

## Sommario

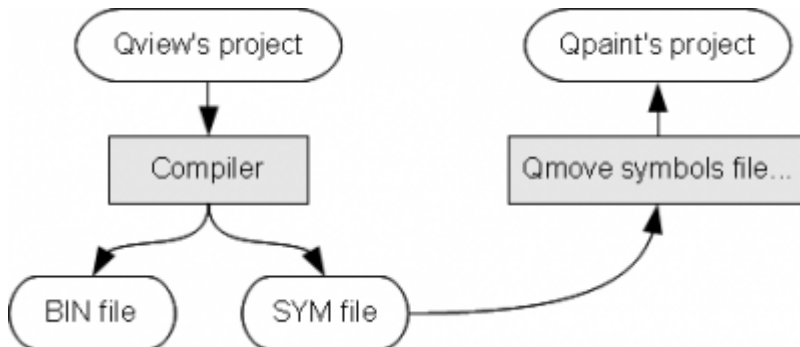
<b>QPaint 5.4</b>	3
<b>0.1 Introduzione</b>	3
<b>0.2 Quick Start</b>	3
0.2.1 Nuovo Progetto	3
0.2.2 Symbols file importing	3
0.2.3 Pagine	4
0.2.4 Oggetti	4
0.2.5 Eventi e Azioni	6
0.2.6 Variabili di terminale	7
<b>1. Controllo da QView</b>	8
<b>1.1 Rilevare la pressione di un tasto</b>	8
<b>1.2 Capire quale pagina è visualizzata</b>	9
<b>1.3 Comandare un cambio pagina</b>	10
<b>1.4 Dataentry attivo</b>	11
<b>2. Esempi</b>	12
<b>2.1 Pulsante grafico</b>	12



## QPaint 5.4

### 0.1 Introduzione

QPaint 5 è un ambiente di sviluppo grafico per la programmazione di una interfaccia operatore QEM. In questo documento sono riportate le caratteristiche principali del programma QPaint. Durante la descrizione dell'ambiente QPaint si faranno molti riferimenti ai concetti dell'ambiente di sviluppo QView che è l'ambiente per sviluppare il software per l'automazione. Il progetto realizzato con QPaint può accedere a tutte le variabili, parametri e altre strutture dati dichiarate nel progetto realizzato con QView. Il meccanismo per "sincronizzare" il progetto QPaint con le variabili di QView è schematizzato nella figura seguente:



Al momento della compilazione del progetto QView, viene generato un file con lo stesso nome del progetto ed estensione ".sym". Il progetto QPaint può utilizzare questo file per poter accedere a tutti i dati.

Il QPaint è un ambiente di sviluppo che permette di creare un'interfaccia operatore senza l'utilizzo di un linguaggio di programmazione, ma tramite un ambiente intuitivo che permette fin da subito di apprezzare la grafica del risultato finale.

### 0.2 Quick Start

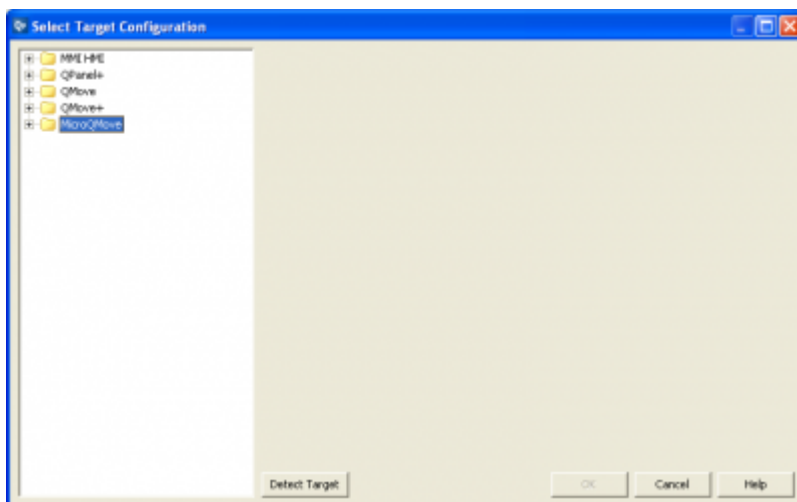
Diamo alcune informazioni fondamentali per un rapido avvio di un nuovo progetto QPaint.

#### 0.2.1 Nuovo Progetto

Si deve selezionare

- File - New Project...

e il QPaint chiedere subito di specificare il modello del prodotto per cui si vuole realizzare il progetto:

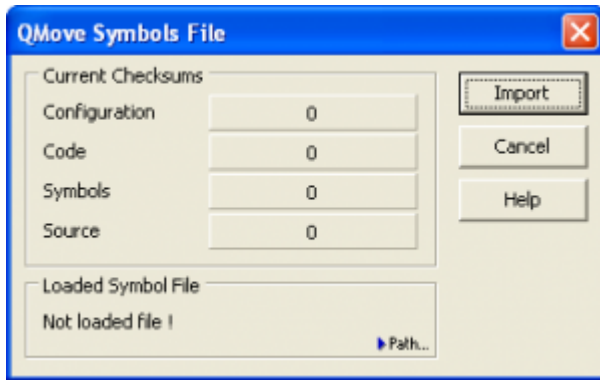


#### 0.2.2 Symbols file importing

Una delle prime operazioni da fare è importare il file che permette di accedere ai simboli dichiarati nel progetto QView associato. Selezionare:

- Project - QMove Symbols File...

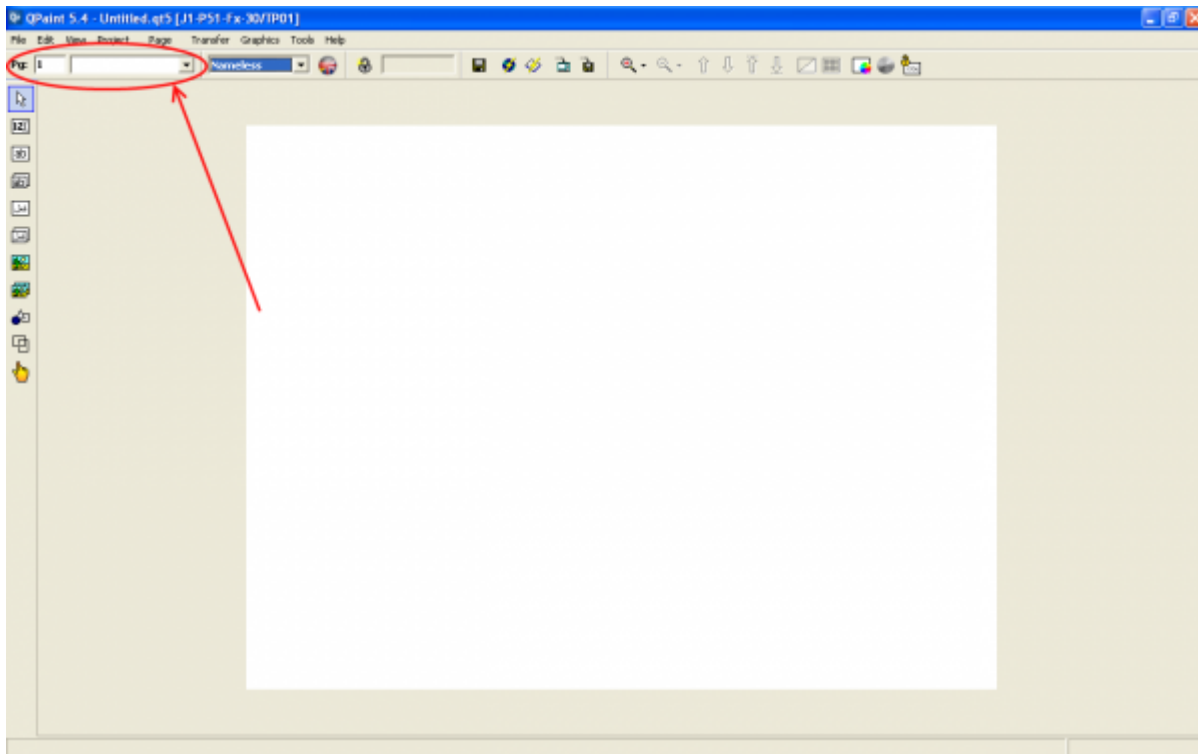
Appare la seguente finestra di dialogo



dopo avere premuto il bottone *Import*, si deve specificare il file dei simboli. Questo file viene generato da QView dopo la compilazione di un progetto. Esso ha lo stesso nome del progetto QView, ma con estensione “.sym”.

### 0.2.3 Pagine

Un progetto QPaint è composto da un certo numero di *Pagine* che sono le visualizzazioni che appaiono sul display e con cui l'operatore dovrà interagire. La lista delle pagine presenti è nell'angolo in alto a sinistra, esse possono essere identificate con un numero o un nome univoco:

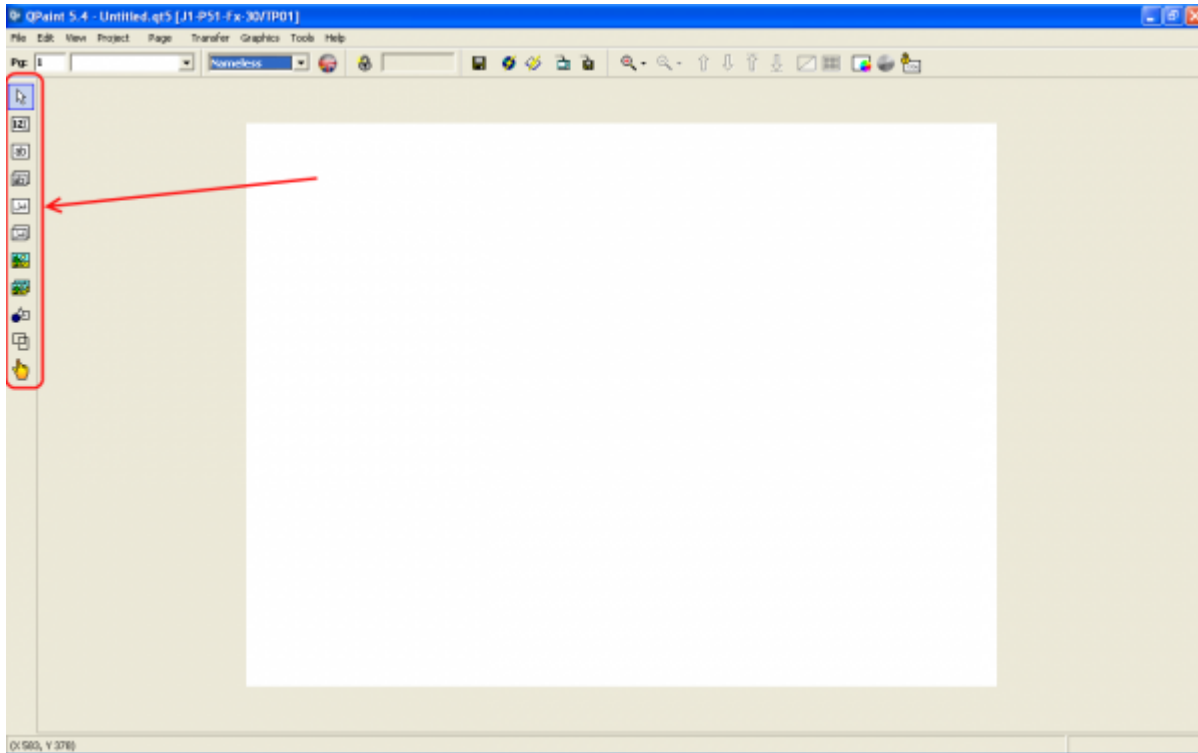


Esiste il menù *Page* per poter gestire le pagine del progetto:

- inserire,
- aggiungere,
- togliere,
- copiare,
- incollare,
- importare,
- esportare,
- impostare il colore dello sfondo,
- programmare gli eventi e le azioni di una pagina.

### 0.2.4 Oggetti

Nelle pagine è possibile inserire un certo numero di oggetti grafici per vari scopi. Per selezionare quale oggetto inserire nella pagina si usa la barra verticale indicata nella seguente figura:



#### 0.2.4.1 Value Object

È l'oggetto che serve per visualizzare il valore di una variabile o di un parametro.

#### 0.2.4.2 String Object

È l'oggetto che serve per scrivere del testo fisso sulla pagina.

#### 0.2.4.3 ValString Object

Con questo oggetto è possibile far apparire un messaggio variabile dipendente dal valore di una variabile del progetto QView.

#### 0.2.4.4 UniString Object

È un oggetto con la stessa funzionalità del *String Object* ma con la possibilità di usare i font installati sul PC ed in particolare i font Unicode che permettono l'uso di alfabeti diversi da quello latino: cirillico, greco, arabico, cinese, ...

#### 0.2.4.5 UniValString Object

È un oggetto con la stessa funzionalità del *ValString Object* ma con la stessa caratteristica del *UniString Object*.

#### 0.2.4.6 Image Object

Questo oggetto permette di inserire un'immagine nella pagina. L'immagine deve essere presente nella libreria delle immagini presente nel progetto. Per accedere alla libreria delle immagini selezionare

- Graphics - Image Manager

L'*Image Manager* serve per poter caricare le immagini nella libreria ed eventualmente fare alcune modifiche relative alle dimensioni.

### 0.2.4.7 VallImage Object

Questo oggetto permette di aggiungere sulla pagina una immagine variabile in funzione del valore di una variabile del progetto QView. Tutte le immagini utilizzate in un oggetto di questo tipo devono essere delle stesse dimensioni. Le immagini devono essere scelte dalla libreria delle immagini già inserite in *Image Manager*.

### 0.2.4.8 Vector Image Object

Questo oggetto permette di riservare un'area della pagina dedicata al disegno di forme elementari. Questo oggetto deve obbligatoriamente essere associato ad un array di word. Esistono delle funzioni di QView che forniscono delle primitive di disegno (linee, rettangoli, archi, cerchi,...).

### 0.2.4.9 Box Object

È un oggetto che permette di realizzare delle cornici di forma rettangolare.

### 0.2.4.10 Touch Area

L'oggetto *Touch Area* crea un'area di tocco che permette di attivare degli eventi legati al tocco dello schermo da parte dell'operatore. Questo oggetto potrà essere utilizzato solamente nei terminali operatore dotati di touch screen. Gli eventi possono essere:

- **On Touch Press**, è il momento in cui il dito entra in contatto con lo schermo;
- **On Touch Release**, è il momento in cui il dito rilascia lo schermo,
- **On Touch Press Double**, è un doppio tocco ravvicinato nel tempo.

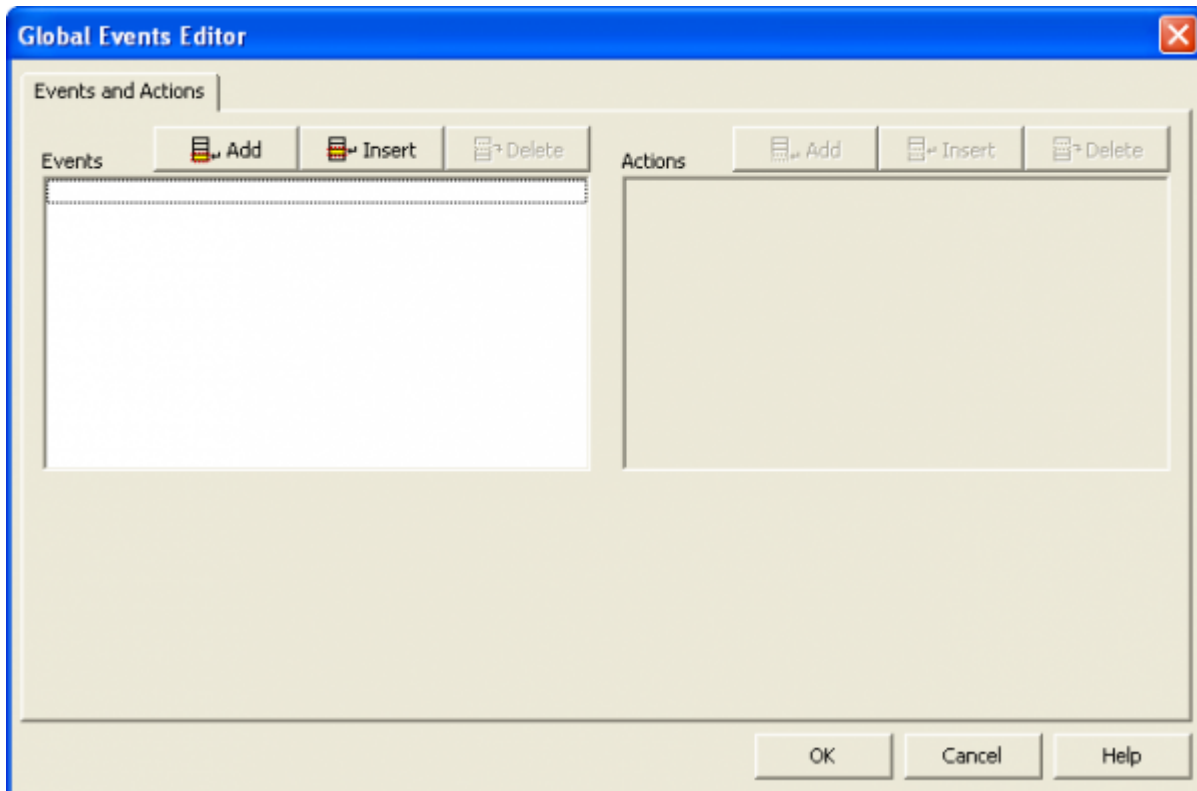
ad ognuno di questi eventi possono essere associate delle azioni. Le azioni sono di tipo generico e non legate al concetto di touch screen e quindi verranno descritte in seguito.

## 0.2.5 Eventi e Azioni

Oltre alla composizione della grafica, è possibile anche implementare nel progetto una semplice programmazione. Il principio è quello di avere a disposizione una serie di *eventi* da cui scegliere. Ad ogni evento io posso associare delle *azioni*. Inoltre posso programmare queste associazioni *evento-azioni* in modo che abbia effetto in tutte le pagine oppure solamente nella pagina visualizzata in quel momento. Per questo motivo esistono due contesti:

1. Project - Global Events Editor
2. Page - Page Events Editor

Entrambi gli *Editors* sono composti di due aree come mostrati in figura



A sinistra si inseriscono gli *eventi* e a destra si inseriscono le *azioni* associate all'evento selezionato a sinistra.

### 0.2.5.1 Eventi

La lista degli eventi possibili sono:

- **On Key**, evento che si ripete continuamente finché è premuto il tasto specificato;
- **On Press**, evento che si attiva alla pressione del tasto specificato;
- **On Release**, evento che si attiva al rilascio del tasto specificato;
- **On Always**, evento che si ripete continuamente;
- **On Page In**, evento che si attiva all'ingresso della pagina;
- **On Time**, evento che si attiva in un certo momento specificato da una data e un'ora;
- **On Change Var**, evento che si attiva quando la variabile specificata cambia valore;
- **On Var**, evento che si attiva quando la variabile specificata assume il valore specificato.

### 0.2.5.2 Azioni

Ad ogni *evento* è possibile associare più di una *azione*. Le *azioni* possibili sono:

- **Goto Page**, vai alla pagina specificata;
- **Next Page**, vai alla pagina successiva;
- **Previous Page**, vai alla pagina precedente;
- **Begin Data Entry**, attiva l'inserimento del dato;
- **Send Command**, invia un comando ad un device dichiarato nel progetto QView;
- **Set Variable**, assegna un valore ad una variabile del progetto QView;
- **Led On**, attiva il LED specificato (solitamente i LED sono associati ai tasti funzione);
- **Led Off**, disattiva il LED specificato;
- **Led Blink**, attiva il lampeggio del LED specificato;
- **Backup**, invia un comando di Backup (verificare se questo comando è supportato dall'hardware);
- **Restore**, invia un comando di Restore (verificare se questo comando è supportato dall'hardware).

### 0.2.6 Variabili di terminale

Ogni progetto QPaint contiene delle variabili sempre presenti che forniscono una serie di informazioni relative al terminale stesso. Queste variabili hanno già un nome prefissato, alcune sono in sola lettura e altre sono anche scrivibili. Facciamo un breve elenco delle variabili più importanti:

Nome	Tipo	Read/Write	Descrizione
\$KEY	L	R	Codice del tasto premuto. Ad ogni bit corrisponde un tasto della tastiera.
\$KEYF	L	R	Codice del tasto funzione premuto. Ad ogni bit corrisponde un tasto della tastiera.
\$KEYF2	L	R	Codice del tasto funzione premuto (Secondo gruppo). Ad ogni bit corrisponde un tasto della tastiera.
\$LED5	L	RW	Codice per attivare/disattivare i LED. Ad ogni bit corrisponde un LED.
\$LED52	L	RW	Codice per attivare/disattivare i LED (Secondo gruppo). Ad ogni bit corrisponde un LED.
\$HOUR, \$MIN, \$SEC, \$DAY, \$MONTH, \$YEAR	B, B, B, B, B, W	RW	Ora e data.
\$DATAENTRYON	F	R	Segnala che in questo momento è attivo l'inserimento di un valore
\$PAGE	L	R	Valore della pagina visualizzata in questo momento.

Ce ne sono molte altre ma queste sono le più utilizzate ed essenziali per poter sviluppare i progetti.

## 1. Controllo da QView

La gestione dell'interfaccia operatore può essere realizzata anche tramite uno scambio di informazioni con il progetto realizzato in QView. Le operazioni più importanti per la gestione di un'interfaccia operatore sono:

- rilevare la pressione di un tasto,
- capire che pagina è visualizzata in quel momento,
- comandare un cambio di una pagina,
- verificare se l'inserimento del dato (dataentry) è attivo (in questo caso non è possibile cambiare la pagina).

Spesso si preferisce gestire l'interfaccia operatore controllando direttamente le operazioni basilari dal progetto QView. Vediamo come questo sia possibile.

### 1.1 Rilevare la pressione di un tasto

Le variabili che permettono di conoscere se un tasto è premuto sono

```
$KEY, $KEYF, $KEYF2.
```

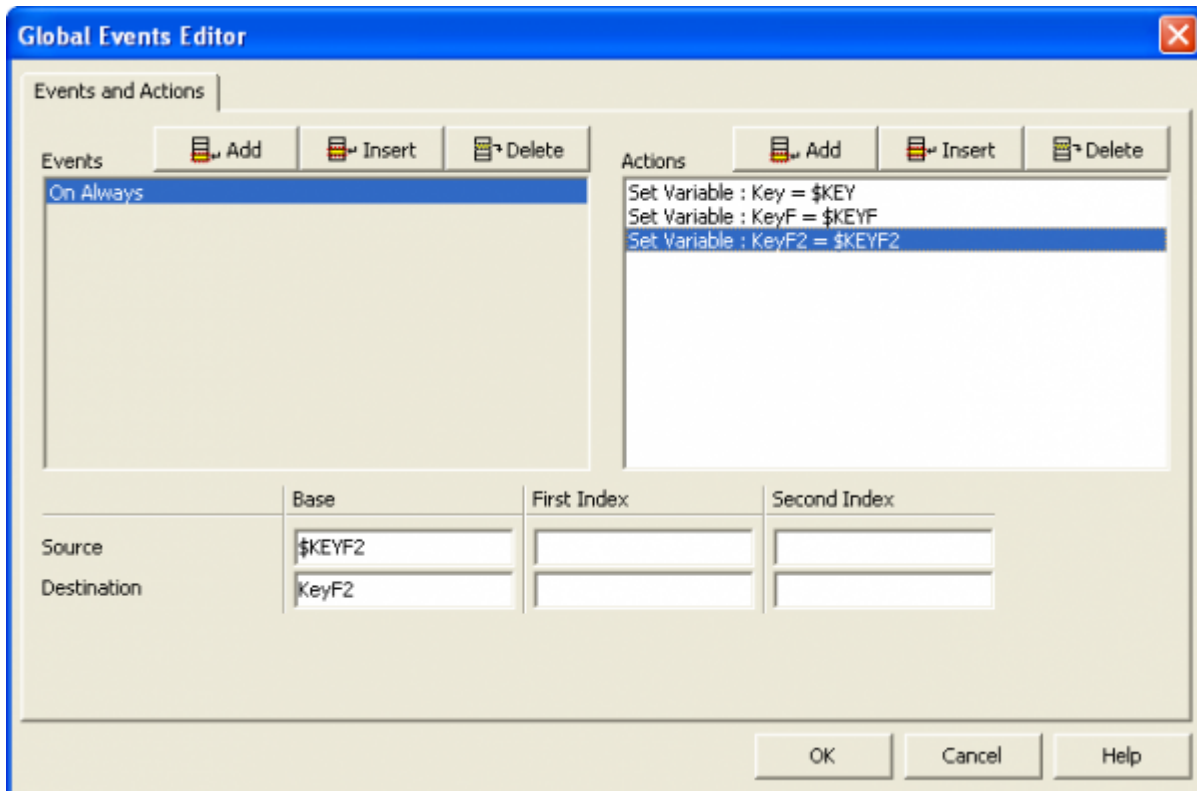
Ogni bit di queste variabili è assegnato ad un tasto della tastiera. Queste variabili sono accessibili solo tramite QPaint e non sono visibili nel progetto QView. Per poter trasferire il valore di queste variabili al progetto QView è necessario prima di tutto predisporre tre variabili nel progetto QView:

```
GLOBAL
Key    L    ;This will be match to $KEY
KeyF   L    ;This will be match to $KEYF
KeyF2  L    ;This will be match to $KEYF2
```

Nel progetto QPaint si seleziona:

- Project - Global Events Editor





Si deve inserire un evento *OnAlways* e associare ad esso tre azioni *SetVariable* come mostrato nella figura. In questo modo il progetto QPaint copierà continuamente (*OnAlways*) i valori delle sue tre variabili nelle variabili di QView. Nel progetto QView è possibile dichiarare le costanti relative alla pressione di ogni tasto:

```

CONST
KEY_1      268435456 ; "1"
KEY_2      1048576  ; "2"
KEY_3      4096     ; "3"
KEY_4      536870912 ; "4"
KEY_5      2097152  ; "5"
KEY_6      8192     ; "6"
KEY_7      1073741824 ; "7"
KEY_8      4194304  ; "8"
KEY_9      16384    ; "9"
KEY_0      8388608  ; "0"
KEY_CLR    -2147483648 ; "CLR"
KEY_ENTER  128      ; "ENTER"
KEY_HELP   64       ; "HELP"
KEY_DECP   32       ; "DECPT"
KEY_SIGN   16       ; "+/-"
KEY_ESC    33554432 ; "ESC"
KEY_UP     131072   ; "UP"
KEY_PGUP   512      ; "PGUP"
KEY_LEFT   67108864 ; "LEFT"
KEY_NEXT   262144   ; "NEXT"
KEY_RIGHT  1024     ; "RIGHT"
KEY_INS    134217728 ; "INS"
KEY_DOWN   524288   ; "DOWN"
KEY_PGDN   2048     ; "PGDN"

KEY_F1     33554432 ; "F1"
KEY_F2     67108864 ; "F2"
KEY_F3     134217728 ; "F3"
KEY_F4     268435456 ; "F4"
KEY_F5     536870912 ; "F5"
KEY_F6     131072   ; "F6"
KEY_F7     262144   ; "F7"
KEY_F8     524288   ; "F8"
KEY_F9     1048576  ; "F9"
KEY_F10    2097152  ; "F10"
KEY_F11    1        ; "F11"
KEY_F12    2        ; "F12"
KEY_F13    4        ; "F13"
KEY_F14    8        ; "F14"
KEY_F15    16       ; "F15"
KEY_F16    32       ; "F16"
KEY_F17    64       ; "F17"
KEY_F18    128      ; "F18"
KEY_F19    256      ; "F19"
KEY_F20    512      ; "F20"
KEY_F21    1024     ; "F21"
KEY_F22    2048     ; "F22"
KEY_F23    4096     ; "F23"
KEY_F24    8192     ; "F24"
KEY_F25    16384    ; "F25"
KEY_F26    32768    ; "F26"

```

Quindi il codice per eseguire una certa azione se viene premuto un tasto sarà

```

IF Key EQ KEY_1
;Put here the code to do when the operator press "1" key
ENDIF

```

## 1.2 Capire quale pagina è visualizzata

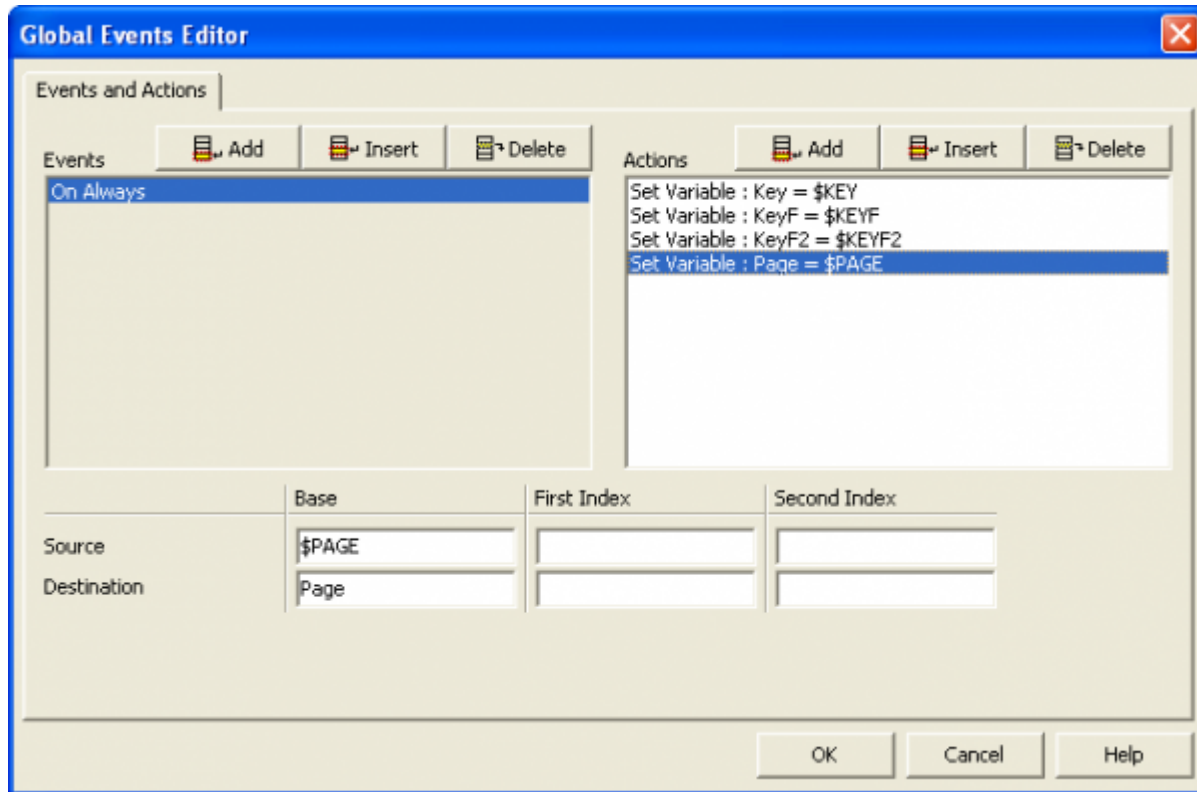
Analogamente al capitolo precedente, anche in questo caso devo predisporre una variabile nel progetto Qview per poter accogliere la variabile

```
$PAGE
```

del progetto QPaint. Quindi dichiaro:

```
GLOBAL
Page L ;This will be match to $PAGE
```

e poi aggiungo un'azione *SetVariable* associata all'evento *OnAlways*:



In questo modo posso eseguire operazioni diverse a seconda di quale pagina è visualizzata:

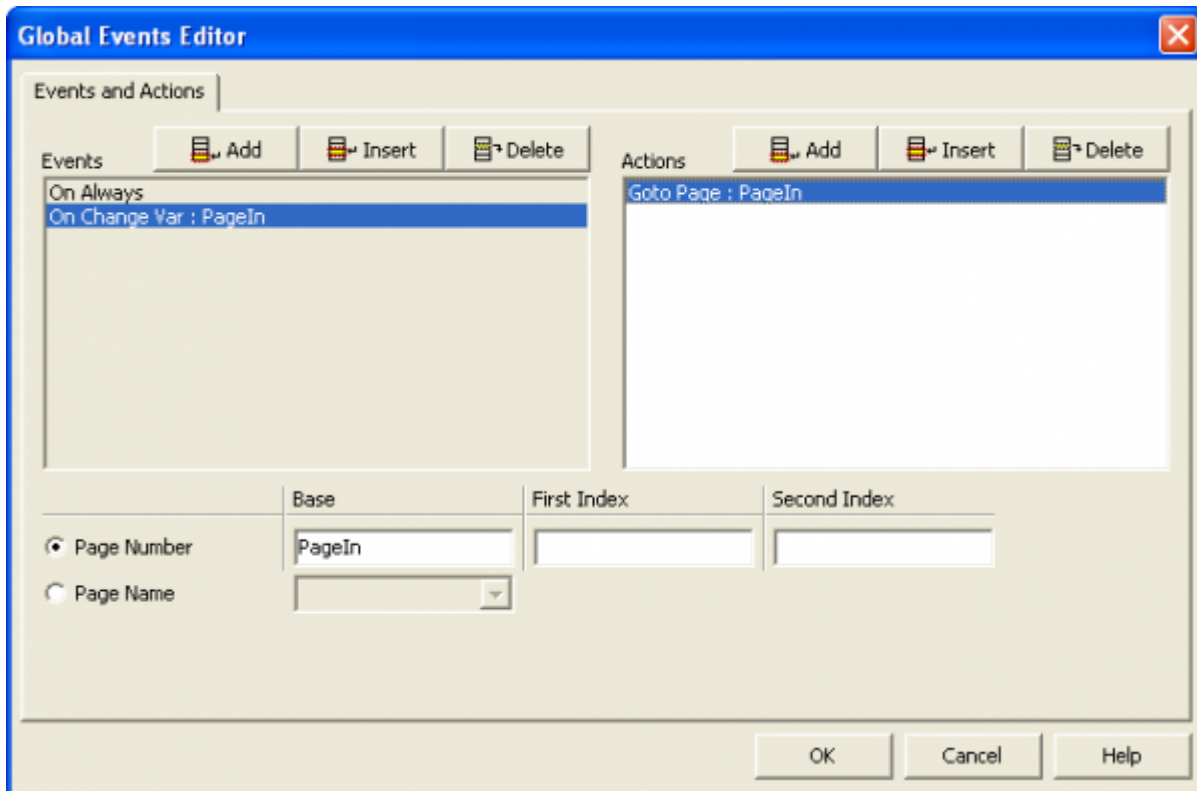
```
IF Page EQ MENU_PAGE
  IF Key EQ KEY_1
    ;Put here the code to do when the operator press "1" key on the "MENU_PAGE" page
  ENDIF
ENDIF
```

## 1.3 Comandare un cambio pagina

Per comandare un cambio pagina dell'interfaccia operatore tramite una sola riga di codice nel progetto QView, è necessario prima di tutto dichiarare una variabile QView :

```
GLOBAL
PageIn L ;This will be used to request the page change
```

è necessario poi aggiungere un evento *OnChangeVar* e associare ad esso un'azione *GoToPage* come in figura:



Quindi quando il valore della variabile *PageIn* viene cambiato in una riga di codice del progetto QView, viene attivata l'azione *Goto Page* che cambia la pagina e quindi viene visualizzata la pagina con indice il nuovo valore della variabile *PageIn*.

Il codice QCL sarà quindi:

```
IF Page EQ MENU_PAGE
  IF Key EQ KEY_1
    PageIn = PAGE_1 ;Change page request: go to "PAGE_1" page
    WAIT(Page EQ PAGE_1) ;Wait the change page has done
  ENDIF
ENDIF
```

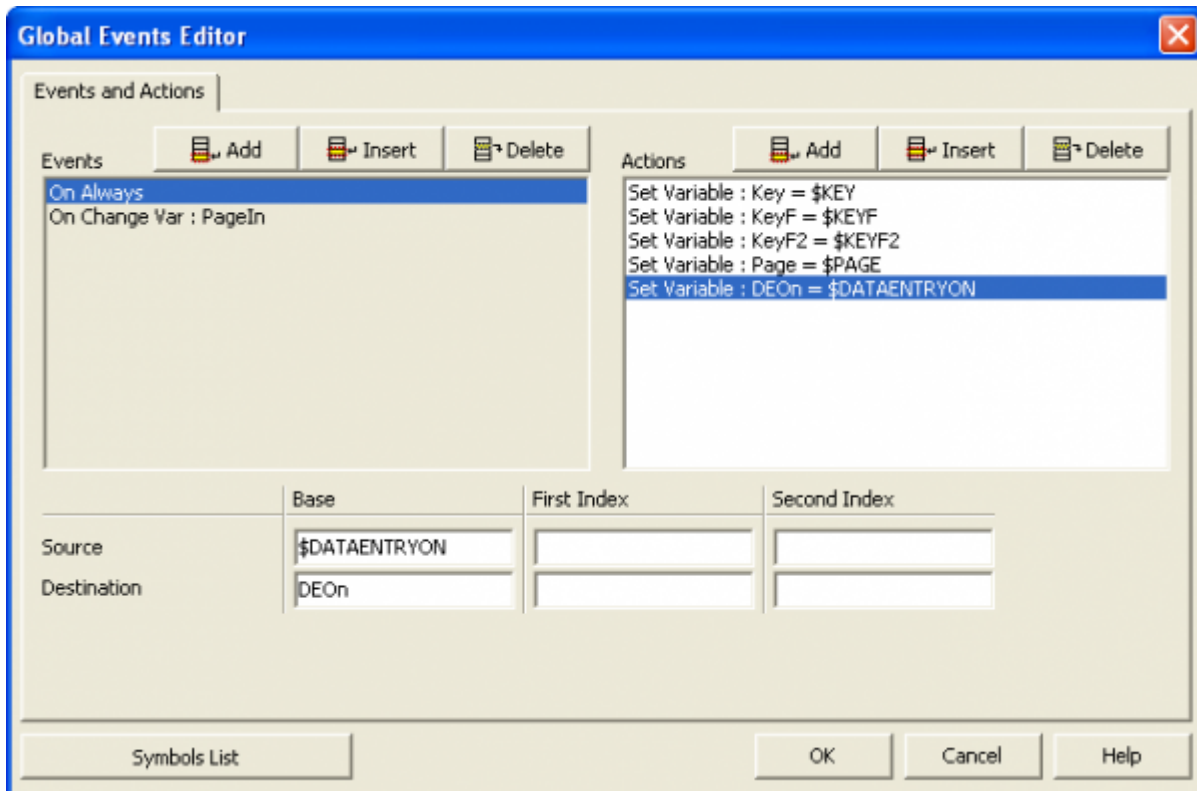
Come si può osservare, viene aggiunta una istruzione *WAIT* per attendere che effettivamente il cambio di pagina richiesto sia avvenuto prima di procedere con il resto delle linee di codice.

## 1.4 Dataentry attivo

Per conoscere lo stato dell'inserimento di un dato (Dataentry) si deve predisporre un'altra variabile nel progetto QVIEW:

```
GLOBAL
DEOn F ;This will match to $DATAENTRYON
```

aggiungere un ulteriore azione *SetVariable* associata all'evento *OnAlways*:



quindi se non posso cambiare pagine durante l'inserimento del dato, devo cambiare il codice in

```
IF Page EQ MENU_PAGE
  IF (Key EQ KEY_1) AND (DEOn EQ 0)
    PageIn = PAGE_1 ;Change page request: go to "PAGE_1" page
    WAIT (Page EQ PAGE_1) ;Wait the change page has done
  ENDIF
ENDIF
```

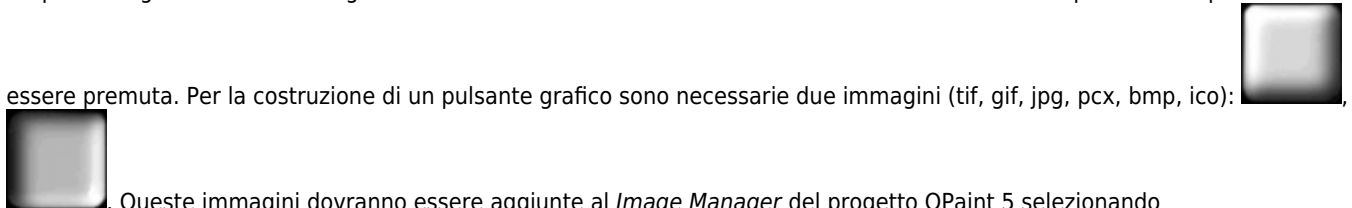
## 2. Esempi

Forniamo degli esempi di programmazione del QPaint 5 per le operazioni più diffuse. Spesso queste operazioni sono ottenute tramite la cooperazione tra la programmazione in QPaint 5 e alcune righe di codice scritte in QView.

- Pulsante grafico.

### 2.1 Pulsante grafico

Un pulsante grafico è una immagine a forma di bottone che una volta toccata modifica la sua forma per dare l'impressione di



essere premuta. Per la costruzione di un pulsante grafico sono necessarie due immagini (tif, gif, jpg, pcx, bmp, ico):

- Grafica - Gestore delle immagini (Graphics - Image Manager)

Per creare il bottone è necessario utilizzare due oggetti:

- un VallImage object e
- un TouchArea object.

Selezionare il VallImage e posizionare sulla pagina del progetto:

**VallImage Object Properties Editor**

Generals | Variables

First Image: 0

Max Images: 2

Image Number: 0

Image Name: butt\_60x60\_off

Coordinates and Sizes

Left: 108

Top: 142

Width: 60

Height: 60


Attributes

☐ Blink Mode

☐ Modify Mode

☐ Reverse Mode

Preview



OK Cancel Help

inserire come *Prima Immagine (First Image)* il valore 0 e come *Numero Immagini (Max Images)* il valore 2. Associare, quindi al valore 0 l'immagine *butt\_60x60\_off* e al valore 1 l'immagine *butt\_60x60\_on*. È necessario quindi associare a questo oggetto una variabile del terminale:

**VallImage Object Properties Editor**

Generals | Variables

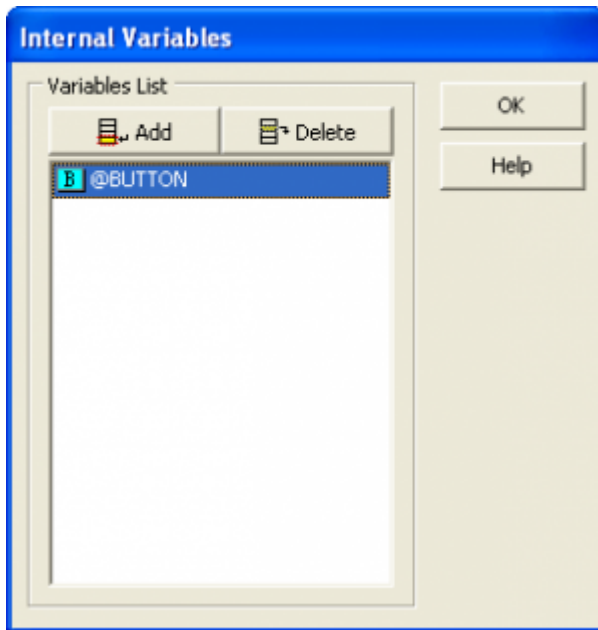
	Base	First Index	Second Index
Variable	@BUTTON		

Symbols List

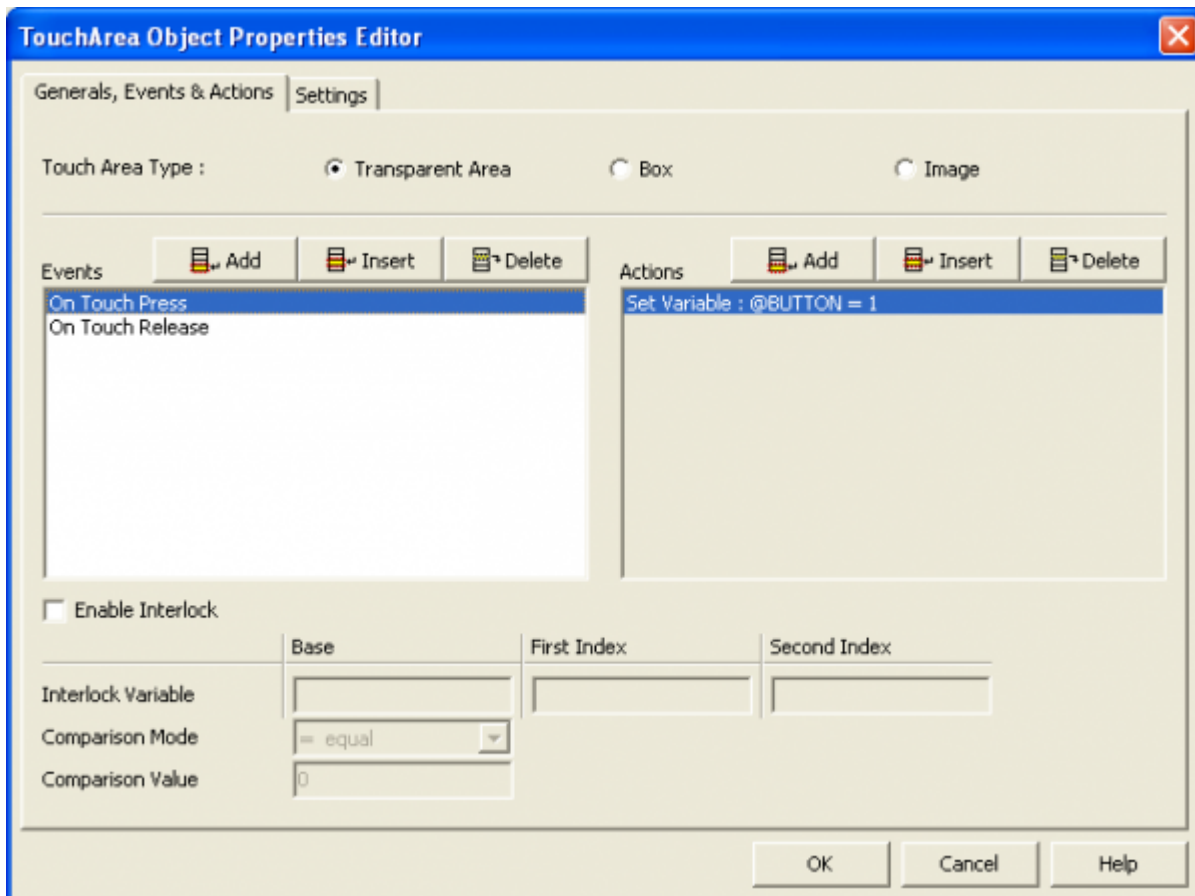
OK Cancel Help

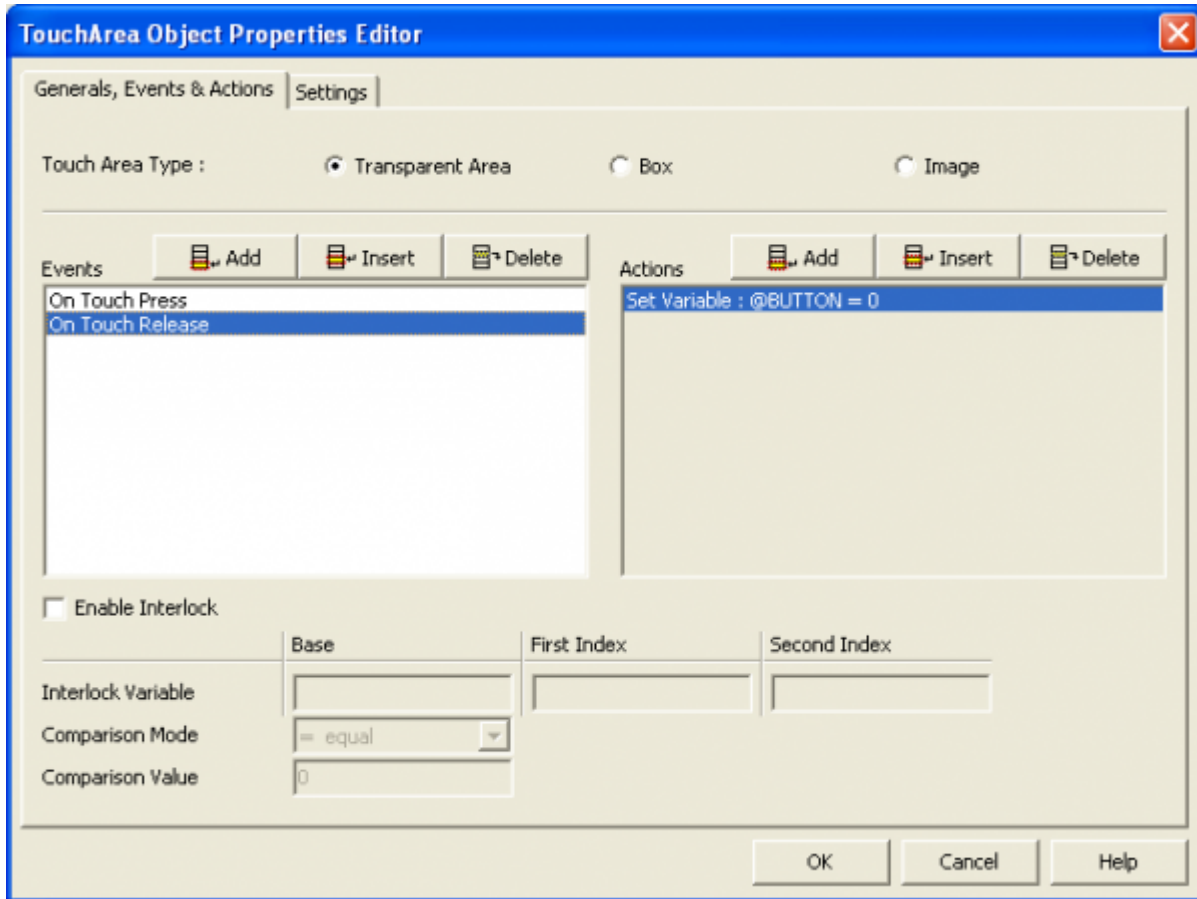
La variabile `@BUTTON` deve essere dichiarata in

- Progetto - Variabili Interne (Project - Internal variables)



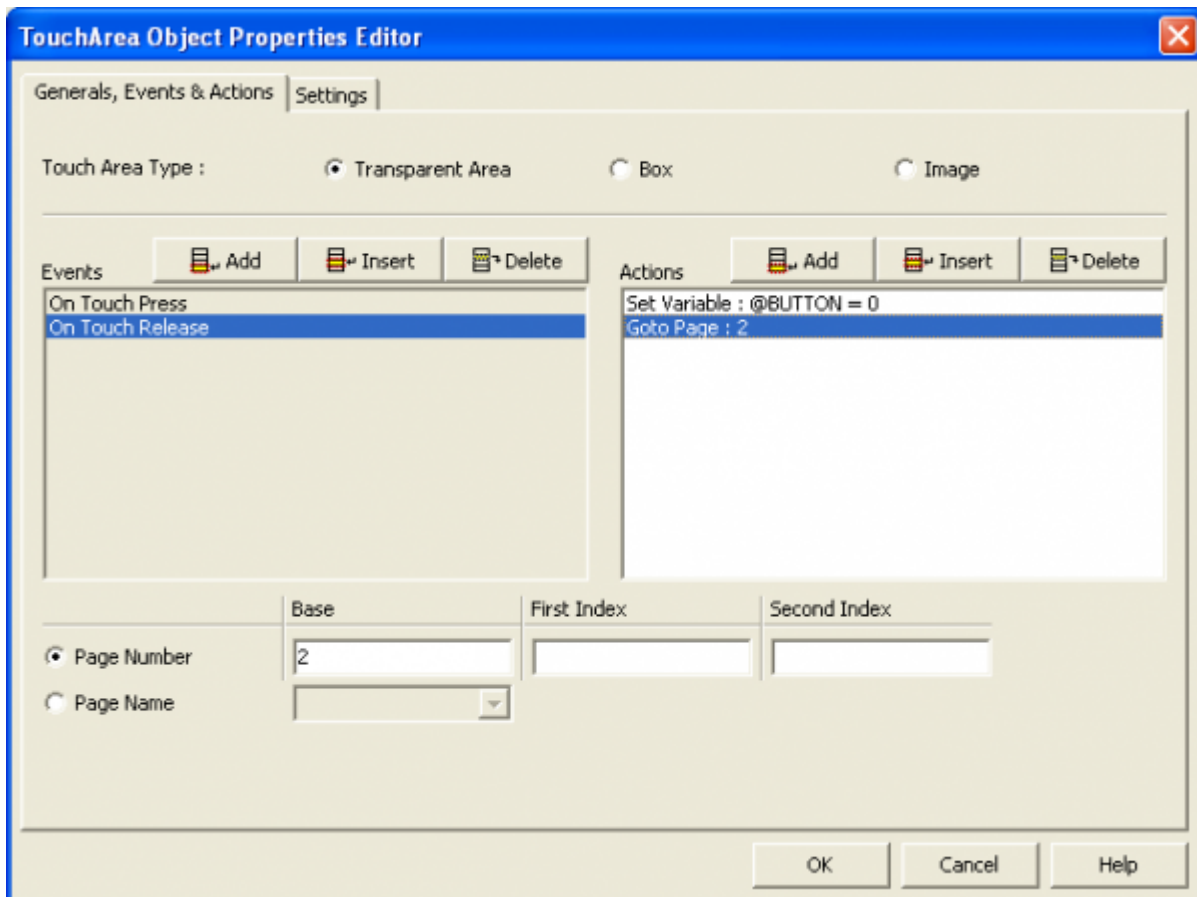
A questo punto è necessario aggiungere l'oggetto TouchArea sopra il VallImage e programmare il tocco e il rilascio in questo modo:



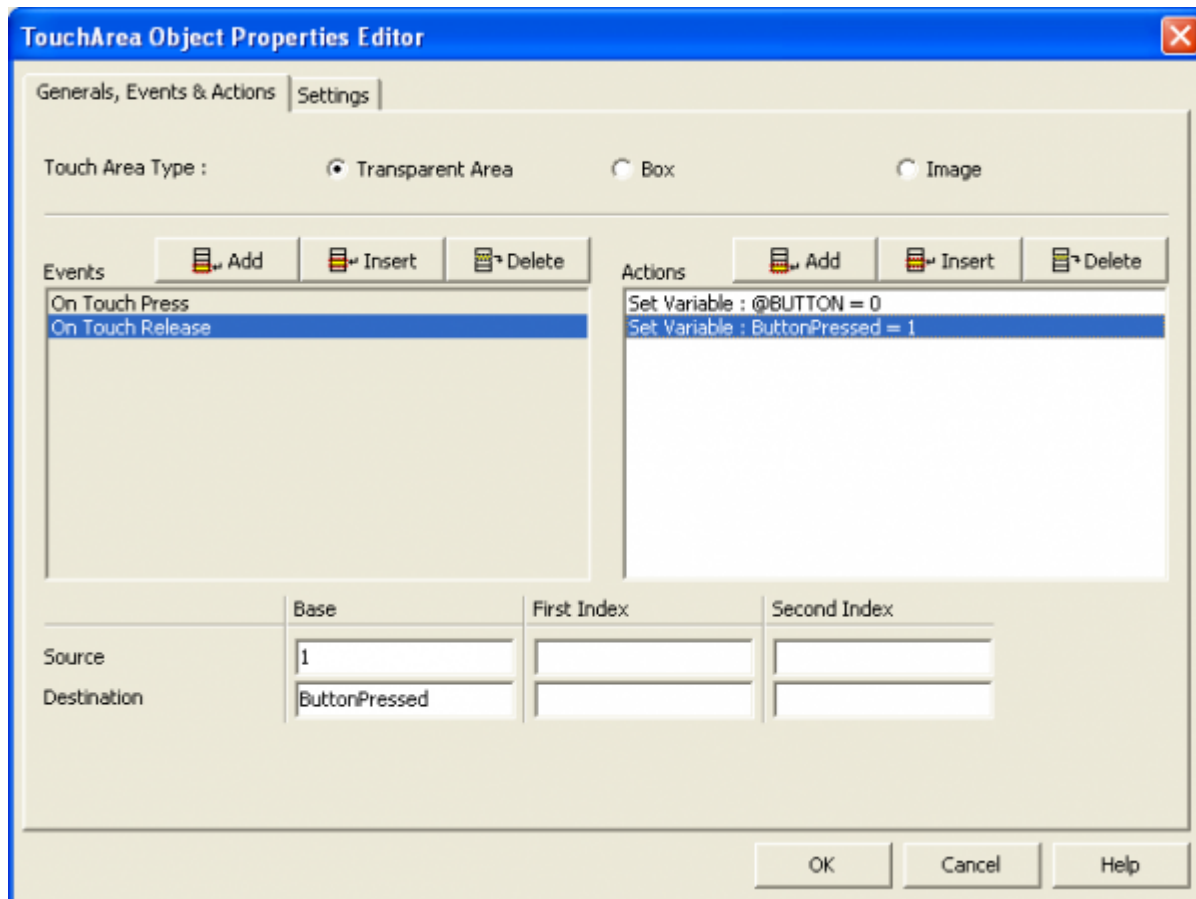


Quindi alla pressione dell'area di tocco viene assegnato il valore 1 alla variabile `@BUTTON` e quindi l'immagine cambierà in `butt_60x60_on`. Al rilascio dell'area di tocco viene assegnato 0 a `@BUTTON` e quindi l'immagine ritorna a `butt_60x60_off`.

A questo punto il pulsante grafico è fatto. Per l'utilizzo di questo pulsante è necessario associare alla pressione e/o al rilascio una azione aggiuntiva che permetta di utilizzare il bottone. Per esempio:



si vede che al rilascio del bottone, viene comandato il cambio pagina con l'azione *Go to page*. Altro utilizzo potrebbe essere quello di assegnare un valore ad una variabile di un progetto QView. Tale variabile poi potrà essere utilizzata nel progetto QView per eseguire altre operazioni.



Documento generato automaticamente da **Qem Wiki** - <https://wiki.qem.it/>

Il contenuto wiki è costantemente aggiornato dal team di sviluppo, è quindi possibile che la versione online contenga informazioni più recenti di questo documento.