand*{\todaysname}{#1}}
```

`\newdaylanguage` We write a macro to define the name of the days.

`\define@languagenamewarning` (*intern*) But before this, we have to define a once only warning.

```
55 \newcommand*\scrdate@languagenamewarning{%
56   \PackageWarningNoLine{scrdate}
57     {\string\languagenamewarning space not
58     defined, using \string\language.\MessageBreak
59     This may result in use of wrong language!\MessageBreak
60     You should use a compatible language
61     package!\MessageBreak
62     (e.g. 'babel', 'german', 'french', ...)}%
63   \let\scrdate@languagenamewarning\relax
64 }
```

```
65 \newcommand\newdaylanguage[8]{%
```

First we check, if the language is defined at the format, the user uses. If it is not defined, we do not define the name of the days and warn.

```
66 \scr@ifundefinedorrelax{1@#1}{%
67   \PackageInfoNoLine{scrdate}{Language #1\space not defined.\MessageBreak
68                                     \protect\dayname@#1\space skipped}}%
```

If it is defined, we define the name-selection-macro `\dayname@<language>`. First we define the new macro `\dayname@<language>`:

```
69 }{%
70   \@namedef{dayname@#1}##1{%
71     \ifcase\numexpr \XmodY{##1}{7}\relax
72       #8\or #2\or #3\or #4\or #5\or #6\or #7\fi%
73   }%
```

Then we define, what to do at `\begin{document}`:

```
74   \AtBeginDocument{%
```

There we first have to test, if `\date<language>` is defined (e.g. using package `german`). If not, we have to warn once more.

```
75     \scr@ifundefinedorrelax{date#1}{%
76       \PackageInfoNoLine{scrdate}
77           {\protect\date#1\space not defined.\MessageBreak
78           \protect\today'sname\space probably cannot use
79           \protect\dayname@#1}}%
```

But if it is defined, we can use it

```
80     }{%
```

There we first save `\date<language>` as `\D@date<language>`.

```
81       \expandafter\let\csname D@date#1\expandafter\endcsname
82       \csname date#1\endcsname
```

Now we redefine `\date<language>`. It first defines `\@dayname` and then calls saved macro.

```
83       \@namedef{date#1}{%
84         \expandafter\let\expandafter\@dayname\csname dayname@#1\endcsname
85         \@nameuse{D@date#1}}%
```

Last we have to select this new `\date<language>`.

```
86       \@ifundefined{language#1}{%
87         \scrdate@language#warning
88         \ifnum\language=\@nameuse{1@#1}
89           \@nameuse{date#1}%
90         \fi
91       }{%
92         \@ifundefined{date\language#1}%
93           {}%
94           {\@nameuse{date\language#1}}%
95       }%
96     }%
97   }%
98 }%
99 }
```

`\@dayname` (*intern*) This should be named selecting the language. Since I changed the definitions package **german** and equal may be loaded before or after `scrdate` or not.

First we define the usual languages using `\newdaylanguage`:

`\dayname@german` (*intern*)

```
100 \newdaylanguage{german}{Montag}{Dienstag}{Mittwoch}
101      {Donnerstag}{Freitag}{Samstag}{Sonntag}
```

`\dayname@ngerman` (*intern*)

```
102 \newdaylanguage{ngerman}{Montag}{Dienstag}{Mittwoch}
103      {Donnerstag}{Freitag}{Samstag}{Sonntag}
```

`\dayname@naustrian` (*intern*)

```
104 \newdaylanguage{naustrian}{Montag}{Dienstag}{Mittwoch}
105      {Donnerstag}{Freitag}{Samstag}{Sonntag}
```

`\dayname@austrian` (*intern*)

```
106 \newdaylanguage{austrian}{Montag}{Dienstag}{Mittwoch}
107      {Donnerstag}{Freitag}{Samstag}{Sonntag}
```

`\dayname@swissgerman` (*intern*)

```
108 \newdaylanguage{swissgerman}{Montag}{Dienstag}{Mittwoch}
109      {Donnerstag}{Freitag}{Samstag}{Sonntag}
```

`\dayname@nswissgerman` (*intern*)

```
110 \newdaylanguage{nswissgerman}{Montag}{Dienstag}{Mittwoch}
111      {Donnerstag}{Freitag}{Samstag}{Sonntag}
```

`\dayname@american` (*intern*)

```
112 \newdaylanguage{american}{Monday}{Tuesday}{Wednesday}
113      {Thursday}{Friday}{Saturday}{Sunday}
```

`\dayname@australian` (*intern*)

```
114 \newdaylanguage{australian}{Monday}{Tuesday}{Wednesday}
115      {Thursday}{Friday}{Saturday}{Sunday}
```

`\dayname@british` (*intern*)

```
116 \newdaylanguage{british}{Monday}{Tuesday}{Wednesday}
117      {Thursday}{Friday}{Saturday}{Sunday}
```

`\dayname@canadian` (*intern*)

```
118 \newdaylanguage{canadian}{Monday}{Tuesday}{Wednesday}
119      {Thursday}{Friday}{Saturday}{Sunday}
```

`\dayname@english` (*intern*)

```
120 \newdaylanguage{english}{Monday}{Tuesday}{Wednesday}
121      {Thursday}{Friday}{Saturday}{Sunday}
```

`\dayname@newzealand` (*intern*)

```
122 \newdaylanguage{newzealand}{Monday}{Tuesday}{Wednesday}
123      {Thursday}{Friday}{Saturday}{Sunday}
```

`\dayname@UKenglish (intern)`

```

124 \newdaylanguage{UKenglish}{Monday}{Tuesday}{Wednesday}
125      {Thursday}{Friday}{Saturday}{Sunday}

```

`\dayname@ukenglish (intern)`

```

126 \newdaylanguage{ukenglish}{Monday}{Tuesday}{Wednesday}
127      {Thursday}{Friday}{Saturday}{Sunday}

```

`\dayname@USenglish (intern)`

```

128 \newdaylanguage{USenglish}{Monday}{Tuesday}{Wednesday}
129      {Thursday}{Friday}{Saturday}{Sunday}

```

`\dayname@usenglish (intern)`

```

130 \newdaylanguage{usenglish}{Monday}{Tuesday}{Wednesday}
131      {Thursday}{Friday}{Saturday}{Sunday}

```

`\dayname@acadian (intern)` This is a former french alias of **babel**. It will be removed from `scrdate` in future.

```

132 \newdaylanguage{acadian}{Lundi}{Mardi}{Mercredi}
133      {Jeudi}{Vendredi}{Samedi}{Dimanche}

```

`\dayname@canadien (intern)` This is a former french alias of **babel**. It will be removed from `scrdate` in future.

```

134 \newdaylanguage{canadien}{Lundi}{Mardi}{Mercredi}
135      {Jeudi}{Vendredi}{Samedi}{Dimanche}

```

`\dayname@francais (intern)` This is a former french alias of **babel**. It will be removed from `scrdate` in future.

```

136 \newdaylanguage{francais}{Lundi}{Mardi}{Mercredi}
137      {Jeudi}{Vendredi}{Samedi}{Dimanche}

```

`\dayname@french (intern)`

```

138 \newdaylanguage{french}{Lundi}{Mardi}{Mercredi}
139      {Jeudi}{Vendredi}{Samedi}{Dimanche}

```

`\dayname@italian (intern)`

```

140 \newdaylanguage{italian}{Luned\‘\i}{Marted\‘\i}{Mercoled\‘\i}
141      {Gioved\‘\i}{Venerd\‘\i}{Sabato}{Domenica}

```

`\dayname@spanish (intern)`

```

142 \newdaylanguage{spanish}{Lunes}{Martes}{Mi\‘ercoles}
143      {Jueves}{Viernes}{S\‘abado}{Domingo}

```

`\dayname@croatian (intern)`

```

144 \newdaylanguage{croatian}{Ponedjeljak}{Utorak}{Srijeda}
145      {\v{C}etvrtak}{Petak}{Subota}{Nedjelja}

```

`\dayname@dutch (intern)`

```

146 \newdaylanguage{dutch}{Maandag}{Dinsdag}{Woensdag}
147      {Donderdag}{Vrijdag}{Zaterdag}{Zondag}

```

`\dayname@finnish (intern)`

```

148 \newdaylanguage{finnish}{Maanantai}{Tiistai}{Keskiviikko}
149      {Torstai}{Perjantai}{Lauantai}{Sunnuntai}

```

```

\dayname@norsk (intern)
150 \newdaylanguage{norsk}{Mandag}{Tirsdag}{Onsdag}
151      {Torsdag}{Fredag}{L\o{}rdag}{S\o{}ndag}

\dayname@danish (intern)
152 \newdaylanguage{danish}{Mandag}{Tirsdag}{Onsdag}
153      {Torsdag}{Fredag}{L\o{}rdag}{S\o{}ndag}

\dayname@swedish (intern)
154 \newdaylanguage{swedish}{M\aa{}ndag}{Tisdag}{Onsdag}
155      {Torsdag}{Fredag}{L\"ordag}{S\"ondag}

\dayname@polish (intern)
156 \newdaylanguage{polish}{Poniedzia\l{}ek}{Wtorek}{\'Sroda}
157      {Czwartek}{Pi\ao{}tek}{Sobota}{Niedziela}

\dayname@czech (intern)
158 \newdaylanguage{czech}{Pond\v{e}l\'i}{\'Uter\'y}{St\v{r}eda}
159      {\v{C}tvrtek}{P\'atek}{Sobota}{Ned\v{e}le}

\dayname@slovak (intern)
160 \newdaylanguage{slovak}{Pondelok}{Utorok}{Streda}
161      {\v{S}tvrtok}{Piatok}{Sobota}{Nede\v{1}a}

```

If there are no language-definitions, we simply want the US-english names of the days.

```
162 \let\@dayname\dayname@english
```

Last but not least file `scrdate.cfg` has to be included, if it exists.

```

163 \InputIfFileExists{scrdate.cfg}
164 {%
165     \@ifundefined{PackageNote}{\PackageInfo}{\PackageNote}%
166     {scrdate}{Local configuration file scrdate.cfg loaded}%
167 }{}
168 </package>

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

C			
\CenturyPart	 <i>1, 3</i>	\DecadePart	 <i>1, 4</i>
Commands:		\ISODayName	 <i>2, 42</i>
\CenturyPart	 <i>1, 3</i>	\ISODayNumber	... <i>2, 38</i>
\DayName	 <i>2, 41</i>	\ISOToday	 <i>3, 46</i>
\DayNameByNumber	. <i>2, 43</i>	\IsoToday	 <i>3, 49</i>
\DayNumber	 <i>2, 18</i>	\nameday	 <i>3, 54</i>
		\newdaylanguage	. <i>3, 55</i>
		D	
		\DayName	 <i>2, 41</i>
		\DayNameByNumber	... <i>2, 43</i>
		\DayNumber	 <i>2, 18</i>
		\DecadePart	 <i>1, 4</i>
		\todaysname	 <i>3, 52</i>
		\todaysnumber	... <i>3, 53</i>

v3.08 – 2011/01/18	
\dayname@danish: New (thanks to Benjamin Hell)	10
\dayname@swedish: New (thanks to Benjamin Hell)	10
v3.08b – 2011/02/22	
\dayname@austrian: new language “austrian”	8
\dayname@naustrian: new language “naustrian”	8
v3.08b – 2011/03/31	
\newdaylanguage: one % added	7
v3.13 – 2014/01/07	
\dayname@polish: New (thanks to Blandyna Bogdol)	10
v3.13 – 2014/01/23	
\dayname@acadian: new language “acadian”	9
\dayname@american: new language “american”	8
\dayname@australian: new language “australian”	8
\dayname@british: new language “british”	8
\dayname@canadian: new language “canadian”	8
\dayname@canadien: new language “canadien”	9
\dayname@czech: New (thanks to Elke Schubert)	10
\dayname@francais: new language “francais”	9
\dayname@newzealand: new language “newzealand”	8
\dayname@nswissgerman: new language “nswissgerman”	8
\dayname@polish: Fixed (thanks to Elke Schubert)	10
\dayname@swissgerman: new language “swissgerman”	8
\dayname@UKenglish: new language “UKenglish”	9
v3.13 – 2014/02/26	
\dayname@slovak: New (thanks to Elke Schubert)	10
v3.15 – 2014/12/11	
\@dayname: default language definition fixed	10
\newdaylanguage: more robust \ifcase	7
v3.24 – 2017/05/29	
\dayname@ukenglish: new language “ukenglish”	9
\dayname@usenglish: new language “usenglish”	9
v3.36 – 2022/02/07	
General: require package scrlogo instead of defining \KOMAScript	1
switch over from scrdoc to koma-script-source-doc	1
v3.40 – 2023/04/17	
General: guide names changed	1
v3.42 – 2024/02/01	
General: \typeout replaced by either \PackageNote or \PackageInfo	10
v3.50 – 2026-05-18	
General: spin-off	1