QDateTimeEdit Class

QDateTimeEdit 类提供了一个用于编辑日期和时间的小部件。更多的...

Header:	**#include**
CMake:	find_package(Qt6 REQUIRED COMPONENTS Widgets) target_link_libraries(mytarget PRIVATE Qt6::Widgets)
qmake:	QT += widgets
Inherits:	QAbstractSpinBox
Inherited By:	QDateEdit and QTimeEdit

- 所有成员的列表，包括继承的成员

公共类型

enum	**Section { NoSection, AmPmSection, MSecSection, SecondSection, MinuteSection, …, YearSection }**
flags	**Sections**

特性

calendarPopup : bool
currentSection : Section
currentSectionIndex : int
date : QDate
dateTime : QDateTime
displayFormat : QString
displayedSections : const Sections
maximumDate : QDate
maximumDateTime : QDateTime
maximumTime : QTime
minimumDate : QDate
minimumDateTime : QDateTime
minimumTime : QTime
sectionCount : const int
time : QTime
timeSpec : Qt::TimeSpec

公共函数

	QDateTimeEdit(QWidget *parent = nullptr)
	QDateTimeEdit(const QDateTime &*datetime*, QWidget *parent = nullptr)
	QDateTimeEdit(QDate *date*, QWidget *parent = nullptr)
	QDateTimeEdit(QTime *time*, QWidget *parent = nullptr)
virtual	~QDateTimeEdit()
bool	calendarPopup() const

	QDateTimeEdit(**QWidget** ***parent** = **nullptr**)
QCalendarWidget *	**calendarWidget**() const
void	**clearMaximumDate**()
void	**clearMaximumDateTime**()
void	**clearMaximumTime**()
void	**clearMinimumDate**()
void	**clearMinimumDateTime**()
void	**clearMinimumTime**()
QDateTimeEdit::Section	**currentSection**() const
int	**currentSectionIndex**() const
QDate	**date**() const
QDateTime	**dateTime**() const
QString	**displayFormat**() const
QDateTimeEdit::Sections	**displayedSections**() const
QDate	**maximumDate**() const
QDateTime	**maximumDateTime**() const
QTime	**maximumTime**() const
QDate	**minimumDate**() const
QDateTime	**minimumDateTime**() const
QTime	**minimumTime**() const
QDateTimeEdit::Section	**sectionAt**(int *index*) const
int	**sectionCount**() const
QString	**sectionText**(QDateTimeEdit::Section *section*) const
void	**setCalendarPopup**(bool *enable*)
void	**setCalendarWidget**(QCalendarWidget **calendarWidget*)
void	**setCurrentSection**(QDateTimeEdit::Section *section*)
void	**setCurrentSectionIndex**(int *index*)
void	**setDateRange**(QDate *min*, QDate *max*)
void	**setDateTimeRange**(const QDateTime &*min*, const QDateTime &*max*)
void	**setDisplayFormat**(const QString &*format*)

void	**QDateTimeEdit**(QWidget *parent = nullptr)
void	**setMaximumDateTime**(const QDateTime &*dt*)
void	**setMaximumTime**(QTime *max*)
void	**setMinimumDate**(QDate *min*)
void	**setMinimumDateTime**(const QDateTime &*dt*)
void	**setMinimumTime**(QTime *min*)
void	**setSelectedSection**(QDateTimeEdit::Section *section*)
void	**setTimeRange**(QTime *min*, QTime *max*)
void	**setTimeSpec**(Qt::TimeSpec *spec*)
QTime	**time**() const
Qt::TimeSpec	**timeSpec**() const

重载公共函数

virtual void	**clear() override**
virtual bool	**event**(QEvent **event*) override
virtual QSize	**sizeHint**() const override
virtual void	**stepBy**(int *steps*) override

公共槽

void	**setDate(QDate *date*)**
void	**setDateTime**(const QDateTime &*dateTime*)
void	**setTime**(QTime *time*)

信号

void	**dateChanged(QDate *date*)**
void	**dateTimeChanged**(const QDateTime &*datetime*)

void	**dateChanged**(QDate *date*)
void	**timeChanged**(QTime *time*)

protected功能

virtual QDateTime	**dateTimeFromText**(**const QString &*text*) const**
virtual QString	**textFromDateTime**(const QDateTime &*dateTime*) const

重载的protected功能

virtual void	**fixup**(**QString &*input*) const override**
virtual void	**focusInEvent**(QFocusEvent **event*) override
virtual bool	**focusNextPrevChild**(bool *next*) override
virtual void	**initStyleOption**(QStyleOptionSpinBox **option*) const override
virtual void	**keyPressEvent**(QKeyEvent **event*) override
virtual void	**mousePressEvent**(QMouseEvent **event*) override
virtual void	**paintEvent**(QPaintEvent **event*) override
virtual QAbstractSpinBox::StepEnabled	**stepEnabled**() const override
virtual QValidator::State	**validate**(QString &*text*, int &*pos*) const override
virtual void	**wheelEvent**(QWheelEvent **event*) override

详细说明

QDateTimeEdit 允许用户通过使用键盘或箭头键来增加和减少日期和时间值来编辑日期。箭头键可用于在 QDateTimeEdit 框中从一个部分移动到另一个部分。日期和时间按照设置的格式显示；看setDisplayFormat()。

```
QDateTimeEdit *dateEdit = new QDateTimeEdit(QDate::currentDate());
dateEdit->setMinimumDate(QDate::currentDate().addDays(-365));
dateEdit->setMaximumDate(QDate::currentDate().addDays(365));
dateEdit->setDisplayFormat("yyyy.MM.dd");
```

在这里，我们创建了一个新的 QDateTimeEdit 对象，并用今天的日期进行初始化，并将有效日期范围限制为今天正负 365 天。我们将顺序设置为月、日、年。

QDateTimeEdit 的有效值范围由属性控制minimumDateTime,maximumDateTime，以及它们各自的日期和时间组成部分。默认情况下，从 100 CE 开始到 9999 CE 结束的任何日期时间均有效。

使用弹出式日历小部件

QDateTimeEdit 可以配置为允许QCalendarWidget用于选择日期。这是通过设置来启用的calendarPopup财产。此外，您可以通过调用提供自定义日历小部件作为日历弹出窗口setCalendarWidget （） 功能。现有的日历小部件可以通过以下方式检索calendarWidget()。

键盘追踪

什么时候keyboard tracking启用（默认），编辑字段的每次击键都会触发值更改的信号。

当允许的range如果比其末端跨越的某些时间间隔更窄，则键盘跟踪会阻止用户编辑日期或时间以访问该间隔的后面部分。例如，对于从 29.04.2020 到 02.05.2020 的范围以及初始日期 30.04.2020，用户既不能更改月份（5 月 30 日超出范围）也不能更改日期（4 月 2 日超出范围）。

禁用键盘跟踪时，仅当焦点在编辑修改内容后离开文本字段时，才会发出更改信号。这允许用户通过无效的日期时间进行编辑以达到有效的日期时间。

也可以看看QDateEdit,QTimeEdit,QDate， 和QTime。

成员类型文档

枚举 QDateTimeEdit::部分 标志 QDateTimeEdit::部分

持续的	价值
`QDateTimeEdit::NoSection`	`0x0000`
`QDateTimeEdit::AmPmSection`	`0x0001`
`QDateTimeEdit::MSecSection`	`0x0002`
`QDateTimeEdit::SecondSection`	`0x0004`
`QDateTimeEdit::MinuteSection`	`0x0008`
`QDateTimeEdit::HourSection`	`0x0010`
`QDateTimeEdit::DaySection`	`0x0100`
`QDateTimeEdit::MonthSection`	`0x0200`
`QDateTimeEdit::YearSection`	`0x0400`

Sections 类型是QFlags

的类型定义。它存储部分值的 OR 组合。

属性文档

calendarPopup : bool

该属性保存当前日历弹出显示模式。

单击箭头按钮后将显示日历弹出窗口。仅当存在有效的日期显示格式时，此属性才有效。

访问功能：

bool	**calendarPopup() const**
void	**setCalendarPopup**(bool *enable*)

也可以看看setDisplayFormat()。

currentSection : Section

该属性保存旋转框的当前部分。

访问功能：

QDateTimeEdit::Section	**currentSection() const**
void	**setCurrentSection**(QDateTimeEdit::Section *section*)

currentSectionIndex : int

该属性保存旋转框的当前部分索引。

如果格式为“yyyy/MM/dd”，显示文本为“2001/05/21”，且光标位置为 5，则 currentSectionIndex 返回 1。如果光标位置为 3，则 currentSectionIndex 为 0，依此类推。

访问功能：

int	**currentSectionIndex() const**
void	**setCurrentSectionIndex**(int *index*)

也可以看看setCurrentSection（） 和currentSection()。

date : QDate

该物业拥有QDate这是在小部件中设置的。

默认情况下，此属性包含 2000 年 1 月 1 日的日期。

访问功能：

QDate	**date() const**
void	**setDate**(QDate *date*)

通知器信号：

void	**dateChanged(QDate *date*)**

也可以看看time和dateTime。

dateTime : QDateTime

该物业拥有QDateTime即设置在QDateTimeEdit。

设置此属性时，新的QDateTime转换为时间系统QDateTimeEdit，因此保持不变。

默认情况下，此属性设置为公元 2000 年年初。只能设置为有效的QDateTime价值。如果任何操作导致此属性的值无效，则会将其重置为minimumDateTime财产。

如果QDateTimeEdit没有日期字段，设置此属性会将小部件的日期范围设置为此属性的新值的日期开始和结束。

访问功能：

QDateTime	**dateTime() const**
void	**setDateTime**(const QDateTime &*dateTime*)

通知器信号：

void	**dateTimeChanged(const QDateTime &*datetime*)**

也可以看看date,time,minimumDateTime， 和maximumDateTime。

displayFormat : QString

此属性保存用于显示日期时间编辑的时间/日期的格式。

该格式的描述见QDateTime::toString（） 和QDateTime::fromString()

格式字符串示例（假设日期是 1969 年 7 月 2 日）：

格式	结果
年月日	1969年7月2日
MMMM 年年	七月 2 日 69
MMMM 年年	7 月 2 日 69

请注意，如果您指定两位数年份，它将被解释为初始化日期时间编辑的世纪。默认世纪是 21 世纪（2000-2099）。

如果指定无效格式，则不会设置该格式。

访问功能：

QString	displayFormat() const
void	**setDisplayFormat**(const QString &*format*)

也可以看看QDateTime::toString（） 和displayedSections()。

[read-only]displayedSections : const Sections

该属性保存当前显示的日期时间编辑字段。

返回此格式的显示部分的位集。

访问功能：

QDateTimeEdit::Sections	displayedSections() const

也可以看看setDisplayFormat（） 和displayFormat()。

maximumDate : QDate

该属性保存日期时间编辑的最大日期。

更改此属性会更新日期maximumDateTime财产，同时保留maximumTime财产。设置该属性时，minimumDate如有必要，进行调整以确保范围保持有效。当这种情况发生时，minimumTime如果属性大于maximumTime财产。否则，对此属性的更改将保留minimumDateTime财产。

该属性只能设置为有效的QDate描述当前日期的对象maximumTime属性使有效QDateTime目的。setMaximumDate()接受的最晚日期是 9999 CE 结束。这是该属性的默认值。可以使用以下命令恢复此默认值clearMaximumDateTime()。

访问功能:

QDate	maximumDate() const
void	**setMaximumDate**(QDate *max*)
void	**clearMaximumDate**()

也可以看看 minimumDate,maximumTime,maximumDateTime,setDateRange(),QDate::isValid（），和 Keyboard Tracking。

maximumDateTime : QDateTime

此属性保存日期时间编辑的最大日期时间。

更改此属性会隐式更新maximumDate和maximumTime属性分别为此属性的日期和时间部分。设置该属性时，minimumDateTime如有必要，进行调整以确保范围保持有效。否则，更改此属性将保留minimumDateTime财产。

该属性只能设置为有效的QDateTime价值。setMaximumDateTime() 接受的最新日期时间是 9999 CE 结束。这是该属性的默认值。可以使用clearMaximumDateTime()恢复此默认值。

访问功能:

QDateTime	maximumDateTime() const
void	**setMaximumDateTime**(const QDateTime &*dt*)
void	**clearMaximumDateTime**()

也可以看看 minimumDateTime,maximumTime,maximumDate(),setDateTimeRange(),QDateTime::isValid（），和 Keyboard Tracking。

maximumTime : QTime

该属性保存日期时间编辑的最长时间。

更改此属性会更新时间maximumDateTime财产，同时保留minimumDate和maximumDate特性。如果这些日期属性一致，则在设置此属性时，minimumTime如有必要，调整属性以确保范围保持有效。否则，更改此属性将保留minimumDateTime财产。

该属性可以设置为任何有效的QTime价值。默认情况下，此属性包含时间 23:59:59 和 999 毫秒。可以使用clearMaximumTime()恢复此默认值。

访问功能:

QTime	maximumTime() const

QTime	maximumTime() const
void	**setMaximumTime**(QTime *max*)
void	**clearMaximumTime**()

也可以看看 minimumTime,maximumDate,maximumDateTime,setTimeRange(),QTime::isValid（），和 Keyboard Tracking。

minimumDate : QDate

该属性保存日期时间编辑的最小日期。

更改此属性会更新日期minimumDateTime财产，同时保留minimumTime财产。设置该属性时，maximumDate如有必要，进行调整以确保范围保持有效。当这种情况发生时，maximumTime如果属性小于minimumTime财产。否则，对此属性的更改将保留maximumDateTime财产。

该属性只能设置为有效的QDate描述当前日期的对象minimumTime属性使有效QDateTime目的。setMinimumDate() 接受的最早日期是公元 100 年的开始日期。该属性的默认日期是公元 1752 年 9 月 14 日。可以使用以下命令恢复此默认值clearMinimumDateTime()。

访问功能：

QDate	minimumDate() const
void	**setMinimumDate**(QDate *min*)

void	**clearMinimumDate**()

也可以看看 maximumDate,minimumTime,minimumDateTime,setDateRange(),QDate::isValid（），和 Keyboard Tracking。

minimumDateTime : QDateTime

此属性保存日期时间编辑的最小日期时间。

更改此属性会隐式更新minimumDate和minimumTime属性分别为此属性的日期和时间部分。设置该属性时，maximumDateTime如有必要，进行调整以确保范围保持有效。否则，更改此属性将保留maximumDateTime财产。

该属性只能设置为有效的QDateTime价值。setMinimumDateTime() 接受的最早日期时间是 100 CE 的开始。该属性的默认日期是公元 1752 年 9 月 14 日开始。可以使用clearMinimumDateTime()恢复此默认值。

访问功能：

QDateTime	minimumDateTime() const
void	**setMinimumDateTime**(const QDateTime &*dt*)

void	**clearMinimumDateTime**()
QDateTime	**minimumDateTime() const**

也可以看看 maximumDateTime,minimumTime,minimumDate,setDateTimeRange(),QDateTime::isValid（），和 Keyboard Tracking。

minimumTime : QTime

该属性保存日期时间编辑的最短时间。

更改此属性会更新时间minimumDateTime财产，同时保留minimumDate和maximumDate特性。如果这些日期属性一致，则在设置此属性时，maximumTime如有必要，调整属性以确保范围保持有效。否则，更改此属性将保留maximumDateTime财产。

该属性可以设置为任何有效的QTime价值。默认情况下，此属性包含时间 00:00:00 和 0 毫秒。可以使用clearMinimumTime()恢复此默认值。

访问功能：

QTime	minimumTime() const
void	**setMinimumTime**(QTime *min*)
void	**clearMinimumTime**()

也可以看看 maximumTime,minimumDate,minimumDateTime,setTimeRange(),QTime::isValid（），和 Keyboard Tracking。

[read-only]sectionCount : const int

该属性保存显示的部分数量。如果格式为“yyyy/yy/yyyy”，则sectionCount返回3

访问功能：

int	sectionCount() const

time : QTime

该物业拥有QTime这是在小部件中设置的。

默认情况下，此属性包含时间 00:00:00 和 0 毫秒。

访问功能：

QTime	time() const
void	**setTime**(QTime *time*)

通知器信号：

void	**timeChanged**(QTime *time*)

也可以看看date和dateTime。

timeSpec : Qt::TimeSpec

此属性保存日期时间编辑使用的当前时间规范。

访问功能：

Qt::TimeSpec	**timeSpec() const**
void	**setTimeSpec**(Qt::TimeSpec *spec*)

成员函数文档

*[explicit]QDateTimeEdit::QDateTimeEdit(QWidget *parent = nullptr)*

构造一个空的日期时间编辑器*parent*。

*[explicit]QDateTimeEdit::QDateTimeEdit(const QDateTime &datetime, QWidget *parent = nullptr)*

构造一个空的日期时间编辑器*parent*。该值设置为*datetime*。

*[explicit]QDateTimeEdit::QDateTimeEdit(QDate date, QWidget *parent = nullptr)*

构造一个空的日期时间编辑器*parent*。该值设置为*date*。

*[explicit]QDateTimeEdit::QDateTimeEdit(QTime time, QWidget *parent = nullptr)*

构造一个空的日期时间编辑器*parent*。该值设置为*time*。

[virtual]QDateTimeEdit::~QDateTimeEdit()

析构函数。

*QCalendarWidget *QDateTimeEdit::calendarWidget() const*

返回编辑器的日历小部件，如果calendarPopup设置为 true 且 (sections() &DateSections_Mask) != 0。

如果尚未设置，此函数将创建并返回一个日历小部件。

也可以看看setCalendarWidget()。

[override virtual]void QDateTimeEdit::clear()

重新实现：QAbstractSpinBox::clear()。

QDate QDateTimeEdit::date() const

返回日期时间编辑的日期。

注意：属性日期的 Getter 函数。

也可以看看setDate()。

[signal]void QDateTimeEdit::dateChanged(QDate date)

每当日期更改时都会发出此信号。新日期已传入*date*。

注意：属性的通知程序信号date。

也可以看看Keyboard Tracking。

[signal]void QDateTimeEdit::dateTimeChanged(const QDateTime &datetime)

每当日期或时间更改时都会发出此信号。新的日期和时间被传入*datetime*。

注意：属性的通知程序信号dateTime。

也可以看看Keyboard Tracking。

[virtual protected]QDateTime QDateTimeEdit::dateTimeFromText(const QString &text) const

返回给定的适当日期时间*text*。

每当日期时间编辑需要将用户输入的文本解释为值时，就会使用此虚拟函数。

也可以看看textFromDateTime（） 和validate()。

[override virtual]bool QDateTimeEdit::event(QEvent *event)

重新实现：QAbstractSpinBox::event（QEvent *事件）。

[override virtual protected]void QDateTimeEdit::fixup(QString &input) const

重新实现：QAbstractSpinBox::fixup(QString &input) const。

[override virtual protected]void QDateTimeEdit::focusInEvent(QFocusEvent *event)

重新实现：QAbstractSpinBox::focusInEvent（QFocusEvent *事件）。

[override virtual protected]bool QDateTimeEdit::focusNextPrevChild(bool next)

重新实现：QWidget::focusNextPrevChild（接下来是布尔值）。

*[override virtual protected]void QDateTimeEdit::initStyleOption(QStyleOptionSpinBox *option) const*

重新实现：QAbstractSpinBox::initStyleOption(QStyleOptionSpinBox *option) const。

初始化*option*使用此 QDataTimeEdit 中的值。当子类需要时，此方法非常有用QStyleOptionSpinBox，但不想自己填写所有信息。

也可以看看QStyleOption::initFrom()。

*[override virtual protected]void QDateTimeEdit::keyPressEvent(QKeyEvent *event)*

重新实现：QAbstractSpinBox::keyPressEvent（QKeyEvent *事件）。

*[override virtual protected]void QDateTimeEdit::mousePressEvent(QMouseEvent *event)*

重新实现：QAbstractSpinBox::mousePressEvent（QMouseEvent *事件）。

*[override virtual protected]void QDateTimeEdit::paintEvent(QPaintEvent *event)*

重新实现：QAbstractSpinBox::paintEvent（QPaintEvent *事件）。

QDateTimeEdit::Section QDateTimeEdit::sectionAt(int index) const

返回节位于*index*。

如果格式为“yyyy/MM/dd”，则sectionAt(0)返回YearSection，sectionAt(1) 返回MonthSection，sectionAt(2) 返回YearSection,

QString QDateTimeEdit::sectionText(QDateTimeEdit::Section section) const

返回给定的文本*section*。

也可以看看currentSection()。

*void QDateTimeEdit::setCalendarWidget(QCalendarWidget *calendarWidget)*

设置给定的*calendarWidget*作为用于日历弹出窗口的小部件。编辑器不会自动取得日历小部件的所有权。

笔记：calendarPopup在设置日历小部件之前必须设置为 true。

也可以看看calendarWidget （） 和calendarPopup。

void QDateTimeEdit::setDateRange(QDate min, QDate max)

设置日期时间编辑允许的日期范围。

此便利功能设置minimumDate和maximumDate特性。

```
setDateRange(min, max);
```

类似于：

```
setMinimumDate(min);
setMaximumDate(max);
```

如果其中之一*min*或者*max*无效，该函数不执行任何操作。该函数保留了minimumTime财产。如果*max*小于*min*， 新的maximumDateTime财产应为新的minimumDateTime财产。如果*max*等于*min*和maximumTime财产少于minimumTime财产，maximumTime属性设置为minimumTime财产。否则，这将保留maximumTime财产。

如果范围比其末尾所跨越的时间间隔更窄，例如跨越月底的一周，则在禁用键盘跟踪的情况下，用户只能将日期编辑为该范围后半部分的日期。

也可以看看minimumDate,maximumDate,setDateTimeRange(),QDate::isValid （） ， 和Keyboard Tracking。

void QDateTimeEdit::setDateTimeRange(const QDateTime &min, const QDateTime &max)

设置日期时间编辑允许的日期时间范围。

此便利功能设置minimumDateTime和maximumDateTime特性。

```
setDateTimeRange(min, max);
```

类似于：

```
setMinimumDateTime(min);
setMaximumDateTime(max);
```

如果其中之一*min*或者*max*无效，该函数不执行任何操作。如果*max*小于*min*,*min*也用作*max*。

如果范围比其末尾所跨越的时间间隔更窄，例如跨越月底的一周，则在禁用键盘跟踪的情况下，用户只能将日期时间编辑为该范围后面的部分。

也可以看看 minimumDateTime,maximumDateTime,setDateRange(),setTimeRange(),QDateTime::isValid（），和Keyboard Tracking。

void QDateTimeEdit::setSelectedSection(QDateTimeEdit::Section section)

选择*section*。如果*section*当前显示的部分中不存在，该函数不执行任何操作。如果*section*是NoSection，此函数将取消选择编辑器中的所有文本。否则，此功能会将光标和当前部分移动到所选部分。

也可以看看currentSection()。

void QDateTimeEdit::setTimeRange(QTime min, QTime max)

设置日期时间编辑的允许时间范围。

此便利功能设置minimumTime和maximumTime特性。

请注意，这些仅分别限制日期时间编辑的值minimumDate和maximumDate。当这些日期属性不一致时，时间晚于*max*允许在之前的日期maximumDate和之前的时间*min*允许在以下日期使用minimumDate。

```
setTimeRange(min, max);
```

类似于：

```
setMinimumTime(min);
setMaximumTime(max);
```

如果其中之一*min*或者*max*无效，该函数不执行任何操作。该函数保留了minimumDate和maximumDate特性。如果这些属性一致并且*max*小于*min*,*min*用作*max*。

如果范围比其末尾所跨越的时间间隔更窄，例如从十点到一小时到同一小时过十点的间隔，则如果键盘跟踪是，用户只能将时间编辑为该范围的后半部分。禁用。

也可以看看minimumTime,maximumTime,setDateTimeRange(),QTime::isValid（），和Keyboard Tracking。

[override virtual]QSize QDateTimeEdit::sizeHint() const

重新实现：QAbstractSpinBox::sizeHint() const。

[override virtual]void QDateTimeEdit::stepBy(int steps)

重新实现：QAbstractSpinBox::stepBy（整数步骤）。

[override virtual protected]QAbstractSpinBox::StepEnabled QDateTimeEdit::stepEnabled() const

重新实现：QAbstractSpinBox::stepEnabled() const。

[virtual protected]QString QDateTimeEdit::textFromDateTime(const QDateTime &dateTime) const

日期时间编辑在需要显示时使用此虚拟函数*dateTime*。

如果你重新实现这个，你可能还需要重新实现validate()。

也可以看看dateTimeFromText（） 和validate()。

QTime QDateTimeEdit::time() const

返回日期时间编辑的时间。

注意：属性时间的 getter 函数。

也可以看看setTime()。

[signal]void QDateTimeEdit::timeChanged(QTime time)

每当时间改变时就会发出该信号。新的时间已传入*time*。

注意：属性的通知程序信号time。

也可以看看Keyboard Tracking。

[override virtual protected]QValidator::State QDateTimeEdit::validate(QString &text, int &pos) const

重新实现：QAbstractSpinBox::validate(QString &input, int &pos) const。

*[override virtual protected]void QDateTimeEdit::wheelEvent(QWheelEvent *event)*

重新实现：QAbstractSpinBox::wheelEvent（QWheelEvent *事件）。