4D Widgets

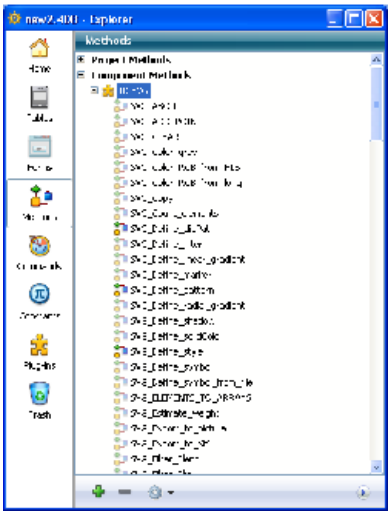
4D Widget components

4D widgets are functionalities provided as components that can be accessed from the Form editor and/or the preconfigured object library. These widgets are compound objects with predefined characteristics. They provide access to standard functions and are very simple to implement.

Three widgets are available:

- SearchPicker: search area with standard appearance.
- DatePicker: date selector.
- TimePicker: time selector.

These widgets can be used with or without programming. You can simply integrate them into your forms and use their default properties. If you want to be able to control and configure them in a more customized manner, you can make use of specific project methods, called "Component methods," which are listed by component on the **Methods** page of the Explorer:



This documentation covers each widget and describes the syntax of the component methods associated with it.

Adding a widget

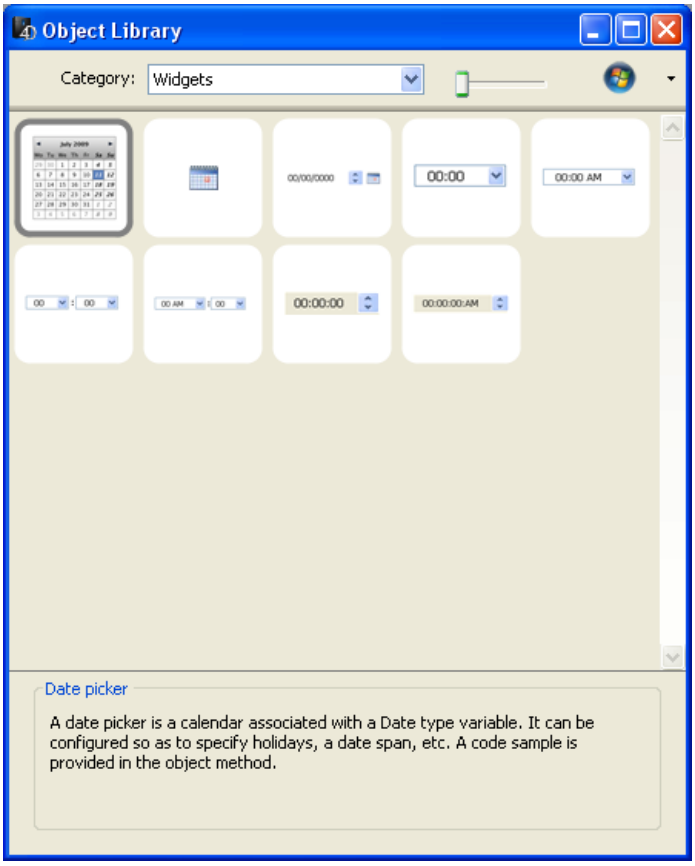
There are two ways to insert a widget area in a form:

- Using the object library
- Using a subform

Via the object library

To insert a widget using the object library:

1. Select the **Object Library** command from the **Design** menu.
   The Object Library window appears.
2. Select **Widgets** in the Category menu.
   All the available widgets are listed:

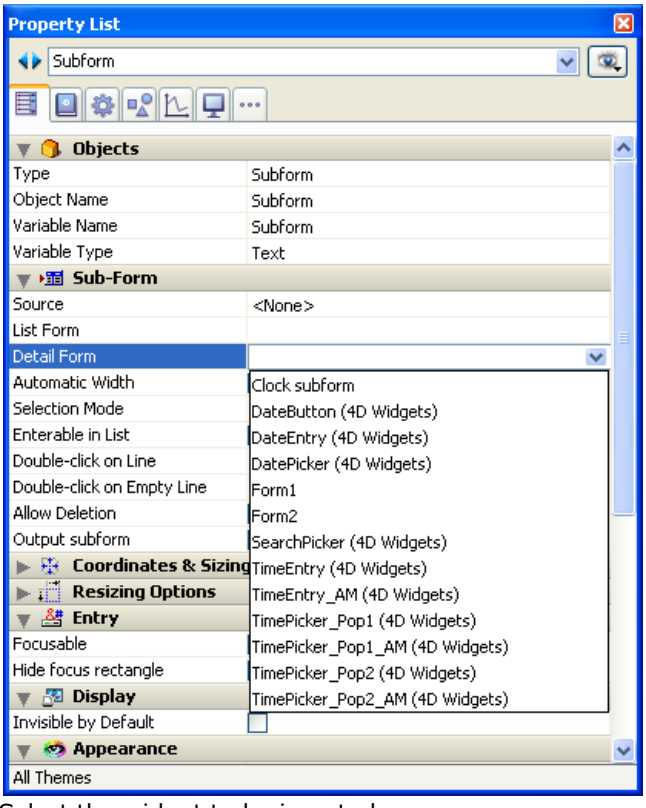


3. Insert the desired widget into your form by dragging and dropping it.
   You can then configure it via the properties of the inserted object or via its object method.

Via a subform

To create a widget via a subform type object:

1. In the Form editor, add a subform object.
   This point is described in the 4D *Design Reference* manual.
2. In the Property List, click on the "Detail Form" menu to scroll through the list of forms that can be used. This list includes the widgets.



3. Select the widget to be inserted.
   You can then configure it via the object properties or the object method of the subform.

DatePicker

- DatePicker and DateEntry
- DatePicker APPLY DEFAULT VALUES
- DatePicker Display Dialog
- DatePicker RESET DEFAULT VALUES
- DatePicker SET DAYS OFF
- DatePicker SET DEFAULT 1ST DAY
- DatePicker SET DEFAULT DAYS OFF
- DatePicker SET DEFAULT MAX DATE
- DatePicker SET DEFAULT MIN DATE
- DatePicker SET MAX DATE
- DatePicker SET MIN DATE
- DatePicker SET WEEK FIRST DAY

DatePicker and DateEntry

The DatePicker widget is an intuitive, easy-to-use object that you can use to make the most of any fields that require dates to be entered or simply represented. This widget is provided in two forms:

- **DatePicker Calendar**: This object can be used either in a subform, or as a pull-down calendar displayed by clicking a button.
- **DateEntry Area**: Date area associated with control buttons. This object can be only be used in a subform.

DatePicker calendar

A DatePicker calendar is an area displaying each month as a table of days. During execution, the user can scroll through the months of the calendar both forwards and backwards by clicking on the arrow buttons. They can also use the arrow keys of the keyboard.

When it is inserted into a subform, a DatePicker object can be used without programming thanks to the mechanism provided by the **bound variable**: you can work with the value of the variable bound with the subform object ("Variable Name" property in the Property List) in order to manage its display and entry.
When the form is executed, this date variable automatically contains the date selected by the user. Conversely, if you modify the value of this variable by programming, it will be shown automatically in the subform.

However, if you want to customize the functioning of the DatePicker or display it as a pop-up menu, you must use the set of component methods that is provided.

Use in a subform

You can insert a DatePicker calendar into a form in two ways:

- By inserting a "Date Picker" object from the preconfigured object library of 4D
- By creating a subform area and assigning the detail form of the **DatePicker** to it.

You can manage this area without programming via the bound variable mechanism (see above).

Use in a pop-up

You can use a DatePicker calendar as a pop-up window. To do this, you can either:

- insert a "Pop up date" object from the preconfigured object library,
- create a subform and assign the **DateButton** detail form to it.
  In both these cases, you can manage its display and entry by binding a date variable to the object.
- create an object that calls the **DatePicker Display Dialog** component method. This method returns the date selected by the user.

DateEntry area

A DateEntry type area facilitates the entry of a date in the form specified in the system preferences (for example

DD/MM/YY).
The area appears as a date type associated with buttons:

During execution, the buttons located to the right of the entry area are only displayed when the object has the focus. The user selects each element of the date (day, month or year) individually by clicking or hitting the Tab key and can scroll them using the numeric stepper or the arrow keys of the keyboard. The calendar icon to the right can be used to select a date from a DatePicker pop-up calendar.

A DateEntry object can be used without programming thanks to the mechanism provided by the bound variable (see the "DatePicker calendar" paragraph). However, if you want to customize the functioning, you can use the set of component methods that is provided. These methods are the same as those of the DatePicker object.

Use in a subform

You can insert a DateEntry area into a form in two ways:

- By inserting a "DateEntry area" object from the preconfigured object library of 4D.
- By creating a subform area and assign the **DateEntry** detail form to it.

DatePicker APPLY DEFAULT VALUES

DatePicker APPLY DEFAULT VALUES (nomObjet)

Parameter	Type		Description
nomObjet	Text	⇨	Nom d'objet sous-formulaire

Description

The **DatePicker APPLY DEFAULT VALUES** command is used to reset all the DatePicker parameters to their default values for the *objectName* subform object.

These default values may be the factory settings but may also have been modified via the SET DEFAULT commands of the component.

The action of this command is immediate: the default values of *objectName* are instantly modified. Note that the variable associated with the object could also be modified in order to take the new values into account. For example, if the new default values set the minimum date to 01/01/2000 and the variable associated with *objectName* was 05/05/1995, its value is automatically returned to 01/01/2000.

The DatePicker parameters include:

- minimum or maximum enterable dates
- the first day of the week
- weekly and yearly "days off" as well as specific holidays

Example

This example resets the parameters of the Date1 object to their default settings:

```
DatePicker APPLY DEFAULT VALUES("Date1")
```

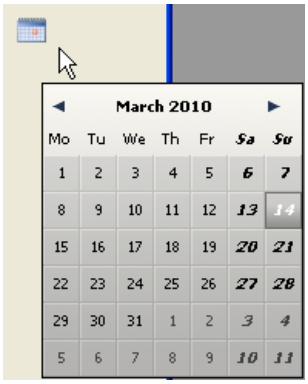
DatePicker Display Dialog

DatePicker Display Dialog {(left ; top {; defaultDate})} -> Function result

Parameter	Type		Description
left	Longint	➡	Location for left side of window
top	Longint	➡	Location for top of window
defaultDate	Date	➡	Date to select by default in the dialog
Function result	Date	⮌	Date selected by user

Description

The **DatePicker Display Dialog** command opens a DatePicker calendar in a pop-up window (a pop-up type window is automatically closed when the user clicks outside the window or hits the **Enter** or **Esc** key).



The two optional *left* and *top* parameters are used to specify the location of the top left corner of the window to be opened. These two parameters must be passed together; if only one is passed, it is ignored. If these parameters are omitted, the window is opened at the location of the click.

You can pass a *defaultDate* date as third parameter in order to set the DatePicker dialog have an associated date pre-selected when displayed to the user.

DatePicker Display Dialog returns the date selected by the user in the DatePicker calendar. If the window is closed without a date being selected by the user, the command returns a blank date (!00/00/00!), even if a *defaultDate* date was passed.

Example

This example displays a DatePicker calendar when a button is clicked:

```
OBJECT GET COORDINATES(*;"MyCalendarButton";$x1;$y1;$x2;$y2)
$MyLocalDate:=DatePicker Display Dialog($x1;$y1)
If($MyLocalDate #!00/00/00!)
   [Event]DateRV:=$MyLocalDate
End if
```

DatePicker RESET DEFAULT VALUES

DatePicker RESET DEFAULT VALUES

Does not require any parameters

Description

The **DatePicker RESET DEFAULT VALUES** command resets the parameters of the DatePicker component to their "factory settings". After executing this component method:

- the minimum or maximum enterable dates are 00/00/00 (meaning there are no limits)
- the first day of the week is 2 (Monday)
- the weekly "days off" are Saturday and Sunday
- no yearly "days off" or specific holidays are set.

Note that this setting is only taken into account for calendars created subsequently and is not applied to any existing calendars. If you want to apply it to existing calendars, you must use the **DatePicker APPLY DEFAULT VALUES** component method.

DatePicker SET DAYS OFF

DatePicker SET DAYS OFF (objectName {; dayType ; ptrDaysOffArray})

Parameter	Type		Description
objectName	Text	⇒	Name of subform object
dayType	Longint	⇒	Types of days off
ptrDaysOffArray	Pointer	⇒	Pointer to date or Boolean array of days off

Description

The **DatePicker SET DAYS OFF** command is used to set the days off to appear in the DatePicker calendar. These days are displayed in bold and italic and remain selectable for the user.

The *objectName* parameter specifies the instance of the subform to which the command must be applied. In this parameter, you must pass the name of a subform object displayed in the current form.

This command can be used to set either recurrent weekly or yearly days off as well as occasional holidays. You specify the type of days off set via the *dayType* parameter:

- 0 = Days off occurring weekly (by default, Saturday and Sunday)
- 1 = Days off that occur every year (such as January 1st or December 25th)
- 2 = Occasional holidays, set for a single year

You set the holidays by creating a array and by passing a pointer to this array as the *ptrDaysOffArray* parameter. The type of array depends on the value passed in *dayType*:

- If you pass 0 in *dayType* (weekly days off), in *ptrDaysOffArray* you must pass a pointer to a Boolean array made up of 7 elements. Each element set to True indicates a weekly day off.
- If you pass 1 or 2 in *dayType* (yearly or occasional days off), in *ptrDaysOffArray* you must pass a pointer to a Date array. In this array, each element must contain a valid date, indicating a day off. The dates must be expressed in the default format corresponding to the system language. If you passed 1 in *dayType* (recurrent days), the year is ignored; you can pass any value.

Example 1

Designation of Friday as the weekly day off (instead of Saturday and Sunday by default):

```
ARRAY BOOLEAN($arrbDaysOff;7)
 //By default, all the elements of a Boolean array are False; thus it is not necessary to add initialization code
$arrbDaysOff{Friday}:=True
DatePicker SET DAYS OFF("mycalendar";0;->$arrbDaysOff)
```

Example 2

Designation of occasional holidays:

```
ARRAY DATE($arrdUniqueDays;0)
 //The year is taken into account
APPEND TO ARRAY($arrdUniqueDays;!02/15/2008!)
APPEND TO ARRAY($arrdUniqueDays;!02/12/2009!)
APPEND TO ARRAY($arrdUniqueDays;!02/17/2010!)
DatePicker SET DAYS OFF(1;->$arrdUniqueDays)
```

DatePicker SET DEFAULT 1ST DAY

DatePicker SET DEFAULT 1ST DAY (dayNum)

Parameter	Type		Description
dayNum	Longint	⇨	First day of the week

Description

The **DatePicker SET DEFAULT 1ST DAY** command is used to set the first day of the week to be displayed by default in the left part of all DatePicker calendars.

In the *dayNum* parameter, pass one of the following 4D constants found in the **Days and Months** theme:

Constant	Type	Value
Sunday	Longint	1
Monday	Longint	2
Tuesday	Longint	3
Wednesday	Longint	4
Thursday	Longint	5
Friday	Longint	6
Saturday	Longint	7

Note that this parameter is only taken into account for calendars created subsequently and does not apply to any existing calendars. If you want to apply it to existing calendars, you must use the **DatePicker APPLY DEFAULT VALUES** component method.

DatePicker SET DEFAULT DAYS OFF

DatePicker SET DEFAULT DAYS OFF (dayType ; ptrDaysOffArray)

Parameter	Type		Description
dayType	Longint	⇒	Types of days off
ptrDaysOffArray	Pointer	⇒	Pointer to date or Boolean array of days off

Description

The **DatePicker SET DEFAULT DAYS OFF** command is used to set the days off that will appear in all the calendars of the DatePicker component. These days are displayed in bold and italic and remain selectable for the user.

Note that this setting is only taken into account for calendars that are created subsequently and does not apply to any existing calendars. If you want to apply it to the existing calendars, you will need to use the **DatePicker APPLY DEFAULT VALUES** component method.

The command can be used to set recurrent weekly or yearly days off as well as occasional holidays. You specify the type of days off set by the method via the *dayType* parameter:

- 0 = Days off occurring weekly (by default, Saturday and Sunday)
- 1 = Days off that occur every year (such as January 1st or December 25th)
- 2 = Occasional holidays, set for a single year

You set the holidays by creating a array and by passing a pointer to this array as the *ptrDaysOffArray* parameter. The type of array depends on the value passed in *dayType*:

- If you pass 0 in *dayType* (weekly days off), in *ptrDaysOffArray* you must pass a pointer to a Boolean array made up of 7 elements. Each element set to True indicates a weekly day off.
- If you pass 1 or 2 in *dayType* (yearly or occasional days off), in *ptrDaysOffArray* you must pass a pointer to a Date array. In this array, each element must contain a valid date, indicating a day off. The dates must be expressed in the default format corresponding to the system language. If you passed 1 in *dayType* (recurrent days), the year is ignored; you can pass any value.

Example

Designation of recurrent holidays (example valid for the USA):

```
ARRAY DATE($arrdRepeatedDays;0)
 //The year is ignored; we use 2000 by default
APPEND TO ARRAY($arrdRepeatedDays;!01/01/2000!)
APPEND TO ARRAY($arrdRepeatedDays;!02/02/2000!)
APPEND TO ARRAY($arrdRepeatedDays;!02/14/2000!)
APPEND TO ARRAY($arrdRepeatedDays;!03/17/2000!)
APPEND TO ARRAY($arrdRepeatedDays;!04/01/2000!)
APPEND TO ARRAY($arrdRepeatedDays;!10/31/2000!)
APPEND TO ARRAY($arrdRepeatedDays;!11/11/2000!)
APPEND TO ARRAY($arrdRepeatedDays;!12/25/2000!)
DatePicker SET DEFAULT DAYS OFF(1;->$arrdRepeatedDays)
```

DatePicker SET DEFAULT MAX DATE

DatePicker SET DEFAULT MAX DATE (maxDate)

Parameter	Type		Description
maxDate	Date	⇨	Upper limit of enterable date

Description

The **DatePicker SET DEFAULT MAX DATE** command is used to set the maximum enterable day for all the calendars of the DatePicker component.

Note that this parameter is only taken into account for calendars created subsequently and does not apply to any existing calendars. If you want to apply it to existing calendars, you must use the **DatePicker APPLY DEFAULT VALUES** component method.

DatePicker SET DEFAULT MIN DATE

DatePicker SET DEFAULT MIN DATE (dateMin)

Parameter	Type		Description
dateMin	Date	⇨	Limite inférieure de date saisissable

Description

The **DatePicker SET DEFAULT MIN DATE** command is used to set the minimum enterable day for all the calendars of the DatePicker component.

Note that this parameter is only taken into account for calendars created subsequently and does not apply to any existing calendars. If you want to apply it to existing calendars, you must use the **DatePicker APPLY DEFAULT VALUES** command.

Example

Designation of minimum date to January 1, 2000:

```
DatePicker SET DEFAULT MIN DATE(!01/01/2000!)
```

DatePicker SET MAX DATE

DatePicker SET MAX DATE (objectName ; maxDate)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
maxDate	Date	⇨	Upper limit of enterable date

Description

The **DatePicker SET MAX DATE** command is used to specify the maximum enterable date in the DatePicker calendar (the days located after this maximum date appear grayed out in the calendar).

The *objectName* specifies the instance of the subform to which the command must be applied. In this parameter, you must pass the name of a subform object displayed in the current form.

The *maxDate* date must be expressed in the default entry format corresponding to the system language. If you pass a blank date (!00/00/00!), all the dates that follow the current date are enterable.

If the maximum enterable date is earlier than the minimum enterable date (see **DatePicker SET MIN DATE**), no date will be enterable.

Example

Disabling all dates after December 31, 2009 in the object named "ReturnDate":

```
DatePicker SET MAX DATE("ReturnDate";!12/31/2009!)
```

DatePicker SET MIN DATE

DatePicker SET MIN DATE (objectName ; dateMin)

Parameter	Type		Description
objectName	Text	⇒	Name of subform object
dateMin	Date	⇒	Lower limit of enterable date

Description

The **DatePicker SET MIN DATE** command is used to set the minimum enterable date in a DatePicker calendar (the days located before this minimum date appear grayed out in the calendar).

The *objectName* parameter specifies the instance of the subform to which the command must be applied. In this parameter, you must pass the name of a subform object displayed in the current form.

The *minDate* date must be expressed in the default entry format corresponding to the system language. If you pass a blank date (!00/00/00!), all the dates preceding the current date will be enterable.

If the minimum enterable date is later than the maximum enterable date (see **DatePicker SET MAX DATE**), no date will be enterable.

Example

The current form contains two DatePicker calendars located in two subform objects named "DP1" and "DP2".

```
 //Disabling all dates before January 1, 2009 in the first calendar
DatePicker SET MIN DATE("DP1";!01/01/2009!)
 //Disabling all dates before March 1st, 2009 in the second calendar
DatePicker SET MIN DATE("DP2";!03/01/2009!)
```

DatePicker SET WEEK FIRST DAY

DatePicker SET WEEK FIRST DAY (objectName ; dayNum)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
dayNum	Longint	⇨	Number of first day to display

Description

The **DatePicker SET WEEK FIRST DAY** command is used to set the first day of the week to display in the left part of a DatePicker calendar. By default, the first day is Monday.

The *objectName* parameter specifies the instance of the subform to which the command must be applied. In this parameter, you must pass the name of a subform object displayed in the current form.

Pass a 4D constant from the **Days and Months** theme in the *dayNum* parameter:

Constant	Type	Value
Sunday	Longint	1
Monday	Longint	2
Tuesday	Longint	3
Wednesday	Longint	4
Thursday	Longint	5
Friday	Longint	6
Saturday	Longint	7

Example 1

Setting first day to Sunday:

```
DatePicker SET WEEK FIRST DAY("mycalendar";Sunday)
```

Example 2

Setting first day to Thursday:

```
DatePicker SET WEEK FIRST DAY("mycalendar";Thursday)
```


March 2010
Th Fr Sa Su Mo Tu We
25 26 27 28 1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
1 2 3 4 5 6 7

SearchPicker

Overview

With the SearchPicker widget, you can easily create standard search areas, similar to those found in browsers or tool bars. The appearance of the area depends on the platform.

Windows: 

Mac OS: 

The text displayed by default in the area can be controlled by programming, using the **SearchPicker SET HELP TEXT** component method.

Functioning

Besides its appearance, a SearchPicker search area is characterized by the following elements: grayed out text, an entry area and a deletion icon.

- The entry area is where the user enters the value to be searched for. This value is automatically and dynamically assigned to the variable that you have bound with the area in the Property list ("Variable Name" property). You use this variable to supply the value searched for to the search method.
- The grayed out text is an aid indicating the field(s) where the search will be carried out to the user. It disappears as soon as the area has the focus. This text can be set via the **SearchPicker SET HELP TEXT** command.
- The deletion button can be used to erase the contents of the area. Its functioning is automatic.

During execution, you can launch your search method by clicking on a custom button in the form or via a form event. The area generates more particularly the On Data Change and On Losing Focus events. You can provide a dynamic search that is reevaluated each time the user enters another character by calling the search method in the On Data Change event.

Creation

You can insert a SearchPicker area in a form in two ways:

- By inserting a "SearchPicker " area from the preconfigured object library of 4D ("Entry areas" theme).
- By creating a subform area and assigning the **SearchPicker** detail form to it.

Then specify the name of the variable bound with the subform ("Variable Name" property in the Property list). When the form is executed, this variable will automatically contain the value searched for by the user. You can then pass this value to your custom search method.

SearchPicker SET HELP TEXT

SearchPicker SET HELP TEXT (objectName ; helpText)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
helpText	Text	⇨	Text to display

Description

The **SearchPicker SET HELP TEXT** command is used to display a non-enterable grayed-out text in the background of the search area specified by *objectName*. This text disappears when the user clicks in the area.

Example

Displays the word "Country" in the area, indicating that the search will concern this variable:

```
SearchPicker SET HELP TEXT("vSearch";"Country")
```

TimePicker

- TimePicker and TimeEntry
- TimePicker APPLY DEFAULT VALUES
- TimePicker DISPLAY SECOND HAND
- TimePicker LCD DISPLAY AMPM
- TimePicker LCD DISPLAY SECONDS
- TimePicker LCD SET COLOR
- TimePicker LCD SET MODE
- TimePicker RESET DEFAULT VALUES
- TimePicker SET DEFAULT LABEL AM
- TimePicker SET DEFAULT LABEL PM
- TimePicker SET DEFAULT MAX TIME
- TimePicker SET DEFAULT MIN TIME
- TimePicker SET DEFAULT STEP
- TimePicker SET LABEL AM
- TimePicker SET LABEL PM
- TimePicker SET MAX TIME
- TimePicker SET MIN TIME
- TimePicker SET STEP

TimePicker and TimeEntry

The TimePicker widget provides easy-to-use objects that you can use to make the most of any fields that require times to be entered or displayed. It can be used in the following forms:

- Single or double pop up menus:

- Time entry areas in the "hh:mm:ss" format associated with a numeric stepper that can be used to increase or decrease the value of the hours, minutes or seconds:

- Clocks (*TimeDisplay*) or digital clocks (*TimeDisplayLCD*):


In addition, each type of TimePicker can display the time in 12-hour (AM-PM) or 24-hour format.

TimePicker object can be used without programming thanks to the mechanisms provided by the bound variable. However, if you want to customize the functioning of TimePicker objects, you can use the set of component methods that is provided.

Creation and use

You can insert a TimePicker area into a form in two ways:

- By inserting a "TimePicker" or "TimeEntry" object from the preconfigured object library of 4D.
- By creating a subform area and assigning the **TimePicker** or **TimeEntry** detail form of your choice to it.

Then specify the name of the variable bound to the subform ("Variable Name" property in the Property List). When the form is executed, this variable will automatically contain the time specified by the user. Conversely, if you modify the value of this variable by programming, it will automatically be shown in the subform. You can also choose not to name the variable in order to benefit from the dynamic variable mechanism.

About clocks (new in v14)

Clock widgets are drawn in SVG, and therefore have a vector path allowing deformations in Application mode (in Design mode, their size is fixed):

Note that:

- For a standard clock, the second hand can be displayed or hidden using the **TimePicker DISPLAY SECOND HAND** method.
- A standard clock automatically changes to "day mode" or "night mode" depending on the time:

The time ranges are 8:00:00 -> 19:59:59 = Day, 20:00 -> 07:59:59 = Night.
- The "digital clock" widget is transparent and has no background, so it can be placed on top of colored objects in order to vary its appearance:

There are several display options for this widget that are available through component methods, which are prefixed by "TimePicker LCD".

Note: Developers can displace this clock drawing and substitute their own creations by replacing the "clock.svg" file found at the first level of the "Resources" folder.

Displaying the current time or a static time

Clocks can either display the current time dynamically, or show a static time.

- To display the current time, associate a **Real** variable to the subform object of the widget (default operation). In this case, the widget automatically displays the current time and operates like a clock.
  You can apply an offset to the displayed time: the value of the number variable indicates the offset in seconds. For example, 3600 = advancing the clock one hour, -1800 = turning the clock back 30 minutes, etc.
- To display a static time, associate a **Time** variable to the subform object of the widget (by means of the **C_TIME** command or the Property List). The clock then displays the value of the this variable.

For example, we want the clock to show 10:10:30:

```
C_TIME(myvar) // myvar is the name of the widget's variable
myvar:=?10:10:30?
```

TimePicker APPLY DEFAULT VALUES

TimePicker APPLY DEFAULT VALUES (objectName)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object

Description

The **TimePicker APPLY DEFAULT VALUES** command is used to reset all the TimePicker parameters to their current default values for the *objectName* subform object.

These default values may be the factory settings but may also have been modified via the SET DEFAULT commands of the component.

The action of this command is immediate: the default values of *objectName* are instantly modified. Note that the bound variable of the object could also be modified in order to take the new values into account. For example, if the new default values set the minimum time to 07:00:00 and the value of the variable bound with *objectName* was 06:00:00, its value is automatically returned to 07:00:00.

The TimePicker parameters include:

- minimum or maximum enterable times,
- the AM and PM labels,
- the steps in minutes.

TimePicker DISPLAY SECOND HAND

TimePicker DISPLAY SECOND HAND (objectName ; secondHand)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
secondHand	Boolean	⇨	True (default) = second hand shown, False = second hand hidden

Description

The **TimePicker DISPLAY SECOND HAND** command displays or hides the second hand in the *objectName* subform object (clock widget only).

By default, the second hand is displayed. To hide it, call this command and pass **False** in the *secondHand* parameter.

TimePicker LCD DISPLAY AMPM

TimePicker LCD DISPLAY AMPM (objectName ; amPm)

Parameter	Type		Description
objectName	Text	⇒	Name of subform object
amPm	Boolean	⇒	True = display AM/PM, False = do not display

Description

The **TimePicker LCD DISPLAY AMPM** displays or hides the AM/PM placed to the right of the *objectName* subform object (digital clock only).

These letters are used to distinguish between the morning and afternoon when the clock is used in 12-hour mode (see **TimePicker LCD SET MODE**).

By default, these letters are displayed. You can pass **False** in *amPm* to hide them.

Example

We want to hide the AM/PM:

```
TimePicker LCD DISPLAY AMPM("Subform1";False)
```

TimePicker LCD DISPLAY SECONDS

TimePicker LCD DISPLAY SECONDS (objectName ; seconds)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
seconds	Boolean	⇨	True = display seconds, False = do not display

Description

The **TimePicker LCD DISPLAY SECONDS** displays or hides the seconds part of the *objectName* subform object (digital clock only).

By default, seconds are displayed. You can pass **False** in *seconds* to hide them.

TimePicker LCD SET COLOR

TimePicker LCD SET COLOR (objectName ; color {; colorG ; colorB})

Parameter	Type		Description
objectName	Text	⇒	Name of subform object
color	Longint	⇒	Value of RGB color (4 bytes) or Value of red component (0..255) if the other parameters are passed
colorG	Longint	⇒	Value of green component (0..255)
colorB	Longint	⇒	Value of blue component (0..255)

Description

The **TimePicker LCD SET COLOR** command sets the colors for the digits in the *objectName* subform object (digital clock only).

This command accepts two syntaxes:

- If you only pass the *color* parameter, you must pass a 4-byte longint whose format (0x00RRGGBB) is described below (the bytes are numbered from 0 to 3, starting from right to left):

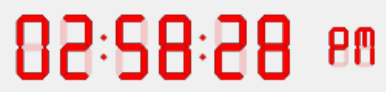
Byte	Description
3	Must be zero for an absolute RGB color
2	Red component of color (0..255)
1	Green component of color (0..255)
0	Blue component of color (0..255)

- You can also pass three parameters: *color*, *colorG* and *colorB*. In this case, each parameter must be a number between 0 and 255, representing a component of the RGB color.

Example

Change the clock digits to red:

```
TimePicker LCD SET COLOR("Subform1";0x00FF0000)
 // can also be written: TimePicker LCD SET COLOR ("Subform1";255;0;0)
```



TimePicker LCD SET MODE

TimePicker LCD SET MODE (objectName ; mode)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
mode	Longint	⇨	12 = display time in 12-hour mode, 24 = display time in 24-hour mode

Description

The **TimePicker LCD SET MODE** sets the display to either 12- or 24-hour mode for the *objectName* subform object (digital clock only).

By default, the object is displayed in 12-hour mode. You can pass the value 24 in the *mode* parameter if you want to switch to 24-hour mode. In this case, it is generally a good idea to hide the AM/PM as well (see the **TimePicker LCD DISPLAY AMPM**).

Example

We want to switch to 24-hour mode and hide the AM/PM:

```
TimePicker LCD SET MODE("Subform1";24)
TimePicker LCD DISPLAY AMPM("Subform1";False)
```

TimePicker RESET DEFAULT VALUES

TimePicker RESET DEFAULT VALUES

Does not require any parameters

Description

The **TimePicker RESET DEFAULT VALUES** command resets the parameters of the TimePicker component to their "factory settings". After execution of this command:

- the minimum enterable time is 08:00:00
- the maximum enterable time is 20:00:00
- the AM and PM labels are the system labels
- the steps in minutes is 00:15:00

Note that this parameter is only taken into account for TimePicker objects created subsequently and does not apply to any existing objects. If you want to apply it to existing objects, you must use the **TimePicker APPLY DEFAULT VALUES** component method.

TimePicker SET DEFAULT LABEL AM

TimePicker SET DEFAULT LABEL AM (label)

Parameter	Type		Description
label	Text	⇨	Label to use for AM

Description

The **TimePicker SET DEFAULT LABEL AM** command is used to modify the default "AM" label in all the TimePicker objects displaying the AM/PM format.

This setting is only taken into account for objects created subsequently and does not apply to any existing objects. If you want to apply it to existing objects, you must use the **TimePicker APPLY DEFAULT VALUES** command.

TimePicker SET DEFAULT LABEL PM

TimePicker SET DEFAULT LABEL PM (label)

Parameter	Type		Description
label	Text	⇨	Label to use for PM

Description

The **TimePicker SET LABEL PM** command is used to modify the default "PM" label in all the TimePicker objects displaying the AM/PM format.

This setting is only taken into account for objects created subsequently and does not apply to any existing objects. If you want to apply it to existing objects, you must use the **TimePicker APPLY DEFAULT VALUES** command.

TimePicker SET DEFAULT MAX TIME

TimePicker SET DEFAULT MAX TIME (maxTime)

Parameter	Type		Description
maxTime	Time	⇨	Upper limit of enterable time

Description

The **TimePicker SET DEFAULT MAX TIME** command is used to specify the maximum enterable time that will be allowed by default for all the TimePicker objects.

This setting is only taken into account for objects created subsequently and does not apply to any existing objects. If you want to apply it to existing objects, you must use the **TimePicker APPLY DEFAULT VALUES** command.

TimePicker SET DEFAULT MIN TIME

TimePicker SET DEFAULT MIN TIME (minTime)

Parameter	Type		Description
minTime	Time	⇨	Lower limit of enterable time

Description

The **TimePicker SET DEFAULT MIN TIME** command is used to specify the minimum enterable time that will be allowed by default for all the TimePicker objects.

This setting is only taken into account for objects created subsequently and does not apply to any existing objects. If you want to apply it to existing objects, you must use the **TimePicker APPLY DEFAULT VALUES** command.

TimePicker SET DEFAULT STEP

TimePicker SET DEFAULT STEP (step)

Parameter	Type		Description
step	Time	⇨	Interval between two time values

Description

The **TimePicker SET DEFAULT STEP** command is used to set the step between time values for all the TimePicker objects.

This setting is only taken into account for objects created subsequently and does not apply to any existing objects. If you want to apply it to existing objects, you must use the **TimePicker APPLY DEFAULT VALUES** command.

TimePicker SET LABEL AM

TimePicker SET LABEL AM (objectName ; label)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
label	Text	⇨	Label to use for AM

Description

The **TimePicker SET LABEL AM** command is used to modify the "AM" label in TimePicker objects displaying the AM/PM format. The command applies to the object designated by *objectName*. By default, the system am/pm labels are used.

Example

Using by default the "in the morning" label instead of the system label for AM:

```
TimePicker SET LABEL AM("clock";"in the morning")
```

TimePicker SET LABEL PM

TimePicker SET LABEL PM (objectName ; label)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
label	Text	⇨	Label to use for PM

Description

The **TimePicker SET LABEL PM** command is used to modify the "PM" label in TimePicker objects displaying the AM/PM format. The command applies to the object designated by *objectName*. By default, the system am/pm labels are used.

Example

Using by default the "in the evening" label instead of the system label for PM:

```
TimePicker SET LABEL PM("clock";"in the evening")
```

TimePicker SET MAX TIME

TimePicker SET MAX TIME (objectName ; maxTime)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
maxTime	Time	⇨	Upper limit of enterable time

Description

The **TimePicker SET MAX TIME** command is used to set the maximum enterable time that will be accepted by the object designated by *objectName*. If a higher time value is entered, it will be rejected.

TimePicker SET MIN TIME

TimePicker SET MIN TIME (objectName ; minTime)

Parameter	Type		Description
objectName	Text	⇨	Name of subform object
minTime	Time	⇨	Lower limit of enterable time

Description

The **TimePicker SET MIN TIME** command is used to set the minimum enterable time that will be accepted by the object designated by *objectName*. If a lower time value is entered, it will be rejected.

TimePicker SET STEP

TimePicker SET STEP (objectName ; step)

Parameter	Type		Description
objectName	Text	➡	Name of subform object
step	Time	➡	Interval between two time values

Description

The **TimePicker SET STEP** command is used to set the step between time values available for the object designated by *objectName*. This parameter only applies to TimePickers displayed as pop-up menus.

The *step* value must be included between 1 minute and 1 hour and must be shown as whole divisions of 60 minutes. In practice, only the values 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 and 60 min are possible. Any other value will automatically be rounded off in order to correspond to this principle.

Example

Configuration of the TimePicker in the form of a pop-up menu named "time1", limitation of enterable times from 8:30 to 16:30 with 10-minute steps:

```
TimePicker SET MIN TIME("time1";?08:30:00?)
TimePicker SET MAX TIME("time1";?16:30:00?)
TimePicker SET STEP("time1";?00:10:00?)
```