

4D Widgets (ウィジェット)

 4Dウィジェットコンポーネント

 DatePicker

 SearchPicker

 TimePicker

 コマンドリスト (文字順)

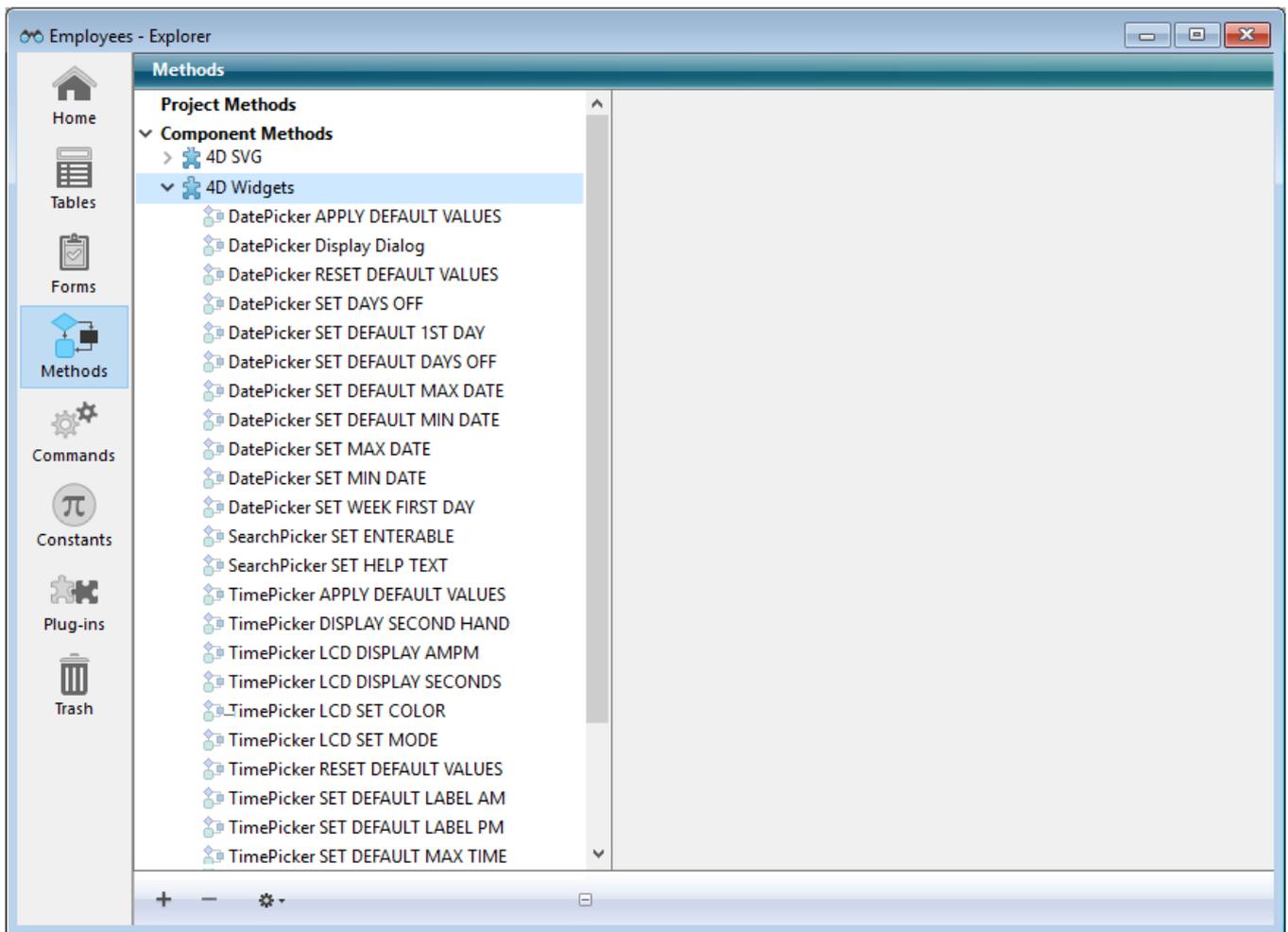
4Dウィジェットコンポーネント

4Dウィジェットはコンポーネントとして提供される機能で、フォームエディタあるいは定義済みのオブジェクトライブラリからアクセスできます。これらのウィジェットは複合オブジェクトで、標準機能へのアクセスを提供する定義済みの特性を持っていて、とても簡単に実装できます。

3つのウィジェットが利用できます:

- SearchPicker: 標準のAPIアランスを持った検索エリア
- DatePicker: 日付セレクト
- TimePicker: 時間セレクト

これらのウィジェットはプログラムなしで使用することができますし、またはプログラムで動作をカスタマイズすることもできます。フォームにウィジェットを配置し、デフォルトのプロパティをそのまま使用できます。動作をカスタマイズし、設定や制御を変更したい場合は、"コンポーネントメソッド"と呼ばれるメソッドを使用できます。コンポーネントメソッドはエクスプローラの**メソッド**ページ内でコンポーネントごとにリスト表示されます:



このドキュメントではそれぞれのウィジェットごとにコンポーネントメソッドのシンタックスを説明します。

ウィジェットの追加

ウィジェットエリアをフォームに挿入するには二通りの方法があります：

- オブジェクトライブラリを使用する方法
- サブフォームを使用する方法

オブジェクトライブラリを使用する方法

ウィジェットをオブジェクトライブラリを使用して挿入するには、以下の手順に従って下さい：

1. **デザインメニュー**から**オブジェクトライブラリ**コマンドをを選択します。
オブジェクトライブラリウィンドウが開きます。
2. カテゴリメニューの中の**ウィジェット**を選択して下さい。
使用可能なウィジェットが全て表示されます：

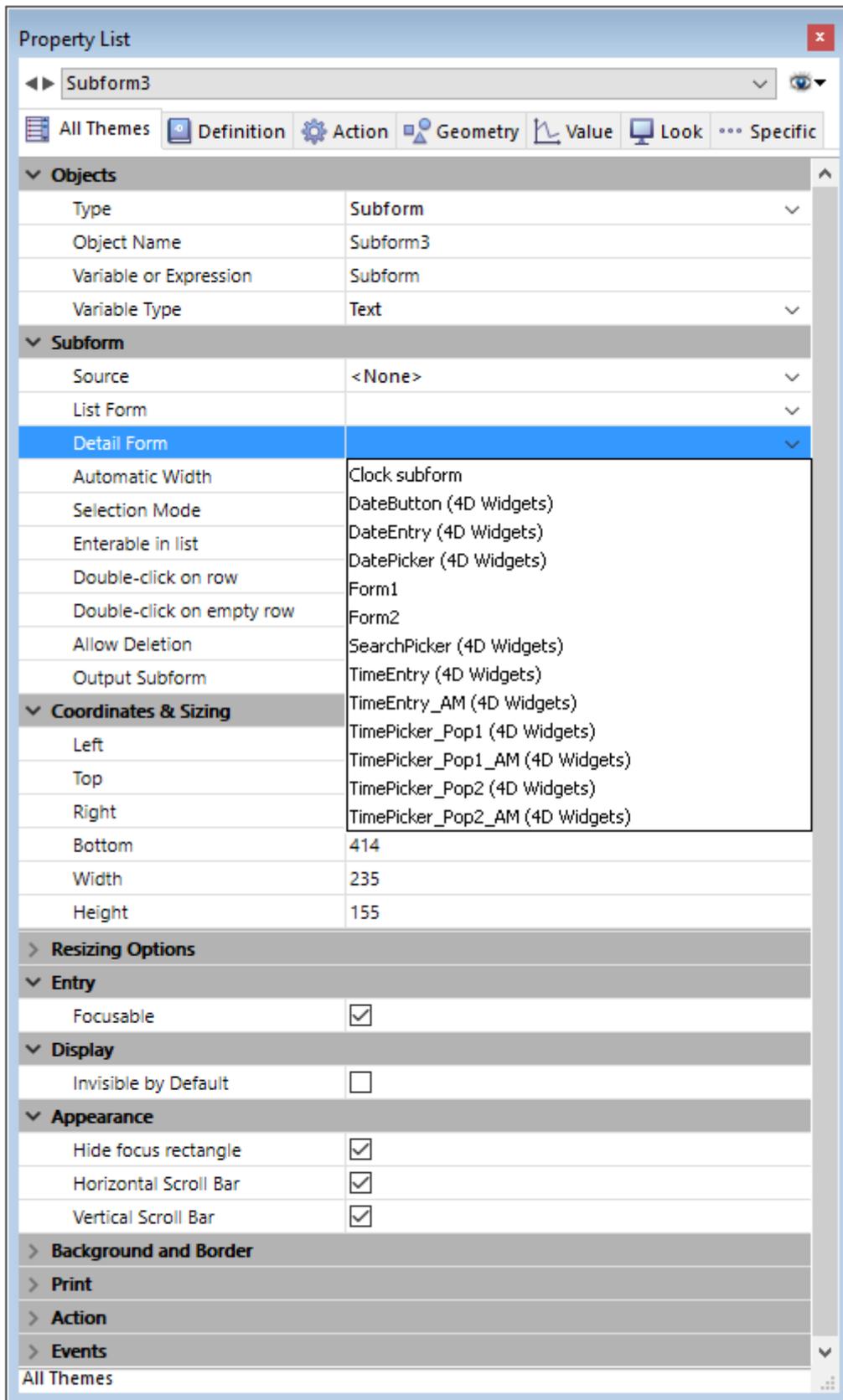


3. 使用したいウィジェットをドラッグ&ドロップでフォームに挿入して下さい。
その後プロパティやオブジェクトメソッドを通じて設定を変更する事ができます。

サブフォームを使用する方法

サブフォームタイプオブジェクトを使用してウィジェットを作成するには、以下の手順に従って下さい:

1. フォームエディター内に、サブフォームオブジェクトを追加して下さい。
これについての詳細は4D *Design Reference* マニュアル内にあります。
2. プロパティリスト内にて、"詳細フォーム"メニューをクリックして使用可能なフォームの一覧を表示します。ウィジェットはこのリスト内にあります。



3. 挿入したいウィジェットを選択して下さい。

その後プロパティやオブジェクトメソッドを通じて設定を変更することができます。

2ページ目以降においてウィジェットを初期化する

ウィジェットは4D サブフォームに基づいており、そのためサブフォームの動作原理に従います。これらの初期化には特に注意する必要があります。

最適化の目的のため、サブフォームオブジェクトは4Dによって、サブフォームが表示されたときにのみインスタンス化されます。これはつまり**EXECUTE METHOD IN SUBFORM** を使用してのメソッドの呼び出しや、サブフォーム内の動的なオブジェクトへのアクセスは、カレントページにおいてサブフォームが表示されるまでは不可能ということです。

ウィジェットをフォームの1ページ目以外に置いた場合、On Load フォームイベント内において初期化メソッド(**TimePicker SET STEP** など)を呼び出すことはできません。実行時にはウィジェットはまだインスタンス化されていないからです。

この場合を管理するために、ウィジェットエリアはロードされて使用準備が整ったことを示すための特定のイベント(値は-1)を返します。このイベントはウィジェット自身のオブジェクトメソッドレベルにおいて、On Data Change、On Load などと同じようにテストされなければなりません。そのため、ウィジェットがフォームの1ページ目以外に置かれている場合、以下のように書く必要があります:

```
If(Form event=-1) // ウィジェットがインスタンス化され初期化可能
    TimePicker SET STEP("myTimePicker";? 00:10:00?) // 例
End if
```

以下のように書いた場合は正しくありません:

```
If(Form event=On Load) // これではウィジェットが1ページ目にある場合にのみ有効
    TimePicker SET STEP("myTimePicker";? 00:10:00?)
End if
```

DatePicker

-  DatePickerと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とDateEntry

DatePickerウィジェットはフォームに日付を入力、表示するための直感的なグラフィックインタフェースオブジェクトを提供します。ウィジェットは2つの形式で提供されます:

- **DatePicker カレンダー:** このオブジェクトはサブフォーム中、あるいはボタンクリックに反応して表示されるプルダウンカレンダーとしても利用できます。
- **DateEntry エリア:** コントロールボタンが割り当てられた日付エリア。このオブジェクトはサブフォームの中でのみ利用できます。

DatePickerカレンダー

DatePickerカレンダーは日付を月ごとに表形式で表示するエリアです。実行中、ユーザーは矢印ボタンをクリックすることでカレンダーの月を前後に移動することができます。キーボードの矢印キーを使用することもできます。

April 2010						
Mo	Tu	We	Th	Fr	Sa	Su
29	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

これがサブフォームに挿入されると、DatePickerオブジェクトは**バインド変数**により提供されるメカニズムのおかげで、プログラミングを行うことなく利用することができます。つまり、サブフォームオブジェクトにバインドされた変数 (プロパティリストの"変数名"プロパティ) の値を使用して表示と入力を管理することができます。

フォームが実行されると、この日付変数にはユーザーが選択した日付が自動で割り当てられます。逆に、プログラムで日付変数の値を変更すると、それがサブフォームにも自動で反映されます。

DatePickerの動作をカスタマイズしたい場合や、ポップアップメニューとして表示したい場合、提供される一連のコンポーネントメソッドを使用します。

サブフォームで使用する

DatePickerカレンダーは2つの方法でフォームに挿入できます:

- 4Dの定義済みオブジェクトライブラリから"Date Picker"オブジェクトを挿入する。
- サブフォームエリアを作成し、**DatePicker**の詳細フォームを割り当てる。

バインド変数のメカニズムを使用すれば、プログラミングなしでこのエリアを管理できます (前述)。

ポップアップで使用する

DatePickerカレンダーをポップアップウィンドウで使用できます。以下のいずれかの方法を使用します:

- 定義済みのオブジェクトライブラリから"Pop up date"を挿入する。
- サブフォームエリアを作成し、**DateButton**詳細フォームを割り当てる。
これらの場合、オブジェクトに日付変数を割り当てることで表示と入力を管理できます。
- **DatePicker Display Dialog**コンポーネントメソッドを呼び出すオブジェクトを作成する。このメソッドからはユーザーが選択した日付が返されます。

DateEntryエリア

DateEntryタイプのエリアはシステム環境設定で定義された形式 (例: YY/MM/DD) での日付入力を容易にします。

このエリアはボタンが割り当てられた日付型として表示されます:



実行中、入力エリアの右にあるボタンはオブジェクトにフォーカスがあるときだけ表示されます。ユーザーはクリックやタブキーを使用して日付の各要素 (年、月、日) を個々に選択でき、数値ステッパーやキーボードの矢印キーを使用してスクロールできます。右側にあるカレンダーアイコンを使用すれば、DatePickerポップアップカレンダーから日付を選択することもできます。バインド変数が提供するメカニズムのおかげで、DateEntryオブジェクトはプログラミングなしで利用できます ("DatePickerカレンダー"の段落参照)。動作をカスタマイズしたい場合は、提供される一連のコンポーネントメソッドを使用できます。これらのメソッドはDatePickerオブジェクトと同じものです。

サブフォームで使用する

DateEntry エリアは2つの方法でフォームに挿入できます:

- 4Dの定義済みオブジェクトライブラリから"DateEntry area"オブジェクトを挿入する。
- サブフォームエリアを作成し、**DateEntry** の詳細 フォームを割り当てる。

🔧 DatePicker APPLY DEFAULT VALUES

DatePicker APPLY DEFAULT VALUES (*objectName*)

引数	型		説明
<i>objectName</i>	テキスト	→	サブフォームオブジェクトの名前

説明

DatePicker APPLY DEFAULT VALUESコマンドを使用して、*objectName*サブフォームオブジェクトのすべてのDatePickerパラメタを、デフォルト値に再設定できます。

これらのデフォルト値は初期値またはコンポーネントのSET DEFAULTコマンド で設定された値です。

このコマンドは即座に効果を及ぼします。*objectName*のデフォルト値は即座に更新されます。オブジェクトに割り当てられた変数は、新しい値を有効にするために更新されるかもしれないことに留意してください。例えば新しい最小日付のデフォルト値が2000/01/01に設定され、*objectName*に割り当てられた変数が1995/05/05のとき、その値は自動で2000/01/01に変更されます。

DatePickerのパラメタには以下があります：

- 入力可能な最小および最大日付
- 週の最初の曜日
- 毎週毎年のお休みおよび特別な休日

例題

この例題ではDate1オブジェクトのパラメタをデフォルト設定に設定します：

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

🌀 DatePicker Display Dialog

DatePicker Display Dialog {{ left ; top {; defaultDate}} } -> 戻り値

引数	型	説明
left	倍長整数	→ ウィンドウの左位置
top	倍長整数	→ ウィンドウの上位置
defaultDate	日付	→ ダイアログ中デフォルトで選択する日付
戻り値	日付	🔄 ユーザが選択した日付

説明

DatePicker Display Dialog コマンドはDatePickerカレンダーをポップアップウィンドウに表示します (ポップアップタイプのウィンドウはユーザがウィンドウの外をクリックするか、**Enter**または**Esc**キーが押されると自動で閉じられます)。



2つのオプションの引数`left`と`top`を使用して、ウィンドウを開く左上の座標を指定できます。これら2つの引数はペアで渡さなければなりません。一つしか渡されない場合、それは無視されます。これらの引数が省略されると、ウィンドウはクリックされた場所で開かれます。

第三引数 `defaultDate` に日付を渡すと、DatePickerダイアログが表示される際にその日付がデフォルトで選択されます。

DatePicker Display Dialog は、ユーザがDatePickerカレンダー内で選択した日付を返します。日付が選択されずにウィンドウが閉じられたとき、コマンドは空の日付 (!00/00/00!) を返します (たとえ`defaultDate`が指定されていた場合でも)。

例題

この例題はボタンがクリックされるとDatePickerカレンダーを表示します:

```
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

このコマンドは引数を必要としません

説明

DatePicker RESET DEFAULT VALUESコマンドはDatePickerコンポーネントのデフォルト値を初期値にリセットします。このコンポーネントメソッド実行後は以下のように設定されます:

- 最小および最大入力可能日付は00/00/00 (制限なし)
- 最初の曜日は2 (月曜)
- 毎週の休日は土曜と日曜
- 毎年の休日と特別な休日はなし

このパラメタは後で作成されるカレンダーに適用され、既存のカレンダーには影響を与えない点に留意してください。既存のカレンダーに適用したい場合は**DatePicker APPLY DEFAULT VALUES**コマンドを使用します。

🌀 DatePicker SET DAYS OFF

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

引数	型	説明
<i>objectName</i>	テキスト	→ サブフォームオブジェクト名
<i>dayType</i>	倍長整数	→ 休日タイプ
<i>ptrDaysOffArray</i>	ポインター	→ 休日の日付またはブールタイプ配列へのポインタ

説明

DatePicker SET DAYS OFFコマンドを使用してDatePickerカレンダーに表示する休日を設定できます。これらの日付はボールドおよびイタリックで表示され、ユーザは依然選択が可能です。*objectName*引数では、コマンドを適用するサブフォームのインスタンスを指定します。この引数にはカレントフォーム中に表示されるサブフォームの名前を渡さなければなりません。

このコマンドでは毎週や毎年の休日や、特別な休日を設定できます。*dayType*引数を使用して休日タイプを指定します:

- 0 = 毎週の休日 (デフォルトで土曜と日曜)
- 1 = 毎年の休日 (1月1日など)
- 2 = 特別な休日でその年のみに設定

配列を作成して、*ptrDaysOffArray*引数にその配列へのポインタを渡すことで、休日を設定します。配列のタイプは*dayType*に渡す値により異なります:

- *dayType*に0 (毎週の休日) を渡した場合、*ptrDaysOffArray*には7要素のブール配列へのポインタを渡します。Trueはその曜日が休日であることを示します。
- *dayType*に1 (毎年の休日) または2 (特別な休日) を渡した場合、*ptrDaysOffArray*には日付配列へのポインタを渡します。この配列では、それぞれの要素に休日を示す有効な日付が格納されていなければなりません。日付はシステム言語に基づくデフォルトフォーマットで表現されなければなりません。1 (毎年の休日) を*dayType*に渡した場合、年は無視されます。有効な日付であればどんな値でも使用できます。

例題 1

デフォルトの土曜日と日曜日に代えて金曜日を毎週の休みに設定する:

```
ARRAY BOOLEAN($arrbDaysOff;7)
```

```
//デフォルトですべてのブール配列要素はFalseです。なので初期化コードを追加する必要はありません。
```

```
$arrbDaysOff{Friday}:=True
```

```
DatePicker SET DAYS OFF("mycalendar";0;->$arrbDaysOff)
```

March 2010						
Mo	Tu	We	Th	Fr	Sa	Su
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	8	9	10	11

例題 2

特別な休日を設定:

```
ARRAY DATE($arrdUniqueDays;0)
```

```
//年も設定される
```

```
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)

引数	型		説明
dayNum	倍長整数	→	最初の曜日

説明

DatePicker SET DEFAULT 1ST DAYコマンドを使用して、すべてのDatePickerカレンダーで、デフォルトで最左に表示される曜日を設定できます。

*dayNum*引数には**Days and Months**テーマの以下の4D定数を渡します:

定数	型	値
Sunday	倍長整数	1
Monday	倍長整数	2
Tuesday	倍長整数	3
Wednesday	倍長整数	4
Thursday	倍長整数	5
Friday	倍長整数	6
Saturday	倍長整数	7

このパラメタは後で作成されるカレンダーに適用され、既存のカレンダーには影響を与えない点に留意してください。既存のカレンダーに適用したい場合は**DatePicker APPLY DEFAULT VALUES**コマンドを使用します。

🌀 DatePicker SET DEFAULT DAYS OFF

DatePicker SET DEFAULT DAYS OFF (*dayType* ; *ptrDaysOffArray*)

引数	型	説明
<i>dayType</i>	倍長整数	→ 休日タイプ
<i>ptrDaysOffArray</i>	ポインター	→ 休日の日付またはブール配列へのポインタ

説明

DatePicker SET DEFAULT DAYS OFFコマンドを使用してDatePickerコンポーネントのすべてのカレンダーに現れる休日を設定できます。これらの日付は太字および斜体で表示され、ユーザーは選択が可能です。

この設定は、メソッド呼び出し後に表示されるカレンダーに対し有効で、すでに表示されたカレンダーには適用されません。既存のカレンダーに適用させたい場合は**DatePicker APPLY DEFAULT VALUES**コマンドを使用します。

このコンポーネントメソッドでは毎週および毎年、さらには特別な休日も設定できます。*dayType*引数を使用して休日のタイプを指定します:

- 0 = 毎週の休み (デフォルトで土日)
- 1 = 毎年の休み (1月1日など)
- 2 = その年だけに設定する特別な休日

配列を作成して*ptrDaysOffArray*にポインタ渡しすることで休日を設定します。配列の型は*dayType*に渡す値に基づきます:

- *dayType*に0を渡す (毎週の休み) 場合、7要素を持つブール配列へのポインタを渡さなければなりません。Trueを渡された要素は毎週の休みに設定されます。
- *dayType*に1または2を渡す (毎年あるいは特別な休日) 場合、*ptrDaysOffArray*には日付配列へのポインタを渡します。この配列にはそれぞれの要素に休日を示す有効な日付が格納されていなければなりません。日付はシステムランゲージのデフォルトフォーマットで表現されていなければなりません。*dayType*に1を渡した場合、年は無視されます。有効な日付であれば、どんな値でも渡すことができます。

例題

毎年の休みを設定する (USAの例):

```
ARRAY DATE($arrdRepeatedDays;0)
//年は無視されるので、仮に2000を使用
```

```
APPEND TO ARRAY($arrdRepeatedDays;!2000/01/01!)
APPEND TO ARRAY($arrdRepeatedDays;!2000/02/02!)
APPEND TO ARRAY($arrdRepeatedDays;!2000/02/14!)
APPEND TO ARRAY($arrdRepeatedDays;!2000/03/17!)
APPEND TO ARRAY($arrdRepeatedDays;!2000/04/01!)
APPEND TO ARRAY($arrdRepeatedDays;!2000/10/31!)
APPEND TO ARRAY($arrdRepeatedDays;!2000/11/11!)
APPEND TO ARRAY($arrdRepeatedDays;!2000/12/25!)
DatePicker SET DEFAULT DAYS OFF(1;->$arrdRepeatedDays)
```

⚙️ DatePicker SET DEFAULT MAX DATE

DatePicker SET DEFAULT MAX DATE (maxDate)

引数	型		説明
maxDate	日付	→	入力可能な日付の上限

説明

DatePicker SET DEFAULT MAX DATEコマンドを使用して、DatePickerコンポーネントのすべてのカレンダーの、入力可能な日付の上限を設定できます。

このパラメタは後で作成されるカレンダーに適用され、既存のカレンダーには影響を与えない点に留意してください。既存のカレンダーに適用したい場合は**DatePicker APPLY DEFAULT VALUES**コマンドを使用します。

⚙️ DatePicker SET DEFAULT MIN DATE

DatePicker SET DEFAULT MIN DATE (dateMin)

引数	型		説明
dateMin	日付	→	入力可能な日付の下限

説明

DatePicker SET DEFAULT MIN DATEコマンドを使用して、DatePickerコンポーネントのすべてのカレンダーの、入力可能な日付の下限を設定できます。

このパラメタは後で作成されるカレンダーに適用され、既存のカレンダーには影響を与えない点に留意してください。既存のカレンダーに適用したい場合は**DatePicker APPLY DEFAULT VALUES**コマンドを使用します。

例題

日付の下限を2000年1月1日にする:

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

🔧 DatePicker SET MAX DATE

DatePicker SET MAX DATE (objectName ; maxDate)

引数	型		説明
objectName	テキスト	→	サブフォームオブジェクト名
maxDate	日付	→	入力可能な日付の上限

説明

DatePicker SET MAX DATEコマンドを使用して、DatePickerカレンダーに入力可能な日付の上限を設定できます (上限より後の日付はカレンダー中で灰色に表示されます)。

*objectName*引数では、コマンドを適用するサブフォームのインスタンスを指定します。この引数にはカレントフォーム中に表示されるサブフォームの名前を渡さなければなりません。

*maxDate*はシステム言語に基づくデフォルト入力フォーマットで表現されなければなりません。空の日付 (!00/00/00!) を渡すと、今日の日付より後の日付はすべて入力可能になります。

日付の上限が下限より小さい場合(**DatePicker SET MIN DATE**参照)、日付の入力はできなくなります。

例題

オブジェクト名"ReturnDate"で、2009/12/31より後を無効にする:

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

🔧 DatePicker SET MIN DATE

DatePicker SET MIN DATE (objectName ; dateMin)

引数	型		説明
objectName	テキスト	→	サブフォームオブジェクト名
dateMin	日付	→	入力可能な日付の下限

説明

DatePicker SET MIN DATEコマンドを使用して、DatePickerカレンダーに入力可能な日付の下限を設定できます (下限より過去の日付はカレンダー中で灰色に表示されます)。

*objectName*引数では、コマンドを適用するサブフォームのインスタンスを指定します。この引数にはカレントフォーム中に表示されるサブフォームの名前を渡さなければなりません。

*minDate*はシステム言語に基づくデフォルト入力フォーマットで表現されなければなりません。空の日付 (!00/00/00!) を渡すと、今日の日付より前の日付はすべて入力可能になります。日付の下限が上限より大きい場合 (**DatePicker SET MAX DATE**参照)、日付の入力はできなくなります。

例題

フォームの2つのサブフォーム"DP1"と"DP2"に2つのDatePickerカレンダーが置かれています。

```
//一番目のカレンダーで2009/01/01以前を無効にする
DatePicker SET MIN DATE("DP1";!2009/01/01!)
//二番目のカレンダーで2009/03/01以前を無効にする
DatePicker SET MIN DATE("DP2";!2009/03/01!)
```

🔧 DatePicker SET WEEK FIRST DAY

DatePicker SET WEEK FIRST DAY (objectName ; dayNum)

引数	型		説明
objectName	テキスト	→	サブフォームオブジェクト名
dayNum	倍長整数	→	先頭に表示する曜日の番号

説明

DatePicker SET WEEK FIRST DAYコマンドを使用してDatePickerカレンダーの最左に表示する曜日を指定できます。デフォルトで最初の曜日は月曜日です。

*objectName*引数では、コマンドを適用するサブフォームのインスタンスを指定します。この引数にはカレントフォーム中に表示されるサブフォームの名前を渡さなければなりません。

*dayNum*引数には**Days and Months**テーマの4D定数を渡します。

定数	型	値
Sunday	倍長整数	1
Monday	倍長整数	2
Tuesday	倍長整数	3
Wednesday	倍長整数	4
Thursday	倍長整数	5
Friday	倍長整数	6
Saturday	倍長整数	7

例題 1

日曜日からの表示に設定する:

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

March 2010						
Sa	Mo	Tu	We	Th	Fr	Sa
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	8	9	10

例題 2

木曜日からの表示設定する:

```
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

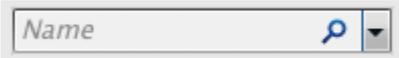
SearchPicker

 概要

 SearchPicker SET HELP TEXT

概要

SearchPickerウィジェットを使用して、ブラウザやツールバーに見られるような標準の検索エリアを簡単に作成できます。エリアのAppearanceはプラットフォームに依存します。

Windows:  Mac OS: 

デフォルトでエリアに表示されるテキストは、**SearchPicker SET HELP TEXT**コンポーネントメソッドを使用して、プログラムでコントロールできます。

動作

Appearanceの他、SearchPicker検索エリアは以下の要素による特徴を持っています: グレイアウトされたテキスト、入力エリア、削除アイコン。

- 入力エリアには検索する値を入力します。この値はプロパティリストの変数名プロパティを使用してエリアにバインドした変数に自動および動的に代入されます。この変数を使用して検索する値を取り出し、検索メソッドに渡すことができます。
- グレイアウトされたテキストは、検索が実行されるフィールドをユーザに補助的に提示するものです。エリアにフォーカスが当たるとテキストは消えます。このテキストは **SearchPicker SET HELP TEXT** コマンドで設定できます。
- 削除ボタンはエリアの内容を消去するために使用します。このボタンは自動で動作します。

実行中、フォーム内カスタムボタンのクリックやフォームイベントを使用して検索メソッドを起動できます。エリアは特に On Data Change や On Losing Focus イベントを生成します。 On Data Change イベントで検索メソッドを呼び出すことで、ユーザが他の文字を入力するたびに再評価を行うような動的検索を提供できます。

作成

フォーム上にSearchPickerエリアを挿入する方法は2つあります:

- 4Dの定義済みオブジェクトライブラリから "SearchPicker "エリアを挿入する ("入力エリア" テーマ)。
- サブフォームエリアを作成し、 **SearchPicker** 詳細フォームを割り当てる。



そのうえで、サブフォームにバインドする変数の名前を指定します (プロパティリストの"変数名"プロパティ)。フォームが実行されると、この変数は自動でユーザーが検索のために入力した値が代入されます。この値をカスタム検索メソッドに渡すことができます。

⚙ SearchPicker SET HELP TEXT

SearchPicker SET HELP TEXT (objectName ; helpText)

引数	型		説明
objectName	テキスト	→	サブフォームオブジェクト名
helpText	テキスト	→	表示するテキスト

説明

SearchPicker SET HELP TEXTコマンドを使用して、 *objectName*で指定した検索エリアのバックグラウンドに、入力不可のグレイアウトされたテキストを表示できます。このテキストはユーザがエリア内をクリックすると消えます。

例題

エリア内に"Country"と表示します。これにより検索がこの変数に関連するものだけであることを示すことができます:

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





TimePicker

-  TimePickerと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

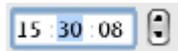
🌱 TimePickerとTimeEntry

TimePickerウィジェットは時間の入力や表示をさらにアトラクティブに行えるフィールドを作成するための簡単に使用できるオブジェクトを提供します。以下の形にて使用することができます:

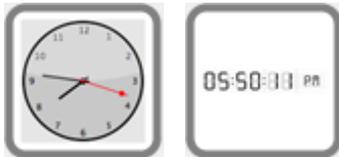
- 1つ、または2つのポップアップ:



- 数値ステッパに関連付けられた "hh:mm:ss" フォーマットの時間入力エリア(数値ステッパは時間、分、秒をそれぞれ増減させるのに使用します):



- 時計(*TimeDisplay*)またはデジタル時計(*TimeDisplayLCD*):



さらにそれぞれのタイプのTimePickerは12時間 (AM-PM) または24時間形式で時間を表示できます。

TimePickerオブジェクトは、バインド変数により提供されるメカニズムのおかげで、プログラムなしで使用できます。しかしTimePickerオブジェクトの動作をカスタマイズしたい場合は、提供されるコンポーネントメソッドを使用しなければなりません。

作成と利用

TimePickerエリアをフォームに挿入する方法は2つあります:

- 4Dの定義済みオブジェクトライブラリから"TimePicker"または"TimeEntry"を挿入する。
- サブフォームエリアを作成し、**TimePicker**または**TimeEntry**詳細フォームを割り当てる。

そのうえで、サブフォームにバインドする変数の名前を指定します (プロパティリストの"変数名"プロパティ)。フォームが実行されると、この変数は自動でユーザーが指定した時間が代入されます。逆にこの変数の値をプログラムで更新すると、自動でサブフォームに反映されます。変数に名前を与えないことにより、フォームローカル変数の利点を生かすこともできます。

時計について(v14よりの新機能)

Clock ウィジェットはSVGで描画されているので、ベクターパスをもち、それ故にアプリケーションモードでは自在に変形することが出来ます(デザインモードではサイズは固定です):



更に、以下の様な機能もあります:

- 時計の秒針は、 **TimePicker DISPLAY SECOND HAND** メソッドを使用することにより表示したり隠したりすることが出来ます。
- 時計は、時刻に応じて"昼間モード"と"夜間モード"を切り替えます:



昼間と夜間の時刻の範囲はそれぞれ、

8:00:00 -> 19:59:59 = 昼間

20:00 -> 07:59:59 = 夜間

となっています。

- "デジタル時計"のウィジェットは透明なので色のあるオブジェクトに重ねて色を変えることも出来ます:



コンポーネントメソッドには表示のオプションに関する様々なメソッドが用意されています。これらのメソッドには"TimePicker LCD" の接頭辞がついています。

注: この時計は"Resources"フォルダーの第一レベルにある"clock.svg" を置き換えることにより削除したりデベロッパ自身の作成したオブジェクトで置き換えたりすることが出来ます。

カレントの時刻または静的な時刻の表示

時計にはカレントの時刻を動的に表示させるか、または静的な時間を表示させることが出来ます。

- カレントの時刻を表示させる場合、ウィジェットのサブフォームオブジェクトを**実数変数**と関連付けて下さい(デフォルトの操作)。この場合、ウィジェットは自動的にカレントの時刻を表示し、時計の様に動作します。
表示される時刻にはオフセットを適用することも出来ます。秒数に換算したオフセットをウィジェットに関連付けられている変数に渡して下さい。例えば、3600 = 時計を1時間進める、-1800 = 時計を30分遅らせる、等です。
- 静的な時刻を表示させる場合、(**C_TIME** コマンドかプロパティリストを使用して)ウィジェットのサブフォームオブジェクトを**時間変数**と関連付けて下さい。時計はこの変数の時刻を表示します。

例えば時計に 10:10:30 と表示させたい場合:

```
C_TIME(myvar) // myvar はウィジェットの変数の名前  
myvar:=?10:10:30?
```



⚙️ TimePicker APPLY DEFAULT VALUES

TimePicker APPLY DEFAULT VALUES (*objectName*)

引数	型		説明
<i>objectName</i>	テキスト	→	サブフォームオブジェクト名

説明

TimePicker APPLY DEFAULT VALUESコマンドを使用して、*objectName*サブフォームオブジェクトのすべてのTimePickerパラメタを現在のデフォルト値に再設定できます。

これらのデフォルト値は初期値またはコンポーネントのSET DEFAULTコマンドで設定された値です。

このコマンドは即座に効果を及ぼします。*objectName*のデフォルト値は即座に更新されます。オブジェクトに割り当てられた変数は、新しい値を有効にするために更新されるかもしれないことに留意してください。例えば新しい最小時間のデフォルト値が07:00:00に設定され、*objectName*に割り当てられた変数が06:00:00のとき、その値は自動で07:00:00に変更されます。

TimePickerパラメタには以下があります:

- 入力可能時間の下限と上限
- AMおよびPMラベル
- 分単位のステップ

⚙️ TimePicker DISPLAY SECOND HAND

TimePicker DISPLAY SECOND HAND (*objectName* ; *secondHand*)

引数	型	説明
<i>objectName</i>	テキスト	→ サブフォームオブジェクト名
<i>secondHand</i>	ブール	→ True (デフォルト) = 秒針を表示 False = 秒針を非表示

説明

TimePicker DISPLAY SECOND HAND コマンドは、*objectName* で指定したサブフォームオブジェクトの、秒針の表示・非表示を設定します(ただしclockウィジェットに限る)。

デフォルトでは、秒針は表示されています。日表示するためには、このコマンドを呼び出して *secondHand* 引数に **False** を渡します。

⚙ TimePicker LCD DISPLAY AMPM

TimePicker LCD DISPLAY AMPM (*objectName* ; *amPm*)

引数	型	説明
<i>objectName</i>	テキスト	→ サブフォームオブジェクト名
<i>amPm</i>	ブール	→ True = AM/PMを表示、 False =非表示

説明

TimePicker LCD DISPLAY AMPM コマンドは、 *objectName* で指定したサブフォームオブジェクトの、AM/PM表示を表示・非表示を設定します(デジタル時計に限る)。

この文字は時計が12時間モードのときに午前か午後を区別するために表示されています。(**TimePicker LCD SET MODE** を参照して下さい。)

デフォルトでは、この文字は表示されています。 *amPm* 引数に **False** を渡すことによって非表示にすることができます。

例題

AM/PM の文字表示を非表示にします:

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

A digital clock display showing the time 03:20:04. The digits are black on a light gray background.

⚙ TimePicker LCD DISPLAY SECONDS

TimePicker LCD DISPLAY SECONDS (*objectName* ; *seconds*)

引数	型		説明
<i>objectName</i>	テキスト	→	サブフォームオブジェクト名
<i>seconds</i>	ブール	→	True = 秒数を表示、 False = 非表示

説明

TimePicker LCD DISPLAY SECONDS *objectName* で指定したサブフォームオブジェクトの、秒数表示の表示・非表示を設定します。(ただしデジタル時計に限る)

デフォルトでは、秒数は表示されています。 *seconds* 引数に **False** を渡す事によって非表示にすることができます。

⚙️ TimePicker LCD SET COLOR

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

引数	型	説明
<i>objectName</i>	テキスト	→ サブフォームオブジェクト名
<i>color</i>	倍長整数	→ RGB カラーの値(4 bytes) または 他の引数が渡された場合には赤成分の値 (0..255)
<i>colorG</i>	倍長整数	→ 緑成分の値 (0..255)
<i>colorB</i>	倍長整数	→ 青成分の値 (0..255)

説明

TimePicker LCD SET COLOR コマンドは、*objectName* で指定したサブフォームオブジェクトの、数字の色を設定します。(ただしデジタル時計に限る)。

このコマンドには二つのシンタックスが使用できます:

- *color* 引数のみを指定する場合、(0x00RRGGBB) のフォーマットに適合する4バイトの倍長整数を渡します。(桁番号は右から左へ数えられ、0から3とナンバリングされています):

バイト 詳細

- 3 絶対RGBカラーを指定するためには0でなければなりません。
- 2 カラーの赤成分 (0..255)
- 1 カラーの緑成分 (0..255)
- 0 カラーの青成分 (0..255)

- それ以外に、3つの引数を渡す事もできます。: *color*, *colorG* そして *colorB* の3つです。この場合、それぞれの引数が0から255の間の数字である必要があり、それぞれがRGBカラーの要素となります。

例題

数字の表示を赤に変更する場合を考えます:

```
TimePicker LCD SET COLOR("Subform1";0x00FF0000)
// 別の記法: TimePicker LCD SET COLOR ("Subform1";255;0;0)
```

02:58:28 PM

⚙️ TimePicker LCD SET MODE

TimePicker LCD SET MODE (objectName ; mode)

引数	型	説明
objectName	テキスト	→ サブフォームオブジェクト名
mode	倍長整数	→ 12 = 時間を12時間モードで表示 24 = 時間を24時間モードで表示

説明

TimePicker LCD SET MODE コマンドは、*objectName* で指定したサブフォームオブジェクトの、12時間モードまたは24時間モードの表示モードを設定します。(ただしデジタル時計に限る)

デフォルトでは、オブジェクトは12時間モードで表示されています。*mode* 引数に24を渡す事によって24時間モードに切り替えることが出来ます。この場合、一般的に言ってAM/PM表示を同時に非表示すると良いです(**TimePicker LCD DISPLAY AMPM** コマンドを参照して下さい)。

例題

24時間表示モードへと切り替えて、同時にAM/PM表示を非表示にする場合:

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



A digital clock display showing the time 15:40:23. The digits are black on a light gray background.

TimePicker RESET DEFAULT VALUES

TimePicker RESET DEFAULT VALUES

このコマンドは引数を必要としません

説明

TimePicker RESET DEFAULT VALUESコマンドはTimePickerコンポーネントのパラメタを初期値にリセットします。このコマンド実行後は以下のように設定されます：

- 入力可能な時間の下限は08:00:00
- 入力可能な時間の上限は20:00:00
- AMとPMラベルはシステムラベル
- 分単位のステップは 00:15:00

この設定は後で作成されるオブジェクトに適用され、既存のオブジェクトには影響を与えない点に留意してください。既存のオブジェクトに適用したい場合は**TimePicker APPLY DEFAULT VALUES**コマンドを使用します

TimePicker SET DEFAULT LABEL AM

TimePicker SET DEFAULT LABEL AM (label)

引数	型		説明
label	テキスト	→	AMに使用されるラベル

説明

TimePicker SET DEFAULT LABEL AMコマンドを使用して、AM/PMフォーマットで表示されるすべてのTimePickerオブジェクトのデフォルトの"AM"ラベルを変更できます。

この設定は後で作成されるオブジェクトに適用され、既存のオブジェクトには影響を与えない点に留意してください。既存のオブジェクトに適用したい場合は**TimePicker APPLY DEFAULT VALUES**コマンドを使用します。

⚙️ TimePicker SET DEFAULT LABEL PM

TimePicker SET DEFAULT LABEL PM (label)

引数	型		説明
label	テキスト	→	PMに使用されるラベル

説明

TimePicker SET DEFAULT LABEL PMコマンドを使用して、AM/PMフォーマットで表示されるすべてのTimePickerオブジェクトのデフォルトの"PM"ラベルを変更できます。

この設定は後で作成されるオブジェクトに適用され、既存のオブジェクトには影響を与えない点に留意してください。既存のオブジェクトに適用したい場合は**TimePicker APPLY DEFAULT VALUES**コマンドを使用します。

⚙️ TimePicker SET DEFAULT MAX TIME

TimePicker SET DEFAULT MAX TIME (maxTime)

引数	型		説明
maxTime	時間	→	入力可能な時間の上限

説明

TimePicker SET DEFAULT MAX TIMEコマンドを使用して、すべてのTimePickerオブジェクトで使用される入力可能な時間の上限を設定できます。

この設定は後で作成されるオブジェクトに適用され、既存のオブジェクトには影響を与えない点に留意してください。既存のオブジェクトに適用したい場合は**TimePicker APPLY DEFAULT VALUES**コマンドを使用します。

TimePicker SET DEFAULT MIN TIME

TimePicker SET DEFAULT MIN TIME (minTime)

引数	型		説明
minTime	時間	→	入力可能な時間の下限

説明

TimePicker SET DEFAULT MIN TIMEコマンドを使用して、すべてのTimePickerオブジェクトで使用される入力可能な時間の下限を設定できます。

この設定は後で作成されるオブジェクトに適用され、既存のオブジェクトには影響を与えない点に留意してください。既存のオブジェクトに適用したい場合は**TimePicker APPLY DEFAULT VALUES**コマンドを使用します。

TimePicker SET DEFAULT STEP

TimePicker SET DEFAULT STEP (step)

引数	型		説明
step	時間	→	2つの時間値の間隔

説明

TimePicker SET DEFAULT STEPコマンドを使用して、すべてのTimePickerオブジェクトの時間と時間の間隔を設定できます。

この設定は後で作成されるオブジェクトに適用され、既存のオブジェクトには影響を与えない点に留意してください。既存のオブジェクトに適用したい場合は**TimePicker APPLY DEFAULT VALUES**コマンドを使用します。

⚙ TimePicker SET LABEL AM

TimePicker SET LABEL AM (objectName ; label)

引数	型		説明
objectName	テキスト	→	サブフォームオブジェクト名
label	テキスト	→	AMに使用するラベル

説明

TimePicker SET LABEL AMコマンドを使用して、AM/PMフォーマットで表示されるTimePickerオブジェクトの"AM"ラベルを変更できます。コマンドは`objectName`で指定されたオブジェクトに適用されます。デフォルトでシステムのam/pmラベルが使用されます。

例題

AMのとき、システムラベルの代わりにデフォルトで"in the morning"を使用する:

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

⚙ TimePicker SET LABEL PM

TimePicker SET LABEL PM (objectName ; label)

引数	型		説明
objectName	テキスト	→	サブフォームオブジェクト名
label	テキスト	→	PMに使用するラベル

説明

TimePicker SET LABEL PMコマンドを使用して、AM/PMフォーマットで表示されるTimePickerオブジェクトの"PM"ラベルを変更できます。コマンドは`objectName`で指定されたオブジェクトに適用されます。デフォルトでシステムのam/pmラベルが使用されます。

例題

PMのとき、システムラベルの代わりにデフォルトで"in the evening"を使用する:

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

TimePicker SET MAX TIME

TimePicker SET MAX TIME (*objectName* ; *maxTime*)

引数	型		説明
<i>objectName</i>	テキスト	→	サブフォームオブジェクト名
<i>maxTime</i>	時間	→	入力可能な時間の上限

説明

TimePicker SET MAX TIME コマンドを使用して、*objectName* で指定したオブジェクトが受け入れる時間の上限を設定できます。それよりも大きな時間が入力されると、受け入れられません。

⚙️ TimePicker SET MIN TIME

TimePicker SET MIN TIME (*objectName* ; *minTime*)

引数	型		説明
<i>objectName</i>	テキスト	→	サブフォームオブジェクト名
<i>minTime</i>	時間	→	入力可能な時間の下限

説明

TimePicker SET MIN TIMEコマンドを使用して、*objectName*で指定したオブジェクトが受け入れる時間の下限を設定できます。それよりも小さな時間が入力されると、受け入れられません。

⚙ TimePicker SET STEP

TimePicker SET STEP (objectName ; step)

引数	型		説明
objectName	テキスト	→	サブフォームオブジェクト名
step	時間	→	2つの時間値の間隔

説明

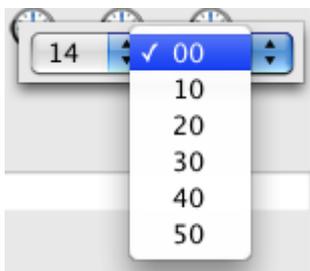
TimePicker SET STEPコマンドを使用して、*objectName*で指定したオブジェクトで利用される時間と時間の間隔のステップを設定できます。このパラメタはポップアップメニューとして表示されるTimePickerにのみ適用されます。

*step*値は1分から1時間の間で、60分を割り切れる整数値のみを指定できます。つまり実際1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30 そして 60が利用可能です。他の値は自動でこの原則に沿うよう丸められます。

例題

"time1"という名前のポップアップメニュー形式のTimePickerを設定します。入力可能な時間は8:30から16:30で、間隔は10分です:

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



4D Widgets (ウィジェット) - コマンドリスト (文字順)

- ⚙ TimePicker DISPLAY SECOND HAND
- ⚙ TimePicker LCD DISPLAY AMPM
- ⚙ TimePicker LCD DISPLAY SECONDS
- ⚙ TimePicker LCD SET COLOR
- ⚙ TimePicker LCD SET MODE