






# 4D Widgets

-  Composants 4D Widgets
-  DatePicker
-  SearchPicker
-  TimePicker
-  Liste alphabétique des commandes

## Composants 4D Widgets

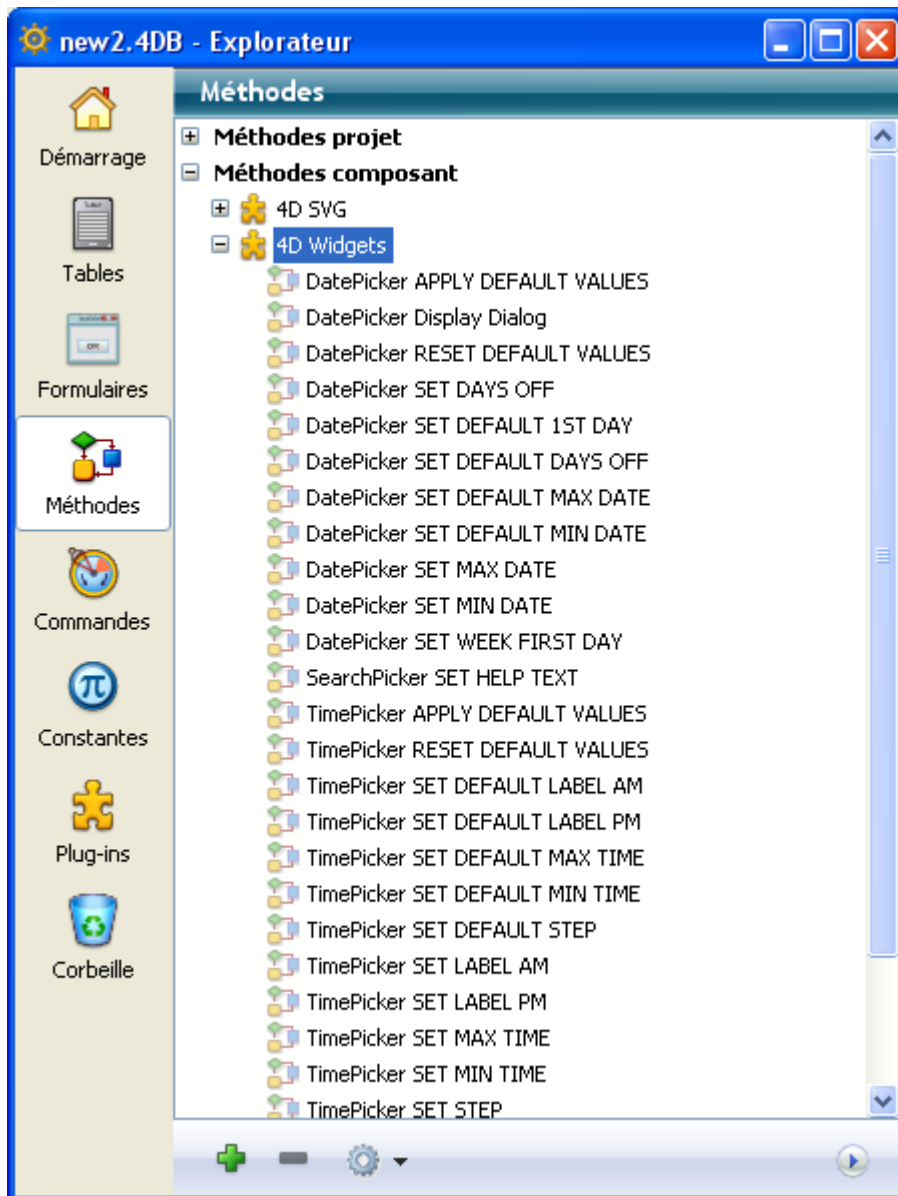
---

Les *widgets 4D* sont des fonctionnalités fournies sous forme de composants. Accessibles depuis l'éditeur de formulaires et la bibliothèque d'objets préconfigurés, ces widgets sont des objets composés disposant de caractéristiques prédéfinies et donnant accès à des fonctionnalités standard très simples à mettre en oeuvre.

Trois widgets sont disponibles :

- SearchPicker : zone de recherche d'apparence standard.
- DatePicker : sélecteur de date.
- TimePicker : sélecteur d'heure.

Ces widgets sont utilisables avec ou sans programmation. Vous pouvez les intégrer simplement dans vos formulaires et utiliser leurs propriétés par défaut. Si vous souhaitez pouvoir les contrôler et les paramétrer de façon plus personnalisée, il vous suffit de faire appel à des méthodes projet spécifiques, appelées "Méthodes composant". Ces méthodes sont listées par composant dans la page **Méthodes** de l'Explorateur :



Cette documentation présente chaque widget et décrit la syntaxe des méthodes composant qui lui sont associées.

## Ajouter un widget

---

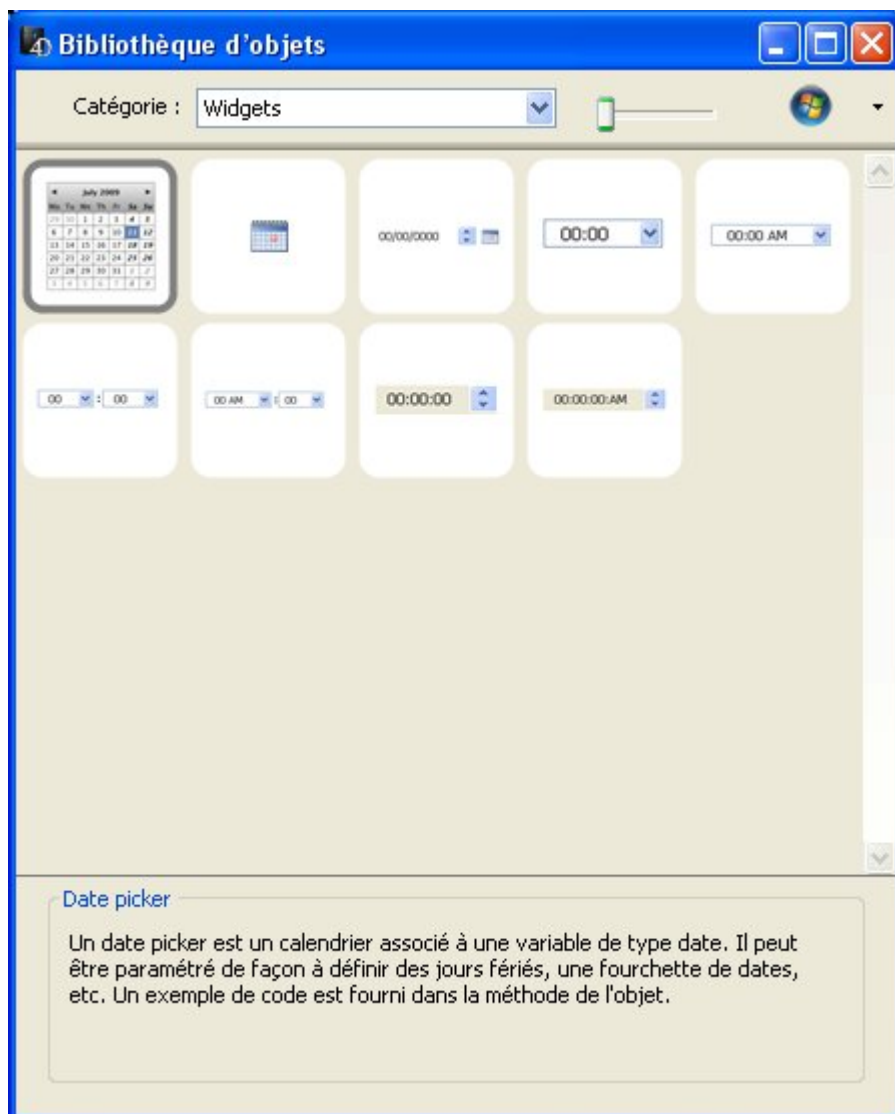
Pour insérer une zone de widget dans un formulaire, vous disposez de deux possibilités :

- Utiliser la bibliothèque d'objets
- Utiliser un sous-formulaire

### Via la bibliothèque d'objets

Pour insérer un widget via la bibliothèque d'objets :

1. Sélectionnez la commande **Bibliothèque d'objets** dans le menu **Développement**.  
La fenêtre de la Bibliothèque d'objets préconfigurés apparaît.
2. Sélectionnez **Widgets** dans le menu Catégories.  
Les widgets disponibles sont listés :

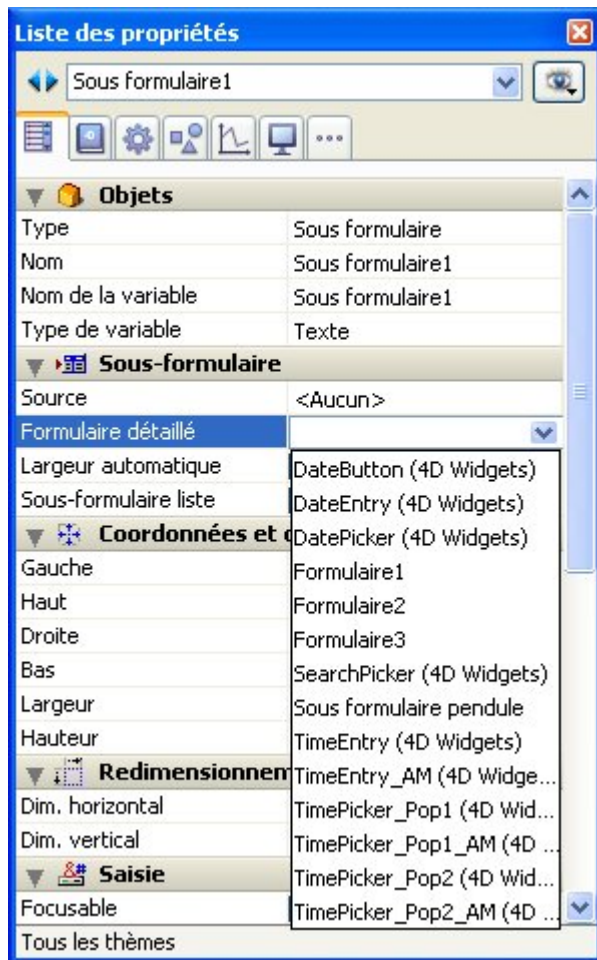


3. Insérez le widget souhaité dans votre formulaire par glisser-déposer. Vous pouvez ensuite le configurer via les propriétés de l'objet inséré ou sa méthode objet.

## Via un sous-formulaire

Pour créer un widget via un objet de type sous-formulaire :

1. Dans l'éditeur de formulaires, ajoutez un objet sous-formulaire. Ce point est décrit dans le manuel *Mode Développement* de 4D.
2. Dans la liste des propriétés, déroulez le menu "Formulaire détaillé" afin d'afficher la liste des formulaires utilisables. Cette liste inclut les widgets.



3. Sélectionnez le widget à insérer.

Vous pouvez ensuite le configurer via les propriétés de l'objet ou la méthode objet du sous-formulaire.

## Initialisation des widgets en page 2 et suivantes

Les widgets sont basés sur des sous-formulaires 4D et sont donc soumis aux mêmes principes de fonctionnement. Vous devez notamment porter une attention particulière à leur initialisation.

Pour des raisons d'optimisation, les objets des sous-formulaires sont instanciés par 4D uniquement lorsque le sous-formulaire est affiché. Cela signifie que les appels de méthodes via **EXECUTE METHOD IN SUBFORM** ou l'accès aux objets dynamiques des sous-formulaires sont impossibles tant que le sous-formulaire n'est pas affiché sur la page courante.

Lorsque vous placez des widgets sur des pages de formulaires autres que la page 1, il n'est donc pas possible d'appeler une méthode d'initialisation (telle que **TimePicker SET STEP**) dans l'événement formulaire On Load, car au moment de son exécution le widget ne sera pas instancié.

Pour traiter ce cas, les zones de widgets retournent un **événement spécifique de valeur -1** pour indiquer qu'ils sont chargés et prêts à être utilisés. Cet événement doit être testé au niveau de la méthode objet du widget lui-même, au même titre que les On Data Change, On Load, etc. Il est donc nécessaire d'écrire, lorsque le widget est situé sur une page de formulaire autre que la page 1 :













```

If(Form event=-1) //le widget est instancié et peut être initialisé
    TimePicker SET STEP("myTimePicker";?00:10:00?) // exemple
End if
  
```

plutôt que :

```
If(Form event=On Load) //fonctionne uniquement si le widget est en page 1
  TimePicker SET STEP("myTimePicker";?00:10:00?)
End if
```

# DatePicker

-  DatePicker et 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 et DateEntry

Le widget DatePicker est un objet intuitif et très simple d'emploi que vous pouvez utiliser pour valoriser tout champ nécessitant la saisie d'une date ou simplement pour représenter une date. Ce widget est proposé sous deux formes :

- **Calendrier DatePicker** : cet objet est utilisable soit dans un sous-formulaire, soit en tant que calendrier déroulant affiché par un clic sur un bouton.
- **Zone DateEntry** : zone de date associée à des boutons de contrôle. Cet objet est utilisable dans un sous-formulaire uniquement.

### Calendrier DatePicker

Un calendrier DatePicker est une zone affichant un mois sous forme de tableau de jours. En exécution, l'utilisateur peut faire défiler les mois du calendrier vers l'avant ou vers l'arrière en cliquant sur des boutons en forme de flèche. Il peut également utiliser les touches fléchées du clavier.



| Mars 2010 |    |    |    |    |           |           |
|-----------|----|----|----|----|-----------|-----------|
| Lu        | Ma | Me | Je | Ve | Sa        | Di        |
| 1         | 2  | 3  | 4  | 5  | <b>6</b>  | <b>7</b>  |
| 8         | 9  | 10 | 11 | 12 | <b>13</b> | <b>14</b> |
| 15        | 16 | 17 | 18 | 19 | <b>20</b> | <b>21</b> |
| 22        | 23 | 24 | 25 | 26 | <b>27</b> | <b>28</b> |
| 29        | 30 | 31 | 1  | 2  | 3         | 4         |
| 5         | 6  | 7  | 8  | 9  | 10        | 11        |

Lorsqu'il est inséré dans un sous-formulaire, un objet DatePicker peut être utilisé sans programmation grâce au mécanisme fourni par la **variable associée** : vous pouvez manipuler la valeur de la variable associée à l'objet sous-formulaire (propriété "Nom de la variable" dans la Liste des propriétés) afin de gérer l'affichage et la saisie.

A l'exécution du formulaire, cette variable date contiendra automatiquement la date sélectionnée par l'utilisateur. A l'inverse, si vous modifiez par programmation la valeur de cette variable, elle sera automatiquement représentée dans le sous-formulaire.

Toutefois, si vous souhaitez personnaliser le fonctionnement du Datepicker ou l'afficher sous forme de pop up menu, vous devez utiliser l'ensemble de méthodes composant qui est fourni.

### Utilisation en sous-formulaire

Vous pouvez insérer un calendrier DatePicker dans un formulaire de deux manières :

- en insérant un objet "Date Picker" depuis la bibliothèque d'objets préconfigurée de 4D



- en créant une zone de sous-formulaire et en lui associant le formulaire détaillé **DatePicker**.

Vous pouvez gérer cette zone sans programmation via le mécanisme de la variable associée (cf. ci-dessus).

### Utilisation dans un pop up

Vous pouvez utiliser un calendrier DatePicker sous forme de fenêtre pop up. Pour cela, vous pouvez soit :

- insérer un objet "Pop up date" de la bibliothèque d'objets préconfigurés,
- créer un sous-formulaire et lui associer le formulaire détaillé **DateButton**,  
Dans ces deux cas, vous pouvez gérer l'affichage et la saisie en associant une variable date à l'objet.
- créer un objet qui appelle la méthode composant **DatePicker Display Dialog**. Cette méthode retourne la date sélectionnée par l'utilisateur.

### Zone DateEntry

---

Une zone de type DateEntry facilite la saisie d'une date sous la forme définie dans les préférences système (par exemple JJ/MM/AA).

La zone se présente sous la forme d'une date de type associée à des boutons :



En exécution, les boutons situés à droite de la zone de saisie ne s'affichent que lorsque l'objet a le focus. L'utilisateur sélectionne individuellement chaque élément de la date (jour, mois ou année) via un clic ou la touche Tabulation et peut le faire défiler en utilisant le stepper numérique ou les touches fléchées du clavier. L'icône calendrier situé à droite permet de sélectionner une date par l'intermédiaire d'un pop up calendrier DatePicker.

Un objet DateEntry peut être utilisé sans programmation grâce au mécanisme fourni par la variable associée (cf. paragraphe "Calendrier DatePicker"). Toutefois, si vous souhaitez en personnaliser le fonctionnement, vous pouvez utiliser l'ensemble de méthodes composant qui est fourni. Ces méthodes sont les mêmes que celles de l'objet DatePicker.

### Utilisation en sous-formulaire

Vous pouvez insérer une zone DateEntry dans un formulaire de deux manières :

- en insérant un objet "Zone de saisie de date" depuis la bibliothèque d'objets préconfigurée de 4D
- en créant une zone de sous-formulaire et en lui associant le formulaire détaillé **DateEntry**.

## DatePicker APPLY DEFAULT VALUES

DatePicker APPLY DEFAULT VALUES ( *nomObjet* )

| Paramètre       | Type  |   | Description                 |
|-----------------|-------|---|-----------------------------|
| <i>nomObjet</i> | Texte | → | Nom d'objet sous-formulaire |

### Description

---

La commande **DatePicker APPLY DEFAULT VALUES** permet de réinitialiser tous les paramètres DatePicker à leur valeur par défaut pour l'objet de sous-formulaire *nomObjet*.

Ces valeurs par défaut peuvent être les réglages d'usine mais peuvent également avoir été modifiées via les commandes SET DEFAULT du composant.

L'action de la commande est immédiate : les valeurs par défaut de *nomObjet* sont instantanément modifiées. A noter que la variable associée à l'objet pourra également être modifiée afin de tenir compte des nouvelles valeurs. Par exemple, si les nouvelles valeurs par défaut fixent la date minimale au 01/01/2000 et que la variable associée à *nomObjet* était le 05/05/1995, sa valeur est automatiquement ramenée au 01/01/2000.

Les paramètres DatePicker incluent :

- les dates saisissables minimale et maximale
- le premier jour de la semaine
- les "jours de congés" hebdomadaires, annuels et spécifiques

### Exemple

---

Cet exemple réinitialise les paramètres de l'objet Date1 à leurs réglages par défaut :

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

## 🔧 DatePicker Display Dialog

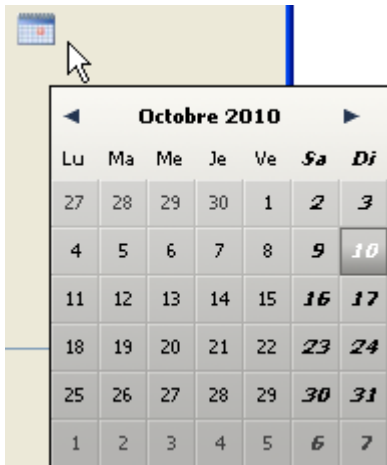
DatePicker Display Dialog {{ gauche ; haut {; dateDéfaut}}} -> Résultat

| Paramètre  | Type        |   | Description                               |
|------------|-------------|---|---|
| gauche     | Entier long | ➔ | Emplacement gauche de la fenêtre          |
| haut       | Entier long | ➔ | Emplacement haut de la fenêtre            |
| dateDéfaut | Date        | ➔ | Date à présélectionner dans le calendrier |
| Résultat   | Date        | ➡ | Date sélectionnée par l'utilisateur       |

### Description

---

La commande **DatePicker Display Dialog** ouvre un calendrier DatePicker dans une fenêtre pop up (une fenêtre de type pop up est automatiquement refermée lorsque l'utilisateur clique en-dehors de la fenêtre ou appuie sur la touche **Entrée** ou **Echap**).



Les deux paramètres optionnels *gauche* et *haut* permettent de définir l'emplacement de l'angle supérieur gauche de la fenêtre à ouvrir. Ces deux paramètres doivent être passés ensemble ; si un seul est passé, il est ignoré. Si ces paramètres sont omis, la fenêtre est ouverte à l'emplacement du clic.

Vous pouvez passer une date dans le paramètre *dateDéfaut* afin d'afficher le calendrier DatePicker avec une date présélectionnée.

**DatePicker Display Dialog** retourne la date sélectionnée par l'utilisateur dans le calendrier DatePicker. Si la fenêtre est refermée sans qu'une date ait été sélectionnée, la commande retourne une date vide (!00/00/00!), même si une date avait été passée dans le paramètre *dateDéfaut*.

### Exemple

---

Cet exemple affiche un calendrier DatePicker en cas de clic sur un bouton :

```
OBJECT GET COORDINATES(*;"MonBtnCalendrier";$x1;$y1;$x2;$y2)
$MaDateLocale:=DatePicker Display Dialog($x1;$y1)
If($MaDateLocale #!00/00/00!)
```

```
[Evenement]DateRV:=$MaDateLocale  
End if
```

## **DatePicker RESET DEFAULT VALUES**

DatePicker RESET DEFAULT VALUES

Ne requiert pas de paramètre

### **Description**

---

La commande **DatePicker RESET DEFAULT VALUES** réinitialise les paramètres du composant DatePicker à leurs "réglages d'usine". Après l'exécution de cette méthode composant :

- les dates saisissables minimale et maximale sont 00/00/00 (ce qui signifie qu'il n'y a pas de limites)
- le premier jour de la semaine est 2 (lundi)
- les "jours de congés" hebdomadaires sont le samedi et le dimanche
- aucun jour férié annuel ni spécifique n'est défini.

A noter que ce paramétrage est pris en compte uniquement pour les calendriers créés par la suite et n'est pas appliqué aux calendriers existants. Si vous souhaitez l'appliquer aux calendriers existants, vous devez utiliser la commande **DatePicker APPLY DEFAULT VALUES**.

## ⚙️ DatePicker SET DAYS OFF

DatePicker SET DAYS OFF ( *nomObjet* {; *typeJour* ; *ptrTabJoursCongés*} )

| Paramètre                | Type           | Description  |
|--------------------------|----------------|--|
| <i>nomObjet</i>          | Texte          | ➔ Nom d'objet sous-formulaire                                  |
| <i>typeJour</i>          | Entier<br>long | ➔ Types de jours fériés  |
| <i>ptrTabJoursCongés</i> | Pointeur       | ➔ Pointeur vers le tableau date ou booléen des jours de congés |

### Description

---

La commande **DatePicker SET DAYS OFF** permet de définir les "jours de congés" à faire figurer dans le calendrier DatePicker. Ces jours sont affichés en gras italique et restent sélectionnables par l'utilisateur.

Le paramètre *nomObjet* désigne l'instance de sous-formulaire à laquelle la commande doit être appliquée. Vous devez passer dans ce paramètre un nom d'objet sous-formulaire affiché dans le formulaire courant.

La méthode composant permet de définir des jours de congés récurrents hebdomadaires ou annuels ainsi que des jours "fériés" ponctuels. Vous précisez le type de jour de congé défini par la méthode via le paramètre *typeJour* :

- 0 = jours de congés répétés chaque semaine (par défaut, le samedi et le dimanche)
- 1 = jours de congés répétés d'année en année (tels que le 1er janvier ou le 25 décembre)
- 2 = jours fériés ponctuels, définis pour une seule année

Vous définissez les jours à l'aide d'un tableau et en passant un pointeur vers ce tableau comme paramètre *ptrTabJoursCongés*. Le type de tableau dépend de la valeur passée dans *typeJour* :

- si vous avez passé 0 dans *typeJour* (jours de congés hebdomadaires), vous devez passer dans *ptrTabJoursCongés* un pointeur vers un tableau booléen comportant 7 éléments. Chaque élément à Vrai désigne un jour de congé hebdomadaire.
- si vous avez passé 1 ou 2 dans *typeJour* (jours de congés annuels ou ponctuels), vous devez passer dans *ptrTabJoursCongés* un pointeur vers un tableau date. Dans ce tableau, chaque élément doit contenir une date valide, indiquant un jour de congé. Les dates doivent être exprimées dans le format par défaut correspondant à la langue du système. Si vous avez passé 1 dans *typeJour* (jours récurrents), l'année est ignorée, vous pouvez passer n'importe quelle valeur.

### Exemple 1

---

Désignation du vendredi comme jour de congé hebdomadaire (au lieu des samedi et dimanche par défaut) :

**ARRAY BOOLEAN**(\$tbJoursCongés;7)

//Par défaut, tous les éléments d'un tableau booléen sont à Faux, il n'est donc pas nécessaire d'ajouter du code d'initialisation

**\$tbJoursCongés**{Friday}:=**True**

**DatePicker SET DAYS OFF**("moncalendrier";0;->**\$tbJoursCongés**)

A calendar for September 2010. The days of the week are abbreviated as Lu, Ma, Me, Je, Ve, Sa, Di. Public holidays are indicated by a red 'X' over the date: 3 (Wednesday), 10 (Monday), 17 (Monday), and 24 (Monday). The calendar shows the following dates in each row: Row 1: 30, 31, 1, 2, 3, 4, 5; Row 2: 6, 7, 8, 9, 10, 11, 12; Row 3: 13, 14, 15, 16, 17, 18, 19; Row 4: 20, 21, 22, 23, 24, 25, 26; Row 5: 27, 28, 29, 30, 1, 2, 3; Row 6: 4, 5, 6, 7, 8, 9, 10.

| Septembre 2010 |    |    |    |    |    |    |
|----------------|----|----|----|----|----|----|
| Lu             | Ma | Me | Je | Ve | Sa | Di |
| 30             | 31 | 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 | 1  | 2  | 3  |
| 4              | 5  | 6  | 7  | 8  | 9  | 10 |

## Exemple 2

---

Désignation de jours fériés ponctuels :

**ARRAY DATE**(\$tdJoursUniques;0)

//L'année est prise en compte

**APPEND TO ARRAY**(\$tdJoursUniques;!15/02/2008!)

**APPEND TO ARRAY**(\$tdJoursUniques;!12/02/2009!)

**APPEND TO ARRAY**(\$tdJoursUniques;!17/02/2010!)

**DatePicker SET DAYS OFF**("moncalendrier";2;->**\$tdJoursUniques**)

## DatePicker SET DEFAULT 1ST DAY

DatePicker SET DEFAULT 1ST DAY ( numJour )

| Paramètre | Type        | Description                         |
|-----------|-------------|-------------------------------------|
| numJour   | Entier long | ➔ Numéro du premier jour à afficher |

### Description

---

La commande **DatePicker SET DEFAULT 1ST DAY** vous permet de désigner le premier jour de la semaine à afficher par défaut dans la partie gauche des calendriers DatePicker.

Passez dans le paramètre *numJour* une des constantes 4D suivantes du thème **Jours et mois** :

| Constante | Type        | Valeur |
|-----------|-------------|--------|
| Sunday    | Entier long | 1      |
| Monday    | Entier long | 2      |
| Tuesday   | Entier long | 3      |
| Wednesday | Entier long | 4      |
| Thursday  | Entier long | 5      |
| Friday    | Entier long | 6      |
| Saturday  | Entier long | 7      |

A noter que ce paramétrage est pris en compte uniquement pour les calendriers créés par la suite et n'est pas appliqué aux calendriers existants. Si vous souhaitez l'appliquer aux calendriers existants, vous devez utiliser la commande **DatePicker APPLY DEFAULT VALUES**.



## DatePicker SET DEFAULT DAYS OFF

DatePicker SET DEFAULT DAYS OFF ( *typeJour* ; *ptrTabJoursCongés* )

| Paramètre                | Type        | Description  |
|--------------------------|-------------|--|
| <i>typeJour</i>          | Entier long | → Types de jours fériés  |
| <i>ptrTabJoursCongés</i> | Pointeur    | → Pointeur vers le tableau date ou booléen des jours de congés |

### Description

---

La commande **DatePicker SET DEFAULT DAYS OFF** vous permet de définir les "jours de congés" à faire figurer dans la totalité des calendriers du composant DatePicker. Ces jours sont affichés en gras italique et restent sélectionnables par l'utilisateur.

A noter que ce paramétrage est pris en compte uniquement pour les calendriers créés par la suite et n'est pas appliqué aux calendriers existants. Si vous souhaitez l'appliquer aux calendriers existants, vous devez utiliser la commande **DatePicker APPLY DEFAULT VALUES**.

La méthode composant permet de définir des jours de congés récurrents hebdomadaires ou annuels ainsi que des jours "fériés" ponctuels. Vous précisez le type de jour de congé défini par la méthode via le paramètre *typeJour* :

- 0 = jours de congés répétés chaque semaine (par défaut, le samedi et le dimanche)
- 1 = jours de congés répétés d'année en année (tels que le 1er janvier ou le 25 décembre)
- 2 = jours fériés ponctuels, définis pour une seule année

Vous définissez les jours à l'aide d'un tableau et en passant un pointeur vers ce tableau comme paramètre *ptrTabJoursCongés*. Le type de tableau dépend de la valeur passée dans *typeJour* :

- si vous avez passé 0 dans *typeJour* (jours de congés hebdomadaires), vous devez passer dans *ptrTabJoursCongés* un pointeur vers un tableau booléen comportant 7 éléments. Chaque élément à Vrai désigne un jour de congé hebdomadaire.
- si vous avez passé 1 ou 2 dans *typeJour* (jours de congés annuels ou ponctuels), vous devez passer dans *ptrTabJoursCongés* un pointeur vers un tableau date. Dans ce tableau, chaque élément doit contenir une date valide, indiquant un jour de congé. Les dates doivent être exprimées dans le format par défaut correspondant à la langue du système. Si vous avez passé 1 dans *typeJour* (jours récurrents), l'année est ignorée, vous pouvez passer n'importe quelle valeur.

### Exemple

---

Désignation des jours fériés récurrents (exemple valide en France) :

**ARRAY DATE(\$tdJoursRépétés;0)**

//L'année est ignorée, nous utilisons 2000 par défaut

**APPEND TO ARRAY(\$tdJoursRépétés;!01/01/2000!)**

**APPEND TO ARRAY(\$tdJoursRépétés;!01/05/2000!)**

**APPEND TO ARRAY(\$tdJoursRépétés;!08/05/2000!)**

**APPEND TO ARRAY(\$tdJoursRépétés;!14/07/2000!)**

**APPEND TO ARRAY(\$tdJoursRépétés;!15/08/2000!)**

**APPEND TO ARRAY(\$tdJoursRépétés;!01/11/2000!)**

**APPEND TO ARRAY(\$tdJoursRépétés;!11/11/2000!)**

**APPEND TO ARRAY(\$tdJoursRépétés;!25/12/2000!)**

***DatePicker SET DEFAULT DAYS OFF(1;->\$tdJoursRépétés)***

## DatePicker SET DEFAULT MAX DATE

DatePicker SET DEFAULT MAX DATE ( dateMax )

| Paramètre | Type   | Description                           |
|-----------|--------|---------------------------------------|
| dateMax   | Date → | Limite supérieure de date saisissable |

### Description

---

La commande **DatePicker SET DEFAULT MAX DATE** vous permet de définir la date saisissable maximale pour la totalité des calendriers du composant DatePicker. A noter que ce paramétrage est pris en compte uniquement pour les calendriers créés par la suite et n'est pas appliqué aux calendriers existants. Si vous souhaitez l'appliquer aux calendriers existants, vous devez utiliser la commande **DatePicker APPLY DEFAULT VALUES**.

## DatePicker SET DEFAULT MIN DATE

DatePicker SET DEFAULT MIN DATE ( dateMin )

| Paramètre | Type |   | Description                           |
|-----------|------|---|---------------------------------------|
| dateMin   | Date | → | Limite inférieure de date saisissable |

### Description

---

La commande **DatePicker SET DEFAULT MIN DATE** vous permet de définir la date saisissable minimale pour la totalité des calendriers du composant DatePicker.

A noter que ce paramétrage est pris en compte uniquement pour les calendriers créés par la suite et n'est pas appliqué aux calendriers existants. Si vous souhaitez l'appliquer aux calendriers existants, vous devez utiliser la commande **DatePicker APPLY DEFAULT VALUES**.

### Exemple

---

Désignation de la date minimale au 1er janvier 2000 :

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

## DatePicker SET MAX DATE

DatePicker SET MAX DATE ( nomObjet ; dateMax )

| Paramètre | Type  |   | Description                           |
|-----------|-------|---|---------------------------------------|
| nomObjet  | Texte | → | Nom d'objet sous-formulaire           |
| dateMax   | Date  | → | Limite supérieure de date saisissable |

### Description

---

La commande **DatePicker SET MAX DATE** permet de définir la date saisissable maximale dans un calendrier DatePicker (les jours situés après cette date maximale apparaissent grisés dans le calendrier).

Le paramètre *nomObjet* désigne l'instance de sous-formulaire à laquelle la commande doit être appliquée. Vous devez passer dans ce paramètre un nom d'objet sous-formulaire affiché dans le formulaire courant.

La date *dateMax* doit être exprimée dans le format de saisie par défaut correspondant à la langue du système. Si vous passez une date vide (!00/00/00!), toutes les dates qui suivent la date courante sont saisissables.

Si la date saisissable maximale est antérieure à la date saisissable minimale (cf. **DatePicker SET MIN DATE**), aucune date ne sera saisissable.

### Exemple

---

Inactivation de toutes les dates après le 31 décembre 2009 dans l'objet nommé "DateRetour" :

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

## DatePicker SET MIN DATE

DatePicker SET MIN DATE ( nomObjet ; dateMin )

| Paramètre | Type  |   | Description                           |
|-----------|-------|---|---------------------------------------|
| nomObjet  | Texte | → | Nom d'objet sous-formulaire           |
| dateMin   | Date  | → | Limite inférieure de date saisissable |

### Description

---

La commande **DatePicker SET MIN DATE** permet de définir la date saisissable minimale dans un calendrier DatePicker (les jours situés avant cette date minimum apparaissent grisés dans le calendrier).

Le paramètre *nomObjet* désigne l'instance de sous-formulaire à laquelle la méthode composant doit être appliquée. Vous devez passer dans ce paramètre un nom d'objet sous-formulaire affiché dans le formulaire courant.

La date *dateMin* doit être exprimée dans le format de saisie par défaut correspondant à la langue du système. Si vous passez une date vide (!00/00/00!), toutes les dates qui précèdent la date courante sont saisissables.

Si la date saisissable minimale est postérieure à la date saisissable maximale (cf. **DatePicker SET MAX DATE**), aucune date ne sera saisissable.

### Exemple

---

Le formulaire courant contient deux calendriers DatePicker situés dans deux objets sous-formulaires nommés "DP1" et "DP2".

```
//Inactivation de toutes les dates avant le 1er janvier 2009 dans le premier calendrier  
DatePicker SET MIN DATE("DP1";!01/01/2009!)  
//Inactivation de toutes les dates avant le 1er mars 2009 dans le deuxième calendrier  
DatePicker SET MIN DATE("DP2";!01/03/2009!)
```

## 🔧 DatePicker SET WEEK FIRST DAY

DatePicker SET WEEK FIRST DAY ( nomObjet ; numJour )

| Paramètre | Type        |   | Description                       |
|-----------|-------------|---|-----------------------------------|
| nomObjet  | Texte       | → | Nom d'objet sous-formulaire       |
| numJour   | Entier long | → | Numéro du premier jour à afficher |

### Description

---

La commande **DatePicker SET WEEK FIRST DAY** permet de désigner le premier jour de la semaine à afficher dans la partie gauche d'un calendrier DatePicker. Par défaut, le premier jour est lundi.

Le paramètre *nomObjet* désigne l'instance de sous-formulaire à laquelle la commande doit être appliquée. Vous devez passer dans ce paramètre un nom d'objet sous-formulaire affiché dans le formulaire courant.

Passez dans le paramètre *numJour* une des constantes 4D suivantes du thème **Jours et mois** :

| Constante | Type        | Valeur |
|-----------|-------------|--------|
| Sunday    | Entier long | 1      |
| Monday    | Entier long | 2      |
| Tuesday   | Entier long | 3      |
| Wednesday | Entier long | 4      |
| Thursday  | Entier long | 5      |
| Friday    | Entier long | 6      |
| Saturday  | Entier long | 7      |

### Exemple 1

---

Paramétrage du premier jour au dimanche :

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



The screenshot shows a calendar for July 2010. The days of the week are labeled as Di, Lu, Ma, Me, Je, Ve, Sa. The dates are arranged in a grid. The first day of the week is Sunday (Di), which is the 27th of the month. The last day of the month is Saturday (Sa), which is the 31st. The dates are: 27, 28, 29, 30, 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.

## Exemple 2

---

Paramétrage du premier jour au jeudi :

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

| Juillet 2010 |    |           |           |    |    |    |
|--------------|----|-----------|-----------|----|----|----|
| Je           | Ve | <i>Sa</i> | <i>Di</i> | Lu | Ma | Me |
| 1            | 2  | <i>3</i>  | <i>4</i>  | 5  | 6  | 7  |
| 8            | 9  | <i>10</i> | <i>11</i> | 12 | 13 | 14 |
| 15           | 16 | <i>17</i> | <i>18</i> | 19 | 20 | 21 |
| 22           | 23 | <i>24</i> | <i>25</i> | 26 | 27 | 28 |
| 29           | 30 | <i>31</i> | <i>1</i>  | 2  | 3  | 4  |
| 5            | 6  | <i>7</i>  | <i>8</i>  | 9  | 10 | 11 |



# SearchPicker

 Présentation

 SearchPicker SET HELP TEXT

## ✚ Présentation

---

Le widget SearchPicker vous permet de créer facilement des zones standard de recherche, semblables à celles que l'on peut trouver dans les navigateurs ou les barres d'outils. L'apparence de la zone dépend de la plate-forme.

Windows :  MacOS : 

Le texte affiché par défaut dans la zone peut être contrôlé par programmation, à l'aide de la méthode composant **SearchPicker SET HELP TEXT**.

## Fonctionnement

---

Hormis son apparence, une zone de recherche SearchPicker se caractérise par les éléments suivants : un texte grisé, une zone de saisie et une icône d'effacement.

- La zone de saisie permet à l'utilisateur de saisir la valeur à rechercher. Cette valeur est automatiquement et dynamiquement affectée à la variable que vous avez associée à la zone dans la Liste des propriétés (propriété "Nom de la variable"). Vous utilisez cette variable pour fournir la valeur recherchée à la méthode de recherche.
- Le texte grisé est une aide indiquant à l'utilisateur le ou les champ(s) où la recherche sera effectuée. Il disparaît dès que la zone a le focus. Ce texte peut être défini via la méthode composant **SearchPicker SET HELP TEXT**.
- Le bouton d'effacement permet d'effacer le contenu de la zone. Son fonctionnement est automatique.

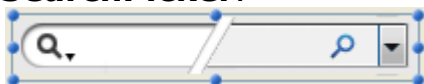
En exécution, vous pouvez lancer votre méthode de recherche via un clic sur un bouton du formulaire ou dans un événement formulaire. La zone génère notamment les événements On Data Change et On Losing Focus. Vous pouvez proposer une recherche dynamique réévaluée à chaque caractère saisi par l'utilisateur en appelant la méthode de recherche dans l'événement On Data Change.

## Création

---

Vous pouvez insérer une zone SearchPicker dans un formulaire de deux manières :

- en insérant un objet "Zone de recherche" depuis la bibliothèque d'objets préconfigurés de 4D (thème "Zones de saisie").
- en créant une zone de sous-formulaire et en lui affectant le formulaire détaillé **SearchPicker**.



Définissez ensuite le nom de la variable associée au sous-formulaire (propriété "Nom de la variable" dans la Liste des propriétés). A l'exécution du formulaire, cette

variable contiendra automatiquement la valeur recherchée par l'utilisateur. Il vous suffit alors de passer cette valeur à votre méthode de recherche personnalisée.

## SearchPicker SET HELP TEXT

SearchPicker SET HELP TEXT ( nomObjet ; texteAide )

| Paramètre | Type  |   | Description                 |
|-----------|-------|---|-----------------------------|
| nomObjet  | Texte | → | Nom d'objet sous-formulaire |
| texteAide | Texte | → | Texte à afficher            |

### Description

---

La commande **SearchPicker SET HELP TEXT** vous permet d'afficher un texte gris non saisissable en arrière-plan de la zone de recherche désignée par *nomObjet*. Ce texte disparaît lorsque l'utilisateur clique dans la zone.

### Exemple



















---

Affichage du mot "Country" dans la zone, indiquant que la recherche portera sur cette variable :

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



# TimePicker

-  TimePicker, TimeEntry et TimeDisplay
-  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, TimeEntry et TimeDisplay

---

Le widget TimePicker propose des objets simples d'emploi que vous pouvez utiliser pour valoriser des champs nécessitant la saisie d'heures ou afficher des heures. Il est utilisable sous différentes formes :

- Pop up menu simple ou double :



- Zone de saisie d'heures au format "hh:mm:ss" associées à un stepper numérique permettant d'augmenter ou de réduire les valeurs d'heures, minutes ou secondes :



- Horloge (*TimeDisplay*) ou horloge digitale (*TimeDisplayLCD*) :



En outre, chaque type de TimePicker peut afficher l'heure sur 12 heures (AM-PM) ou 24 heures.

Un objet TimePicker peut être utilisé sans programmation grâce aux mécanismes fournis par la variable associée. Toutefois, vous pouvez personnaliser le fonctionnement des objets TimePicker à l'aide des méthodes composant qui sont fournies.

## Création et configuration

---

Vous pouvez insérer une zone TimePicker dans un formulaire de deux manières :

- en déposant un objet "TimePicker", "Zone de saisie d'heure" ou "Zone d'affichage d'heure" depuis la bibliothèque d'objets préconfigurés de 4D
- en créant une zone de sous-formulaire et en lui affectant le formulaire détaillé **TimePicker**, **TimeEntry**, **TimeDisplay** ou **TimeDisplayLCD** selon votre choix.

Définissez ensuite le nom de la variable associée au sous-formulaire (propriété "Nom de la variable" dans la Liste des propriétés). A l'exécution du formulaire, cette variable contiendra automatiquement l'heure définie par l'utilisateur. A l'inverse, si vous modifiez par programmation la valeur de cette variable, elle sera automatiquement représentée dans le sous-formulaire. Vous pouvez aussi ne pas nommer de variable afin de bénéficier du mécanisme des variables dynamiques.

## A propos des horloges (nouveau v14)

---

Les widgets "horloge" sont dessinés en SVG, ils bénéficient donc d'un tracé vectoriel autorisant les déformations en mode Application (en mode Développement, leur taille est fixe) :



A noter que :

- Dans l'horloge standard, la trotteuse peut être affichée ou masquée à l'aide de la méthode **TimePicker DISPLAY SECOND HAND**.
- L'horloge standard passe automatiquement en "mode jour" ou "mode nuit" en fonction de l'heure :



Les plages horaires sont 8:00:00 -> 19:59:59 = Jour, 20:00 -> 07:59:59 = Nuit.

- Le widget "horloge digitale" est transparent et sans arrière-plan, il peut donc être placé au-dessus d'objets colorés afin de varier les rendus :



Vous disposez de plusieurs options d'affichage pour ce widget, accessibles via les méthodes composant débutant par "TimePicker LCD".

**Note:** Les développeurs peuvent remplacer le dessin de l'horloge standard par leurs propres créations en remplaçant le fichier "clock.svg" situé au premier niveau du dossier "Resources".

## Afficher l'heure courante ou une heure statique

Les horloges peuvent soit afficher dynamiquement l'heure courante soit représenter une heure statique.

- Pour afficher l'heure courante, associez une variable **Numérique** à l'objet sous-formulaire du widget (fonctionnement par défaut). Dans ce cas, le widget affiche automatiquement l'heure courante et fonctionne comme une pendule. Vous pouvez appliquer un décalage à l'heure affichée : la valeur de la variable numérique indique le décalage en secondes. Par exemple, 3600 = avancer la pendule d'une heure, -1800 = reculer la pendule de 30 mn, etc.
- Pour afficher une heure statique, associez une variable **Heure** à l'objet sous-formulaire du widget (via la commande **C\_TIME** ou la Liste des propriétés). La valeur de la variable est représentée par l'horloge.

Par exemple, on souhaite représenter 10:10:30 :

```
C_TIME(mavar) //mavar est le nom de la variable du widget  
mavar:=?10:10:30?
```





## TimePicker APPLY DEFAULT VALUES

TimePicker APPLY DEFAULT VALUES ( *nomObjet* )

| Paramètre       | Type  |   | Description                 |
|-----------------|-------|---|-----------------------------|
| <i>nomObjet</i> | Texte | → | Nom d'objet sous-formulaire |

### Description

---

La commande **TimePicker APPLY DEFAULT VALUES** permet de réinitialiser tous les paramètres TimePicker à leur valeur par défaut courante pour l'objet de sous-formulaire *nomObjet*.

Ces valeurs par défaut peuvent être les réglages d'usine mais peuvent également avoir été modifiées via les commandes SET DEFAULT du composant.

L'action de la commande est immédiate : les valeurs par défaut de *nomObjet* sont instantanément modifiées. A noter que la variable associée à l'objet pourra également être modifiée afin de tenir compte des nouvelles valeurs. Par exemple, si les nouvelles valeurs par défaut fixent l'heure minimale à 07:00:00 et que la variable associée à *nomObjet* était 06:00:00, sa valeur est automatiquement ramenée à 07:00:00.

Les paramètres TimePicker incluent :

- les heures saisissables minimales et maximales,
- les libellés AM et PM ,
- les intervalles de minutes.

## TimePicker DISPLAY SECOND HAND

TimePicker DISPLAY SECOND HAND ( nomObjet ; trotteuse )

| Paramètre | Type    | Description  |
|-----------|---------|--|
| nomObjet  | Texte   | → Nom d'objet sous-formulaire  |
| trotteuse | Booléen | → Vrai (défaut) = afficher la trotteuse, Faux = masquer la trotteuse |

### **Description**

---

La commande **TimePicker DISPLAY SECOND HAND** permet d'afficher ou de masquer la trotteuse dans l'objet de sous-formulaire *nomObjet* (widget horloge uniquement).

Par défaut, la trotteuse est affichée. Pour la masquer, appelez cette commande et passez **Faux** dans le paramètre *trotteuse*.

## ⚙️ TimePicker LCD DISPLAY AMPM

TimePicker LCD DISPLAY AMPM ( nomObjet ; amPm )

| Paramètre | Type    | Description  |
|-----------|---------|--|
| nomObjet  | Texte   | → Nom d'objet sous-formulaire                          |
| amPm      | Booléen | → Vrai = afficher le symbole, Faux = ne pas l'afficher |

### Description

---

La commande **TimePicker LCD DISPLAY AMPM** permet d'afficher ou de masquer les lettres AM/PM placées à droite de l'objet de sous-formulaire *nomObjet* (horloge digitale uniquement).

Ces lettres sont utiles pour distinguer le matin de l'après midi lorsque l'horloge est en mode 12 heures (cf. **TimePicker LCD SET MODE**).

Par défaut, les lettres sont affichées. Passez **Faux** dans *amPm* pour les masquer.

### Exemple

---

On souhaite masquer les lettres AP/PM :

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



03:20:04

## TimePicker LCD DISPLAY SECONDS

TimePicker LCD DISPLAY SECONDS ( nomObjet ; secondes )

| Paramètre | Type    | Description  |
|-----------|---------|--|
| nomObjet  | Texte   | → Nom d'objet sous-formulaire                              |
| secondes  | Booléen | → Vrai = afficher les secondes, Faux = ne pas les afficher |

### **Description**

---

La commande **TimePicker LCD DISPLAY SECONDS** permet d'afficher ou de masquer la partie des secondes de l'objet de sous-formulaire *nomObjet* (horloge digitale uniquement).

Par défaut, les secondes sont affichées. Passez **Faux** dans *secondes* pour les masquer.

## ⚙ TimePicker LCD SET COLOR

TimePicker LCD SET COLOR ( nomObjet ; coul {; coulV ; coulB} )

| Paramètre | Type        | Description  |
|-----------|-------------|--|
| nomObjet  | Texte       | → Nom d'objet sous-formulaire  |
| coul      | Entier long | → Valeur de la couleur RVB (4 octets) ou Valeur de la composante rouge (0..255) si les autres paramètres sont passés |
| coulV     | Entier long | → Valeur de la composante verte (0..255)   |
| coulB     | Entier long | → Valeur de la composante bleue (0..255)   |

### Description

---

La commande **TimePicker LCD SET COLOR** vous permet de définir les couleurs des chiffres de l'objet de sous-formulaire *nomObjet* (horloge digitale uniquement).

Cette commande admet deux syntaxes :

- Si vous passez uniquement le paramètre *coul*, vous devez passer un entier long de 4 octets dont le format (0x00RRGGBB) est décrit ci-dessous (les octets sont numérotés de 0 à 3 de la droite vers la gauche) :

| Octet | Description |
|-------|-------------|
|-------|-------------|

|   |   |
|---|---|
| 3 | Doit être zéro pour une couleur RVB absolue |
| 2 | Composante rouge de la couleur (0..255)     |
| 1 | Composante verte de la couleur (0..255)     |
| 0 | Composante bleue de la couleur (0..255)     |

- Vous pouvez également passer trois paramètres *coul*, *coulV* et *coulB*. Dans ce cas, chaque paramètre doit être compris entre 0 et 255 et représente une composante de la couleur RVB.

### Exemple

---

Passage des chiffres de l'horloge en rouge :

```
TimePicker LCD SET COLOR("Subform1";0x00FF0000)  
// peut aussi s'écrire TimePicker LCD SET COLOR ("Subform1";255;0;0)
```



02:58:28 PM

## ⚙️ TimePicker LCD SET MODE

TimePicker LCD SET MODE ( nomObjet ; mode )

| Paramètre | Type        | Description  |
|-----------|-------------|--|
| nomObjet  | Texte       | → Nom d'objet sous-formulaire  |
| mode      | Entier long | → 12 = afficher l'heure sur 12 heures, 24 = afficher l'heure sur 24 heures |

### Description

---

La commande **TimePicker LCD SET MODE** permet de configurer l'affichage sur 12 heures ou 24 heures de l'objet de sous-formulaire *nomObjet* (horloge digitale uniquement).

Par défaut, l'objet est affiché sur 12 heures. Si vous souhaitez passer en mode "24 heures", passez la valeur 24 dans *mode*. Dans ce cas, il est généralement judicieux de masquer les lettres AM/PM (cf. commande **TimePicker LCD DISPLAY AMPM**).

### Exemple

---

On souhaite passer en mode 24 heures et masquer les lettres AM/PM :

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



## **TimePicker RESET DEFAULT VALUES**

TimePicker RESET DEFAULT VALUES

Ne requiert pas de paramètre

### **Description**

---

La commande **TimePicker RESET DEFAULT VALUES** réinitialise les paramètres du composant TimePicker à leurs "réglages d'usine". Après l'exécution de cette méthode composant :

- l'heure saisissable minimale est 08:00:00
- l'heure saisissable maximale est 20:00:00
- les libellés AM et PM sont ceux définis dans le système
- l'intervalle de minutes est 00:15:00

A noter que ce paramétrage est pris en compte uniquement pour les objets TimePicker créés par la suite et n'est pas appliqué aux objets existants. Si vous souhaitez l'appliquer aux objets existants, vous devez appeler la commande **TimePicker APPLY DEFAULT VALUES**.

## TimePicker SET DEFAULT LABEL AM

TimePicker SET DEFAULT LABEL AM ( libellé )

| Paramètre | Type  |   | Description                |
|-----------|-------|---|----------------------------|
| libellé   | Texte | → | Libellé à utiliser pour AM |

### Description

---

La commande **TimePicker SET DEFAULT LABEL AM** permet de modifier le libellé "AM" par défaut pour tous les objets TimePicker affichant le format AM/PM.

Ce paramétrage est pris en compte uniquement pour les objets créés par la suite et n'est pas appliqué aux objets existants. Si vous souhaitez l'appliquer aux objets existants, vous devez utiliser la commande **TimePicker APPLY DEFAULT VALUES**.



## TimePicker SET DEFAULT LABEL PM

TimePicker SET DEFAULT LABEL PM ( libellé )

| Paramètre | Type  |   | Description                |
|-----------|-------|---|----------------------------|
| libellé   | Texte | → | Libellé à utiliser pour PM |

### Description

---

La commande **TimePicker SET DEFAULT LABEL PM** permet de modifier le libellé "PM" par défaut pour tous les objets TimePicker affichant le format AM/PM.

Ce paramétrage est pris en compte uniquement pour les objets créés par la suite et n'est pas appliqué aux objets existants. Si vous souhaitez l'appliquer aux objets existants, vous devez utiliser la commande **TimePicker APPLY DEFAULT VALUES**.

## TimePicker SET DEFAULT MAX TIME

TimePicker SET DEFAULT MAX TIME ( heureMax )

| Paramètre | Type    | Description                           |
|-----------|---------|---------------------------------------|
| heureMax  | Heure → | Limite supérieure d'heure saisissable |

### Description

---

La commande **TimePicker SET DEFAULT MAX TIME** permet de définir l'heure saisissable maximale qui sera acceptée par défaut par tous les objets TimePicker. Ce paramétrage est pris en compte uniquement pour les objets créés par la suite et n'est pas appliqué aux objets existants. Si vous souhaitez l'appliquer aux objets existants, vous devez utiliser la commande **TimePicker APPLY DEFAULT VALUES**.

## **TimePicker SET DEFAULT MIN TIME**

TimePicker SET DEFAULT MIN TIME ( heureMin )

| <b>Paramètre</b> | <b>Type</b> | <b>Description</b>                    |
|------------------|-------------|---------------------------------------|
| heureMin         | Heure →     | Limite inférieure d'heure saisissable |

### **Description**

---

La commande **TimePicker SET DEFAULT MIN TIME** permet de définir l'heure saisissable minimale qui sera acceptée par défaut par tous les objets TimePicker. Ce paramétrage est pris en compte uniquement pour les objets créés par la suite et n'est pas appliqué aux objets existants. Si vous souhaitez l'appliquer aux objets existants, vous devez utiliser la commande **TimePicker APPLY DEFAULT VALUES**.

## TimePicker SET DEFAULT STEP

TimePicker SET DEFAULT STEP ( intervalle )

| Paramètre  | Type    | Description                            |
|------------|---------|--|
| intervalle | Heure → | Intervalle entre deux valeurs d'heures |

### Description

---

La commande **TimePicker SET DEFAULT STEP** permet de définir par défaut l'intervalle entre les valeurs d'heures pour tous les objets TimePicker.

Ce paramétrage est pris en compte uniquement pour les objets créés par la suite et n'est pas appliqué aux objets existants. Si vous souhaitez l'appliquer aux objets existants, vous devez utiliser la commande **TimePicker APPLY DEFAULT VALUES**.

## TimePicker SET LABEL AM

TimePicker SET LABEL AM ( nomObjet ; libellé )

| Paramètre | Type  |   | Description                 |
|-----------|-------|---|-----------------------------|
| nomObjet  | Texte | → | Nom d'objet sous-formulaire |
| libellé   | Texte | → | Libellé à utiliser pour AM  |

### Description

---

La commande **TimePicker SET LABEL AM** permet de modifier le libellé "AM" dans les objets TimePicker affichant le format AM/PM. La commande s'applique à l'objet désigné par *nomObjet*. Par défaut, les libellés système am/pm sont utilisés.

### Exemple

---

Utilisation par défaut du libellé "du matin" au lieu du libellé système pour AM :

```
TimePicker SET LABEL AM("horloge";"du matin")
```

## TimePicker SET LABEL PM

TimePicker SET LABEL PM ( nomObjet ; libellé )

| Paramètre | Type  |   | Description                 |
|-----------|-------|---|-----------------------------|
| nomObjet  | Texte | → | Nom d'objet sous-formulaire |
| libellé   | Texte | → | Libellé à utiliser pour PM  |

### Description

---

La commande **TimePicker SET LABEL PM** permet de modifier le libellé "PM" dans les objets TimePicker affichant le format AM/PM. La commande s'applique à l'objet désigné par *nomObjet*. Par défaut, les libellés système am/pm sont utilisés.

### Exemple

---

Utilisation par défaut du libellé "du soir" au lieu du libellé système pour PM :

```
TimePicker SET LABEL PM("horloge";"du soir")
```

## **TimePicker SET MAX TIME**

TimePicker SET MAX TIME ( nomObjet ; heureMax )

| <b>Paramètre</b> | <b>Type</b> |   | <b>Description</b>                    |
|------------------|-------------|---|---------------------------------------|
| nomObjet         | Texte       | → | Nom d'objet sous-formulaire           |
| heureMax         | Heure       | → | Limite supérieure d'heure saisissable |

### **Description**

---

La commande **TimePicker SET MAX TIME** permet de définir l'heure saisissable maximale qui sera acceptée par l'objet désigné par *nomObjet*. Si une valeur d'heure supérieure est saisie, elle sera rejetée.

## **TimePicker SET MIN TIME**

TimePicker SET MIN TIME ( nomObjet ; heureMin )

| <b>Paramètre</b> | <b>Type</b> |   | <b>Description</b>                    |
|------------------|-------------|---|---------------------------------------|
| nomObjet         | Texte       | → | Nom d'objet sous-formulaire           |
| heureMin         | Heure       | → | Limite inférieure d'heure saisissable |

### **Description**

---

La commande **TimePicker SET MIN TIME** permet de définir l'heure saisissable minimale qui sera acceptée par l'objet désigné par *nomObjet*. Si une valeur d'heure inférieure est saisie, elle sera rejetée.



## ⚙ TimePicker SET STEP

TimePicker SET STEP ( nomObjet ; intervalle )

| Paramètre  | Type    | Description                            |
|------------|---------|--|
| nomObjet   | Texte → | Nom d'objet sous-formulaire            |
| intervalle | Heure → | Intervalle entre deux valeurs d'heures |

### Description

---

La commande **TimePicker SET STEP** permet de définir l'intervalle entre les valeurs d'heures disponibles pour l'objet désigné par *nomObjet*. Ce paramétrage s'applique aux TimePickers affichés sous forme de pop up menus uniquement.

La valeur d'*intervalle* doit être comprise entre 1 mn et 1 heure et doit pouvoir être représentée sous forme de divisions entières de 60 mn. En pratique, seules les valeurs de 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 et 60 mn sont possibles. Toute valeur différente sera automatiquement arrondie de manière à correspondre à ce principe.

### Exemple

---

Paramétrage du TimePicker en pop up menu nommé "heure1", limitation des heures saisissables de 8h30 à 16h30 et intervalles de 10 mn :

```
TimePicker SET MIN TIME("heure1";?08:30:00?)  
TimePicker SET MAX TIME("heure1";?16:30:00?)  
TimePicker SET STEP("heure1";?00:10:00?)
```

