datetime2 v1.1: date and time formats

Nicola L. C. Talbot http://www.dickimaw-books.com/

2015-09-15

The year $\langle YYYY \rangle$ can't be negative in \DTMdate or \DTMDate. Use \DTMdisplaydate or \DTMDisplaydate instead.

These commands internally use \DTMdisplaydate and \DTMDisplaydate, respectively. If the datetime2-calc package has been loaded, the day of the week will be computed, otherwise the day of the week will be set to -1. Another benefit of the datetime2-calc package is that it allows additional formats permitted by the pgfcalendar package:

- $\langle YYYY \rangle \langle MM \rangle$ -last (the last day of the given month).
- $\langle YYYY \rangle \langle MM \rangle \langle DD \rangle + \langle n \rangle$ ($\langle n \rangle$ days before the given date).
- $\langle YYYY \rangle \langle MM \rangle \langle DD \rangle + \langle n \rangle$ ($\langle n \rangle$ days after the given date).
- $\langle YYYY \rangle \langle MM \rangle$ -last+- $\langle n \rangle$ ($\langle n \rangle$ days before the last day of the given month).
- $\langle YYYY \rangle \langle MM \rangle$ -last+ $\langle n \rangle$ ($\langle n \rangle$ days after the last day of the given month).

See the pgfcalendar package for further details.

If you want to be able to use a date in an expandable context that can perform these calculations, consider first saving the date using one of the commands described in Section 4 and then use one of the expandable commands such as \DTMuse to display the date.

An error or unexpected results may occur if you try using one of these extended formats without loading the datetime2-calc package. An example that only works with datetime2-calc:

\DTMdate{2015-03-last}

An example that works with or without datetime2-calc:

\DTMdate{2015-03-31}

In this second case, you'll only notice a difference in the output if the style should show the day of the week.

The style of the date is the same as for \DTMdisplaydate and \DTMDisplaydate (which \DTMdate and \DTMDate internally use, as mentioned above).

A time can be displayed using

\DTMdisplaytime

 $\label{local_decomposition} $$ DTMdisplaytime {\langle hour \rangle} {\langle minute \rangle} {\langle sec \rangle} $$$

 $^{^1\}mbox{Well},$ actually it can if you put it in braces and don't use datetime 2-calc.

where the arguments are all numerical (using 24 hours). The *time style* currently in effect determines how the time is formatted. The command is designed to be used in an expandable context so the styles should take care to protect any fragile commands.

Note that this command doesn't display the time zone. To display the time zone, you need to use

\DTMdi spl ayzone

$\label{eq:def:def:DTMdisplayzone} $$ \operatorname{TZh} {\langle TZm \rangle} $$$

where $\langle TZh \rangle$ is the hour offset and $\langle TZm \rangle$ is the minute offset. The display is governed by the *zone style*. Again, the style should protect any fragile commands in case this is used in an expandable context.

The current time (as set at the start of the document build) can be displayed using

\DTMcurrenttime

\DTMcurrenttime

This internally just uses \DTMdisplaytime and so is designed for use in an expandable context.

The current zone can be displayed using

\DTMcurrentzone

\DTMcurrentzone

This internally just uses \DTMdisplayzone and so is designed for use in an expandable context.

If the PDFTEX primitive \pdfcreationdate is defined, the current time information is obtained from that, which includes the seconds and time zone. LuaTeX also defines this command but XaTeX doesn't, and in that case the only way to determine the current time is from TeX's \time primitive which only contains the number of minutes since midnight, which means that the seconds and time zone are unavailable. Therefore if XaTeX is used, the showseconds and showzone options are automatically switched off.

There is also a non-expandable robust command to display the time:

\DTMtime

$\DTMtime{\langle tm \rangle}$

where $\langle tm \rangle$ must be in the 24 hour format $\langle hh \rangle$: $\langle mm \rangle$: $\langle ss \rangle$ (colon-separated numerical arguments). Take care if you use babel with a language setting that

makes the colon character active. You will have to switch off the shorthands in order to use this command correctly.

The full date, time and zone (if available) can be displayed using

\DTMdi spl ay

```
\label{eq:dayof} $$ \operatorname{DTMdisplay}{\langle year\rangle}_{\langle month\rangle}_{\langle day\rangle}_{\langle day of week\rangle}_{\langle hh\rangle}_{\langle mm\rangle}_{\langle ss\rangle}_{\langle TZh\rangle}_{\langle TZm\rangle}$
```

The arguments are all numerical. The way the information is displayed in the document is governed by the *full style* (or *date-time style*). Typically the full style will redefine this command to use \DTMdisplaydate, \DTMdisplaytime and (optionally) \DTMdisplayzone. The Showzone setting may govern whether or not to display the time zone (although a style may ignore this setting). The separators between the date and time and between the time and zone are governed by the style.

There is also an analogous version if capitalisation is required:

\DTMDi spl ay

Some styles may simply make this equivalent to \DTMdisplay. Other styles may use a similar format to \DTMdisplay but replace \DTMdisplaydate with \DTMDisplaydate.

The full current date, time and (optionally) zone can be displayed using:

\DTMnow

\DTMnow

which uses \DTMdisplay or

\DTMNow

\DTMNow

which uses \DTMDisplay.

4 Storing and Using Dates and Times

Date, time and zone information can be saved for later use. Note that the information is always saved numerically. The style is only applied when the information is later used. The commands that save the information are robust and not expandable. The commands that use the data are typically expandable although there may be some exceptions. Take care that the colon (:) and hyphen (-) characters haven't had their normal category code changed. (For example, through babel's shortcuts.) In the commands below, the *(name)* (no active characters) is a name that uniquely identifies the information.

Dates are saved using

\DTMsavedate

$\DTMsavedate{\langle name \rangle} {\langle date \rangle}$

where \(\lambda t \text{ate} \) is in the same format as for \DTMdate. As with \DTMdate (and \DTMDate) the format can be extended with the datetime2-calc package. If you want to access the day of week, you must make sure that datetime2-calc has been loaded before you save the date. (Remember that the calc and showdow package options will automatically load datetime2-calc.) If datetime2-calc has been loaded, the day of week number will be calculated and saved. Whether or not it is displayed in the document when the date is later used depends on the settings when the date is displayed not when it's saved.

This command will override any previously defined date saved with this \(\lambda name \rangle\). If a time or zone hasn't been edith t(is) TJ/F85 10.0364 Tf2272.034 0 Td (h) TJ/F83 10.0364 Tf 3.5 (ill)-325((l)-1(l)-325bne)-325(set)-224toether-185(w)-1(sme)-225(th)1(ye)-226(wi)-1(ll)-205(r)19(maint)-224(u)-1(n)1c(ha)1ngsed.

T(H**okavee**fe

$\DTMsavetime{\langle name \rangle} {\langle time \rangle}$

where the $\langle time \rangle$ is in the same format as for \DTMtime.

This command will override any previously defined time saved with this $\langle name \rangle$. If a date or zone hasn't been defined with this $\langle name \rangle$, the date and zone elements will all be set to 0 (or -1 for the day of week) otherwise they will remain unchanged.

Times and zone are saved using

\DTMsavetimezn

$\verb|\DTMsavetimezn{|\langle name\rangle|}{\langle time\ and\ zone\rangle|}$

where the $\langle time \ and \ zone \rangle$ is in the form

```
\langle hh \rangle : \langle mm \rangle : \langle ss \rangle \langle TZh \rangle : \langle TZm \rangle
```

(Note the space between the seconds and the hour offset.)

This command will override any previously defined time and zone saved with this $\langle name \rangle$. If a date hasn't been defined with this $\langle name \rangle$, the year, month and day will be set to zero and the day of the week to -1 otherwise they will remain unchanged.

All date, time and zone information can be saved at the same time using:

\DTMsavetimestamp

$\texttt{DTMsavetimestamp}\{\langle name \rangle\}\{\langle data \rangle\}$

where $\langle data \rangle$ is in the format:

```
\langle YYYY\rangle - \langle MM\rangle - \langle DD\rangle T\langle hh\rangle : \langle mm\rangle : \langle ss\rangle \langle zone\rangle
```

The $\langle zone \rangle$ may either be Z or in the form $\langle TZh \rangle$: $\langle TZm \rangle$ (for example, -03:00 or -3:0). This will override any date, time or zone data previously saved with this $\langle name \rangle$.

The current date and time can be saved using:

\DTMsavenow

$\verb|\DTMsavenow{|}\langle name \rangle|$

There is also a command that can be used to save the modification date of a file, but it's not available for some T_FX engines:

\DTMsavefilemoddate

 $\DTMsavefilemoddate{\langle name \rangle} {\langle file name \rangle}$

where \(\file name \) is the name of the file (remember to use forward slashes \(\) for the directory divider). If you build your document using PDF\(\text{PDF}\(\text{LE}\(\text{X} \), this command will use the PDFT\(\text{LE}\(\text{X} \) primitive \pdffilemoddate. If you use LuaT\(\text{LE}\(\text{X} \) this command will attempt to use os.date but it uses \(\text{Z} \) for the time zone, which may not work on some operating systems. If you use X\(\text{LE}\(\text{X} \) this command will generate a warning and will assume a date of 0000-00-00T00:00:00Z.

The above commands are all localised to the current scope. If the data is required after the end of the scope, you can make the assignments global using:

\DTMmakegI obal

\DTMmakeglobal{\(\langle\)}

For example:

\DTMsavenow{mydate}\DTMmakeglobal {mydate}

A previously saved date can be displayed using the current style with

\DTMusedate

$\DTMusedate{\langle name \rangle}$

This just uses \DTMdisplaydate . An error will occur if $\langle name \rangle$ hasn't been defined. Alternatively for the capitalised version:

\DTMUsedate

\DTMUsedate{\langle name \range \}

which uses \DTMDisplaydate instead.

A previously saved time can be displayed using the current style with

\DTMusetime

$\DTMusetime{\langle name \rangle}$

This just uses \DTMdisplaytime . An error will occur if (name) hasn't been defined

A previously saved zone can be displayed using the current style with

\DTMusezone

\DTMusezone{\langle name \range \}

This just uses \DTMdisplayzone . An error will occur if $\langle name \rangle$ hasn't been defined.

The entire date, time and zone can be displayed in the current style with

\DTMuse

\DTMuse{\langle name \range \}

This uses \DTMdisplay . An error will occur if (name) hasn't been defined. Alternatively,

\DTMUse

 $\DTMUse{\langle name \rangle}$

will use \DTMDisplay instead.

You can determine if a given (name) has been defined using

\DTMi fsaveddate

 $\label{local_def} $$ \operatorname{DTMifsaveddate}(\langle name \rangle) {\langle true \rangle} {\langle false \rangle} $$$

The individual numerical elements can be fetched using one of the following commands. These don't check if the given data identified by $\langle name \rangle$ has been defined and will expand to \relax if the name isn't recognised.

\DTMfetchyear

\DTMfetchyear{\langle name \rangle}

This expands to the year.

\DTMfetchmonth

\DTMfetchmonth{\langle name \rangle}

This expands to the month number.

\DTMfetchday

\DTMfetchday{\(\langle\)}

This expands to the day of the month.

\DTMfetchdow

 $\verb|\DTMfetchdow{<| (name)|}|$

This expands to the day of the week number (-1 if unknown).

\DTMfetchhour

\DTMfetchhour{\langle name \rangle}

This expands to the hour.

\DTMfetchmi nute

\DTMfetchminute{\langle name \range \}

This expands to the minute.

\DTMfetchsecond

 $\verb|\DTMfetchsecond{|\langle name \rangle|}$

This expands to the second.

\DTMfetchTZhour

 $\verb|\DTMfetchTZhour{|} \langle name \rangle|$

This expands to the hour offset.

\DTMfetchTZmi nute

 $\verb|\DTMfetchTZminute{<| (name |) |}|$

This expands to the minute offset.

5 Styles

If you want to just change the date style use:

\DTMsetdatestyle

\DTMsetdatestyle{\(\langle\)}

where $\langle name \rangle$ identifies the style. For example:

\DTMsetdatestyle{iso}

This will just change the date style (\DTMdisplaydate and \DTMDisplaydate), not the time or zone styles. Note that \DTMdisplay typically uses \DTMdisplaydate so this will also change the date element of \DTMdisplay.

If you want to just change the time style use:

\DTMsettimestyle

\DTMsettimestyle{\(\langle\)}

where $\langle name \rangle$ identifies the style. For example:

\DTMsettimestyle{iso}

This will just change the time style (\DTMdisplaytime), not the date or zone styles. Note that \DTMdisplay typically uses \DTMdisplaytime so this will also change the time element of \DTMdisplay.

If you want to just change the zone style use:

\DTMsetzonestyle

$\DTMsetzonestyle{\langle name \rangle}$

where $\langle name \rangle$ identifies the style. For example:

\DTMsetzonestyle{iso}

This will just change the zone style (\DTMdisplayzone), not the date or time styles. Note that \DTMdisplay typically uses \DTMdisplayzone so this will also change the zone element of \DTMdisplay.

If you want to change the full style use:

\DTMsetstyle

\DTMsetstyle{\(\lame\range\)}

where $\langle name \rangle$ identifies the style. For example:

\DTMsetstyle{iso}

Note that in this case this does more than simply

\DTMsetdatestyle{iso}\DTMsettimestyle{iso}\DTMsetzonestyle{iso}

as it also changes \DTMdisplay and \DTMDisplay. If the style \(name \) is only a partial style, a warning will be issued for any partial styles that aren't defined for the given name as well as a warning for the undefined full style. An error will occur if there are neither partial nor full styles with the given \(name \).

The predefined styles listed in Section 5.1.1 are all *full styles*. This means that they change the date, time, zone and full format, so any of them can be used in \DTMsetdatestyle, \DTMsettimestyle, \DTMsetzonestyle or \DTMsetstyle. However it's possible for a style to be only a *partial style*, such as those described in Section 5.1.2.

For example, if foo is a date style and a time style but isn't a zone style or a full style then you can use

\DTMsetdatestyle{foo}

and

\DTMsettimestyle{foo}

but you can't use \DTMsetzonestyle. You can use

\DTMsetstyle{foo}

but this will now only be equivalent to

\DTMsetdatestyle{foo}\DTMsettimestyle{foo}

and while \DTMdisplay and \DTMDisplay will typically use these date and time settings, the way that the date, time and zone are arranged will be governed by the full style setting that was already in effect before the date and time style changed.

The style changes are all local and so are affected by the current scope.

5.1 Predefined Styles

The base datetime2 package provides a number of predefined numerical styles. Section 5.1.1 lists the full styles, which can be used with \DTMsetstyle, \DTMsettatestyle, \DTMsettatestyle and \DTMsetzonestyle. Section 5.1.2 lists the predefined (partial) times styles, which can be used with \DTMsettimestyle and \DTMsetstyle.

5.1.1 Full Styles

The following are predefined full styles that are provided by the base datetime2 package. Additional styles are available through the language modules (see Section 6).

default The default style displays the date in the form

```
\langle YYYY \rangle \langle YMsep \rangle \langle MM \rangle \langle MDsep \rangle \langle DD \rangle
```

where the month $\langle MM \rangle$ and day of the month $\langle DD \rangle$ numbers are formatted as two digits. The separators $\langle YMsep \rangle$ and $\langle MDsep \rangle$ default to a hyphen but can be changed using the options yearmonthsep, monthdaysep or datesep (either through the package options or using \DTMsetup).

The time is displayed in the form:

```
\langle hh \rangle \langle HMsep \rangle \langle mm \rangle \langle MSsep \rangle \langle ss \rangle
```

where the hour, month and seconds are formatted as two digits. The final $\langle MSsep \rangle \langle ss \rangle$ is omitted if the option showseconds has been set to false. The separators $\langle HMsep \rangle$ and $\langle MSsep \rangle$ default to a colon (:) but these may be changed using the options hourminsep, minsecsep or datetimesep.

The zone is displayed in form

```
\langle TZh \rangle \langle HMsep \rangle \langle TZm \rangle
```

or just Z if the option showisoZ is set to true and both $\langle TZh \rangle$ and $\langle TZm \rangle$ are zero. The separator $\langle HMsep \rangle$ is the same as used for the time format. The final $\langle HMsep \rangle \langle TZm \rangle$ is omitted if the option showzoneminutes is set to false. The hour offset $\langle TZh \rangle$ is formatted as two digits proceeded by either + or - and the minute offset is formatted as two digits. Note that since one of the main purposes of this package is to provide expandable date commands that can be used to write information to external files, no attempt is made to convert the hyphen - (for negative offsets) into a minus sign. If you want it rendered correctly in your document, consider placing the time zone command in math mode and adjust the separators as necessary.

The full style is in the form

```
\langle date \rangle \langle DTsep \rangle \langle time \rangle \langle TZsep \rangle \langle zone \rangle
```

The $\langle date \rangle \langle DTsep \rangle$ part is omitted if the option showdate is set to false, and the $\langle TZsep \rangle \langle zone \rangle$ part is omitted if the option showzone is set to false. The separator between the date and time $\langle DTsep \rangle$ defaults to \space but may be changed using the datetimesep option. The separator between the time and zone $\langle TZsep \rangle$ defaults to nothing but may be changed using the timezonesep option.

- iso The iso style is like the default style but the separators can't be changed. The separators used in the date format are fixed as hyphens and the separators used in the time and zone formats are fixed as colons. In the full format, the separator between the date and time is fixed as T and there's no separator between the time and zone. The only options that can change the iso style are showseconds, showdate, showzone, showzoneminutes and showisoZ.
- yyyymd This is like the default style except that the month and date aren't forced into a two-digit format.
- ddmmyyyy This is like the default style except that the date is formatted in the reverse order

```
\langle DD \rangle \langle MDsep \rangle \langle MM \rangle \langle YMsep \rangle \langle YYYY \rangle
```

The day and month are displayed as two-digits and the separators are as for the default style. The options that modify the default style similarly modify this style.

- dmyyyy This is like the ddmmyyyy style except that it doesn't force the day and month into a two-digit format. The options that modify the default style similarly modify this style.
- dmyy This is like the dmyy style except that it only displays the final two digits of the year. The options that modify the default style similarly modify this style.
- mmddyyyy This is like the ddmmyyyy style except the day and month numbers are reversed. The separator between the month and day is still given by the monthdaysep or datesep options. The separator between the day and year is given by the dayyearsep or datesep options.
- mdyyyy This is like the mmddyyyy style except that it doesn't force the day and month into a two-digit format.
- mdyy This is like the mdyyyy style except that the year only has the final two digits displayed.

pdf This formats the date, time and zone so that the full style is in the form required by the date settings in \pdfinfo. The date format is

```
D:\langle YYYY\rangle\langle MM\rangle\langle DD\rangle
```

where the month and day numbers are displayed as two digits.

The time format is

```
\langle hh\rangle\langle mm\rangle\langle ss\rangle
```

where the numbers are displayed as two digits.

The zone format is

```
\langle hh \rangle' \langle mm \rangle'
```

or Z for zero time offset if the option ShowisoZ is used. (The ShowisoZ option is the only option that modifies the pdf style.) The hour and minutes are displayed as two digits where the hour has the sign present (either + or -).

The full style is a concatenation of the date, time and zone.

```
D:\langle YYYY\rangle\langle MM\rangle\langle DD\rangle\langle hh\rangle\langle mm\rangle\langle ss\rangle\langle hh\rangle'\langle mm\rangle'
```

5.1.2 Time Styles

There's only one predefined time (partial) style provided by the base datetime2 package. This style can be used to override the time format part of full styles. For example, to use the default full style with the hmmss time style:

```
\DTMsetstyle{default}\DTMsettimestyle{hmmss}
```

hmmss The hmmss style is like the time style provided by the full default style except that the hour isn't forced into two digits.

5.1.3 Zone Styles

The following are predefined zone (partial) styles that are provided by the base datetime2 package. These styles can be used to override the zone format part of full styles. For example, to use the default full style with the map zone style:

\DTMsetstyle{default}\DTMsetzonestyle{map}

map The map style uses \DTMusezonemapordefault to display the mapping, if one exists, or use the default style, if a mapping doesn't exist. For example:

\DTMNatoZoneMaps
\DTMsetzonestyle{map}

This first defines the NATO mappings and then switches to the map style.

hhmm The hhmm style displays the time zone in the form

 $\langle TZh \rangle \langle HMsep \rangle \langle TZm \rangle$

where $\langle HMsep \rangle$ is given by the hourminsep option. This style honours the showzoneminutes option but ignores the showisoZ option. The hour is always prefixed by the sign.

5.2 Defining New Styles

A new date style can be defined using:

\DTMnewdatestyle

 $\verb|\DTMnewdatestyle{<|name||} {<|definition||}$

This defines a partial style that should only modify \DTMdisplaydate and \DTMDisplaydate. The redefinition of these commands should be placed in \(\definition \rangle \).

A new time style can be defined using:

\DTMnewtimestyle

 $\DTMnewtimestyle{\langle name \rangle} {\langle definition \rangle}$

This defines a partial style that should only modify \DTMdisplaytime . The redefinition should be placed in $\langle definition \rangle$.

A new zone style can be defined using:

\DTMnewzonestyle

 $\verb|\DTMnewzonestyle{<|name|}| {\langle definition|}|$

This defines a partial style that should only modify \DTMdisplayzone . The redefinition should be placed in $\langle definition \rangle$.

A new full style can be defined using:

\DTMnewstyle

This does

and finally (full style definition) should redefine \DTMdisplay and \DTMDisplay.

Remember to use a double-hash to reference the parameters (##1, ##2 etc) within $\langle definition \rangle$ in all the above. In each case $\langle name \rangle$ is the label identifying the style and shouldn't contain active characters.

There are some helper commands provided that you might want to use in the style definitions.

\DTMtwodigits

\DTMtwodigits{\(\lamber\rangle\)}

This displays (number) so that it has *only* two digits. Unlike MTEX's \two@digits this will check for a negative number and will trim a number whose absolute value is greater than 100. This command is expandable.

\DTMcentury

\DTMcentury{\langle year\rangle}

This converts $\langle year \rangle$ to the century. If $\langle year \rangle$ is negative it does:

```
-\DTMcentury{-\(\sqrt{year}\)\}
```

Example:

\DTMcentury{1945}

expands to 20 (not 19). Note that

\DTMcentury{1900}

expands to 19.

\DTMdi vhundred

\DTMdivhundred{\(\langle number \rangle\)}

This expands to $\lfloor \langle number \rangle / 100 \rfloor$ (integer division by 100 rounded down). For example:

\DTMdi vhundred{1945}

expands to 19 and

\DTMdi vhundred{1900}

expands to 19.

\DTMtexorpdfstring

\DTMtexorpdfstring $\{\langle T_E X \rangle\}\{\langle PDF \rangle\}$

If hyperref is loaded, this is equivalent to \texorpdfstring otherwise it just does the first argument and ignores the second. (The check for hyperref is deferred until the start of the document environment, so it doesn't matter if hyperref is loaded after datetime2.) This command may be used to provide alternative text to use if the date/time/zone is displayed in the PDF bookmarks.

\DTMsep

$\DTMsep{\langle tag \rangle}$

This accesses the value of the $\langle tag \rangle$ sep base package option. (Not the language module options.) For example

\DTMsep{yearmonth}

expands to the value supplied by the yearmonthsep package option.

\DTMusezonemap

$\label{eq:definition} $$ \operatorname{DTMusezonemap} \{\langle TZh \rangle\} \{\langle TZm \rangle\} $$$

This expands to the time zone abbreviation or \relax if no mapping has been set for the given time zone.

You can define a time zone mapping using

\DTMdefzonemap

```
\label{eq:def_def_def} $$ \operatorname{TZh} {\langle TZh \rangle} {\langle TZm \rangle} {\langle abbr \rangle} $$
```

For example

\DTMdefzonemap{00}{00}{GMT} \DTMdefzonemap{01}{00}{BST}

Note that datetime2 doesn't know anything about daylight saving, so this is only really designed for dates and times in a specific location. This overwrites any previous mapping for this time zone.

The base datetime2 package provides

\DTMNatoZoneMaps

\DTMNatoZoneMaps

This defines the military/NATO mappings from A (Alpha time) to Z (Zulu time). You can use this command if you want these time zones (but remember to set an appropriate time zone style that uses the zone mappings).

The language modules may provide mappings that are enabled when you switch to that style. For example, the en-GB language module provides the mapzone option which, if set to true, will map +00:00 to GMT and +01:00 to BST. See the documentation for the language module for further details.

\DTMcI earmap

 $\label{eq:decomposition} $$\operatorname{DTMclearmap}_{\langle TZh\rangle}_{\langle TZm\rangle}$$$

Clears the time zone mapping. The regional time zone styles should use

\DTMresetzones

\DTMresetzones

before applying any regional mappings. This defaults to nothing which means that any mappings previously defined by other styles won't be cleared. You can redefine this command if you want to clear any mappings that aren't relevant for other regions.

You can test if a mapping is defined using

\DTMhaszonemap

 $\label{eq:definition} $$ DTMhaszonemap{\langle TZh\rangle}_{\langle TZm\rangle}_{\langle true\rangle}_{\langle false\rangle}$$$

This will do $\langle true \rangle$ if there is a mapping defined for that time zone or $\langle false \rangle$ otherwise.

\DTMusezonemapordefault

\DTMusezonemapordefault $\{\langle TZh\rangle\}\{\langle TZm\rangle\}$

This will use the mapping if its defined otherwise it will expand to the format $\langle TZh\rangle\langle HMsep\rangle\langle TZm\rangle$ where $\langle HMsep\rangle\langle TZm\rangle$ is omitted if the option <code>ShowZone-minutes</code> is set to false. The separator $\langle HMsep\rangle$ is as given by the hourminsep option. (The <code>ShowisoZ</code> option isn't used here so UTC+00:00 will be displayed as +00:00 or +00 if there's no mapping.)

Here's an example of a simple date style that just displays the year and month as two digits but uses the yearmonthsep option:

```
\newdatestyle
{mmyy}% label
{% definitions
\renewcommand*{\DTMdisplaydate}[4]{%
\DTMtwodigits{##2}\DTMsep{yearmonth}\DTMtwodigits{##1}}%
\renewcommand*{\DTMDisplaydate}{\DTMdisplaydate}%
}
```

If you want to distribute your new styles, just put the definitions in a package and upload it to CTAN. For example (replace mystylename with something more appropriate, and also change the date in the \ProvidesPackage line):

```
\NeedsTeXFormat{LaTeX2e}
\Provi desPackage{mystyl ename}[2014/03/24 v1.0]
\Requi rePackage{dateti me2}
% style definitions here
\endinput
```

Save the file as mystylename.sty, add some documentation about the style (or styles) provided and read the instructions at http://www.ctan.org/upload and http://www.ctan.org/file/help/ctan/CTAN-upload-addendum. The upload location for additions to the datetime2 package (either for packages defining new styles or for language modules) should be /macros/latex/contrib/datetime2-contrib/mystylename (remember to replace mystylename as appropriate).

6 Multi-Lingual Support

If you want to use datetime2 with babel or polyglossia, make sure you load babel/polyglossia before you load datetime2 otherwise their \date\language\ will overwrite \datetime's definition of \today. Additionally you need to make sure you install the relevant datetime2 language modules. These modules are automatically loaded, if required, by datetime2 but only if they are already installed. Remember that if you use XqETEX you won't have the seconds or time zone available for the current date and time.

If the required language modules aren't installed or datetime2 is loaded before babel/polyglossia then datetime2's definition of \today will be overridden and may no longer match the currently selected date style.

Each language module defines a textual style (where the month is displayed as a word) for that language or region which can be used in the argument of \DTMsetstyle, \DTMsetdatestyle, \DTMsettimestyle or \DTMsetzonestyle. The language module may also define a numeric style. In the ambiguous cases where the language name alone doesn't indicate the region (for example, english instead of UKenglish or USenglish) the module should use the default numeric style (see Section 5.1.1).

The textual style provided by the module will automatically be set using \DTMsetstyle if the useregional option is set to text. By default useregional is false, unless the language/region is passed via the datetime2 package option list. (The useregional option is unaffected if the setting is passed through the document class option list.) The numeric style provided by the module will automatically be set if the useregional option is set to numeric. See the descriptions for the useregional and style options in Section 7.

Be careful not to mix the language/region options between the document class option list and the babel/polyglossia interface. For example:

```
\documentclass[en-GB]{article}
\usepackage[canadien, british]{babel}
```

This will prevent the tracklang package from picking up the babel setting and it will only detect the en-GB option. Use only the document class options or only the babel package option list or duplicate *all* the babel package options with analogous tracklang options in the document class. For example

```
\documentcl ass[canadi en, bri ti sh]{arti cl e}
\usepackage{babel}
```

or

```
\documentclass{article}
\usepackage[canadien, british]{babel}
```

or

```
\documentclass[fr-CA, en-GB]{article} \usepackage[canadien, british]{babel}
```

Language modules may be used without babel or polyglossia. For example:

```
\documentclass{article}
\usepackage[en-GB]{datetime2}
\begin{document}
\today
\end{document}
```

If you have more than one language or region you will need to switch styles using \DTMsetstyle etc:

```
\documentcl ass{article}
\usepackage[en-GB, en-CA]{datetime2}
\begin{document}
\DTMsetstyle{en-GB}\today.
\DTMsetstyle{en-CA}\today
\end{document}
```

If you want to change the number separators for the *regional* numeric styles, you need to use \DTMlangsetup. If you want to change the number separators for the base datetime2 predefined numeric styles (see Section 5.1) then you need to use \DTMsetup or the package options. You therefore need to use \DTMsetup for the ambiguous regionless language numeric settings since they just use the default style. Check the module documentation to find out if the default style is used.

Examples of use:

1. Language option specified through the document class and picked up by tracklang (which is loaded by datetime2). This setting is also picked up by babel which is loaded before datetime2.

```
\documentclass[british]{article}
\usepackage{babel}
\usepackage{datetime2}
\begin{document}
\today
\end{document}
```

The date is displayed in the default format 2015-03-01.

In this case, the en-GB language module is loaded which defines the text style en-GB and the numeric style en-GB-numeric. Since useregional hasn't been set, \today uses datetime2's default numerical format. If babel was loaded after datetime2, the babel's hook management system would overwrite datetime2's definition of \today so that it no longer used \DTMdisplaydate. A similar result is obtained if in the above example babel is replaced with polyglossia (where the language is set in the document class option).

You can change the useregional setting either through datetime2's package options or using \DTMsetup however it will only have an effect during the module loading (when the value is changed via the package option) and when \date(language) is used.

For example, in the document below, the date is displayed using the default numeric format because useregional has been changed *after* babel uses \datebritish to set the language at the start of the document.

```
\documentclass[british]{article}
\usepackage{babel }
\usepackage{datetime2}
\begin{document}
\DTMsetup{useregional }
\today
\end{document}
```

So here \today again displays the date in the form 2015-03-01.

If the setting is moved to the preamble:

```
\documentclass[british]{article}
\usepackage{babel }
\usepackage{datetime2}

\DTMsetup{useregional }
\begin{document}
\today
\end{document}
```

then the useregional setting is checked at the beginning of the document when babel uses \datebritish. So in this case \today will display the date in the form 1st March 2015.

2. Language setting specified through babel's package option list:

```
\documentclass{article}
\usepackage[british]{babel}
\usepackage{datetime2}
\begin{document}
\today
\end{document}
```

This has the same result as placing british in the document class option list, so the date is again displayed in the default format 2015-03-01 but, as in the previous example, the en-GB and en-GB-numeric styles are both defined if required.

However a problem occurs if babel is replaced by polyglossia:

```
\documentcl ass{article}
\usepackage{fontspec}
\usepackage{pol ygl ossi a}
\setdefaul tl anguage[vari ant=uk]{engl i sh}
\usepackage{dateti me2}
\begi n{document}
\today
\end{document}
```

In this case tracklang is unable to pick up the variant and can only detect the root language, so it will load the generic english module instead of the en-GB module. This means that the en-GB and en-GB-numeric styles are no longer available. However, since useregional is false the date is still displayed using the default numeric style in the form 2015-03-01.

3. As mentioned above neither babel nor polyglossia are required in order to use the datetime2 language modules. You can simply supply the language setting in the package option list:

```
\documentclass{article}
\usepackage[british]{datetime2}
\begin{document}
\today
\end{document}
```

This additionally sets useregional=true (since the language is in the package option list not the document class option list) so the date produced by \today now uses the en-GB date style in the form 1st March 2015.

4. The regional numeric format can be used instead if useregional is set to numeric:

```
\documentclass{article}
\usepackage[british, useregional =numeric]{datetime2}
\begin{document}
\today
\end{document}
```

This now displays the date in the form 1/3/2015.

Many of the language options have synonyms. In addition to the babel synonyms (such as british or UKenglish) the tracklang package provides options in ISO form, such as en-GB. Note that the style name provided by each language module is independent of the package option used to select that style. So regardless of whether you use british, UKenglish or en-GB, the text style name is en-GB and the numeric style name is en-GB-numeric. If just english is used, the text style name is english but the numeric style is default.

Languages where the region is automatically implied, such as SCOttish, provide a text style with the root language name (scottish in this instance) and a numeric style in the form $\langle language \rangle$ -numeric (such as scottish-numeric). Note that irish has regionless styles irish and irish-numeric but also has regional styles ga-IE and ga-IE-numeric (for the Republic of Ireland) and ga-GB and ga-GB-numeric (for Northern Ireland). In this case the regionless style has a numeric style instead of using the default style since both ga-IE-numeric and ga-GB-numeric are the same so there's no ambiguity. The only difference in the three modules datetime2-irish, datetime2-ga-IE and datetime2-ga-GB is the time zone mappings.

The language modules may provide additional settings that can be applied using

\DTMI angsetup

$\texttt{\DTMlangsetup}[\langle language\ list\rangle] \{\langle options\rangle\}$

where $\langle language\ list \rangle$ is a comma-separated list of language modules that have been loaded (such as en-GB, en-US) and $\langle options \rangle$ is a $\langle key \rangle = \langle value \rangle$ list of options. If $\langle language\ list \rangle$ is omitted, then the list of all loaded language modules is assumed. The modules may also provided user commands to further customise the style. These settings should all be described in the module's documentation, which should be accessible via texdoc datetime2- $\langle language \rangle$ where $\langle language \rangle$ is the root language name in lower case (such as english).

Note that although I maintain the datetime2 English language module, I don't maintain the other modules. If you have an issue with one of the other modules, please contact the module maintainer. If there is no maintainer, feel free to volunteer to take over the maintenance (send me a message). If there's no module for your language you can create your own module and upload it to CTAN in the /macros/latex/contrib/datetime2-contrib/datetime2-contrib/datetime2-dianguage directory.

You can use the English or Irish modules as a template for a language with multiple regions. Just download the English source files datetime2-english.dtx and datetime2-english.ins or the Irish source files datetime2-irish.dtx and datetime2-irish.ins from CTAN and make the appropriate modifications. Alternatively you can use the Scottish module as a template for a single-region language. Just download the Scottish source files datetime2-scottish.dtx and datetime2-scottish.ins from CTAN and make the appropriate modifications. (Don't forget to provide a README file.)

Each language module should be in a file named datetime2- $\langle lang \rangle$.ldf where $\langle lang \rangle$ is the language name or $\langle language\ ISO\ code \rangle$ - $\langle country\ ISO\ code \rangle$. (See the tracklang documentation for further details of the naming scheme.)

If you want to provide a language module don't assume all users want to use the same input encoding or babel shortcuts as you. Use \LaTeX commands for non-ASCII characters (and remember to use protect where necessary).

As an addendum to the above warning, LuaTeX and XeTeX support UTF-8 characters without the need to make them active, so I recommend you provide two files: one with the LateX commands, such as \c, for (PDF) LateX users, and one with UTF-8 characters for LuaLateX and XetAeX users. For example, the fr-FR module could start with:

\Provi desDateTi meModul e{fr-FR}

```
\Requi rePackage{i fxetex, i fl uatex}
\i fxetex
\Requi reDateTi meModul e{french-utf8}
\el se
\i fl uatex
\Requi reDateTi meModul e{french-utf8}
\el se
\Requi reDateTi meModul e{french-asci i }
\fi
\fi
```

This helps provide fully expandable dates for Lual TeX and XeVTeX users. (See the Scottish or Irish modules.)

7 Package Options

The following package options are provided. Most of these are $\langle key \rangle = \langle value \rangle$ options, unless stated otherwise.

Settings that govern the predefined numerical styles (not including the fixed styles iso and pdf):

- yearmonthsep This sets the separator between the year and month for the big-endian and little-endian styles. Default: (hyphen).
- monthdaysep This sets the separator between the month and day. Default: (hyphen).
- dayyearsep This sets the separator between the day and year for the middle-endian styles. Default: (hyphen).
- datesep This sets the separators between the day and month, the month and year, and the day and year. Example:

```
\usepackage[datesep=/]{datetime2}
```

This is equivalent to:

\usepackage[yearmonthsep=/, monthdaysep=/, dayyearsep=/]{datetime2}

- hourminsep This sets the separator between the hour and minute. (Both for the time and for the zone.) Default: : (colon).
- minsecsep This sets the separator between the minute and seconds. Default: : (colon).
- timesep This sets the separators between the hour and minute and between the minute and seconds. Example:

```
\usepackage[timesep=:]{datetime2}
```

This is equivalent to:

```
\usepackage[hourmi nsep=: , mi nsecsep=: ]{dateti me2}
```

The following settings are used by the predefined numerical styles when displaying the full date, time and zone (excluding the fixed styles iso and pdf) with commands that use \DTMdisplay or \DTMDisplay.

datetimesep Sets the separator between the date and time. Default: \space.

timezonesep Sets the separator between the time and zone. Default: empty.

The following settings are used by the predefined styles and may also be used by the language modules.

- showseconds Boolean key to determine whether or not to show the seconds when the time is displayed. The iso style honours this setting but the pdf style ignores it. Default: true unless X_TT_EX is used.
- Showdate Boolean key to determine whether or not to show the date with commands that use \DTMdisplay or \DTMDisplay. (Some styles may ignore this.) The iso style honours this setting but the pdf style ignores it. Default: true unless X_TT_FX is used.
- Showzone Boolean key to determine whether or not to show the time zone with commands that use \DTMdisplay or \DTMDisplay. (Some styles may ignore this.) The iso style honours this setting but the pdf style ignores it. Default: true unless XaTeX is used.
- Showzoneminutes Boolean key to determine whether or not to show the zone offset minutes. The iso style honours this setting but the pdf style ignores it. This setting is ignored if Showzone is false. Default: true.
- ShowisoZ Boolean key to determine whether or not to show UTC+00:00 as Z instead of numerically. This option may be ignored by zone styles that use the zone mappings. If you want all the time zones in military form, you can use \DTMNatoZoneMaps to set up the time zone abbreviations and then use a zone style that uses the mappings. Default: true.

General settings:

useregional Allowed values: false, text or numeric. You may also use num as an abbreviation for numeric. If no value is supplied text is assumed.

This key determines whether or not to *use* the loaded regional settings and, if the regional setting should be used, it determines whether the text style (months as words) or numeric style should be used. If you haven't loaded babel or polyglossia, this key only has an effect when used as a package option. If you have loaded one of those packages, the change comes into effect at module load time and whenever \date \language \in is used (which includes at the beginning of the document environment). If you want to switch the style at any other time, you need to use \DTMsetstyle but unless useregional=false the next instance of \date \language \in will change the style.

Note that setting this option to false doesn't prevent the modules from being loaded. It just prevents them from automatically setting the style and prevents \date \(\language\rangle\) from changing the style if you are using babel or polyglossia.

The default value is false unless the language or region is passed to the datetime2 package option list. However, using style will set useregional to false.

Examples:

\documentcl ass[bri ti sh]{arti cl e}
\usepackage{babel }
\usepackage{dateti me2}

In the above useregional is false.

\documentclass{article}
\usepackage[british]{datetime2}

In the above useregional is text.

\documentclass{article}
\usepackage[british, style=iso]{datetime2}

In the above useregional is false. (The british option implements usenumerical = text but the style option then implements usenumerical = false.)

```
\documentclass{article}
\usepackage[style=iso, british]{datetime2}
```

In the above useregional is text. (The style option implements usenumerical=false but the british option then implements usenumerical=text.)

style Sets the current style using \DTMsetstyle when the datetime2 package has finished loading. This also sets useregional=false but that setting can be overridden later in the option list.

Default value: empty (use the default style or the regional style, according to the value of useregional).

This key isn't available in \DTMsetup. Use \DTMsetstyle instead.

calc Load the datetime2-calc package. This will allow the day of week to be computed and allow you to use the pgfcalendar offset style date formats in commands like \DTMdate as well as defining the commands described in Section 8. This option doesn't take a value. It can't be switched off. This option can't be used in \DTMsetkeys. The default is to not load datetime2-calc.

showdow This is a boolean key that determines whether or not to show the day of week in styles that support this. Note that Showdow=true will automatically load datetime2-calc so

\usepackage[showdow]{datetime2}

is equivalent to

\usepackage[showdow, calc]{datetime2}

This option may be used in \DTMsetup, but if you attempt to switch it on in the document environment you'll get an error if the datetime2-calc package hasn't been loaded. Default: false.

warn This is a boolean key. If true (default) datetime2 warnings will be displayed. If false, the warnings will be suppressed. Default: true.

Any additional option passed to the datetime2 package (not through \DTMsetup) will be considered a tracklang option and will be passed to \TrackPredefinedDialect. (See the tracklang documentation for further details of that command.)

Apart from Calc, Style and the regional options, all the above options can also be set using:

\DTMsetup

\DTMsetup{\langle option \list\rangle}

The language modules may additionally provide options which can be set using:

\DTMI angsetup

$\texttt{\DTMlangsetup[}\langle module\ list\rangle]\{\langle option\ list\rangle\}$

This will set the $\langle option \ list \rangle$ for each module listed in $\langle module \ list \rangle$. Unknown options will generate a warning rather than an error message. The default value of $\langle module \ list \rangle$ is the list of all loaded modules.

Example:

```
\documentcl ass{article}
\usepackage[british]{datetime2}
\DTMI angsetup{mapzone}
```

The module list here is english-base, en-GB and since the english-base doesn't have a mapzone option, this will result in a warning:

```
Package datetime2 Warning: Region 'english-base' has ignored (datetime2) the following settings: (datetime2) mapzone
```

You can either ignore the warning or use the optional argument to exclude the english-base module:

```
\documentclass{article}
\usepackage[british]{datetime2}
\DTMlangsetup[en-GB]{mapzone}
```

Note that some modules may have options with the same name as the above listed package options, but the keys are defined in different families (see XkeyVal documentation) so you need to take care to use \DTMsetup for package-wide settings and \DTMlangsetup for the module-specific settings.

For example, the datesep package option described above is used by the predefined numerical styles but regional modules that provide their own numerical styles may use a different date separator that matches their region so they may also provide a datesep option independent of the base datesep option.

Examples:

```
\documentclass[british]{article}
\usepackage[datesep=.]{datetime2}
\begin{document}
\today
\end{document}
```

The above displays the date in the form 2015.09.15 since the default style is in use and datesep is used as a package option.

```
\documentcl ass[bri ti sh]{article}
\usepackage{dateti me2}
\DTMsetup{datesep=. }
\begi n{document}
\today
\end{document}
```

The above displays the date in the form 2015.09.15 since the default style is in use and datesep is used in \DTMsetup.

```
\documentcl ass[bri ti sh]{article}
\usepackage{dateti me2}
\DTMI angsetup{datesep=. }
\begi n{document}
\today
\end{document}
```

The above displays the date in the form 2015-09-15 since the default style is in use but datesep is used in \DTMlangsetup, which only influences the en-GB-numeric style, which isn't the current style.

```
\documentcl ass[bri ti sh]{article}
\usepackage[useregi onal =numeri c]{dateti me2}
```

```
\DTMI angsetup{datesep=. }
\begin{document}
\today
\end{document}
```

The above displays the date in the form 15.9.2015 since the en-GB-numeric style is in use and datesep is used in \DTMlangsetup.

```
\documentclass[british]{article}
\usepackage[useregional]{datetime2}
\DTMlangsetup{datesep=.}
\begin{document}
\today
\end{document}
```

The above displays the date in the form 15th September 2015 since the en-GB style is in use and datesep is used in \DTMlangsetup, which only influences the en-GB-numeric style.

```
\documentcl ass[bri ti sh]{article}
\usepackage[useregi onal =numeri c]{dateti me2}
\DTMsetup{datesep=. }
\begi n{document}
\today
\end{document}
```

The above displays the date in the form 15/9/2015 since the en-GB-numeric style is in use but datesep is used in \DTMsetup which influences the base predefined numeric styles not the regional styles.

8 The datetime2-calc Package

The datetime2-calc package can be loaded after datetime2 in the usual way:

\usepackage{dateti me2}
\usepackage{dateti me2-cal c}

or using the calc package option to datetime2:

\usepackage[calc]{datetime2}

or by using showdow = true:

\usepackage[showdow]{datetime2}

This package loads the pgfcalendar package which provides a way of computing the day of week from a given date. Once datetime2-calc has been loaded, you can enable or disable the weekday in dates where the style supports this, but note that not all styles support this, even if the datetime2-calc package has been loaded.

As with the commands in Section 4, the commands described below that save date/time information will *overwrite* any previously defined date/time data with the same identifying ⟨*name*⟩. However, they may only overwrite specific elements of the data (for example, just the year, month, day and day of week elements) and leave the other elements unchanged. Where the remaining elements are undefined they'll be set to zero, except for the day of week element, which will be set to -1.

In addition to enabling the weekday calculations, the datetime2-calc package also provides the following commands:

\DTMsavej ul i anday

 $\label{lem:decomposition} $$ DTMsavejulianday{\langle name \rangle} {\langle number \rangle}$$

This uses $\protect{\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\protect\$

\DTMsavej ul i anday{mydate}{2457023}

\DTMsaveddatetoj ul i andate

\DTMsaveddatetojulianday{\(\rangerrow\)} \{\(\register\)\}

This uses $\protect\operatorname{pgfcalendardatetojulian}$ to convert a previously saved date (identified by $\langle name \rangle$) to a Julian day. The result is stored in $\langle register \rangle$ which should be a count register (not a $\protect\operatorname{MTFX}$ counter name). Example:

```
\newcount\myct
\DTMsaveddatetoj ul i anday{mydate}{\myct}
```

\DTMsaveddateoffsettoj ul i andate

```
\label{lem:decomposition} $$ DTMsaveddateoffsettojulianday{$\langle name \rangle$} {\langle offset \rangle$} {\langle register \rangle$} $$
```

This is like the previous command but converts the date obtained by incrementing the saved date with $\langle \textit{offset} \rangle$. The result is stored in $\langle \textit{register} \rangle$. This is equivalent to

```
\pgfcalendardatetojulian\{\langle v \rangle - \langle m \rangle - \langle d \rangle + \langle offset \rangle\} \{\langle register \rangle\}
```

where $\langle y \rangle$, $\langle m \rangle$ and $\langle d \rangle$ are the year, month and day fetched from the saved date. A negative $\langle offset \rangle$ indicates an earlier date. Example:

```
\DTMsaveddateoffsettoj ul i anday{mydate}{2}{\myct}
```

or

\DTMsaveddateoffsettoj ul i anday{mydate}{-7}{\myct}

\DTMi fdate

```
\label{local_def} $$ \operatorname{DTMifdate}_{\langle name\rangle}_{\langle test\rangle}_{\langle true\rangle}_{\langle false\rangle} $$
```

This is just a convenient interface to \pgfcalendarifdate for a saved date (identified by $\langle name \rangle$). The remaining arguments are the same as the final three arguments of \pgfcalendarifdate . Note that the equals, at least, at most and between keywords available in $\langle test \rangle$ need to be in the format specified by the pgf manual, but remember that you can use commands like \pTMfetchyear . Example:

```
Is \texttt{mydate2} (\DTMusedate{mydate2}) before
\texttt{mydate} (\DTMusedate{mydate})?
\DTMi fdate
{mydate2}
{at most=
 \DTMfetchyear{mydate}-\DTMfetchmonth{mydate}-\DTMfetchday{mydate}}
{yes}{no}.
```

\DTMsaveddatediff

 $\verb|\DTMsavedatediff{<| name 1 \rangle} {<| name 2 \rangle} {<| register \rangle}|$

Computes the difference (in days) between two saved dates and stores the result in the given count register. The first date is identified by $\langle name1 \rangle$ and the second date is identified by $\langle name2 \rangle$. The dates are converted to their respective Julian day numbers $\langle J1 \rangle$ and $\langle J2 \rangle$ and the result is given by $\langle J1 \rangle - \langle J2 \rangle$.

Note that the time and zone are not taken into account, even if they were provided when the dates were stored.

Example:

\DTMsaveddatediff{mydate}{mydate2}{\myct}

```
\DTMusedate{mydate} is
\ifnum\myct=0
  the same day as
\else
  \ifnum\myct<0
   \number-\myct\space day\ifnum\myct<-1s\fi\space before
  \else
   \number\myct\space day\ifnum\myct>1s\fi\space after
  \fi
\fi
\DTMusedate{mydate2}.
```

The pgfcalendar package also provides a variety of useful date-related commands. See the documentation (part of the pgf manual) for further details. Note that the language modules don't use pgfcalendar month and weekday names as the pgfcalendar package isn't loaded by default.

The datetime2-calc package also provides commands that convert a datetime instance into Zulu¹ time (UTC+00:00).

\DTMsaveaszulutime

This converts the given datetime instance into UTC+00:00 and saves the result. You can then use the date with commands like \DTMuse described in Section 4. The $\langle name \rangle$ argument is the label identifying the saved data. The other arguments are all numbers. Example:

 $\DTMsaveaszulutime{mydate}{2014}{6}{3}{20}{45}{0}{6}{0}$

\DTMtozul u

```
\DTMtozulu{\langle name1\rangle}{\langle name2\rangle}
```

 $^{^{1}\}mbox{That}\mbox{'s}$ Zulu as in the NATO alphabet representation of the letter Z.

Uses \DTMsaveaszulutime to convert the datetime stored in $\langle name1 \rangle$ and saves it to $\langle name2 \rangle$. Example:

\DTMsavetimestamp{mydate}{2014-05-01T03:55:00 -06:00}

Original date: \DTMuse{mydate}.

\DTMtozulu{mydate}{mydate2}
UTC+00: 00: \DTMuse{mydate2}.

The above produces (using the default format):

Original date: 2014-05-01 03:55:00-06:00.

UTC+00:00: 2014-05-01 09:55:00Z.

9 The Code

9.1 datetime2.sty code

```
1\NeedsTeXFormat{LaTeX2e}
```

2\ProvidesPackage{datetime2}[2015/09/15 v1.1 (NLCT) date and time formats]

Use tracklang to find out what languages have been loaded.

3 \Requi rePackage{trackl ang}

Also require etoolbox.

4 \RequirePackage{etool box}

Need xkeyval for $\langle key \rangle = \langle value \rangle$ interface.

5 \RequirePackage{xkeyval}[2006/11/18]

\dtm@yearmonthsep Separator between year and month for numeric dates.

6\newcommand*{\dtm@yearmonthsep}{-}

\dtm@monthdaysep Separator between month and day for numeric dates.

7\newcommand*{\dtm@monthdaysep}{-}

\dtm@dayyearsep Separator between day and year for numeric middle-endian dates.

8\newcommand*{\dtm@dayyearsep}{-}

\dtm@hourmi nsep Separator between the hour and minute for times.

9\newcommand*{\dtm@hourmi nsep}{:}

\dtm@mi nsecsep Separator between the minute and second for times.

10 \newcommand*{\dtm@mi nsecsep}{:}

\dtm@ti mezonesep Separator between the date and time.

11\newcommand*{\dtm@datetimesep}{\space}%

\dtm@ti mezonesep Separator between the time and time zone.

12 \newcommand*{\dtm@timezonesep}{}

datesep Set year/month and month/day separator.

13 \define@key{datetime2.sty}{datesep}{%

14 \renewcommand*{\dtm@yearmonthsep}{#1}%

15 \renewcommand*{\dtm@monthdaysep}{#1}%

16 \renewcommand*{\dtm@dayyearsep}{#1}%

17 }

```
Set year/month separator.
yearmonthsep
               18 \define@key{datetime2.sty}{yearmonthsep}{%
                  \renewcommand*{\dtm@yearmonthsep}{#1}%
               20 }
monthdaysep
              Set month/day separator.
               21 \define@key{datetime2.sty}{monthdaysep}{%
                  \renewcommand*{\dtm@monthdaysep}{#1}%
               23 }
  dayyearsep
              Set day/year separator for middle-endian dates.
               24 \defi ne@key{dateti me2. sty}{dayyearsep}{%
                  \renewcommand*{\dtm@dayyearsep}{#1}%
               26 }
              Set hour/minute and minute/second separator.
     timesep
               27 \define@key{datetime2.sty}{timesep}{%
               28 \renewcommand*{\dtm@hourmi nsep}{#1}%
                  \renewcommand*{\dtm@minsecsep}{#1}%
               30 }
  hourmi nsep
              Set hour/minute separator.
               31 \define@key{datetime2.sty}{hourminsep}{%
                  \renewcommand*{\dtm@hourminsep}{#1}%
              33 }
              Set minute/second separator.
   mi nsecsep
               34 \defi ne@key{dateti me2. sty}{mi nsecsep}{%
               35 \renewcommand*{\dtm@mi nsecsep}{#1}%
               36 }
              Set separator between the time and the time zone (used in \DTMnow).
 ti mezonesep
               37 \defi ne@key{dateti me2. sty}{ti mezonesep}{%
               38 \renewcommand*{\dtm@timezonesep}{#1}%
               39 }
datetimesep
              Set separator between the date and the time (used in \DTMnow).
               40 \define@key{datetime2.sty}{datetimesep}{%
                   \renewcommand*{\dtm@datetimesep}{#1}%
               42 }
showseconds
              Boolean key to determine whether or not to show the seconds.
               43 \define@bool key{dateti me2. sty}[DTM]{showseconds}[true]{}
              Boolean key to determine whether or not to show the date in \DTMdisplay and
    showdate
               \DTMDisplay.
               44 \defi ne@bool key{dateti me2. sty}[DTM]{showdate}[true]{}
               45 \DTMshowdatetrue
```

showzone

Boolean key to determine whether or not to show the time zone in \DTMdisplay and \DTMDisplay.

```
46\define@boolkey{datetime2.sty}[DTM]{showzone}[true]{}
```

showi soZ

Boolean key to determine whether or not to use Z instead of +00:00 for UTC in the default, iso or pdf styles. (Other styles may also use this.)

```
47 \defi ne@bool key{dateti me2. sty}[DTM]{showi soZ}[true]{} 48 \DTMshowi soZtrue
```

Switch off seconds and time zone if \pdfcreationdate isn't defined, otherwise switch on.

```
49 \ifdef\pdfcreationdate
50 {%
51 \DTMshowsecondstrue
52 \DTMshowzonetrue
53 }%
54 {%
55 \DTMshowsecondsfalse
56 \DTMshowzonefalse
57 }%
```

showzonemi nutes

Boolean key to determine whether or not to show the time zone minutes. (If \DTMshowzonefalse then this option is irrelevant.)

```
58 \ensuremath{\mbox{ The pool key}} \ensuremath{\mbox{ Sty}[DTM]_{showzonemi nutes}[true]_{59 \ensuremath{\mbox{ DTMshowzonemi nutestrue}}}
```

\DTMi fcaseregi onal

```
\label{lem:definition} $$ \operatorname{DTMifcaseregional}_{\langle false\rangle}_{\langle text\rangle}_{\langle numeric\rangle} $$
```

Determines if the user wants the language modules to set the regional format. The first argument $\langle false \rangle$ indicates that they don't want the regional format set, the second argument $\langle text \rangle$ indicates they want the textual format (e.g. 1st March, 2015 or March 1, 2005) and the third argument $\langle numeric \rangle$ indicates they want the numeric format (e.g. 1/3/2015 or 3/1/2015). A change in the setting will only have an affect when the module is loaded and when $\adtackspace(language)$ is used to set the style. The default is false.

```
60 \newcommand*{\DTMi fcaseregi onal }[3]{#1}
```

useregi onal

Setting to determine whether or not to use the regional settings (if any are loaded).

```
61 \defi ne@choi cekey{dateti me2. sty}{useregi onal }[\val \nr]%
62 {fal se, text, numeri c, num}[text]%
63 {%
64 \i fcase\nr\rel ax
65 \renewcommand*{\DTMi fcaseregi onal }[3]{##1}%
```

```
66  \or
67  \renewcommand*{\DTMi fcaseregi onal }[3]{##2}%
68  \or
69  \renewcommand*{\DTMi fcaseregi onal }[3]{##3}%
70  \or
71  \renewcommand*{\DTMi fcaseregi onal }[3]{##3}%
72  \fi
73 }
```

\@dtm@setusecalc

```
74 \newcommand*{\@dtm@setusecalc}{%
75 \renewcommand*{\@dtm@usecalc}{\RequirePackage{datetime2-calc}}%
76}
```

\@dtm@usecalc

77 \newcommand*{\@dtm@usecalc}{}

Disable attempt to load datetime2-calc in the document.

```
78 \AtBeginDocument{%
   \@i fpackagel oaded{dateti me2-cal c}%
      \renewcommand*{\@dtm@setusecalc}{}%
81
   }%
82
    {%
83
      \renewcommand*{\@dtm@setusecalc}{%
84
        \PackageError{datetime2}{You must load 'datetime2-calc'
85
        package to use option 'showdow' }{Try one of the following: ^^J
86
        pass 'calc' option to 'datetime2' package when you load it^^J
87
        or move 'showdow' option to 'datetime2' package option list^^J
        or move \string\DTLsetup\space to the preamble. }%
89
      }%
90
    }%
91
92 }
```

cal c This option will load the datetime2-calc which uses the pgfcalendar package to compute the day of week and offsets. The package is loaded at the end of this one.

93 \DeclareOptionX{calc}{\@dtm@setusecalc}

showdow

Boolean key to determine whether or not to show the day of week for the styles that can show the day of week. If this is switched on, then datetime2-calc is required. If this key is set later in the document with \DTMsetup, then the datetime2-calc package must previously be loaded for it to have an effect.

```
94 \defi ne@bool key{dateti me2. sty}[DTM]{showdow}[true]{%
95  \i fDTMshowdow \@dtm@setusecalc \fi
96 }
97 \DTMshowdowfal se
```

```
\@dtm@warning Warning messages.
                      98 \newcommand* { \@dtm@warni ng} [1] {%
                          \if@dtm@warn
                            \PackageWarni ng{dateti me2}{#1}%
                     101
                          \fi
                     102 }
               warn Allow user to suppress package warnings.
                     103 \define@bool key{dateti me2. sty}[@dtm@]{warn}[true]{}
                     104 \@dtm@warntrue
\@dtm@initialstyle
                     105 \newcommand*{\@dtm@i ni ti al style}{}
              style Set the style. This automatically sets useregional = false.
                     106\define@key{datetime2.sty}{style}{%
                          \renewcommand*{\@dtm@initialstyle}{#1}%
                          \ifstrempty{#1}%
                     108
                     109
                          {}%
                     110
                           \renewcommand*{\DTMi fcaseregi onal }[3]{##1}%
                     111
                     112
                          }%
                     113 }
                        Pass any unknown options to tracklang. This will automatically switch the
                      useregional setting to text.
                     114 \Decl areOpti onX*{%
                          \TrackPredefi nedDi al ect{\CurrentOpti on}%
                          \renewcommand*{\DTMi fcaseregi onal }[3]{#2}%
                     116
                     117 }
                        Process options passed to this package:
                     118 \ProcessOptionsX
                        Disable calc option. If it's required, just load datetime2-calc with \usepackage.
                     119 \di sabl e@keys{dateti me2. sty}{cal c}
                      Disable style option. If it's required, just use \DTMsetup.
                     120 \di sabl e@keys{dateti me2. sty}{styl e}
                        Provide a way to set options after package has been loaded.
          \DTMsetup
                     121 \newcommand*{\DTMsetup}[1]{%
                     122 \def\@dtm@usecalc{}%
                     123 \setkeys{datetime2.sty}{#1}%
                     124 \@dtm@usecalc
```

125 }

9.1.1 Defaults

This section sets up the defaults.

\@dtm@parsedate

Parse date in the format $\langle year \rangle$ - $\langle month \rangle$ - $\langle day \rangle$. The arguments are expanded. (This is redefined by datetime2-calc.)

```
126 \def\@dtm@parsedate#1-#2-#3\@dtm@endparsedate{%
127 \edef\@dtm@year{\number#1}%
128 \edef\@dtm@month{\number#2}%
129 \edef\@dtm@day{\number#3}%
130 \def\@dtm@dow{-1}%
131}
```

\@dtm@parsetime

Define command to parse time in the format $\langle h \rangle : \langle m \rangle : \langle s \rangle$. The results are stored in \@dtm@hour, \@dtm@minute and \@dtm@second. The arguments are expanded.

```
132 \def\@dtm@parseti me#1: #2: #3\@dtm@endparseti me{%
133 \edef\@dtm@hour{\number#1}%
134 \edef\@dtm@mi nute{\number#2}%
135 \edef\@dtm@second{\number#3}%
136}
```

\@dtm@parsetimezn

Define command to parse time in the format $\langle h \rangle : \langle m \rangle : \langle s \rangle \langle znh \rangle : \langle znm \rangle$. The results are stored in \d minute, \d minute, \d message and \d message are expanded.

```
137 \def\@dtm@parseti mezn#1: #2: #3 #4\@dtm@endparseti mezn{%
138 \@dtm@parseti me#1: #2: #3\@dtm@endparseti me
139 \@dtm@parsezone{#4}%
140 }
```

\@dtm@parsezone

Define command to parse time zone in the format Z or $\langle znh \rangle$: $\langle znm \rangle$. The results are stored in \@dtm@timezonehour and \@dtm@timezoneminute. The arguments are expanded in the event that registers are used.

```
141 \newcommand*{\@dtm@parsezone}[1]{%
142
    \ifstrequal {#1}{Z}%
143
      \def\@dtm@timezonehour{+00}%
144
      \def\@dtm@timezoneminute{00}%
145
    }%
146
147
      \@dtm@parse@zone#1\@dtm@endparse@zone
148
149
150 }
151 \def\@dtm@parse@zone#1: #2\@dtm@endparse@zone{%
    \edef\@dtm@timezonehour{\number#1}%
    \edef\@dtm@timezoneminute{\number#2}%
153
154 }
```

@dtm@parsetimestamp

Parse date and time in ISO format $\langle YYYY \rangle - \langle MM \rangle - \langle DD \rangle T \langle hh \rangle : \langle mm \rangle : \langle sec \rangle \langle time zone \rangle$ where $\langle time zone \rangle$ may be Z or in the form $\langle hh \rangle : \langle mm \rangle$ (where $\langle hh \rangle$ includes the sign).

```
155 \def\@dtm@parseti mestamp#1-#2-#3T#4: #5: #6#7#8\@dtm@endparseti mestamp{%
156 \@dtm@parsedate#1-#2-#3\@dtm@endparsedate
157 \@dtm@parseti me#4: #5: #6#7\@dtm@endparseti me
158 \@dtm@parsezone{#8}%
159 }
```

\DTMsavefilemoddate

Not available for some engines.

```
160 \newcommand*{\DTMsavefilemoddate}[2]{%
    \@dtm@warning{Your TeX engine doesn't support accessing
    file modification dates}%
    \cslet{@dtm@#1@year}{0}%
163
    \cslet{@dtm@#1@month}{0}%
    \cslet{@dtm@#1@day}{0}%
    \csl et{@dtm@#1@dow}{-1}%
166
    \csl et{@dtm@#1@hour}{0}%
167
    \cslet{@dtm@#1@minute}{0}%
168
    \csl et{@dtm@#1@second}{0}%
    \cslet{@dtm@#1@TZhour}{0}%
170
    \csl et{@dtm@#1@TZmi nute}{0}%
171
172 }
```

Find out the current time. If PDFMTEX is being used, then it can be fetched from \pdfcreationdate

```
173 \i fdef\pdfcreati ondate
174 {%
```

Define commands to parse \pdfcreationdate

```
\def\@dtm@parsepdfdatetime#1: #2#3#4#5#6#7#8#9{%
176
       \def\@dtm@year{#2#3#4#5}%
       \def\@dtm@month{#6#7}%
177
       \def\@dtm@day{#8#9}%
178
       \@dtm@parsepdftime
179
180
     \def\@dtm@parsepdftime#1#2#3#4#5#6#7\@dtm@endparsepdfdatetime{%
181
       \def\@dtm@hour{#1#2}%
182
       \def\@dtm@mi nute{#3#4}%
       \def\@dtm@second{#5#6}%
184
       \ifstrequal \{\#7\}\{Z\}\%
185
       {%
186
         \def\@dtm@timezonehour{00}%
187
         \def\@dtm@timezoneminute{00}%
188
       }%
189
       {%
190
         \@dtm@parsepdftimezone#7%
       }%
192
    }
193
```

```
194 \def\@dtm@parsepdfti mezone#1' #2' {%
195 \def\@dtm@ti mezonehour{#1}%
196 \def\@dtm@ti mezonemi nute{#2}%
197 }%
```

Now parse \pdfcreationdate

uss \expandafter\@dtm@parsepdfdatetime\pdfcreationdate\@dtm@endparsepdfdatetime

Save the values.

```
199 \let\@dtm@currentyear\@dtm@year
200 \let\@dtm@currentmonth\@dtm@month
201 \let\@dtm@currentday\@dtm@day
202 \let\@dtm@currenthour\@dtm@hour
203 \let\@dtm@currentminute\@dtm@minute
204 \let\@dtm@currentsecond\@dtm@second
205 \let\@dtm@currenttimezonehour\@dtm@timezonehour
206 \let\@dtm@currenttimezoneminute\@dtm@timezoneminute
207%
```

LuaTFX doesn't provide \pdffilemoddate (but it does provide \pdfcreationdate).

```
\ifdef\pdffilemoddate
208
209
       \renewcommand*{\DTMsavefilemoddate}[2]{%
210
211
         \expandafter\@dtm@parsepdfdatetime\pdffilemoddate{#2}\@dtm@endparsepdfdatetime
         \cslet{@dtm@#1@year}{\@dtm@year}%
212
213
         \csl et{@dtm@#1@month}{\@dtm@month}%
         \cslet{@dtm@#1@day}{\@dtm@day}%
         \cslet{@dtm@#1@dow}{\@dtm@dow}%
215
         \cslet{@dtm@#1@hour}{\@dtm@hour}%
216
         \csl et{@dtm@#1@mi nute}{\@dtm@mi nute}%
217
218
         \csl et{@dtm@#1@second}{\@dtm@second}%
         \cslet{@dtm@#1@TZhour}{\@dtm@timezonehour}%
219
         \cslet{@dtm@#1@TZminute}{\@dtm@timezoneminute}%
220
      }
221
    }%
222
    {%
223
```

Lua time zone information provided by %z is OS dependent, so this might not work.

```
\i fdef\di rectl ua
224
225
         \renewcommand*{\DTMsavefilemoddate}[2]{%
           \expandafter\@dtm@parseluadatetime
227
             \di rectl ua{tex. pri nt(os. date(
228
                 "\expandafter\@gobble\string\%Y-%
229
                  \expandafter\@gobble\string\\m-\%
230
                  \expandafter\@gobble\string\%d-%
231
                  \expandafter\@gobble\string\%w
232
                  \expandafter\@gobble\string\%H: %
233
                  \expandafter\@gobble\string\\M: %
                  \expandafter\@gobble\string\%S
235
```

```
\expandafter\@gobble\string\%z",
236
                Ifs. attributes("#2"). modification))}%
237
            \@dtm@endparseluadatetime
238
            \cslet{@dtm@#1@year}{\@dtm@year}%
239
            \cslet{@dtm@#1@month}{\@dtm@month}%
            \cslet{@dtm@#1@day}{\@dtm@day}%
241
            \csl et{@dtm@#1@dow}{\@dtm@dow}%
242
            \cslet{@dtm@#1@hour}{\@dtm@hour}%
243
            \csl et{@dtm@#1@mi nute}{\@dtm@mi nute}%
244
            \cslet{@dtm@#1@second}{\@dtm@second}%
245
            \cslet{@dtm@#1@TZhour}{\@dtm@TZhour}%
246
            \csl et{@dtm@#1@TZmi nute}{\@dtm@TZmi nute}%
248
         \def\@dtm@parseluadatetime#1-#2-#3-#4 #5:#6:#7 #8\@dtm@endparseluadatetime{%
249
            \edef\@dtm@year{\number#1}%
250
            \edef\@dtm@month{\number#2}%
251
            \edef\@dtm@day{\number#3}%
252
            \edef\@dtm@dow{\number#4}%
253
            \edef\@dtm@hour{\number#5}%
254
            \edef\@dtm@mi nute{\number#6}%
            \edef\@dtm@second{\number#7}%
256
            \@dtm@parsel uati mezone#8000000\@dtm@endparsel uati mezone
257
258
         \def\@dtm@parseluatimezone#1#2#3#4#5#6{%
259
            \ifstrequal {#1}{+}%
260
            {%
261
               \def\@dtm@TZhour{#1#2#3}%
262
263
              \ifstrequal {#4}{:}%
               {%
264
                  \def\@dtm@TZminute{#5#6}%
265
               }%
267
               {%
                  \def\@dtm@TZmi nute{#4#5}%
268
              }%
269
            }%
270
            {%
271
               \ifstrequal {#1}{-}%
272
               {%
273
274
                  \def\@dtm@TZhour{#1#2#3}%
                  \ifstrequal {#4}{:}%
275
                  {%
276
                     \def\@dtm@TZminute{#5#6}%
277
                  }%
278
                  {%
279
                     \def\@dtm@TZminute{#4#5}%
280
                  }%
281
               }%
282
               {%
283
                 \ifstrequal {#1}{Z}%
284
```

```
{%
285
                    \def\@dtm@TZhour{0}%
286
                    \def\@dtm@TZmi nute{0}%
287
                  }%
288
                  {%
289
                    \def\@dtm@TZhour{#1#2}%
290
                    \ifstrequal {#3}{:}%
291
292
                    {%
                        \def\@dtm@TZminute{#4#5}%
293
                    }%
294
                    {%
295
                        \def\@dtm@TZminute{#3#4}%
296
                    }%
297
                  }%
298
               }%
299
             }%
300
             \@@dtm@parseluatimezone
301
          }
302
          \def\@@dtm@parseluatimezone#1\@dtm@endparseluatimezone{%
303
304
305
306
       {}
     }%
307
308 }%
309 {%
```

\pdfcreationdate not defined. By a process of elimination, the TeX engine is either XeTeX or it's very old. (LuaTeX recognises \pdfcreationdate.) In this case, the seconds and time zone can't be obtained. The hour and minute need to be calculated from TeX's \time primitive.

```
\count@=\time\relax
310
311
      \divide\count@ by 60\relax
      \edef\@dtm@currenthour{\number\count@}%
312
      \multiply\count@ by -60\relax
313
314
      \advance\count@ by \time\relax
      \edef\@dtm@currentmi nute{\number\count@}%
315
      \newcommand*{\@dtm@currentsecond}{00}%
316
      \newcommand\@dtm@currentti mezonehour{00}%
317
      \newcommand\@dtm@currenttimezoneminute{00}%
318
```

Get the day, month and year from T_FX's primitives.

```
319 \edef\@dtm@currentyear{\number\year}%
320 \edef\@dtm@currentmonth{\number\month}%
321 \edef\@dtm@currentday{\number\day}%
322}
```

Make \DTMsavefilemoddate robust.

323 \robustify\DTMsavefilemoddate

Current day of week defaults to -1 (that is, ignore it).

```
\@dtm@currentdow
```

324 \newcommand*{\@dtm@currentdow}{-1}

Allow XALTEX users a way of manually setting the current time zone.

```
\DTMsetcurrentzone
                      325 \newcommand*{\DTMsetcurrentzone}[2]{%
                            \renewcommand\@dtm@currenttimezonehour{#1}%
                            \renewcommand\@dtm@currenttimezoneminute{#2}%
                      328 }
              \today
                      329 \renewcommand*{\today}{%
                      330 \DTMdisplaydate
                            {\@dtm@currentyear}%
                           {\@dtm@currentmonth}%
                           {\@dtm@currentday}%
                            {\@dtm@currentdow}%
                      334
                      335 }
              \Today First letter upper case version.
                      зз6 \newcommand*{\Today}{%
                      337 \DTMDisplaydate
                      338 {\@dtm@currentyear}%
                      339
                           {\@dtm@currentmonth}%
                            {\@dtm@currentday}%
                            {\@dtm@currentdow}%
                      342 }
                         \label{lem:decomposition} $$ \operatorname{DTMdisplaydate}_{\langle year\rangle}_{\langle month\rangle}_{\langle day\rangle}_{\langle day \ of \ week\rangle}$$
    \DTMdi spl aydate
                        Display the given date. If the day of week is negative, ignore it. The default style
                        ignores it regardless.
                       343 \newcommand*\DTMdisplaydate[4]{%
                       344 \number#1\dtm@yearmonthsep\DTMtwodigits{#2}\dtm@monthdaysep\DTMtwodigits{#3}%
                      345 }%
   \DTMDi spl aydate First letter upper case version. Defaults to \DTMdisplaydate.
```

346\newcommand*{\DTMDi spl aydate}{\DTMdi spl aydate}

\DTMdate Display date where date is specified in the format $\langle yyyy \rangle - \langle mm \rangle - \langle dd \rangle$. Use \expandafter in case argument is a control sequence containing the date.

This command isn't expandable

```
347\newrobustcmd*{\DTMdate}[1]{%
348 \expandafter\@dtm@parsedate#1\@dtm@endparsedate
349 \DTMdisplaydate{\@dtm@year}{\@dtm@month}{\@dtm@day}{\@dtm@dow}%
350}
```

```
\DTMDate Upper case version.
                  351 \newrobustcmd*{\DTMDate}[1]{%
                       \expandafter\@dtm@parsedate#1\@dtm@endparsedate
                       \DTMDisplaydate{\@dtm@year}{\@dtm@month}{\@dtm@day}{\@dtm@dow}%
                  354 }
\DTMcurrenttime Display the current time.
                  355 \newcommand*{\DTMcurrenttime}{%
                  356 \DTMdisplaytime
                  357 {\@dtm@currenthour}%
                  358 {\@dtm@currentminute}%
                       {\@dtm@currentsecond}%
                  360 }
\DTMdisplaytime
                     \label{lem:decomposition} $$ \operatorname{DTMdisplaytime}_{\langle hour \rangle}_{\langle minute \rangle}_{\langle sec \rangle} $$
                   Display the given time.
                  361 \newcommand*\DTMdisplaytime[3]{%
                  362 \DTMtwodigits{#1}\dtm@hourminsep\DTMtwodigits{#2}%
                       \ifDTMshowseconds\dtm@minsecsep\DTMtwodigits{#3}\fi
                  364 }%
                  Display date where time is specified in the format \( \lambda our \rangle : \lambda minute \rangle : \lambda seconds \rangle .
        \DTMtime
                   This uses \expandafter in case argument is a control sequence containing
                   the time. Not expandable.
                  365 \newrobustcmd*{\DTMtime}[1]{%
                  366 \@dtm@parsetime#1\@dtm@endparsetime
                       \DTMdisplaytime{\@dtm@hour}{\@dtm@minute}{\@dtm@second}%
                  368 }
\DTMcurrentzone
                  Display the current time zone.
                  369 \newcommand*{\DTMcurrentzone}{%
                  370 \DTMdi spl ayzone
                  371 {\@dtm@currenttimezonehour}%
                  372 {\@dtm@currenttimezoneminute}%
                  373 }
374 \newcommand* {\DTMdisplayzone} [2] {%
                  375 \ifboolexpe
                  376 { bool {DTMshowi soZ}
                        and test{\ifnumequal {#1}{0}}
```

and test{\ifnumequal {#2}{0}}

379 }% 380 {%

```
Z%
        381
        382 }%
        383 {%
             \in fnum#1<0\leq se+fi DTMtwodigits{#1}%
        384
             \ifDTMshowzonemi nutes\dtm@hourmi nsep\DTMtwodi gi ts{#2}\fi
        386 }%
        387 }
\DTMnow Current date, time and time zone.
        388 \newcommand*{\DTMnow}{%
        389 \DTMdisplay
        390 {\@dtm@currentyear}
        391 {\@dtm@currentmonth}
        392 {\@dtm@currentday}
        393 {\@dtm@currentdow}
        394 {\@dtm@currenthour}%
        395 {\@dtm@currentminute}%
        396 {\@dtm@currentsecond}%
        397 {\@dtm@currenttimezonehour}%
        398 {\@dtm@currenttimezoneminute}%
        399 }
\DTMNow Current date, time and time zone.
        400 \newcommand*{\DTMNow}{%
        401 \DTMDisplay
        402 {\@dtm@currentyear}
        403 {\@dtm@currentmonth}
        404 {\@dtm@currentday}
        405 {\@dtm@currentdow}
        406 {\@dtm@currenthour}%
        407 {\@dtm@currentminute}%
        408 {\@dtm@currentsecond}%
        409 {\@dtm@currenttimezonehour}%
        410 {\@dtm@currenttimezoneminute}%
        411 }
```

\DTMdi spl ay

 $\label{eq:lower_lower} $$ \operatorname{DTMdisplay}_{\langle YYYY\rangle}_{\langle MM\rangle}_{\langle DD\rangle}_{\langle DOW\rangle}_{\langle hh\rangle}_{\langle mm\rangle}_{\langle ss\rangle}_{\langle TZh\rangle}_{\langle TZm\rangle}$$

Display the date and time.

```
412 \newcommand*{\DTMdi spl ay}[9]{%
413 \ifDTMshowdate
414 \DTMdi spl aydate{#1}{#2}{#3}{#4}%
415 \dtm@dateti mesep
416 \fi
417 \DTMdi spl ayti me
```

```
{#5}%
418
    {#6}%
419
    {#7}%
420
421 \ifDTMshowzone
422
    \dtm@timezonesep
     \DTMdi spl ayzone
423
      {#8}%
424
     {#9}%
425
426 \fi
427 }
```

\DTMDi spl ay

First letter upper case version. Defaults to \DTMdisplay.

428 \newcommand* {\DTMDi spl ay} {\DTMdi spl ay}

9.1.2 Styles

Provide user level commands for displaying number as two digits. (Truncate if over 99, to allow for converting year to two digits).

\DTMtwodigits

```
429 \newcommand* {\DTMtwodigits}[1]{%
430 \i fnum#1<0
    -\DTMtwodigits{-#1}%
431
432 \else
      \i fnum#1<100
433
        \i fnum#1<10
434
           0\number#1
435
        \el se
436
           \number#1
437
        \fi
438
      \el se
439
```

\numexpr rounds rather than truncates integer division, which is a little awkward to get around in an expandable context.

\DTMcentury Expands to the given number divided by 100 rounded upwards (in absolute terms). Provided in case the user just wants the century.

```
448 \newcommand* {\DTMcentury}[1]{%
     \i fnum#1<0
449
       -\DTMcentury{-#1}%
450
451
     \el se
452
       \i fnum\numexpr#1-(#1/100)*100<1
           \number\numexpr#1/100\relax
453
       \el se
454
           \mbox{number}\mbox{numexpr}(\#1/100) + 1\mbox{rel ax}
       \fi
456
     \fi
457
458 }
```

\DTMdi vhundred Expands to the given number modulo 100.

```
459 \newcommand*{\DTMdi vhundred}[1]{%
    \i fnum#1<0
460
     -\DTMdi vhundred{-#1}%
461
462
    \el se
      \ifnum\numexpr#1-(#1/100)*100<0
463
         \mbox{number}(\#1)/100-1\relax
464
      \el se
465
         466
      \fi
467
    \fi
468
469 }
```

\DTMtexorpdfstring Provide user with a command that will use hyperref's \texorpdfstring if hyperref has been loaded. If hyperref isn't loaded it just does the first argument.

```
470 \newcommand* {\DTMtexorpdfstring}[2]{#1}
471 \AtBegi nDocument{%
    \@i fpackagel oaded{hyperref}%
472
473
       \renewcommand*{\DTMtexorpdfstring}{\texorpdfstring}%
474
     }%
475
476
    {}%
477 }
```

Access separator:

\DTMsep

478 \newcommand* {\DTMsep}[1] {\csname dtm@#1sep\endcsname}

Date-only styles are stored internally as \@dtm@datestyle@\label\, timeonly styles are stored internally as \@dtm@timestyle@\(label\), zone-only styles are stored internally as \@dtm@zonestyle@\\ label\.

\DTMnewdatestyle Define a new date-only style. This should only redefine \DTMdisplaydate and

\DTMDisplaydate, which may or may not use the separators \dtm@yearmonthsep and \dtm@monthdaysep.

```
479 \newcommand* {\DTMnewdatestyle}[2]{%
    \ifcsdef{@dtm@datestyle@#1}%
481
       \PackageError{datetime2}{Date style '#1' already exists}{}%
482
    }%
483
    {%
484
        \csdef{@dtm@datestyle@#1}{#2}%
485
    }%
486
487 }
```

\DTMnewti mestyle Define a new time-only style. This should only redefine \DTMdisplaytime, which may or may not use the separators \dtm@hourminsep and \dtm@minsecsep.

```
488 \newcommand* { \DTMnewtimestyle}[2]{%
     \ifcsdef{@dtm@timestyle@#1}%
489
490
       \PackageError{datetime2}{Time style '#1' already exists}{}%
491
     }%
492
     {%
493
        \csdef{@dtm@timestyle@#1}{#2}%
494
     }%
495
496 }
```

\DTMnewti mezone Define a new zone-only style. This should only redefine \DTMdisplayzone, which may or may not use the separator \dtm@hourminsep.

```
497 \newcommand* { \DTMnewzonestyle} [2] {%
     \ifcsdef{@dtm@zonestyle@#1}%
499
       \PackageError{datetime2}{Zone style '#1' already exists}{}%
500
     }%
501
502
        \csdef{@dtm@zonestyle@#1}{#2}%
503
     }%
504
505 }
```

Zone styles may use mappings to use a regional time zone (such as GMT or BST). It's up to the language modules to define these mappings. A mapping for time zone $\langle TZh \rangle$: $\langle TZm \rangle$ is stored in $\backslash \mathbb{Q}dtm\mathbb{Q}zonemap\mathbb{Q}\langle TZh \rangle$: $\langle TZm \rangle$.

\DTMdefzonemap

```
\DTMdefzonemap\{\langle TZh\rangle\}\{\langle TZm\rangle\}\{\langle map\rangle\}
```

This will override any previous mapping for the given time zone.

```
506\newcommand*{\DTMdefzonemap}[3]{%
507 \csdef{@dtm@zonemap@\DTMtwodigits{#1}:\DTMtwodigits{#2}}{#3}%
```

```
508 }
```

```
usezonemapordefault
                                                                       Expands to the mapping or the default if not defined.
                                                                       509 \newcommand*{\DTMusezonemapordefault}[2]{%
                                                                                       \ifcsundef{@dtm@zonemap@\DTMtwodigits{#1}:\DTMtwodigits{#2}}%
                                                                       511
                                                                                               \ifnum#1<0\else+\fi
                                                                       512
                                                                                              \DTMtwodigits{#1}%
                                                                       513
                                                                                              \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodigits{#2}\fi
                                                                       514
                                                                       515
                                                                       516
                                                                                       {\csname @dtm@zonemap@\DTMtwodigits{#1}: \DTMtwodigits{#2}\endcsname}%
                                                                       517 }
                \DTMusezonemap
                                                                      Expands to the mapping. (No check if defined.)
                                                                       518 \newcommand* {\DTMusezonemap}[2]{%
                                                                                       \csname @dtm@zonemap@\DTMtwodigits{#1}: \DTMtwodigits{#2}\endcsname
                                                                       520 }
                \DTMhaszonemap
                                                                       521 \newcommand*{\DTMhaszonemap}[4]{%
                                                                                      \ifcsundef{@dtm@zonemap@\DTMtwodigits{#1}:\DTMtwodigits{#2}}{#4}{#3}%
                                                                       523 }
                                                                        Undefines the given zone mapping. No check is made to determine if the map
                       \DTMclearmap
                                                                          exists.
                                                                       524 \newcommand* {\DTMclearmap} [2] {%
                                                                       \colored{1} \col
                                                                       526 }
                          \DTMshowmap Debugging command.
                                                                       527 \newcommand*{\DTMshowmap}[2]{%
                                                                       \colone{10} \col
                                                                       529 }
                \DTMresetzones
                                                                         Regional modules should use this before setting their local zones, so that users
                                                                          can unset previously defined zones that are outside the region if they require.
                                                                          By default this does nothing, so no modifications are made.
                                                                       530 \newcommand*{\DTMresetzones}{}
                                                                        Provide a command to set the time zone abbreviations to the military/NATO
         \DTMNatoZoneMaps
                                                                          style.
                                                                       531 \newcommand*{\DTMNatoZoneMaps}{%
                                                                                        \defzonemap{01}{00}{A}% Alpha time zone
                                                                                        \defzonemap{02}{00}{B}% Bravo time zone
                                                                       533
                                                                                       \defzonemap{03}{00}{C}% Charlie time zone
                                                                       534
                                                                                        \defzonemap{04}{00}{D}% Delta time zone
                                                                                       \defzonemap{05}{00}{E}% Echo time zone
                                                                                       \defzonemap{06}{00}{F}% Foxtrot time zone
```

```
\defzonemap{07}{00}{G}\% Golf time zone
    \defzonemap{08}{00}{H}% Hotel time zone
539
    \defzonemap{09}{00}{I}% India time zone
540
    \defzonemap{10}{00}{K}% Kilo time zone
541
    \defzonemap{11}{00}{L}% Lima time zone
    \defzonemap{12}{00}{M}% Mike time zone
543
    \defzonemap{-01}{00}{N}% November time zone
544
    \defzonemap{-02}{00}{0}% Oscar time zone
545
    \defzonemap{-03}{00}{P}% Papa time zone
546
    \defzonemap{-04}{00}{Q}% Quebec time zone
547
    \defzonemap{-05}{00}{R}% Romeo time zone
548
    \defzonemap{-06}{00}{S}% Sierra time zone
549
550
    \defzonemap{-07}{00}{T}% Tango time zone
551
    \defzonemap{-08}{00}{U}% Uniform time zone
    \defzonemap{-09}{00}{V}% Victor time zone
552
    \defzonemap{-10}{00}{W}% Whiskey time zone
553
    \defzonemap{-11}{00}{X}% X-ray time zone
554
    \defzonemap{-12}{00}{Y}% Yankee time zone
555
    \defzonemap{00}{00}{Z}\% Zulu time zone
556
557 }
```

\DTMnewstyle

 $\label{label} $$ \operatorname{DTMnewstyle}(\langle label\rangle)_{\langle date\ style\ definition\rangle}_{\langle cone\ style\ definition\rangle}_{$

Define a new style. The full format redefines \DTMdisplay and \DTMDisplay.

```
558 \newcommand*{\DTMnewstyle}[5]{%
     \ifcsdef{@dtm@style@#1}%
     {%
560
       \PackageError{datetime2}{Style '#1' already exists}{}%
561
     }%
562
     {%
563
       \DTMnewdatestyle{#1}{#2}%
564
       \DTMnewtimestyle{#1}{#3}%
565
       \DTMnewzonestyle{\#1}{\#4}%
566
        \csdef{@dtm@style@#1}{%
567
           \csuse{@dtm@datestyle@#1}%
568
           \csuse{@dtm@timestyle@#1}%
569
           \csuse{@dtm@zonestyle@#1}%
570
           #5%
571
        }%
572
     }%
573
574 }
```

\DTMsetdatestyle

```
575 \newrobustcmd*{\DTMsetdatestyle}[1]{%
576 \ifcsdef{@dtm@datestyle@#1}%
```

```
{\csuse{@dtm@datestyle@#1}}%
                    578
                         {%
                           \PackageError{datetime2}{Date style '#1' not defined}{}%
                    579
                         }%
                    580
                    581 }
  \DTMsettimestyle
                    582 \newrobustcmd*{\DTMsettimestyle}[1]{%
                         \ifcsdef{@dtm@timestyle@#1}%
                         {\csuse{@dtm@timestyle@#1}}%
                    584
                         {%
                    585
                    586
                           \PackageError{datetime2}{Time style '#1' not defined}{}%
                         }%
                    587
                    588 }
  \DTMsetzonestyle
                    589 \newrobustcmd*{\DTMsetzonestyle}[1]{%
                         \ifcsdef{@dtm@zonestyle@#1}%
                         {\csuse{@dtm@zonestyle@#1}}%
                    591
                         {%
                    592
                           \PackageError{datetime2}{Zone style '#1' not defined}{}%
                    593
                         }%
                    594
                    595 }
      \DTMsetstyle
                    596 \newrobustcmd*{\DTMsetstyle}[1]{%
                         \ifcsdef{@dtm@style@#1}%
                         {\csuse{@dtm@style@#1}}%
                         {%
                    599
                    600
                            \let\dtm@unknownstyle\@dtm@unknownstyle
                            \ifcsdef{@dtm@datestyle#1}%
                    601
                               {\csuse{@dtm@datestyle@#1}\let\dtm@unknownstyle\@dtm@unknown@style}%
                               {\@dtm@warning{No date style '#1' defined}}%
                    603
                             \ifcsdef{@dtm@timestyle#1}%
                    604
                               {\csuse{@dtm@timestyle@#1}\let\dtm@unknownstyle\@dtm@unknown@style}%
                    605
                               {\@dtm@warning{No time style '#1' defined}}%
                    606
                             \ifcsdef{@dtm@zonestyle#1}%
                    607
                               \label{lem:constyle} $$ \csup {\csup e=\#1}\le t^2 e^{\csup e=\#1}\le t^2 e^{\csup e=\#1}. $$
                    608
                               {\@dtm@warning{No zone style '#1' defined}}%
                    609
                    610
                             \dtm@unknownstyle{#1}%
                    611
                         }%
                    612 }
\@dtm@unknownstyle
                    613 \newcommand* { \@dtm@unknownstyle} [1] {%
                         \PackageError{datetime2}{Unknown style '#1'}{}%
                    615 }
```

∖@dtm@unknown@style

```
616 \newcommand* { \@dtm@unknown@style} [1] {%
     \@dtm@warning{No full style '#1' defined}{}%
618 }
   Define default style:
619 \DTMnewstyle
620 {default}%label
621
   {% date style
       \renewcommand*\DTMdisplaydate[4]{%
622
623
         \number##1\DTMsep{yearmonth}\DTMtwodigits{##2}%
         \DTMsep{monthday}\DTMtwodigits{##3}%
624
       }%
625
       \renewcommand*{\DTMDisplaydate}{\DTMdisplaydate}%
627 }%
   {% time style
628
       \renewcommand*\DTMdisplaytime[3]{%
629
         \DTMtwodigits{##1}\DTMsep{hourmin}\DTMtwodigits{##2}%
630
         \ifDTMshowseconds\DTMsep{mi nsec}\DTMtwodi gi ts{##3}\fi
631
632
       }%
633 }%
    {% zone style
634
      \renewcommand*{\DTMdisplayzone}[2]{%
635
        \i fbool expe
636
        { bool {DTMshowi soZ}
637
          and test{\ifnumequal{##1}{0}}
638
          and test{\ifnumequal {##2}{0}}
639
        }%
640
        {%
641
          Z%
642
        }%
643
        {%
644
         \in fnum##1<0\leq se+fi DTMtwodigits{##1}%
         \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodi gi ts{##2}\fi
646
        }%
647
      }%
648
   }%
649
    {% full style
650
      \renewcommand*{\DTMdisplay}[9]{%
651
       \i fDTMshowdate
652
653
        \DTMdi spl aydate{##1}{##2}{##3}{##4}%
        \DTMsep{datetime}%
654
       \fi
655
       \DTMdisplaytime
656
        {##5}%
657
        {##6}%
658
        {##7}%
659
```

\i fDTMshowzone

\DTMsep{timezone}%

\DTMdi spl ayzone

660

661

662

```
{##8}%
663
         {##9}%
664
       \fi
665
      }%
666
      \renewcommand*{\DTMDisplay}{\DTMdisplay}%
667
668 }
   Define iso style which ignores the separator settings:
669 \DTMnewstyle
   {i so}%l abel
    {% date style
       \renewcommand*\DTMdisplaydate[4]{%
672
         \number##1-\DTMtwodigits{##2}-\DTMtwodigits{##3}%
673
674
675
       \renewcommand*{\DTMDisplaydate}{\DTMdisplaydate}%
   }%
676
    {% time style
677
       \renewcommand*\DTMdisplaytime[3]{%
678
         \DTMtwodigits{##1}: \DTMtwodigits{##2}%
679
         \ifDTMshowseconds: \DTMtwodigits{##3}\fi
680
       }%
681
682 }%
    {% zone style
683
      \renewcommand*{\DTMdisplayzone}[2]{%
684
        \i fbool expe
685
686
        { bool {DTMshowi soZ}
          and test{\ifnumequal {##1}{0}}
687
          and test{\ifnumequal {##2}{0}}
688
        }%
689
        {%
690
          Z%
691
        }%
692
        {%
693
         \infty fnum##1<0\le se+fi DTMtwodigits{##1}%
694
         \ifDTMshowzonemi nutes: \DTMtwodi gi ts{##2}\fi
695
        }%
696
      }%
697
698 }%
    {% full style
699
      \renewcommand*{\DTMdisplay}[9]{%
700
       \ifDTMshowdate
701
        \DTMdi spl aydate{##1}{##2}{##3}{##4}%
702
        T%
703
       \fi
704
       \DTMdisplaytime
705
        {##5}%
706
        {##6}%
707
        {##7}%
708
       \i fDTMshowzone
        \DTMdi spl ayzone
710
```

```
{##8}%
711
         {##9}%
712
       \fi
713
      }%
714
      \renewcommand*{\DTMDi spl ay}{\DTMdi spl ay}%
715
716 }
   Define pdf style which converts into a format that can be used in \pdfinfo:
717 \DTMnewstyle
718 {pdf}%| abel
    {% date style
       \renewcommand*\DTMdisplaydate[4]{%
720
         D: \number##1 % space intended
721
         \DTMtwodigits{##2}\DTMtwodigits{##3}%
722
723
       \renewcommand*{\DTMDi spl aydate}{\DTMdi spl aydate}%
724
725 }%
    {% time style
726
       \renewcommand*\DTMdisplaytime[3]{%
727
728
         \DTMtwodi gi ts{##1}\DTMtwodi gi ts{##2}\DTMtwodi gi ts{##3}%
       }%
729
730 }%
731 {% zone style
      \renewcommand*{\DTMdisplayzone}[2]{%
732
        \i fbool expe
733
734
        { bool {DTMshowi soZ}
          and test{\ifnumequal {##1}{0}}
735
          and test{\ifnumequal {##2}{0}}
736
        }%
737
        {%
738
          Z%
739
        }%
740
        {%
741
         \pi4 \i fnum##1<0\el se+\fi \DTMtwodi gi ts{##1}' \DTMtwodi gi ts{##2}' %
742
        }%
743
      }%
744
745 }%
746 {% full style
      \renewcommand*{\DTMdisplay}[9]{%
747
       \DTMdi spl aydate{##1}{##2}{##3}{##4}%
748
       \DTMdi spl ayti me{##5}{##6}{##7}%
749
       \DTMdi spl ayzone{##8}{##9}%
750
751
      }%
      \renewcommand*{\DTMDisplay}{\DTMdisplay}%
752
753 }
   Define yyyymd style:
754 \DTMnewstyle
755 {yyyymd}%label
756 {% date style
```

```
\renewcommand*\DTMdisplaydate[4]{%
757
         \number##1
758
         \DTMsep{yearmonth}%
759
         \number##2
760
         \DTMsep{monthday}%
761
         \number##3
762
       }%
763
       \renewcommand*{\DTMDi spl aydate}{\DTMdi spl aydate}%
764
765
    {% time style
766
       \renewcommand*\DTMdisplaytime[3]{%
767
         \DTMtwodi gi ts{##1}\DTMsep{hourmi n}\DTMtwodi gi ts{##2}%
768
769
         \ifDTMshowseconds\DTMsep{minsec}\DTMtwodigits{##3}\fi
770
       }%
771 }%
772 {% zone style
      \renewcommand*{\DTMdisplayzone}[2]{%
        \i fbool expe
774
        { bool {DTMshowi soZ}
775
          and test{\ifnumequal{##1}{0}}
776
          and test\{ \inf \{\#2\} \{0\} \}
777
        }%
778
        {%
779
          Z%
        }%
781
        {%
782
         \ifnum##1<0\else+\fi\DTMtwodigits{##1}%
783
         \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodi gi ts{##2}\fi
784
785
        }%
      }%
786
787 }%
788
    {% full style
      \renewcommand*{\DTMdisplay}[9]{%
789
       \ifDTMshowdate
790
        \DTMdi spl aydate{##1}{##2}{##3}{##4}%
791
792
        \DTMsep{datetime}%
       \fi
793
       \DTMdisplaytime
794
        {##5}%
        {##6}%
796
        {##7}%
797
       \i fDTMshowzone
798
799
        \DTMsep{timezone}%
        \DTMdi spl ayzone
800
         {##8}%
801
         {##9}%
802
       \fi
803
      }%
804
      \renewcommand*{\DTMDi spl ay}{\DTMdi spl ay}%
805
```

```
806 }
   Define ddmmyyyy style:
807 \DTMnewstyle
808 {ddmmyyyy}%label
809 {% date style
       \renewcommand*\DTMdisplaydate[4]{%
         \DTMtwodigits{##3}\DTMsep{monthday}%
811
         \DTMtwodigits{##2}\DTMsep{yearmonth}%
812
         \number##1
813
814
       \renewcommand*{\DTMDi spl aydate}{\DTMdi spl aydate}%
815
   }%
816
    {% time style
818
       \renewcommand*\DTMdisplaytime[3]{%
         \DTMtwodi gi ts{##1}\DTMsep{hourmi n}\DTMtwodi gi ts{##2}%
819
         \ifDTMshowseconds\DTMsep{minsec}\DTMtwodigits{##3}\fi
820
       }%
821
822
    }%
823
    {% zone style
      \renewcommand*{\DTMdisplayzone}[2]{%
824
        \i fbool expe
825
        { bool {DTMshowi soZ}
826
          and test{\ifnumequal{##1}{0}}
827
          and test{\inftyifnumequal{##2}{0}}
828
        }%
829
        {%
830
          Z%
831
        }%
832
        {%
833
         \infty fnum##1<0\le se+fi DTMtwodigits{##1}%
834
         \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodigits{##2}\fi
835
        }%
836
      }%
837
838
    {% full style
839
      \renewcommand*{\DTMdisplay}[9]{%
840
       \ifDTMshowdate
        \DTMdi spl aydate{##1}{##2}{##3}{##4}%
842
        \DTMsep{datetime}%
843
       \fi
844
       \DTMdisplaytime
845
        {##5}%
846
        {##6}%
847
848
        {##7}%
       \i fDTMshowzone
849
        \DTMsep{timezone}%
850
        \DTMdi spl ayzone
851
         {##8}%
852
         {##9}%
853
```

```
\fi
854
      }%
855
      \renewcommand*{\DTMDisplay}{\DTMdisplay}%
856
857 }
   Define dmyyyy style:
858 \DTMnewstyle
859 {dmyyyy}%label
   {% date style
       \renewcommand*\DTMdisplaydate[4]{%
861
         \number##3
862
         \DTMsep{monthday}%
863
         \number##2
864
         \DTMsep{yearmonth}%
865
866
         \number##1
867
       \renewcommand*{\DTMDi spl aydate}{\DTMdi spl aydate}%
868
   }%
869
870
    {% time style
871
       \renewcommand*\DTMdisplaytime[3]{%
         \DTMtwodi gi ts{##1}\DTMsep{hourmi n}\DTMtwodi gi ts{##2}%
872
         \ifDTMshowseconds\DTMsep{minsec}\DTMtwodigits{##3}\fi
873
       }%
874
875 }%
    {% zone style
876
      \renewcommand*{\DTMdisplayzone}[2]{%
877
        \i fbool expe
878
        { bool {DTMshowi soZ}
879
          and test{\ifnumequal {##1}{0}}
880
          and test{\ifnumequal{##2}{0}}
881
        }%
882
        {%
883
          Z%
884
        }%
885
        {%
886
         \in fnum##1<0\le se+fi DTMtwodigi ts{##1}%
887
         \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodigits{##2}\fi
888
        }%
889
      }%
890
891 }%
    {% full style
892
      \renewcommand*{\DTMdisplay}[9]{%
893
       \ifDTMshowdate
894
        \DTMdi spl aydate{##1}{##2}{##3}{##4}%
895
896
        \DTMsep{datetime}%
897
       \DTMdisplaytime
898
        {##5}%
899
        {##6}%
900
        {##7}%
901
```

```
\i fDTMshowzone
        \DTMsep{timezone}%
903
        \DTMdi spl ayzone
904
         {##8}%
905
         {##9}%
906
       \fi
907
      }%
908
      \renewcommand*{\DTMDisplay}{\DTMdisplay}%
909
910
   }
   Define dmyy style:
911 \DTMnewstyle
912 {dmyy}%| abel
   {% date style
914
       \renewcommand*\DTMdisplaydate[4]{%
         \number##3 % space intended
915
         \DTMsep{monthday}%
916
         \number##2 % space intended
917
918
         \DTMsep{yearmonth}%
919
         \DTMtwodigits{##1}%
       }%
920
       \renewcommand*{\DTMDi spl aydate}{\DTMdi spl aydate}%
921
922 }%
923 {% time style
       \renewcommand*\DTMdisplaytime[3]{%
924
         \DTMtwodigits{##1}\DTMsep{hourmin}\DTMtwodigits{##2}%
925
         \i fDTMshowseconds\DTMsep{mi nsec}\DTMtwodi gi ts{##3}\fi
926
       }%
927
928 }%
    {% zone style
      \renewcommand*{\DTMdisplayzone}[2]{%
930
        \i fbool expe
931
        { bool {DTMshowi soZ}
932
          and test{\ifnumequal{##1}{0}}
933
          and test{\ifnumequal {##2}{0}}
934
        }%
935
        {%
936
          Z%
937
        }%
938
        {%
939
         \in fnum##1<0\le se+fi DTMtwodigi ts{##1}%
940
         \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodi gi ts{##2}\fi
941
942
        }%
      }%
943
944
   }%
    {% full style
945
      \renewcommand*{\DTMdisplay}[9]{%
946
       \ifDTMshowdate
947
        \DTMdi spl aydate{##1}{##2}{##3}{##4}%
948
        \DTMsep{datetime}%
949
```

```
\fi
       \DTMdisplaytime
951
        {##5}%
952
        {##6}%
953
        {##7}%
954
       \i fDTMshowzone
955
        \DTMsep{timezone}%
956
        \DTMdi spl ayzone
957
         {##8}%
958
         {##9}%
959
       \fi
960
      }%
961
962
      \renewcommand*{\DTMDisplay}{\DTMdisplay}%
963
   Define mmddyyyy style:
964 \DTMnewstyle
   {mmddyyyy}%label
966
    {% date style
967
       \renewcommand*\DTMdisplaydate[4]{%
         \DTMtwodigits{##2}\DTMsep{monthday}%
968
         \DTMtwodigits{##3}\DTMsep{dayyear}%
969
         \number##1
970
       }%
971
       \renewcommand*{\DTMDi spl aydate}{\DTMdi spl aydate}%
972
973
   }%
    {% time style
974
       \renewcommand*\DTMdisplaytime[3]{%
975
         \DTMtwodigits{##1}\DTMsep{hourmin}\DTMtwodigits{##2}%
976
977
         \ifDTMshowseconds\DTMsep{minsec}\DTMtwodigits{##3}\fi
       }%
978
    }%
979
    {% zone style
980
      \renewcommand*{\DTMdisplayzone}[2]{%
981
        \i fbool expe
982
        { bool {DTMshowi soZ}
983
          and test{\inftyifnumequal{##1}{0}}
984
          and test\{ \inf \{\#2\} \{0\} \}
985
        }%
986
        {%
987
          Z%
988
        }%
989
        {%
990
         \in fnum##1<0\le se+fi DTMtwodigi ts{##1}%
991
         \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodigits{##2}\fi
992
        }%
993
      }%
994
    }%
995
    {% full style
996
      \renewcommand*{\DTMdisplay}[9]{%
997
```

```
\ifDTMshowdate
         \DTMdi spl aydate{##1}{##2}{##3}{##4}%
999
         \DTMsep{datetime}%
1000
1001
        \DTMdisplaytime
1002
         {##5}%
1003
         {##6}%
1004
         {##7}%
1005
        \i fDTMshowzone
1006
         \DTMsep{timezone}%
1007
         \DTMdi spl ayzone
1008
          {##8}%
1009
1010
          {##9}%
        \fi
1011
       }%
1012
       \renewcommand*{\DTMDisplay}{\DTMdisplay}%
1013
1014 }
   Define mdyyyy style:
1015 \DTMnewstyle
1016 {mdyyyy}%label
1017 {% date style
        \renewcommand*\DTMdisplaydate[4]{%
1018
          \number##2 % space intended
1019
          \DTMsep{monthday}%
1020
1021
          \number##3 % space intended
          \DTMsep{dayyear}%
1022
          \number##1 % space intended
1023
1024
1025
        \renewcommand*{\DTMDisplaydate}{\DTMdisplaydate}%
    }%
1026
    {% time style
1027
        \renewcommand*\DTMdisplaytime[3]{%
1028
          \DTMtwodi gi ts{##1}\DTMsep{hourmi n}\DTMtwodi gi ts{##2}%
1029
          \i fDTMshowseconds\DTMsep{mi nsec}\DTMtwodi gi ts{##3}\fi
1030
       }%
1031
1032 }%
    {% zone style
1033
      \renewcommand*{\DTMdisplayzone}[2]{%
1034
         \i fbool expe
1035
         { bool {DTMshowi soZ}
1036
           and test{\ifnumequal{##1}{0}}
1037
           and test{\ifnumequal {##2}{0}}
1038
         }%
1039
         {%
1040
           Z%
1041
         }%
1042
         {%
1043
          \in fnum##1<0\leq se+fi DTMtwodigits{##1}%
1044
          \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodi gi ts{##2}\fi
1045
```

```
}%
1046
       }%
1047
1048 }%
    {% full style
1049
       \renewcommand*{\DTMdisplay}[9]{%
1050
        \i fDTMshowdate
1051
         \DTMdi spl aydate{##1}{##2}{##3}{##4}%
1052
         \DTMsep{datetime}%
1053
        \fi
1054
        \DTMdisplaytime
1055
         {##5}%
1056
         {##6}%
1057
1058
         {##7}%
        \i fDTMshowzone
1059
         \DTMsep{timezone}%
1060
         \DTMdi spl ayzone
1061
          {##8}%
1062
          {##9}%
1063
        \fi
1064
       }%
1065
       \renewcommand*{\DTMDi spl ay}{\DTMdi spl ay}%
1066
1067 }
   Define mdyy style:
1068 \DTMnewstyle
1069 {mdyy}%| abel
    {% date style
1070
        \renewcommand*\DTMdisplaydate[4]{%
1071
          \number##2 % space intended
1072
          \DTMsep{monthday}%
1073
          \number##3 % space intended
1074
          \DTMsep{dayyear}%
1075
          \DTMtwodigits{##1}%
1076
1077
        \renewcommand*{\DTMDi spl aydate}{\DTMdi spl aydate}%
1078
1079 }%
    {% time style
1080
        \renewcommand*\DTMdisplaytime[3]{%
1081
          \DTMtwodi gi ts{##1}\DTMsep{hourmi n}\DTMtwodi gi ts{##2}%
1082
          \ifDTMshowseconds\DTMsep{mi nsec}\DTMtwodi gi ts{##3}\fi
1083
        }%
1084
    }%
1085
    {% zone style
1086
       \renewcommand*{\DTMdisplayzone}[2]{%
1087
1088
         \i fbool expe
         { bool {DTMshowi soZ}
1089
           and test{\ifnumequal {##1}{0}}
1090
           and test{\ifnumequal {##2}{0}}
1091
         }%
1092
         {%
1093
```

```
Z%
1094
         }%
1095
         {%
1096
          \in fnum##1<0\leq se+fi DTMtwodigits{##1}%
1097
          \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodi gi ts{##2}\fi
1098
         }%
1099
       }%
1100
    }%
1101
     {% full style
1102
       \renewcommand*{\DTMdisplay}[9]{%
1103
        \ifDTMshowdate
1104
         \DTMdi spl aydate{##1}{##2}{##3}{##4}%
1105
1106
         \DTMsep{datetime}%
1107
        \DTMdisplaytime
1108
        {##5}%
1109
         {##6}%
1110
         {##7}%
1111
        \i fDTMshowzone
1112
         \DTMsep{timezone}%
1113
         \DTMdi spl ayzone
1114
1115
          {##8}%
          {##9}%
1116
        \fi
1117
       }%
1118
       \renewcommand*{\DTMDi spl ay}{\DTMdi spl ay}%
1119
1120
    Define hmmss time style
1121 \DTMnewtimestyle
1122 {hmmss}% label
    {%
1123
        \renewcommand*\DTMdisplaytime[3]{%
1124
          \number##1
1125
          \DTMsep{hourmin}\DTMtwodigits{##2}%
1126
          \ifDTMshowseconds\DTMsep{mi nsec}\DTMtwodi gi ts{##3}\fi
1127
        }%
1128
1129 }%
    Define map zone style
1130 \DTMnewzonestyle
1131 {map}% label
1132
    {%
        \renewcommand*\DTMdisplaytime[3]{%
1133
          \DTMusezonemapordefaul t{##1}{##2}%
1134
        }%
1135
1136 }%
    Define hhmm zone style
1137 \DTMnewzonestyle
1138 {hhmm}% label
```

```
1139 {%
       \renewcommand*\DTMdisplaytime[3]{%
1140
          \in fnum##1<0\le se+fi DTMtwodigits{##1}%
1141
          \ifDTMshowzonemi nutes\DTMsep{hourmin}\DTMtwodigits{##2}\fi
1142
1143
1144 }
```

9.1.3 Saving and Using Dates

Date and time information is stored in control sequences in the form \@dtm@\\ label\\ @\\ tag\\, where $\langle label \rangle$ is the label uniquely identifying the information and $\langle tag \rangle$ is the element (year, month, day, dow, hour, minute, second, TZhour and TZminute). Missing information is stored as 0 (except for the day of week, which is stored as -1).

\DTMsavedate

Save the date specified in the format $\langle yyyy \rangle - \langle mm \rangle - \langle dd \rangle$. \expandafter is used in case the argument is a control sequence storing the date. This will redefine an existing saved date with the same label. The first argument is the label.

```
1145 \newrobustcmd*{\DTMsavedate}[2]{%
    \expandafter\@dtm@parsedate#2\@dtm@endparsedate
1146
     \cslet{@dtm@#1@year}{\@dtm@year}%
1147
     \cslet{@dtm@#1@month}{\@dtm@month}%
1148
     \cslet{@dtm@#1@day}{\@dtm@day}%
1149
     \cslet{@dtm@#1@dow}{\@dtm@dow}%
1150
     \i fcsundef{@dtm@#1@hour}{\csdef{@dtm@#1@hour}{0}}{}%
1151
     \i fcsundef{@dtm@#1@mi nute}{\csdef{@dtm@#1@mi nute}{0}}{}%
1152
    1153
    \i fcsundef{@dtm@#1@TZhour}{\csdef{@dtm@#1@TZhour}{0}}{}%
     \ifcsundef{@dtm@#1@TZminute}{\csdef{@dtm@#1@TZminute}{0}}{}%
1155
1156 }
```

\DTMsavenoparsedate Save the date without parsing the $\langle YYYY \rangle - \langle MM \rangle - \langle DD \rangle$ format.

```
1157 \newrobustcmd* {\DTMsavenoparsedate} [5] {%
     \csedef{@dtm@#1@year}{\number#2}%
     \csedef{@dtm@#1@month}{\number#3}%
1159
     \csedef{@dtm@#1@day}{\number#4}%
1160
     \csedef{@dtm@#1@dow}{\number#5}%
1161
1162
     \i fcsundef{@dtm@#1@hour}{\csdef{@dtm@#1@hour}{0}}{}%
    \ifcsundef{@dtm@#1@mi nute}{\csdef{@dtm@#1@mi nute}{0}}}{}%
1163
    1164
    \i fcsundef{@dtm@#1@TZhour}{\csdef{@dtm@#1@TZhour}{0}}{}%
     \ifcsundef{@dtm@#1@TZminute}{\csdef{@dtm@#1@TZminute}{0}}{}%
1166
1167 }
```

\DTMsavetime

Save the time specified in the format $\langle hh \rangle:\langle mm \rangle:\langle ss \rangle$. \expandafter is used in case the argument is a control sequence storing the date. This will redefine an existing saved date with the same label. The first argument is the label.

1168 \newrobustcmd*{\DTMsavetime}[2]{%

```
\expandafter\@dtm@parsetime#2\@dtm@endparsetime
                 1169
                       \cslet{@dtm@#1@hour}{\@dtm@hour}%
                 1170
                       \csl et{@dtm@#1@mi nute}{\@dtm@mi nute}%
                 1171
                       \cslet{@dtm@#1@second}{\@dtm@second}%
                 1172
                       \i fcsundef{@dtm@#1@year}{\csdef{@dtm@#1@year}{0}}{}%
                 1173
                       \i fcsundef{@dtm@#1@month}{\csdef{@dtm@#1@month}{0}}{}%
                 1174
                       \ifcsundef{@dtm@#1@day}{\csdef{@dtm@#1@day}{0}}{}%
                 1175
                       \ifcsundef{@dtm@#1@dow}{\csdef{@dtm@#1@dow}{-1}}{}%
                 1176
                       \i fcsundef{@dtm@#1@TZhour}{\csdef{@dtm@#1@TZhour}{0}}{}%
                 1177
                       \ifcsundef{@dtm@#1@TZmi nute}{\csdef{@dtm@#1@TZmi nute}{}}}}
                 1178
                 1179 }
                  Save the time (including zone) specified in the format \langle hh \rangle:\langle mm \rangle:\langle ss \rangle \langle tzh \rangle:\langle tzm \rangle.
\DTMsavetimezn
                  \expandafter is used in case the argument is a control sequence storing the
                  date. This will redefine an existing saved date with the same label. The first
                  argument is the label.
                 1180 \newrobustcmd*{\DTMsavetimezn}[2]{%
                       \expandafter\@dtm@parsetimezn#2\@dtm@endparsetimezn
```

```
1182
     \cslet{@dtm@#1@hour}{\@dtm@hour}%
     \csl et{@dtm@#1@mi nute}{\@dtm@mi nute}%
1183
     \csl et{@dtm@#1@second}{\@dtm@second}%
1184
     \cslet{@dtm@#1@TZhour}{\@dtm@timezonehour}%
1185
     \csl et{@dtm@#1@TZmi nute}{\@dtm@ti mezonemi nute}%
1186
     \i fcsundef{@dtm@#1@year}{\csdef{@dtm@#1@year}{0}}{}%
1187
     \i fcsundef{@dtm@#1@month}{\csdef{@dtm@#1@month}{0}}{}%
1188
     \ifcsundef{@dtm@#1@day}{\csdef{@dtm@#1@day}{0}}{}%
1189
     \ifcsundef{@dtm@#1@dow}{\csdef{@dtm@#1@dow}{-1}}{}%
1190
1191 }
```

\DTMsaveti mestamp Save the time (including zone) specified in the format $\langle YYYY\rangle - \langle MM\rangle - \langle DD\rangle T\langle hh\rangle : \langle mm\rangle : \langle ss\rangle \langle time zone\rangle$

```
1192 \newrobustcmd*{\DTMsavetimestamp}[2]{%
     \expandafter\@dtm@parsetimestamp#2\@dtm@endparsetimestamp
1193
     \cslet{@dtm@#1@year}{\@dtm@year}%
1194
     \cslet{@dtm@#1@month}{\@dtm@month}%
1195
     \cslet{@dtm@#1@day}{\@dtm@day}%
1196
     \cslet{@dtm@#1@dow}{\@dtm@dow}%
1197
     \csl et{@dtm@#1@hour}{\@dtm@hour}%
1198
     \csl et{@dtm@#1@mi nute}{\@dtm@mi nute}%
1199
     \csl et{@dtm@#1@second}{\@dtm@second}%
1200
     \cslet{@dtm@#1@TZhour}{\@dtm@timezonehour}%
1201
     \csl et{@dtm@#1@TZmi nute}{\@dtm@ti mezonemi nute}%
1202
1203 }
```

\DTMsavenow Save the current time.

```
1204 \newrobustcmd{\DTMsavenow}[1]{%
1205 \csl et{@dtm@#1@year}{\@dtm@currentyear}%
1206 \csl et{@dtm@#1@month}{\@dtm@currentmonth}%
```

```
\cslet{@dtm@#1@day}{\@dtm@currentday}%
                1207
                     \cslet{@dtm@#1@dow}{\@dtm@currentdow}%
                1208
                     \cslet{@dtm@#1@hour}{\@dtm@currenthour}%
                1209
                     \csl et{@dtm@#1@mi nute}{\@dtm@currentmi nute}%
                1210
                      \csl et{@dtm@#1@second}{\@dtm@currentsecond}%
                     \cslet{@dtm@#1@TZhour}{\@dtm@currenttimezonehour}%
                1212
                     \cslet{@dtm@#1@TZmi nute}{\@dtm@currentti mezonemi nute}%
                1213
                1214 }
                 Globally set the stored information.
 \DTMmakeglobal
                1215 \newrobustcmd{\DTMmakeglobal}[1]{%
                      \qlobal\csletcs{@dtm@#1@year}{@dtm@#1@year}%
                      \qlobal\csletcs{@dtm@#1@month}{@dtm@#1@month}%
                     \global\csletcs{@dtm@#1@day}{@dtm@#1@day}%
                1218
                     \global\csletcs{@dtm@#1@dow}{@dtm@#1@dow}%
                1219
                     \global\csletcs{@dtm@#1@hour}{@dtm@#1@hour}%
                1220
                      \qlobal\csletcs{@dtm@#1@minute}{@dtm@#1@minute}%
                1221
                     \qlobal\csletcs{@dtm@#1@second}{@dtm@#1@second}%
                1222
                     \global\csletcs{@dtm@#1@TZhour}{@dtm@#1@TZhour}%
                1223
                     \global\csletcs{@dtm@#1@TZminute}{@dtm@#1@TZminute}%
                1224
                1225 }
                    Expandable ways of fetching saved data. (No check for existence performed.)
                 The argument is the label.
  \DTMfetchyear
                1226 \newcommand*{\DTMfetchyear}[1]{\csname @dtm@#1@year\endcsname}
 \DTMfetchmonth
                1227\newcommand*{\DTMfetchmonth}[1]{\csname @dtm@#1@month\endcsname}
   \DTMfetchday
                1228 \newcommand* {\DTMfetchday}[1] {\csname @dtm@#1@day\endcsname}
   \DTMfetchdow
                1229 \newcommand*{\DTMfetchdow}[1]{\csname @dtm@#1@dow\endcsname}
  \DTMfetchhour
                1230 \newcommand*{\DTMfetchhour}[1]{\csname @dtm@#1@hour\endcsname}
\DTMfetchmi nute
                1231 \newcommand*{\DTMfetchmi nute}[1]{\csname @dtm@#1@mi nute\endcsname}
\DTMfetchsecond
                1232 \newcommand*{\DTMfetchsecond}[1]{\csname @dtm@#1@second\endcsname}
\DTMfetchTZhour
                1233 \newcommand*{\DTMfetchTZhour}[1]{\csname @dtm@#1@TZhour\endcsname}
```

1234\newcommand*{\DTMfetchTZmi nute}[1]{\csname @dtm@#1@TZmi nute\endcsname}

\DTMusedate

\DTMusedate{\label\}

Displays the previously saved date using \DTMdisplaydate .

```
1235 \newcommand*\DTMusedate[1]{%
     \ifcsundef{@dtm@#1@year}%
1236
1237
     {%
         \PackageError{datetime2}{Undefined date '#1'}{}%
1238
     }%
1239
     {%
1240
         \DTMdi spl aydate
1241
          {\csname @dtm@#1@year\endcsname}%
1242
          {\csname @dtm@#1@month\endcsname}%
1243
          {\csname @dtm@#1@day\endcsname}%
1244
          {\csname @dtm@#1@dow\endcsname}%
1245
1246
     }%
1247 }%
```

\DTMUsedate

 $\DTMUsedate{\langle label \rangle}$

Displays the previously saved date using \DTMDisplaydate.

```
1248 \newcommand*\DTMUsedate[1]{%
     \ifcsundef{@dtm@#1@year}%
1249
     {%
1250
         \PackageError{datetime2}{Undefined date '#1'}{}%
1251
     }%
1252
     {%
1253
         \DTMDi spl aydate
1254
          {\csname @dtm@#1@year\endcsname}%
1255
          {\csname @dtm@#1@month\endcsname}%
1256
          {\csname @dtm@#1@day\endcsname}%
1257
          {\csname @dtm@#1@dow\endcsname}%
1258
     }%
1259
1260 }%
```

\DTMusetime

 $\DTMusetime{\langle label \rangle}$

Displays the previously saved time using \DTMdisplaytime.

1261 \newcommand*\DTMusetime[1]{%

```
\ifcsundef{@dtm@#1@hour}%
1262
1263
     {%
         \PackageError{datetime2}{Undefined time '#1'}{}%
1264
     }%
1265
     {%
1266
         \DTMdisplaytime
1267
          {\csname @dtm@#1@hour\endcsname}%
1268
          {\csname @dtm@#1@minute\endcsname}%
1269
          {\csname @dtm@#1@second\endcsname}%
1270
     }%
1271
1272 }%
```

\DTMusezone

 $\DTMusezone{\langle label \rangle}$

Displays the previously saved date using \DTMdisplayzone.

```
1273 \newcommand*\DTMusezone[1]{%
     \ifcsundef{@dtm@#1@TZhour}%
1274
     {%
1275
         \PackageError{datetime2}{Undefined time '#1'}{}%
1276
     }%
1277
     {%
1278
         \DTMdi spl ayzone
1279
1280
          {\csname @dtm@#1@TZhour\endcsname}%
          {\csname @dtm@#1@TZminute\endcsname}%
1281
     }%
1282
1283 }%
```

\DTMuse

\DTMuse{\label\}

Displays the previously saved date and time.

```
1284 \newcommand*\DTMuse[1]{%
     \ifcsundef{@dtm@#1@year}%
1285
1286
         \PackageError{datetime2}{Undefined date-time '#1'}{}%
1287
     }%
1288
     {%
1289
        \DTMdi spl ay
1290
          {\csname @dtm@#1@year\endcsname}%
1291
          {\csname @dtm@#1@month\endcsname}%
1292
          {\csname @dtm@#1@day\endcsname}%
1293
          {\csname @dtm@#1@dow\endcsname}%
1294
1295
          {\csname @dtm@#1@hour\endcsname}%
          {\csname @dtm@#1@minute\endcsname}%
1296
1297
          {\csname @dtm@#1@second\endcsname}%
```

\DTMUse

\DTMUse{\label\}

Displays the previously saved date and time.

```
1302 \newcommand*\DTMUse[1]{%
     \ifcsundef{@dtm@#1@year}%
1303
1304
     {%
1305
         \PackageError{datetime2}{Undefined date-time '#1'}{}%
     }%
1306
     {%
1307
        \DTMDi spl ay
1308
          {\csname @dtm@#1@year\endcsname}%
1309
          {\csname @dtm@#1@month\endcsname}%
1310
          {\csname @dtm@#1@day\endcsname}%
1311
1312
          {\csname @dtm@#1@dow\endcsname}%
1313
          {\csname @dtm@#1@hour\endcsname}%
          {\csname @dtm@#1@minute\endcsname}%
1314
          {\csname @dtm@#1@second\endcsname}%
1315
1316
          {\csname @dtm@#1@TZhour\endcsname}%
          {\csname @dtm@#1@TZminute\endcsname}%
1317
     }%
1318
1319 }%
```


9.1.4 Language Module Loading

Define commands to load regional settings.

Nedtmerequi remodule Use tracklang interface to find the associated file for the given dialect.

```
1323 \newcommand*{\@dtm@requiremodule}[1]{%
     \IfTrackedLanguageFileExists{#1}%
1324
     {datetime2-}% prefix
1325
     {.ldf}% suffix
1326
     {%
1327
       \RequireDateTimeModule{\CurrentTrackedTag}%
1328
     }%
1329
     {%
1330
       \@dtm@warning{Date-Time Language Module '#1' not installed}%
1331
```

```
}%
1332
1333 }
```

Nedtmel oadedregions List of loaded datetime 2 language modules.

1334 \newcommand* { \@dtm@l oadedregi ons } { }

qui reDateTi meModul e

Input the language file, if not already loaded. Should only be used with \@dtm@requiremodule which sets commands like \CurrentTrackedDialect. Since the language modules are loaded within \@dtm@requiremodule they may use this command to load dependent modules.

```
1335 \newcommand* {\RequireDateTimeModule} [1] {%
1336 \i fundef\CurrentTrackedDi al ect
1337 {%
      \PackageError{datetime2}%
1338
       {\string\RequireDateTimeModule\space not permitted here}%
1339
       {This command is only permitted inside datetime2 language
1340
       modules. }%
1341
1342 }%
1343
       \ifcsundef{ver@datetime2-#1.ldf}%
1344
1345
         \input{datetime2-#1.ldf}%
1346
1347
         \ifdefempty\@dtm@loadedregions
1348
           \edef\@dtm@loadedregions{#1}%
1349
         }%
1350
         {%
1351
           \edef\@dtm@l oadedregi ons{\@dtm@l oadedregi ons, #1}%
1352
         }%
1353
       }%
1354
       {}%
1355
1356 }%
1357 }
```

videsDateTimeModule For use in language module to identify itself.

```
1358 \newcommand* {\ProvidesDateTimeModule} [1] {%
1359
     \ProvidesFile{datetime2-#1.ldf}%
1360 }
```

\DTMdefkey

Used by language modules to define a key.

1361 \newcommand*{\DTMdefkey}[1]{\define@key[dtm]{#1}}

\DTMdefchoi cekey

 $\label{limited} $$ \operatorname{DTMdefchoicekey}_{\langle region\rangle}_{\langle key\rangle}_{\langle bin\rangle}_{\langle choice|list\rangle}_{\langle default\rangle}_{\langle func\rangle}$$$

Used by language modules to define a choice key.

1362 \newcommand* {\DTMdefchoi cekey}[1] {\defi ne@choi cekey[dtm]{#1}}

\DTMdefbool key

Used by language modules to define a boolean key.

1363 \newcommand*{\DTMdefbool key}[1]{\defi ne@bool key[dtm]{#1}}

\DTMi fbool

 $\label{eq:definition} $$ \operatorname{DTMifbool}_{\langle region\rangle}_{\langle key\rangle}_{\langle true\rangle}_{\langle false\rangle} $$$

Test boolean key that was defined using \DTMdefboolkey

 $1364 \end{4} {\bf M} if bool {\bf d} fool {\bf$

\DTMsetbool

 $\verb|\DTMsetbool{| (region)| { (key) } { (value) }}| \\$

Set boolean key that was defined using \DTMdefboolkey

1365 \newcommand*{\DTMsetbool}[3]{\setbool{dtm@#1@#2}{#3}}

\DTMI angsetup

Set up options for language modules. The optional argument is a list of language/regions. If omitted all loaded regions are iterated over. (I'm not sure why \setkeys doesn't work if the same key is present in multiple families, so this iterates over the families instead.)

```
1366 \newcommand* {\DTMI angsetup} [2] [\@dtm@l oadedregi ons] {%
    \@for\@dtm@region: =#1\do{%
       \setkeys*+[dtm]{\@dtm@region}{#2}%
1368
      \ifdefempty\XKV@rm{}%
1369
1370
         \@dtm@warning{Region '\@dtm@region' has ignored
1371
          \MessageBreak the following settings: \MessageBreak
1372
          \XKV@rm
1373
          ^^J}%
1374
1375
1376 }%
1377 }
```

```
Now load all the required modules (if installed) using the tracklang interface. (Language packages, such as babel or polyglossia must be loaded before this.)
```

```
1378 \AnyTrackedLanguages
1379 {%
      \ForEachTrackedDi al ect{\thi s@di al ect}%
1380
1381
1382
        \@dtm@requi remodul e\thi s@di al ect
      }%
1383
1384 }
1385 { %
 No tracked languages. The default is already set up, so nothing to do here.
    Load datetime2-calc if required.
1387 \@dtm@usecalc
    Use the Style package option, if set.
1388 \i fdefempty\@dtm@i ni ti al styl e\{\DTMsetstyl e\{\edtm@i ni ti al styl e\}\}
```

9.2 datetime2-calc.sty code

```
1389 \NeedsTeXFormat{LaTeX2e}
1390 \Provi desPackage{dateti me2-cal c}[2015/09/15 v1.1 (NLCT)]
Load other required packages
```

1391 \Requi rePackage {pgfkeys} 1392 \Requi rePackage {pgfcal endar}

\@dtm@j ul i anday Register for storing Julian day number.

1393 \newcount\@dtm@j ul i anday

\@dtm@parsedate

Redefine \@dtm@parsedate so that it uses pgfcalendar to compute the required information. This allows for offsets, the use of last and also determine the day of week.

```
1394 \def\@dtm@parsedate#1-#2-#3\@dtm@endparsedate{%
1395 \pgfcal endardatetoj ul i an{#1-#2-#3}{\@dtm@j ul i anday}%
1396 \pgfcal endarj ul i antodate{\@dtm@j ul i anday}{\@dtm@year}{\@dtm@month}{\@dtm@day}%
1397 \pgfcal endarj ul i antoweekday{\@dtm@j ul i anday}{\count@}%
1398 \edef\@dtm@dow{\number\count@}%
1399 }
```

Set the current day of week

\@dtm@currentdow

```
1400 \pgfcal endardatetoj ul i an
1401 {\@dtm@currentyear-\@dtm@currentmonth-\@dtm@currentday}%
1402 {\@dtm@j ul i anday}%
1403 \pgfcal endarj ul i antoweekday{\@dtm@j ul i anday}{\count@}%
1404 \edef\@dtm@currentdow{\number\count@}%
```

\DTMsavej ul i anday Save the date obtained from the Julian day number.

```
1405 \newrobustcmd*{\DTMsavejulianday}[2]{%
     \pgfcal endari uliantodate{#2}{\@dtm@year}{\@dtm@month}{\@dtm@day}%
1406
     \pgfcal endarj ul i antoweekday{#2}{\count@}%
1407
    \csedef{@dtm@#1@dow}{\number\count@}%
1408
     \cslet{@dtm@#1@year}{\@dtm@year}%
1409
    \csl et{@dtm@#1@month}{\@dtm@month}%
1410
     \cslet{@dtm@#1@day}{\@dtm@day}%
1411
    \ifcsundef{@dtm@#1@hour}{\csdef{@dtm@#1@hour}{0}}{}%
1412
1413
    \ifcsundef{@dtm@#1@minute}{\csdef{@dtm@#1@minute}{0}}{}%
    1414
    \ifcsundef{@dtm@#1@TZhour}{\csdef{@dtm@#1@TZhour}{}}}}%
1415
    \ifcsundef{@dtm@#1@TZmi nute}{\csdef{@dtm@#1@TZmi nute}{}}}
1417 }
```

aveddatetoj ul i anday

Converts a saved date to a Julian day number. The first argument is the name referencing the saved date, the second is a count register in which to store the result.

```
1418 \newrobustcmd* {\DTMsaveddatetojulianday}[2]{%
     \ifcsundef{@dtm@#1@year}%
1420
         \PackageError{datetime2-calc}{Unknown date '#1'}{}%
1421
     }%
1422
     {%
1423
        \pgfcal endardatetoj ul i an
1424
         {\csname @dtm@#1@year\endcsname
1425
         -\csname @dtm@#1@month\endcsname
1426
         -\csname @dtm@#1@day\endcsname}
1427
1428
         {#2}%
     }%
1429
1430 }
```

teoffsettoj ul i anday

Converts an offset from the saved date to a Julian day number. The first argument is the name referencing the saved date, the second is the offset increment and the third is a count register in which to store the result.

```
1431 \newrobustcmd*{\DTMsaveddateoffsettojulianday}[3]{%
     \ifcsundef{@dtm@#1@year}%
1432
1433
         \PackageError{datetime2-calc}{Unknown date '#1'}{}%
1434
     }%
1435
     {%
1436
        \pgfcal endardatetoj ul i an
1437
         {\csname @dtm@#1@year\endcsname
1438
         -\csname @dtm@#1@month\endcsname
1439
         -\csname @dtm@#1@day\endcsname
1440
         +#2}
1441
         {#3}%
1442
     }%
1443
1444 }
```



```
1445 \newrobustcmd*{\DTMifdate}[4]{%
     \ifcsundef{@dtm@#1@year}%
1446
1447
         \PackageError{datetime2-calc}{Unknown date '#1'}{}%
1448
     }%
1449
     {%
1450
        \pgfcal endari fdate
1451
         {\csname @dtm@#1@year\endcsname
1452
1453
         -\csname @dtm@#1@month\endcsname
         -\csname @dtm@#1@day\endcsname}
1454
         {#2}{#3}{#4}%
1455
    }%
1456
1457 }
```

\DTMsaveddatediff Computes the difference between two saved dates. The result is stored in the third argument, which should be a count register.

```
1458 \newrobustcmd* {\DTMsaveddatediff}[3]{%
     \ifcsundef{@dtm@#1@year}%
1459
1460
         \PackageError{datetime2-calc}{Unknown date '#1'}{}%
1461
     }%
1462
      {%
1463
         \ifcsundef{@dtm@#2@year}%
1464
1465
            \PackageError{datetime2-calc}{Unknown date '#1'}{}%
1466
         }%
1467
         {%
1468
           \pgfcal endardatetoj ul i an
1469
            {\csname @dtm@#1@year\endcsname
1470
            -\csname @dtm@#1@month\endcsname
1471
            -\csname @dtm@#1@day\endcsname}
1472
            {#3}%
1473
           \pgfcal endardatetoj ul i an
1474
            {\csname @dtm@#2@year\endcsname
1475
1476
            -\csname @dtm@#2@month\endcsname
            -\csname @dtm@#2@day\endcsname}
1477
            {\@dtm@j ul i anday}%
1478
           \advance#3 by -\@dtm@julianday\relax
1479
1480
     }%
1481
1482 }
```

\DTMtozul u Converts the datetime data referenced by the first argument into Zulu time and saves it to data referenced by the second argument.

```
1483 \newrobustcmd*{\DTMtozulu}[2]{%
1484 \ifcsundef{@dtm@#1@year}%
1485 {%
1486 \PackageError{datetime2-calc}{Unknown date '#1'}{}%
```

```
1487
     }%
     {%
1488
        \DTMsaveaszulutime{#2}%
1489
        {\DTMfetchyear{#1}}%
1490
        {\DTMfetchmonth{#1}}%
1491
        {\DTMfetchday{#1}}%
1492
        {\DTMfetchhour{#1}}%
1493
        {\DTMfetchminute{#1}}%
1494
        {\DTMfetchsecond{#1}}%
1495
        {\DTMfetchTZhour{#1}}%
1496
        {\DTMfetchTZminute{#1}}%
1497
1498
     }%
1499 }
```

\DTMsaveaszul uti me Converts the given datetime into Zulu (+00:00) and saves the result.

```
\label{lem:line} $$ \operatorname{DTMsavetozulutime}_{\langle name\rangle}_{\langle year\rangle}_{\langle month\rangle}_{\langle day\rangle}_{\langle hour\rangle}_{\langle minute\rangle}_{\langle second\rangle}_{\langle tzh\rangle}_{\langle tzm\rangle}_{}}
```

```
1500 \newrobustcmd* {\DTMsaveaszul uti me} [9] {%
     \edef\@dtm@year{\number#2}%
1501
     \edef\@dtm@month{\number#3}%
1502
     \edef\@dtm@day{\number#4}%
1503
     \edef\@dtm@hour{\number#5}%
1504
     \edef\@dtm@mi nute{\number#6}%
1505
1506
     \edef\@dtm@second{\number#7}%
     \edef\@dtm@TZhour{\number#8}%
1507
     \edef\@dtm@TZminute{\number#9}%
1508
     \pgfcal endardatetoj ul i an{\@dtm@year-\@dtm@month-\@dtm@day}{\@dtm@j ul i anday}%
1509
 First adjust the minute offset if non-zero
     \i fnum\@dtm@TZmi nute=0\relax
1510
1511
     \el se
        \count@=\@dtm@mi nute\relax
1512
 Add or subtract the offset minute
        \ifnum\@dtm@TZhour<0\relax
1513
          \advance\count@ by \@dtm@TZminute\relax
1514
       \el se
1515
          \advance\count@ by -\@dtm@TZminute\relax
1516
1517
       \edef\@dtm@mi nute{\number\count@}%
 Does the hour need adjusting?
       \ifnum\count@<0\relax
1519
          \advance\count@ by 60\relax
1520
          \edef\@dtm@mi nute{\number\count@}%
1521
```

Need to subtract 1 from the hour but does the day need adjusting?

Subtract 1 from the hour

Minute isn't negative. Is it \geq 60?

```
\ifnum\count@>59\relax
1532 \advance\count@ by -60\relax
1533 \edef\@dtm@mi nute{\number\count@}%
```

Add 1 to the hour

```
1534 \count@ = \@dtm@hour\relax
1535 \advance\count@ by 1\relax
1536 \edef\@dtm@hour{\number\count@}%
```

Does the day need adjusting?

```
1537 \ifnum\@dtm@hour=24\relax
1538 \def\@dtm@hour{00}%
1539 \advance\@dtm@julianday by 1\relax
1540 \fi
1541 \fi
1542 \fi
1543 \fi
```

Now adjust the hour offset if non-zero

```
1544 \i fnum\@dtm@TZhour=0\relax
1545 \else
1546 \count@=\@dtm@hour\relax
1547 \advance\count@ by -\@dtm@TZhour\relax
```

Does the day need adjusting?

```
1548
       \ifnum\count@<0\relax
          \advance\count@ by 24\relax
1549
          \edef\@dtm@hour{\number\count@}%
1550
          \advance\@dtm@julianday by -1\relax
1551
       \el se
1552
          \ifnum\count@>23\relax
1553
          \advance\count@ by -24\relax
1554
          \edef\@dtm@hour{\number\count@}%
1555
          \advance\@dtm@julianday by 1\relax
1556
          \el se
1557
            \edef\@dtm@hour{\number\count@}%
1558
          \fi
1559
```

```
\fi
1560
                          \fi
1561
                           \pgfcal endarj uliantodate{\edtm@julianday}{\edtm@year}{\edtm@month}{\edtm@day}%
1562
                           \pgfcal endarj ul i antoweekday{\@dtm@j ul i anday}{\count@}%
1563
      Save the results.
                           \csedef{@dtm@#1@dow}{\number\count@}%
1564
                           \cslet{@dtm@#1@year}{\@dtm@year}%
1565
                           \cslet{@dtm@#1@month}{\edtm@month}%
1566
                           \c \ensuremath{\c Cslet} = {\c dtm@\#1@day}{\c dtm@day}%
1567
                           \cslet{@dtm@#1@hour}{\@dtm@hour}%
1568
                           \cslet{@dtm@\#1@minute}{\@dtm@minute}\%
1569
                           \verb|\cs||\ et{@dtm@\#1@second}{\wedge}| % \label{lem:cslet} % $$ \cslett{@dtm@\#1@second}| % $$ \cslett{@dtm@gsecond}| % $$ \cslett{@dtmggsecond}| % $$ \cslett{@dtmggsecond
1570
                           \csdef{@dtm@#1@TZhour}{0}%
1571
                           \csdef{@dtm@#1@TZminute}{0}%
1572
```

1573 }

Index

Symbols \@dtm@currentdow	\dtm@hourmi nsep 45 \dtm@mi nsecsep 45 \dtm@monthdaysep 45 \dtm@ti mezonesep 45 \dtm@yearmonthsep 45 \DTMcentury 24,59 \DTMcl earmap 26,61 \DTMcurrentti me 11,56 \DTMcurrentzone 11,56 \DTMDate 9,56 \DTMdate 9,55 \DTMdefbool key 82 \DTMdefchoi cekey 82
\@dtm@unknownstyle 63 \@dtm@usecalc 48 \@dtm@warning 49	\DTMdefkey 81 \DTMdefzonemap 25,60 \DTMDi spl ay 12,58 \DTMdi spl ay 12,57
B babel package	\DTMDi spl aydate
D date style	\DTMfetchday 16,77 \DTMfetchdow 16,77 \DTMfetchhour 16,77 \DTMfetchmi nute 16,77
datesep (option)	\DTMfetchmonth
datetimesep (option) 46 dayyearsep (option) 46 ddmmyyyy style 21,68 defaul t style	\DTMfetchyear 16,77 \DTMhaszonemap 26,61 \DTMi fbool 82 \DTMi fcaseregi onal 47
. 5, 7, 20–22, 28–30, 32, 44, 47, 64 display style 8 dmyy style 21, 70 dmyyyy style 21, 69 document (environment) . 9, 25, 36, 38 \dtm@dayyearsep 45	\DTMi fdate

\ DTMtull -	
\DTMnewstyle \dots 23,62	environments:
\DTMnewtimestyle 23,60	document 9, 25, 36, 38
\DTMnewtimezone 60	etoolbox package 45
\DTMnewzonestyle 23	F
\DTMNow 12, 57	
\DTMnow 12, 57	full style 12, 18, 19, 23
\DTMresetzones 26,61	Н
\DTMsaveaszulutime 43,86	hhmm style 23,74
\DTMsavedate 13,75	hmmss style
\DTMsaveddatediff 42,85	hmmss package
\DTMsaveddateoffsettoj uliandate	hourminsep (option)
42	hyperref package
\DTMsaveddateoffsettoj ulianday	hyperrei package 23,39
	Ī
\DTMsaveddatetojuliandate 41	i so style 21, 35, 36, 47, 65
\DTMsaveddatetojulianday 84	130 style 21,00,00, 11,00
\DTMsavefilemoddate 14,51	M
\DTMsavej ul i anday 41,84	map style
\DTMsavenoparsedate 13,75	mdyy style 21, 73
\DTMsavenow	mdyyyy style
\DTMsavetime	mfirstuc package 8
\DTMsavetime \documents 13,73	mi nsecsep (option) 46
\DTMsavetimestamp 14,76	mmddyyyy style 21,71
\DTMsep 25, 59	monthdaysep (option) 46
\DTMsetbool 82	5 1 1 7
	0
\DTMsetcurrentzone 55	O options (definitions):
\DTMsetcurrentzone 55 \DTMsetdatestyle 18,62	_
\DTMsetcurrentzone	options (definitions): 48 cal c 45
\DTMsetcurrentzone	options (definitions): 48 cal c 45 datesep 45 datetimesep 46
\DTMsetcurrentzone	options (definitions): cal c 48 datesep 45 datetimesep 46 dayyearsep 46
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 datetimesep 46 dayyearsep 46 hourminsep 46
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46 showdate 46
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 datetimesep 46 dayyearsep 46 hourminsep 46 minsecsep 46 monthdaysep 46 showdate 46 showdow 48
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47
\DTMsetcurrentzone	options (definitions): 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47 showseconds 46
\DTMsetcurrentzone	options (definitions): 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47 showzone 47
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47 showzone 46 showzonemi nutes 47
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47 showzone 47 showzonemi nutes 47 styl e 49
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 datetimesep 46 dayyearsep 46 hourminsep 46 minsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47 showseconds 46 showzone 47 showzoneminutes 47 style 49 timesep 46
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 datetimesep 46 dayyearsep 46 hourminsep 46 minsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47 showzone 46 showzoneminutes 47 styl e 49 ti mesep 46 ti mezonesep 46
\DTMsetcurrentzone \	options (definitions): 48 cal c 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47 showzone 46 showzonemi nutes 47 styl e 49 ti mesep 46 ti mezonesep 46 useregi onal 47
\DTMsetcurrentzone	options (definitions): 48 datesep 45 datetimesep 46 dayyearsep 46 hourminsep 46 minsecsep 46 monthdaysep 46 showdate 46 showdow 48 showisoZ 47 showzone 46 showzone 47 style 49 timesep 46 timezonesep 46 useregional 47 warn 49
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 dateti mesep 46 dayyearsep 46 hourmi nsep 46 mi nsecsep 46 monthdaysep 46 showdate 46 showdow 48 showi soZ 47 showzone 46 showzonemi nutes 47 styl e 49 ti mesep 46 ti mezonesep 46 useregi onal 47
\DTMsetcurrentzone	options (definitions): 48 datesep 45 datetimesep 46 dayyearsep 46 hourminsep 46 minsecsep 46 monthdaysep 46 showdate 46 showisoZ 47 showseconds 46 showzone 47 showzoneminutes 47 style 49 timesep 46 timezonesep 46 useregional 47 warn 49 yearmonthsep 46
\DTMsetcurrentzone	options (definitions): 48 cal c 48 datesep 45 datetimesep 46 dayyearsep 46 hourminsep 46 minsecsep 46 monthdaysep 46 showdate 46 showdow 48 showisoZ 47 showzone 47 showzoneminutes 47 style 49 timesep 46 timezonesep 46 timezonesep 46 useregional 47 warn 49 yearmonthsep 46
\DTMsetcurrentzone	options (definitions): 48 datesep 45 datetimesep 46 dayyearsep 46 hourminsep 46 minsecsep 46 monthdaysep 46 showdate 46 showisoZ 47 showseconds 46 showzone 47 showzoneminutes 47 style 49 timesep 46 timezonesep 46 useregional 47 warn 49 yearmonthsep 46

calc 9, 13, 37, 38, 41	\ProvidesDateTimeModule81
datesep 20, 21, 35, 39, 40	
datetimesep	R
dayyearsep	\RequireDateTimeModule 81
en-GB 32	_
english	S
hourminsep 20, 23, 26, 35	scotti sh style
mapzone 26	scottish-numericstyle 32
minsecsep 20, 35	showdate (option) 46
monthdaysep	showdow (option) 48
scottish 32	showi soZ (option) 47
showdate	showseconds (option) 46
showdow	showzone (option) 47
true	showzonemi nutes (option) 47
showisoZ 20–23, 26, 36	style (option)
false 5	an.
showseconds 11, 20, 21, 36	T
false 5	\time 11
showzone	time style
false 5	timesep (option)
showzoneminutes . 20, 21, 23, 26, 36	timezonesep (option) 46
style 28, 37, 38, 83	\Today
timesep 35	\today 8, 55
timezonesep	tracklang package
UKenglish 32	29–33, 38, 45, 49, 80, 83
usenumerical	U
false	ukdate package
text 37	useregional (option)
useregional 6, 28, 30–32, 36, 37	acor egrenar (epiton) Trentition
false	W
true 32	warn (option) 49
warn	
yearmonthsep 20, 25, 27, 35	X
partial style	xkeyval package
pdf style 22, 35, 36, 47, 66	
\pdfcreationdate11	Y
\pdfi nfo	yearmonthsep (option) 46
pgf package 42, 43	yyyymd style 21,66
pgfcalendar package	Z
polyglossia package 9, 28–32, 36, 37, 83	zone style 11, 18, 19, 23