



## BADGEREADERX2 CONTROL

Il controllo BADGEREADERX permette un semplice interfacciamento al lettore di badge Nexus da parte di qualsiasi applicazione scritta utilizzando un linguaggio che supporti ActiveX.

Il pacchetto fornito è composto di 3 files:

1. BadgereaderX2Control.ocx: codice binario del controllo
2. BadgereaderX2Control.tlb: type library per lo sviluppo con ambienti che supportano questa tecnologia.
3. BadgereaderX2Control.lic: file di licenza per l'utilizzo a design time.

Vengono inoltre forniti due programmi di esempio, uno scritto con Delphi e l'altro con VB6.

### Utilizzo del componente

Dopo aver effettuato la registrazione necessaria per ogni componente ActiveX ed averne effettuato l'importazione all'interno dell'ambiente di sviluppo utilizzato, BADGEREADERX2 può essere inserito in qualsiasi progetto nelle modalità previste dal tool utilizzato.

Visivamente il componente si presenta come un rettangolo colorato (ogni colore ha un preciso significato, come specificato in seguito), che permette, a design time o tramite codice, di essere dimensionato a piacere e di personalizzarne vari aspetti grafici.

Le **proprietà** salienti per il funzionamento sono:

#### Design time/run time – read/write

- **COM:** **string** indica la porta utilizzata per il collegamento al lettore di badge Nexus. Per il lettore con interfaccia USB questo deve essere sempre impostato a "USB ANY"

#### Run time only – read/write

- **ACTIVE:** **BOOL** permette di attivare/disattivare il colloquio con il dispositivo.

#### Run time only – read only

- **STATUS:** **int** indica lo stato corrente del dispositivo:
- **1** : lettore in corretto colloquio con il PC
- **2** : colloquio non funzionante o Active = False

I colori che può assumere il componente sono i seguenti:

- **Grigio:** Colloquio disattivato (Active = false)
- **Verde:** Colloquio attivo e in corretto funzionamento
- **Rosso:** Colloquio attivo ma non in corretto funzionamento (timeout)

A run time, una serie di **eventi** vengono generati in corrispondenza di variazioni di stato:

- **OnReadedBadge(const tessera: String)**

Lettura corretta di un badge: il parametro **tessera** contiene il valore letto.

- **OnChangedBadgeStatus(newstato: Int)**

Variazione dello stato della tessera nel lettore, il parametro **newstato** può avere i seguenti valori:

- **1** = il lettore non ha tessere inserite
- **2** = una tessera è inserita fino in fondo al lettore
- **3** = una tessera è inserita solo parzialmente nel lettore.

1. **OnBadCRCTrama**

Ricevuta una trama dal lettore con il crc sbagliato, nessun parametro.



- **OnBadCRCData**  
Ricevuta una trama dal lettore con il crc dei dati sbagliato, nessun parametro.
- **OnBadInsertBadge**  
La tessera è stata inserita con il verso sbagliato, nessun parametro
- **OnFailReadBadge**  
Il lettore non è stato in grado di leggere correttamente la tessera.
- **OnTimeout**  
Caduta del colloquio fra il lettore e il pc, nessun parametro.
- **OnRestore**  
Ripristino del colloquio fra il lettore ed il PC, nessun parametro.
- **OnProtocolEvent(const mess: String)**  
Si è verificato un diverso evento relativo al colloquio, il parametro **mess** ne contiene la descrizione.

Altre proprietà ed eventi hanno unicamente lo scopo di gestire l'aspetto visuale del componente, trattandosi di elementi standard, non si ritiene necessario dettagliarne il funzionamento in questo luogo.