

Versione 2.6



(C) 2002-2017 Simone Zanella Productions  
Tutti i diritti riservati.

## **ATTENZIONE – COPIA SERIALIZZATA**

Il pacchetto BCF® è protetto dalle leggi sul diritto d'autore ed il suo utilizzo è soggetto all'approvazione, da parte del cliente, del contratto di licenza che specifica chiaramente l'ambito di utilizzo. **In nessun caso il pacchetto deve essere distribuito o trasmesso a terzi in violazione dei limiti imposti dal contratto di licenza. Poiché ogni copia del prodotto è serializzata, la SZP è sempre in grado a partire da una copia abusiva di risalire al licenziatario originale, contro il quale saranno esercitate tutte le azioni legali previste per la violazione della legge sul diritto d'autore e del contratto di licenza.**

## SOMMARIO

INTRODUZIONE .....	5
INSTALLAZIONE .....	7
UTILIZZO DI BCF ASSISTANT .....	8
DESCRIZIONE DEI FONTS .....	11
SPECIFICITÀ DELLE VARIE SIMBOLOGIE .....	12
Codice 39 .....	12
Codice 93 .....	12
UCC-ITF 14.....	12
Interleaved 2 di 5, parifica 0 .....	12
Industrial 2 di 5.....	13
5 barre 2 di 5 .....	13
3 barre Matrix 2 di 5 .....	13
Codice 11 Matrix.....	13
BCD Matrix 2 di 5.....	13
Codice 2 di 5 Invertito .....	14
Codice 2 di 5 Compresso .....	14
MSI.....	14
Plessey .....	14
Delta-Distance-A .....	14
EAN13.....	14
ISBN .....	15
ISBN 13.....	15
EAN 8.....	15
UPC A .....	15
UPC E .....	16
Add on di 2 cifre per codici EAN/UPC .....	16
Add on di 5 cifre per codici EAN/UPC .....	16
Codice 32 farmaceutico italiano .....	16

<b>Codice C.I.P. (farmaceutico francese).....</b>	<b>17</b>
<b>Codice Codabar/Monarch.....</b>	<b>17</b>
<b>Codice 128 .....</b>	<b>17</b>
<b>EAN 128.....</b>	<b>18</b>
<b>Codice 39 ‘Full Ascii’ e Codice 93 ‘Full Ascii’ .....</b>	<b>18</b>
<b>Codice Postnet (postale americano) .....</b>	<b>19</b>
<b>UTILIZZO CON VISUAL BASIC® .....</b>	<b>21</b>
<b>UTILIZZO CON VISUAL BASIC® .NET .....</b>	<b>24</b>
<b>UTILIZZO CON VISUAL C++® .....</b>	<b>27</b>
<b>UTILIZZO CON VISUAL OBJECTS®.....</b>	<b>28</b>
<b>UTILIZZO CON CRYSTAL REPORTS® .....</b>	<b>29</b>
<b>UTILIZZO CON CRYSTAL REPORTS® .NET .....</b>	<b>30</b>
<b>UTILIZZO CON ALTRI LINGUAGGI .....</b>	<b>31</b>
<b>UTILIZZO CON MICROSOFT WORD®.....</b>	<b>32</b>
<b>UTILIZZO CON MICROSOFT ACCESS® .....</b>	<b>34</b>
<b>UTILIZZO CON MICROSOFT EXCEL® .....</b>	<b>35</b>
<b>LICENZA D’USO.....</b>	<b>37</b>

## Introduzione

BCF® è un pacchetto costituito da una collezione di tipi di carattere per la produzione di codici a barre ed una libreria a collegamento dinamico (DLL), necessaria per il calcolo della transcodifica da testo a barcode. Come strumento aggiuntivo viene fornito BCF Assistant, un applicativo che permette di creare codici a barre nelle simbologie supportate in modo semplice e immediato, per poter incollare poi quanto prodotto all'interno di altri programmi.

I font inclusi, in standard True Type™ per Windows, permettono di realizzare barcode nelle seguenti simbologie (il numero effettivo di simbologie disponibili dipende dal tipo di licenza acquistato):

- Codice 39
- Codice 39 Full Ascii
- Codice 93
- Codice 93 Full Ascii
- ITF-14
- Interleaved 2/5
- Industrial 2/5
- 5 Barre 2/5
- 3 Barre 2/5
- BCD Matrix 2/5
- Codice 11 Matrix
- 2/5 Invertito
- 2/5 Compresso
- MSI
- Plessey
- Delta-Distance-A
- EAN-13 (con e senza add-on di 2 e 5 cifre)
- EAN-8 (con e senza add-on di 2 e 5 cifre)
- UPC-A (con e senza add-on di 2 e 5 cifre)
- UPC-E (con e senza add-on di 2 e 5 cifre)
- Codice 32 (con e senza scritta leggibile in tipo OCR)
- Codice C.I.P.
- Codabar/Monarch
- Codice 128
- EAN 128
- Postnet
- ISBN
- ISBN 13

I codici a barre possono essere utilizzati in qualsiasi pacchetto Windows che supporta la possibilità di selezionare il tipo di carattere e le dimensioni del testo. I codici prodotti seguendo quanto riportato in questo manuale possono essere letti con i tradizionali strumenti di acquisizione, quali penne ottiche, lettori CCD e scanner laser.

La DLL fornita è necessaria per produrre codici leggibili in quanto non è possibile semplicemente applicare il tipo di carattere all'informazione da codificare. Infatti, un barcode è sempre contraddistinto da:

- barre di inizio

- corpo del codice
- eventuale carattere di controllo
- barre di fine

Le barre di inizio e fine sono dei simboli speciali che istruiscono il decodificatore sul tipo di simbologia con la quale il codice a barre è rappresentato; non contengono informazione significativa per l'utente e sono normalmente mappate nei font del pacchetto sui caratteri “\*” (start) e “#” (stop).

Il corpo del codice, nelle simbologie più semplici, corrisponde più o meno direttamente all'informazione da codificare; alcune simbologie, tuttavia, come l'Interleaved 2/5, la famiglia EAN/UPC ed i codici 128 fanno eccezione, in quanto ottimizzano il contenuto informativo con speciali strategie, per poter utilizzare gli spazi tra le barre oltre che le barre stesse.

Il carattere di controllo, obbligatorio per alcune simbologie (come il codice 128 e la famiglia EAN), raccomandato per altre (Interleaved 2/5) oppure facoltativo (es. codice 39), viene calcolato con un algoritmo matematico sulla base del contenuto da codificare e fornisce al decodificatore un'informazione addizionale sulla base della quale è possibile stabilire se il contenuto è stato interpretato correttamente, con un elevato grado di sicurezza.

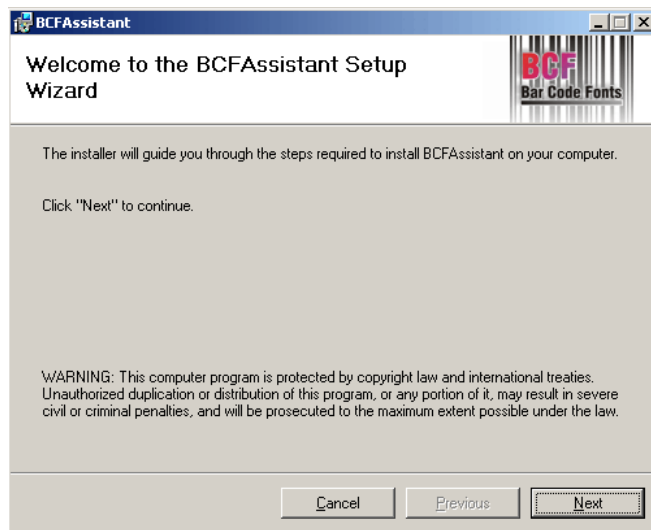
Ogni simbologia è contraddistinta da alcune peculiarità, tra cui:

- alfabeto codificabile – tutte le simbologie ammettono le cifre 0..9, alcune permettono di includere caratteri dell'alfabeto e caratteri speciali, altre infine contemplano l'intero set di caratteri ASCII;
- lunghezza – fissa o variabile, limitata solo dalle dimensioni del simbolo risultante;
- compattezza – lo stesso contenuto, espresso in simbologie diverse, produce simboli di lunghezze diverse; inoltre alcune simbologie, a parità di lunghezza del testo da codificare, producono codici di lunghezza diversa in quanto dipendenti anche dal contenuto e dalla sua disposizione (es. codice 128);
- sicurezza – alcune simbologie devono essere accompagnate dal carattere di controllo per essere lette con sicurezza (es. Interleaved 2/5); le simbologie più complesse prevedono barre di diversi spessori, per cui è necessario che gli strumenti di stampa e di lettura siano in grado di gestirli correttamente.

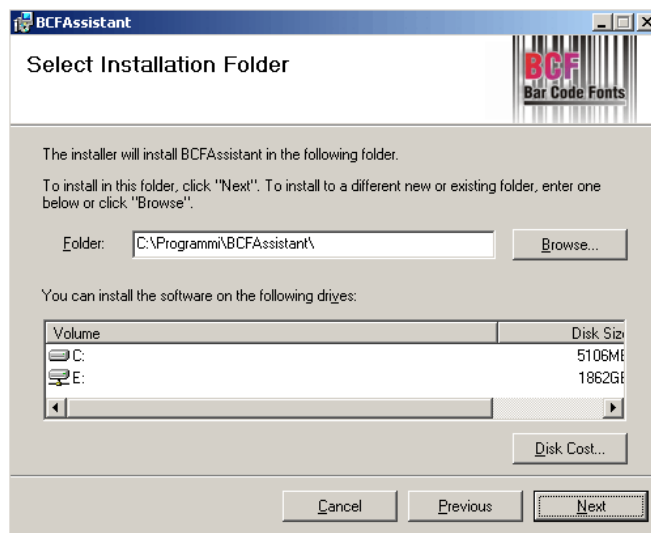
In ogni caso, si tenga sempre presente che il campo di applicazione del codice vincola spesso la scelta del tipo di simbologia.

## Installazione

Per installare il pacchetto, inserire il cd-rom nel lettore del proprio PC, equipaggiato con sistema operativo Windows. Nel caso in cui non parta in automatico il programma di installazione, aprire da “Gestione Risorse” il disco corrispondente al cd-rom e fare doppio click su index.html, seguendo successivamente il collegamento per BCF Assistant®.



Premere “Next”, leggere e confermare la licenza (pulsante “I Agree”) e premere ancora “Next”.



Nella schermata successiva, premere il pulsante “Browse” per modificare il percorso di installazione; premere il pulsante “Next” per confermare e ancora “Next” per avviare l’installazione nella directory indicata.

Il pacchetto provvede ad installare le DLL a 32 bit necessarie, oltre a BCF Assistant e a questo manuale.

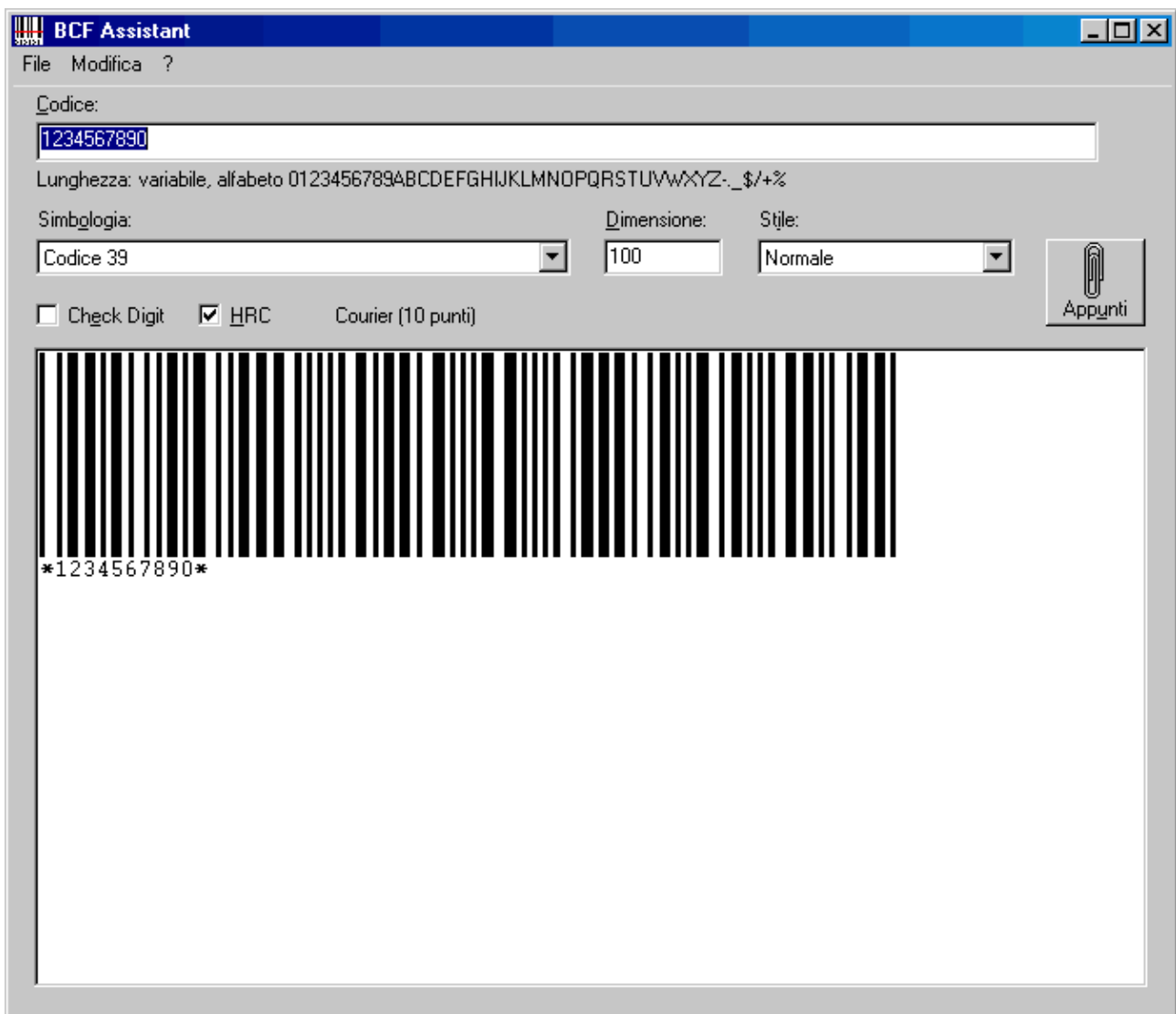
I font devono essere installati manualmente, attraverso il pannello di controllo di Windows, icona "Tipi di carattere". Selezionare "File, Installa nuovo tipo di carattere". Specificare la lettera relativa al cd-rom nel quale si trova il cd e la cartella TTF; premere il pulsante "Seleziona tutto" e confermare con OK.

La DLL a 64 bit BCFDLL64.dll va copiata manualmente in C:\Windows\Sysnative sui sistemi operativi a 64 bit (e qualora si debbano invocare le funzioni della DLL da applicativi a 64 bit).

## Utilizzo di BCF Assistant

BCF Assistant è un applicativo che permette di creare codici a barre nelle simbologie supportate in modo semplice e immediato, per poter incollare poi quanto prodotto all'interno di altri programmi. Al termine dell'installazione, premere il pulsante Start/Avvio di Windows e selezionare dalla lista dei programmi "BCF Assistant".

La figura seguente mostra l'interfaccia del programma.

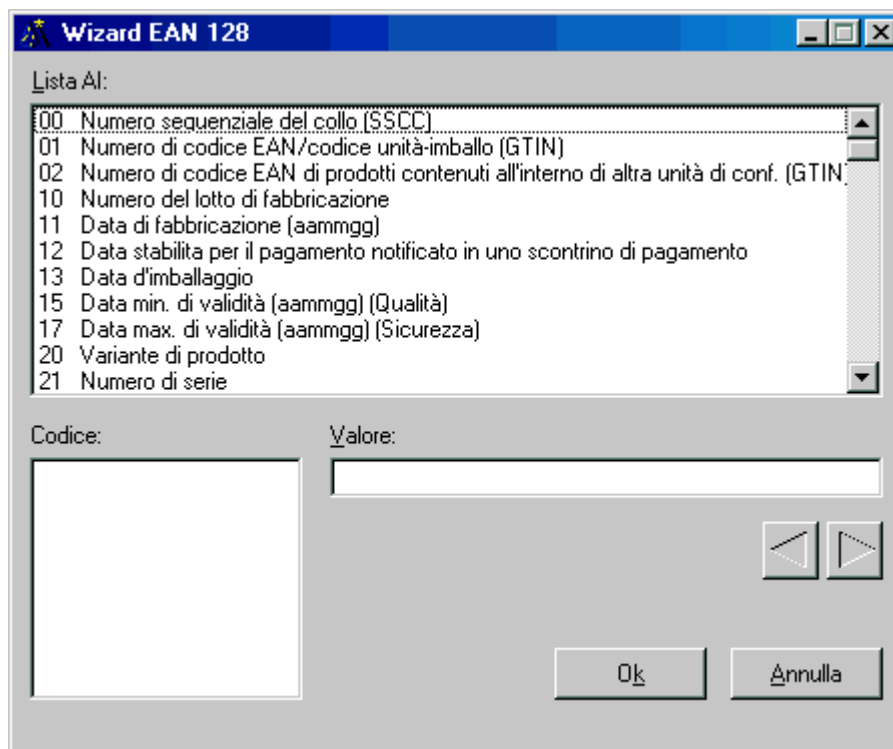


Il funzionamento è molto semplice.

Nel campo "Codice" si digita l'informazione da inserire nel barcode; sotto il campo compare una descrizione che specifica i caratteri ammessi e la lunghezza massima. Premendo Invio, compare nel grande riquadro sottostante il codice a barre corrispondente, purché le dimensioni lo consentano. Qualora il testo contenuto non sia codificabile, comparirà un messaggio. Per la sola simbologia EAN 128, alla destra del campo compare il simbolo di una bacchetta magica; clickando su questo simbolo si aprirà la finestra per la composizione assistita del codice (wizard).



Scegliere dalla lista degli AI (Application Identifier) il tipo di campo da introdurre; digitare nella casella “Valore” il contenuto, seguendo le indicazioni sottostanti riguardo alla lunghezza e al tipo di dato richiesto. Premere il pulsante con la freccia verso destra per introdurre il campo; usare il pulsante con la freccia a sinistra per tornare indietro. Premere “Ok” per accettare il codice EAN-128 così composto.



Il campo “Simbologia” permette di scegliere il tipo di barcode da creare; alcune simbologie limitano le altre impostazioni (HRC, check digit).

Il campo “Dimensione” specifica il numero di punti del carattere barcode da creare.

Lo “Stile” cambia in base alla simbologia; “normale” e “corto” fanno riferimento all’altezza del barcode (la versione “corta” è alta la metà della normale); “stretto” e “largo” fanno riferimento alla larghezza, essendo la versione stretta più compatta e la versione larga leggermente “oversized”.

La casella “Check Digit” abilita/disabilita il carattere di controllo; per alcune simbologie, questa casella non è selezionabile. Ovviamente, il check digit abilitato comporta una maggiore lunghezza del barcode risultante (se la simbologia prevede una lunghezza variabile).

La casella “HRC” (Human Readable Code) abilita/disabilita la stampa del testo in caratteri leggibili sotto il barcode. Il tipo di carattere utilizzato si seleziona attraverso l’opzione del menu “Modifica, Font HRC” e compare a destra della casella di selezione. Alcune simbologie (come EAN/UPC) hanno sempre il testo in caratteri leggibili; altre, come il Code 32, hanno un proprio font e non rispettano le impostazioni selezionate dall’utente.

Premere il pulsante “Appunti” per copiare nella clipboard di Windows il contenuto della casella di anteprima. Dalle applicazioni che lo consentono, utilizzare “Modifica, Incolla” oppure premere Ctrl + V per inserire il contenuto degli appunti nel documento.



## Descrizione dei fonts

La tabella seguente riassume la lista dei font disponibili nel pacchetto (il numero effettivo dipende dal tipo di licenza acquistato):

Nome file	Nome font	Descrizione
szp39__.ttf	SZP39	Codice 39 normale
szp39n__.ttf	SZP39 Narrow	Codice 39 stretto
szp39w__.ttf	SZP39 Wide	Codice 39 largo
szp39s__.ttf	SZP39 Short	Codice 39 corto
szp39ns_.ttf	SZP39 Narrow Short	Codice 39 stretto e corto
szp39ws_.ttf	SZP39 Wide Short	Codice 39 largo e corto
szp93__.ttf	SZP93	Codice 93 normale
szp93n__.ttf	SZP93 Narrow	Codice 93 stretto
szp93w__.ttf	SZP93 Wide	Codice 93 largo
szp93s__.ttf	SZP93 Short	Codice 93 corto
szp93ns_.ttf	SZP93 Narrow Short	Codice 93 stretto e corto
szp93ws_.ttf	SZP93 Wide Short	Codice 93 largo e corto
szp25__.ttf	SZP25 ITF	Interleaved 2/5 normale
szp25s__.ttf	SZP25 ITF Short	Interleaved 2/5 corto
szpind25.ttf	SZPIND25	Industrial 2/5 normale
szpin25s.ttf	SZPIND25 Short	Industrial 2/5 corto
szp5bar_.ttf	SZP5BARS	5 barre 2/5 normale
szp5bars.ttf	SZP5BARS Short	5 barre 2/5 corto
szp3bar_.ttf	SZP3BARS	3 barre 2/5 normale
szp3bars.ttf	SZP3BARS Short	3 barre 2/5 corto
szpbcd_.ttf	SZPBCD	BCD Matrix 2/5 normale
szpbcds_.ttf	SZPBCD Short	BCD Matrix 2/5 corto
szpmsi_.ttf	SZPMSI	MSI normale
szpmsis_.ttf	SZPMSI Short	MSI corto
szp32__.ttf	SZP32	Codice 32 (incluso HRC)
szpcbar_.ttf	SZPCODABAR	Codabar/Monarch normale
szpcbars.ttf	SZPCODABAR Short	Codabar/Monarch corto
szpean_.ttf	SZPEAN	EAN/UPC/ADD-ON/ISBN normale
szpeans_.ttf	SZPEAN Short	EAN/UPC/ADD-ON/ISBN corto
szp128__.ttf	SZP128	Codice 128/EAN 128 normale
szp128s_.ttf	SZP128 Short	Codice 128/EAN 128 corto
szp11__.ttf	SZP11	Codice 11 Matrix normale
szp11s_.ttf	SZP11 Short	Codice 11 Matrix corto
szp25com.ttf	SZP25 COMP	Codice 2/5 Compresso normale
szp25cos.ttf	SZP25 COMP Short	Codice 2/5 Compresso corto
szp25inv.ttf	SZP25 INV	Codice 2/5 Invertito normale
szp25ins.ttf	SZP25 INV Short	Codice 2/5 Invertito corto
szpdelta.ttf	SZPDELTA DA	Codice Delta-Distance-A normale
szpdelts.ttf	SZPDELTA DA Short	Codice Delta-Distance-A corto
szpitf14.ttf	SZPITF14	Codice ITF-14 normale
szpitfs_.ttf	SZPITF14 Short	Codice ITF-14 corto
szpplesy.ttf	SZPPLESSEY	Codice Plessey normale

szppless.ttf	SZPPLESSEY Short	Codice Plessey corto
szppost.ttf	SZPPOSTNET	Codice Postnet

## Specificità delle varie simbologie

Descriviamo brevemente di seguito le varie simbologie supportate dal pacchetto.

### **Codice 39**

È una delle simbologie più diffuse, grazie alla sua versatilità e alla sicurezza della codifica, sebbene la densità di informazione non sia elevata (i simboli risultanti sono abbastanza grandi).

Le caratteristiche salienti sono:

- codice a lunghezza variabile;
- alfabeto: 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ-.\_\$/+% (lo spazio deve essere specificato con il carattere di sottolineatura ‘\_’);
- di norma il Codice 39 è rappresentato **senza** carattere di controllo, che è appunto opzionale.

La DLL converte automaticamente in maiuscolo eventuali caratteri minuscoli prima di procedere con la codifica.

### **Codice 93**

È una simbologia molto versatile, sviluppata come successore del Codice 39, del quale mantiene lo stesso alfabeto, pur essendo più compatta.

Le caratteristiche salienti sono:

- codice a lunghezza variabile;
- alfabeto: 0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ-.\_\$/+% (lo spazio deve essere specificato con il carattere di sottolineatura ‘\_’);
- di norma il codice 93 è rappresentato **con** carattere di controllo.

La DLL converte automaticamente in maiuscolo eventuali caratteri minuscoli prima di procedere con la codifica.

### **UCC-ITF 14**

È una simbologia molto diffusa e compatta, esclusivamente numerica, di lunghezza fissa pari a 14 cifre. È immediatamente riconoscibile in quanto il codice è racchiuso da una cornice nera sui quattro lati. È molto utilizzata per il tracciamento merci in transito.

Le caratteristiche salienti sono:

- lunghezza 13 (se con check digit) oppure 14 cifre; ; se la stringa da codificare ha lunghezza insufficiente, viene corretta dalla DLL con l'aggiunta di "0" a sinistra;
- sono ammesse solo le cifre da 0 a 9;
- il check digit è **obbligatorio**.

### **Interleaved 2 di 5, parifica 0**

È una simbologia molto diffusa e compatta, esclusivamente numerica. Poiché le cifre sono codificate a coppie, devono essere in numero pari se non viene incluso il carattere di controllo, dispari altrimenti.

Le caratteristiche salienti sono:

- lunghezza variabile;

- sono ammesse solo le cifre da 0 a 9;
- il check digit è **raccomandato**, in quanto altrimenti il decodificatore può interpretare in modo parziale il codice.

La DLL aggiunge automaticamente uno “0” a sinistra, se necessario, per ottenere lunghezza pari del testo da codificare.

### ***Industrial 2 di 5***

È una simbologia che sfrutta la stessa codifica dell’Interleaved 2/5, usando esclusivamente le barre (e non gli spazi). In questo modo, la sicurezza della decodifica è maggiore, sebbene crescano anche le dimensioni del simbolo risultante.

Le caratteristiche salienti sono:

- lunghezza variabile;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è opzionale.

### ***5 barre 2 di 5***

È una simbologia utilizzata prevalentemente nei laboratori di sviluppo fotografici. È esclusivamente numerica, a bassa densità d’informazione.

Le caratteristiche salienti sono:

- lunghezza variabile;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è opzionale.

### ***3 barre Matrix 2 di 5***

È una simbologia esclusivamente numerica, a media densità d’informazione.

Le caratteristiche salienti sono:

- lunghezza variabile;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è opzionale.

### ***Codice 11 Matrix***

È una simbologia che consente la codifica di 11 simboli, a media densità d’informazione.

Le caratteristiche salienti sono:

- lunghezza variabile;
- sono ammesse solo le cifre da 0 a 9 ed il carattere “-”;
- il carattere di controllo è **opzionale**; se generato automaticamente, fino a 10 caratteri viene aggiunto un solo check digit, per codici più lunghi i caratteri di controllo divengono due.

### ***BCD Matrix 2 di 5***

È una simbologia esclusivamente numerica, a media densità d’informazione.

Le caratteristiche salienti sono:

- lunghezza variabile;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è opzionale.

### **Codice 2 di 5 Invertito**

È una simbologia esclusivamente numerica, a bassa densità d'informazione; veniva utilizzata per la sua possibilità di essere riprodotta con stampanti a catena; oggi è obsoleta.

Le caratteristiche salienti sono:

- lunghezza variabile;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **opzionale**.

### **Codice 2 di 5 Compresso**

È una simbologia esclusivamente numerica, ad alta densità d'informazione, basata sul 2/5 3 barre Matrix.

Le caratteristiche salienti sono:

- lunghezza variabile;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **opzionale**.

### **MSI**

È una simbologia esclusivamente numerica, a bassa densità d'informazione.

Le caratteristiche salienti sono:

- lunghezza variabile, massimo 15 cifre;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo (doppio) è **raccomandato**.

### **Plessey**

È una simbologia alfanumerica, a bassa densità d'informazione.

Le caratteristiche salienti sono:

- lunghezza variabile, massimo 15 caratteri;
- sono ammesse solo le cifre da 0 a 9 ed i caratteri A, B, C, D, E, F;
- il carattere di controllo (doppio) è **obbligatorio**.

### **Delta-Distance-A**

È una simbologia alfanumerica, a bassa densità d'informazione.

Le caratteristiche salienti sono:

- lunghezza variabile, massimo 15 (se con check digit) o 16 caratteri;
- sono ammesse solo le cifre da 0 a 9 ed i caratteri K, L, M, O;
- il carattere di controllo è **raccomandato**.

### **EAN13**

È una simbologia esclusivamente numerica, ad alta densità d'informazione, che viene utilizzata per l'etichettatura di beni al consumo distribuiti nell'ambito dell'Unione Europea.

Le caratteristiche salienti sono:

- lunghezza 12 (se con carattere di controllo) oppure 13 cifre; se la stringa da codificare ha lunghezza insufficiente, viene corretta dalla DLL con l'aggiunta di "0" a sinistra;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **obbligatorio**.

### **ISBN**

È una simbologia esclusivamente numerica, ad alta densità d'informazione, che viene utilizzata per l'etichettatura di libri. Consiste in un codice EAN13 sopra il quale appare la dicitura "ISBN" seguita dal codice (completo di check digit) opportunamente sillabato attraverso l'introduzione del carattere "-".

Le caratteristiche salienti sono:

- lunghezza 9 caratteri; se la stringa da codificare ha lunghezza insufficiente, viene corretta dalla DLL con l'aggiunta di "0" a sinistra;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **obbligatorio** e viene aggiunto automaticamente; tale carattere può assumere anche il valore "X";
- la DLL restituisce due righe, corrispondenti alla scritta sopra al codice ed al barcode; ad entrambe va applicato il font EAN per ottenere il simbolo corretto.

### **ISBN 13**

È una simbologia esclusivamente numerica, ad alta densità d'informazione, che viene utilizzata per l'etichettatura di libri. Consiste in un codice EAN13 sopra il quale appare la dicitura "ISBN" seguita dal codice (completo di check digit) opportunamente sillabato attraverso l'introduzione del carattere "-".

Le caratteristiche salienti sono:

- lunghezza 12 caratteri; se la stringa da codificare ha lunghezza insufficiente, viene corretta dalla DLL con l'aggiunta di "0" a sinistra;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **obbligatorio** e viene aggiunto automaticamente;
- la DLL restituisce due righe, corrispondenti alla scritta sopra al codice ed al barcode; ad entrambe va applicato il font EAN per ottenere il simbolo corretto.

### **EAN 8**

È una simbologia esclusivamente numerica, ad alta densità d'informazione, che viene utilizzata per l'etichettatura di beni al consumo distribuiti nell'ambito dell'Unione Europea; viene utilizzata qualora le dimensioni dell'oggetto non permettano l'adozione dell'EAN13.

Le caratteristiche salienti sono:

- lunghezza 7 (se con carattere di controllo) oppure 8 cifre; se la stringa da codificare ha lunghezza insufficiente, viene corretta dalla DLL con l'aggiunta di "0" a sinistra;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **obbligatorio**.

### **UPC A**

È una simbologia esclusivamente numerica, ad alta densità d'informazione, che viene utilizzata per l'etichettatura di beni al consumo distribuiti nell'ambito degli Stati Uniti.

Le caratteristiche salienti sono:

- lunghezza 11 (se con\_carattere di controllo) oppure 12 cifre; se la stringa da codificare ha lunghezza insufficiente, viene corretta dalla DLL con l'aggiunta di "0" a sinistra;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **obbligatorio**.

### **UPC E**

È una simbologia esclusivamente numerica, ad alta densità d'informazione, che viene utilizzata per l'etichettatura di beni al consumo distribuiti nell'ambito degli Stati Uniti; viene utilizzata qualora le dimensioni dell'oggetto non permettano l'adozione dell'UPC-A.

Le caratteristiche salienti sono:

- lunghezza 7 (se con\_carattere di controllo) oppure 8 cifre; se la stringa da codificare ha lunghezza insufficiente, viene corretta dalla DLL con l'aggiunta di "0" a sinistra;
- sono ammesse solo le cifre da 0 a 9;
- deve iniziare **necessariamente** per 0 o 1;
- il carattere di controllo è **obbligatorio**.

### **Add on di 2 cifre per codici EAN/UPC**

Le simbologie EAN/UPC possono essere espanse attraverso un codice aggiuntivo di 2 cifre posto a 5 moduli di distanza dalle barre destre del simbolo. L'Add-on di 2 cifre viene utilizzato molto in ambito editoriale, ad esempio sui mensili.

Le caratteristiche salienti sono:

- lunghezza 2;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è sempre calcolato automaticamente, indipendentemente da quanto richiesto dall'utente;
- la stringa di transcodifica inizia già con un carattere che corrisponde allo spazio di separazione tra un codice EAN/UPC e l'inizio dell'Add-on, per cui è sufficiente sommare questa stringa alla precedente ottenuta per il codice EAN/UPC per ottenere un codice completo correttamente formattato.

### **Add on di 5 cifre per codici EAN/UPC**

Le simbologie EAN/UPC possono essere espanse attraverso un codice aggiuntivo di 5 cifre posto a 5 moduli di distanza dalle barre destre del simbolo. L'Add-on di 5 cifre viene utilizzato molto in ambito editoriale, ad esempio sui quotidiani.

Le caratteristiche salienti sono:

- lunghezza 5;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è sempre calcolato automaticamente, indipendentemente da quanto richiesto dall'utente;
- la stringa di transcodifica inizia già con un carattere che corrisponde allo spazio di separazione tra un codice EAN/UPC e l'inizio dell'Add-on, per cui è sufficiente sommare questa stringa alla precedente ottenuta per il codice EAN/UPC per ottenere un codice completo correttamente formattato.

### **Codice 32 farmaceutico italiano**



È una simbologia studiata appositamente per la generazione di codici da imprimere sulle confezioni delle specialità medicinali in Italia, conformemente a quanto richiesto dal D.L. del 10 giugno 1983. In pratica, si tratta di un sistema di codifica basato sul Codice 39.

Le caratteristiche salienti sono:

- lunghezza 8 (se con carattere di controllo) oppure 9;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **obbligatorio**;
- se viene richiesta la transcodifica con HRC, viene aggiunta la scritta sottostante contenente il codice in caratteri leggibili (prefissato da "A").

### **Codice C.I.P. (farmaceutico francese)**

È una simbologia utilizzata in Francia dal "Club Inter-Pharmaceutique".

In pratica, si tratta di un sistema di codifica basato sul Codice 39.

Le caratteristiche salienti sono:

- lunghezza 6 (se con carattere di controllo) oppure 7;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **obbligatorio**;
- dato il particolare algoritmo di calcolo del check digit (modulo 11), **non tutte le sequenze di 6 cifre sono codificabili** - devono essere scartate tutte le sequenze che comportano il modulo del checksum pari a 10.

### **Codice Codabar/Monarch**

È una simbologia che comprende le dieci cifre, sei caratteri aggiuntivi e quattro caratteri speciali di start/stop. È molto utilizzata soprattutto in ambito medico.

Le caratteristiche salienti sono:

- lunghezza variabile, maggiore o uguale a 3;
- deve iniziare e finire per un carattere incluso in "ABCDEMT\*";
- deve essere costituito, per il rimanente, da caratteri inclusi in "0123456789-\$/./+";
- il carattere di controllo è facoltativo.

### **Codice 128**

Questa simbologia deve il suo nome alla possibilità di poter codificare tutti i 128 caratteri del set ASCII; ha un'alta densità di informazione ed affidabilità. La densità è accentuata dal fatto che le cifre numeriche sono codificate a coppie.

Le caratteristiche salienti sono:

- lunghezza variabile;
- permette la codifica dei caratteri con codice ASCII compreso tra 0 e 127;
- ammette i seguenti caratteri addizionali:
  - 128 = NUL
  - 129 = FNC1
  - 130 = FNC2
  - 131 = FNC3
  - 132 = FNC4
- per rappresentare i caratteri non stampabili, si deve utilizzare la notazione C-like:
  - \a o \A = 7
  - \b o \B = 8
  - \f o \F = 12
  - \n o \N = 10

\r o \R = 13

\t o \T = 9

\v o \V = 11

\\ = 92 = "\"

\? = 93 = "?"

\' = 39 = "'"

\" = 34 = doppi apici

\dnnn = chr(nnn), dove nnn è un numero decimale compreso tra 000 e 255

\ooo = carattere corrispondente al numero ooo espresso in base ottale (cifre 0..7), da 000 a 377

\xhh = carattere corrispondente al numero hh espresso in base esadecimale (0..9, A..F), da 00 a FF

Pertanto, se la stringa da codificare contiene il carattere "\" questo andrà raddoppiato per non essere interpretato come l'inizio di una sequenza estesa;

- o il carattere di controllo è obbligatorio ed è sempre aggiunto automaticamente, indipendentemente da quanto richiesto dall'utente.

### ***EAN 128***

Questa simbologia è basata sul Codice 128; in pratica, permette di codificare le informazioni in campi contraddistinti da un codice numerico (AI = Application Identifier), per cui successivamente si possono estrarre in maniera automatica solo le informazioni rilevanti contenute nel simbolo.

Il calcolo degli AI spetta all'utente; si ricorda solamente che qualora il campo seguente all'AI:

- o sia a lunghezza variabile;
- o sia più corto della massima lunghezza ammessa;
- o non si trovi alla fine del codice,

deve essere separato dal successivo AI dal carattere Chr(129) = FNC1 = "\201" = "\x81".

Il carattere di controllo è obbligatorio ed è sempre aggiunto automaticamente, indipendentemente da quanto richiesto dall'utente.

### ***Codice 39 "Full Ascii" e Codice 93 "Full Ascii"***

Queste due simbologie sono basate, rispettivamente, sul Codice 39 e sul Codice 93. Attraverso combinazioni di caratteri, permettono di codificare l'intero set ASCII.

Le caratteristiche salienti sono:

- o lunghezza variabile;
- o permettono la codifica dei caratteri con codice ASCII compreso tra 0 e 127;
- o per rappresentare i caratteri non stampabili, si deve utilizzare la notazione C-like:

\a o \A = 7

\b o \B = 8

\f o \F = 12

\n o \N = 10

\r o \R = 13

\t o \T = 9

\v o \V = 11

\\ = 92 = "\"

\? = 93 = "?"

\' = 39 = "'"

\" = 34 = doppi apici

\dnnn = chr(nnn), dove nnn è un numero decimale compreso tra 000 e 255

\ooo = carattere corrispondente al numero ooo espresso in base ottale (cifre 0..7), da 000 a 377

\xhh = carattere corrispondente al numero hh espresso in base esadecimale (0..9, A..F), da 00 a FF

Pertanto, se la stringa da codificare contiene il carattere "\" questo andrà raddoppiato per non essere interpretato come l'inizio di una sequenza estesa.

### ***Codice Postnet (postale americano)***

È una simbologia utilizzata negli Stati Uniti per la marcatura della posta in transito.

In pratica, si tratta di un sistema di codifica basato su barre di due differenti altezze.

Le caratteristiche salienti sono:

- lunghezza fissa a 5, 9 o 11 cifre;
- sono ammesse solo le cifre da 0 a 9;
- il carattere di controllo è **obbligatorio**.



## Utilizzo con Visual Basic®

BCF rende disponibile una libreria a collegamento dinamico (BCFDLL.DLL) che viene automaticamente copiata nella cartella di sistema all'installazione del pacchetto. Questa libreria serve per poter calcolare la transcodifica del testo a cui applicare il font per ottenere codici a barre leggibili.

BCFDLL espone due funzioni:

- SZPStringToBarcodeDLL, per calcolare la transcodifica;
- SZPValidateStringDLL, per verificare se il testo può essere codificato con una determinata simbologia.

Per poter utilizzare queste funzioni, è sufficiente dichiararle in un modulo di Visual Basic o di VBA, in questo modo (vedi anche la cartella Examples\VisualBasic sul CD):

```
Declare Function SZPStringToBarcodeDLL Lib "BCFDLL" (ByVal codice As String, _  
ByVal simbologia As String, ByVal ConCD As Integer, _  
ByVal barcode As String) As Integer
```

```
Declare Function SZPValidateStringDLL Lib "BCFDLL" (ByVal codice As String, _  
ByVal simbologia As String, ByVal ConCD As Integer) As Boolean
```

La funzione SZPStringToBarcodeDLL scrive nel buffer barcode la transcodifica di codice in base a simbologia e al flag del carattere di controllo (ConCD), restituendo il numero di caratteri copiati in barcode.

La funzione SZPValidateStringDLL verifica se è possibile calcolare la transcodifica di codice in base a simbologia e al flag del carattere di controllo (ConCD), restituendo True se è possibile e False se è impossibile.

Descriviamo i parametri delle due funzioni:

codice

rappresenta il testo da codificare; dovranno essere utilizzate le convenzioni previste dalla simbologia selezionata

simbologia

stringa alfanumerica che specifica quale simbologia utilizzare; le simbologie possibili sono riassunte dalla seguente tabella:

C39	codice 39
C39FULL	codice 39 Full Ascii
C93	codice 93
C93FULL	codice 93 Full Ascii
ITF	UCC-ITF 14
C25	Interleaved 2/5
CIND25	Industrial 2/5
C5BARS	5 Barre 2/5
C3BARS	3 Barre 2/5
CBCD	BCD Matrix 2/5
C11	codice 11 Matrix
C25INV	codice 2/5 Invertito

C25COMP	codice 2/5 Compresso
CMSI	MSI
PLESSEY	Plessey
DELTA	Delta-Distance-A
EAN13	EAN 13
EAN8	EAN 8
UPCA	UPC A
UPCE	UPC E
ADD2	add-on di due cifre per EAN/UPC
ADD5	add-on di cinque cifre per EAN/UPC
C32	codice 32 senza scritta leggibile
C32HRC	codice 32 con scritta leggibile
CIP	codice C.I.P. (farmaceutico francese)
CBAR	codabar/monarch
C128	codice 128
E128	EAN 128
POSTNET	Postnet (codice postale americano)
ISBN	ISBN
ISBN13	ISBN a 13 cifre

ConCD

valore che indica se aggiungere o meno il carattere di controllo specifico per la simbologia; 1 = aggiungi carattere di controllo, 0 = non aggiungi

Barcode (solo per SZPStringToBarcodeDLL)

stringa di lunghezza sufficiente a contenere la transcodifica; suggeriamo di allocare un minimo di 64 bytes, e comunque, detta  $n$  la lunghezza del testo da codificare, un minimo di  $4 * n + 10$  bytes  
 Chiaramente, in base alla simbologia scelta sarà necessario applicare l'opportuno tipo di carattere alla transcodifica per ottenere il barcode cercato, seguendo la tabella riportata nella sezione "Descrizione dei fonts".

*Esempi:*

' Questa funzione calcola la transcodifica con una singola chiamata, restituendola

```
Function CodiceBCF(s As String, symb As String, con_cd As Boolean) As String
```

```
Dim result As String * 255, lung As Integer
```

```
lung = SZPStringToBarcodeDLL(s, symb, IIf(con_cd, 1, 0), result)
```

```
CodiceBCF = Left(result, lung)
```

```
End Function
```

---

' Esempio di chiamata diretta per codificare in EAN13 con check digit il  
 ' codice specificato; è importante notare che vengono specificate 12 cifre,  
 ' in quanto la cifra di controllo è aggiunta automaticamente (flag ConCd = 1)

```
Dim result As String * 255, lung As Integer
```

```
[..]
```

```
lung = SZPStringToBarcodeDLL("123456789012", "EAN13", 1, result)
```

' Left(result, lung) conterrà adesso la transcodifica

---

```
' Questa funzione verifica se è possibile calcolare la transcodifica di s in  
' simbologia symb e con_cd  
  
Function VerificaBCF(s As String, symb As String, con_cd As Boolean) As Boolean  
  
VerificaBCF = SZPValidateStringDLL(s, symb, IIf(con_cd, 1, 0))  
  
End Function
```

La DLL a 64 bit espone le medesime funzioni e prende gli stessi parametri.

## Utilizzo con Visual Basic® .Net

La libreria a collegamento dinamico (BCFDLL.DLL o BCFDLL64.DLL) può essere utilizzata in Visual Basic .Net in modo del tutto analogo a Visual Basic.

Per poter utilizzare le due funzioni esposte, è sufficiente dichiararle in un modulo in questo modo (vedi anche la cartella `Examples\VisualBasicNet` sul CD):

```
Imports System.Text
Imports System.Runtime.InteropServices

Module Module1

    ' DLL a 32 bit
    Public Declare Function SZPStringToBarcodeDLL Lib "BCFDLL" _
        (ByVal codice As String, ByVal simbologia As String, _
        ByVal ConCD As Integer, <MarshalAs(UnmanagedType.LPStr)> _
        ByVal barcode As StringBuilder) As Integer

    Public Declare Function SZPValidateStringDLL Lib "BCFDLL" _
        (ByVal codice As String, ByVal simbologia As String, _
        ByVal ConCD As Integer) As Boolean

    ' DLL a 64 bit
    Public Declare Function SZPStringToBarcodeDLL64 Lib "BCFDLL64" Alias _
        "SZPStringToBarcodeDLL" (ByVal codice As String, _
        ByVal simbologia As String, ByVal ConCD As Integer, _
        <MarshalAs(UnmanagedType.LPStr)> ByVal barcode As StringBuilder) As Integer

    Public Declare Function SZPValidateStringDLL64 Lib "BCFDLL64" Alias _
        "SZPValidateStringDLL" (ByVal codice As String, _
        ByVal simbologia As String, ByVal ConCD As Integer) As Boolean

    [...]
End Module
```

### *Esempi:*

' Questa funzione calcola la transcodifica con una singola chiamata, restituendola

```
Function CodiceBCF(s As String, symb As String, con_cd As Boolean) As String
```

```
Dim result As StringBuilder = New StringBuilder(255)
```

```
Dim lung As Integer, i As Integer
```

```
If con_cd Then
```

```
    i = 1
```

```
Else
```

```
    i = 0
```

```
End If
```

```
' DLL a 32 bit
```

```
lung = SZPStringToBarcodeDLL(s, symb, i, result)
```

```
' DLL a 64 bit
```

```
' lung = SZPStringToBarcodeDLL64(s, symb, i, result)
```

```
CodiceBCF = result.ToString(0, lung)
```

```
End Function
```

' Questa funzione verifica se è possibile calcolare la transcodifica di s in



```
' simbologia symb e con_cd

Function VerificaBCF(s As String, symb As String, con_cd As Boolean) As Boolean

If con_cd Then
    i = 1
Else
    i = 0
End If
' DLL a 32 bit
VerificaBCF = SZPValidateStringDLL(s, symb, i)
' DLL a 64 bit
' VerificaBCF = SZPValidateStringDLL64(s, symb, i)

End Function
```



## Utilizzo con Visual C++<sup>®</sup>

Valgono molte delle considerazioni espone nella sezione “Utilizzo con Visual Basic”; ci limitiamo a riportare le differenze. I prototipi delle funzioni contenute nella DLL sono i seguenti (vedi anche la cartella Examples\VisualCPP sul CD):

```
int APIENTRY SZPStringToBarcodeDLL(char *codice, char *simbologia, int con_cd,
                                   char *barcode);
```

```
int APIENTRY SZPValidateStringDLL(char *newcod, char *simbologia, int con_cd);
```

Per invocare dinamicamente le funzioni è sufficiente caricare la libreria, assegnare ad un puntatore a funzione l'indirizzo della funzione nella libreria ed utilizzarla, come in questo esempio:

```
#include <windows.h>
#include <winbase.h>

typedef int (APIENTRY *BCF1)(char *, char *, int);
typedef int (APIENTRY *BCF2)(char *, char *, int, char *);

int main(int argc, char* argv[])
{
    HINSTANCE hLibrary;
    BCF1 lpFunc1;
    BCF2 lpFunc2;

    hLibrary = LoadLibrary("BCFDLL.DLL");
    if (hLibrary != NULL)
    {
        char result[255];
        int lung;

        lpFunc1 = (BCF1) GetProcAddress(hLibrary, "SZPValidateStringDLL");
        lpFunc2 = (BCF2) GetProcAddress(hLibrary, "SZPStringToBarcodeDLL");

        if (lpFunc1 != NULL)
            if (lpFunc1("123456789012", "EAN13", 1))
            {
                // Transcodifica possibile
                if (lpFunc2 != NULL)
                {
                    lung = lpFunc2("123456789012", "EAN13", 1, result);
                    result[lung] = 0;
                    printf("%s\r\n", result);
                }
            }
        FreeLibrary(hLibrary);
    }
    return 0;
}
```

## Utilizzo con Visual Objects®

Valgono molte delle considerazioni espone nella sezione “Utilizzo con Visual Basic”; ci limitiamo a riportare le differenze. Le funzioni esterne vanno così dichiarate (vedi anche la cartella Examples\VisualObjects sul CD):

```
_DLL FUNCTION SZPStringToBarcodeDLL(codice AS PSZ, simbologia AS PSZ, ConCD AS  
DWORD, barcode AS PSZ) AS DWORD PASCAL:BCFDLL.SZPStringToBarcodeDLL
```

```
_DLL FUNCTION SZPValidateStringDLL(codice AS PSZ, simbologia AS PSZ, ConCD AS  
DWORD) AS DWORD PASCAL:BCFDLL.SZPValidateStringDLL
```

Per invocare dinamicamente le funzioni è sufficiente utilizzarle in questo modo:

```
LOCAL pszBuffer AS PSZ  
LOCAL dwLen AS DWORD  
LOCAL sValue AS STRING  
LOCAL lSuccess AS LOGIC  
  
IF SZPValidateStringDLL(PSZ("12345"), PSZ("C128"), 1)  
    pszBuffer := PSZ(MemAlloc(255))  
    dwLen := SZPStringToBarcodeDLL(PSZ("12345"), PSZ("C128"), 1, pszBuffer)  
    sValue := Left(PSZ2String(pszBuffer), dwLen)  
ENDIF
```

## Utilizzo con Crystal Reports®

Crystal Reports® è un software per la generazione di report molto popolare, sia come applicativo autonomo che come designer in diversi ambienti di sviluppo.

Il modo più semplice per ottenere la necessaria conversione dei dati da rendere in barcode è l'utilizzo della libreria CRUFLbcf.dll, DLL/Active-X installata con BCFAssistant oppure disponibile nella directory Examples\Cystal del CD; la DLL è anche installabile manualmente copiandola in:

```
C:\WINDOWS\SYSTEM (Windows 95/98/ME)
C:\WINNT\SYSTEM32 (Windows NT/2000)
C:\WINDOWS\SYSTEM32 (Windows XP/2003)
```

La DLL deve essere registrata, nel caso di copia del file, digitando la seguente istruzione da un prompt di Dos:

```
REGSVR32 C:\WINNT\SYSTEM32\CRUFLbcf.dll
```

A questo punto, per ottenere un barcode correttamente formattato introdurre il campo da codificare nel report; selezionare "Ometti" nella scheda di formattazione per non farlo stampare; quindi, creare una nuova formula ed utilizzare il seguente codice:

```
Formula = IIF(bcfBarcodeVerify ({Articoli.Codice}, "C128", True),
             bcfBarcodeEncode({Articoli.Codice}, "C128", True), "")
```

Oppure, se è noto che la rappresentazione è sicuramente possibile:

```
Formula = bcfBarcodeEncode({Articoli.Codice}, "C128", True)
```

*N.B. Nell'esempio si assume che il campo da rendere in barcode si chiami Codice, si trovi nella tabella Articoli e che la simbologia sia Code 128 con check digit abilitato*

La formula può quindi essere introdotta come campo nel report, al quale si andrà ad applicare il font di dimensioni opportune per ottenere il barcode della grandezza richiesta.

**Nota bene:** CRUFLbcf.dll utilizza BCFDLL.DLL per tutte le operazioni di codifica; tale DLL deve, pertanto, essere anch'essa installata.

## Utilizzo con Crystal Reports® .Net

Il modo più semplice per ottenere la necessaria conversione dei dati da rendere in barcode è l'utilizzo della libreria CRUFL\_VB\_BCF.dll (disponibile nella directory Examples\CrystalNet), che deve essere registrata aprendo il prompt dei comandi di Visual Studio 2005 ed eseguendo il seguente comando:

```
gacutil -if [percorso]CRUFL_VB_BCF.dll
```

A questo punto, per ottenere un barcode correttamente formattato introdurre il campo da codificare nel report; selezionare "Ometti" nella scheda di formattazione per non farlo stampare; quindi, creare una nuova formula ed utilizzare il seguente codice:

```
Formula = IIF(VBBCFBCFVerify({Articoli.Codice}, "C128", True),  
             VBBCFBCFEncode({Articoli.Codice}, "C128", True), "")
```

Oppure, se è noto che la rappresentazione è sicuramente possibile:

```
Formula = VBBCFBCFEncode({Articoli.Codice}, "C128", True)
```

*N.B. Nell'esempio si assume che il campo da rendere in barcode si chiami Codice, si trovi nella tabella Articoli e che la simbologia sia Code 128 con check digit abilitato*

La formula può quindi essere introdotta come campo nel report, al quale si andrà ad applicare il font di dimensioni opportune per ottenere il barcode della grandezza richiesta.

**Nota bene:** CRUFL\_VB\_BCF.dll utilizza BCFDLL.DLL per tutte le operazioni di codifica; tale DLL deve, pertanto, essere anch'essa installata.

## Utilizzo con altri linguaggi

Se è necessario integrare BCF con altri linguaggi, il sistema più semplice è l'utilizzo dell'Active-X di supporto (BCFActiveX.ocx); questo controllo viene automaticamente installato con BCFAssistant, oppure è disponibile nella directory `Examples\ActiveX` del CD; il controllo è anche installabile manualmente copiandolo in:

```
C:\WINDOWS\SYSTEM (Windows 95/98/ME)
```

```
C:\WINNT\SYSTEM32 (Windows NT/2000)
```

```
C:\WINDOWS\SYSTEM32 (Windows XP/2003)
```

L'Active-X deve essere registrato, nel caso di copia del file, digitando la seguente istruzione da un prompt di Dos:

```
REGSVR32 C:\WINNT\SYSTEM32\BCFActiveX.ocx
```

L'Active-X si introduce nei propri progetti come un qualsiasi altro controllo e rende disponibili due metodi:

```
Encode(codice As String, simbologia As String, checkdigit As Boolean) As String  
Verify(codice As String, simbologia As String, checkdigit As Boolean) As Boolean
```

Ad esempio, per ottenere in barcode la rappresentazione della stringa "ABCDEF" in Code128 con check digit, detto BCFActiveX1 il controllo istanziato, la sintassi è la seguente:

```
barcode = BCFActiveX1.Encode("ABCDEF", "C128", True)
```

**Nota bene:** BCFActiveX.ocx utilizza BCFDLL.DLL per tutte le operazioni di codifica; tale DLL deve, pertanto essere anch'essa installata.

## Utilizzo con Microsoft Word®

La seguente procedura descrive come utilizzare BCF con Microsoft Word® per realizzare la stampa in fusione di moduli che devono comprendere un barcode.

Il metodo descritto prevede due passi:

- stampa unione in nuovo documento;
- esecuzione di una macro per la sostituzione dei dati con la rappresentazione in barcode.

Procedere quindi in questo modo:

### 1) **installazione del codice utilizzato per la sostituzione dei barcode:**

- se non fosse stato installato il supporto VBA (Visual Basic for Applications) eseguire il programma di installazione di Word ed aggiungere tale opzione;
- dal menu di Word selezionare *Strumenti*, quindi *Macro* → *Editor di Visual Basic*;
- dall'editor di Visual Basic, verificare che sia visualizzata la finestra **Progetto - Project**; se fosse nascosta, dal menu scegliere *Visualizza* → *Gestione progetti*;
- selezionare nella finestra **Progetto** la voce **Normal**; premere il tasto destro e dal menu contestuale scegliere *Importa file*; alla richiesta del nome del file, selezionare `BCFReplace.bas` (disponibile nella cartella `Examples\Word` del CD di BCF);
- se successivamente dovesse essere necessario cambiare la dimensione del barcode sostituito, modificare la riga:

```
Const SZPFontSize = 36
```

all'inizio del modulo `BCFReplace` (doppio click per aprirlo dalla Gestione progetti);

- salvare le modifiche effettuate e chiudere l'editor di Visual Basic; avendo importato il modulo `BCFReplace` nel modello **Normal** (modello di base per tutti i documenti), la macro in esso contenuta sarà ora disponibile in tutti i documenti di Word;

### 2) **creare il documento modello per la stampa fusione:** l'unica avvertenza, in questo caso, è quella di racchiudere ogni stringa da codificare tra "[SBC]" ed "[EBC]"; ad esempio, se il campo da trasformare in barcode è <<Codice>>, in Word si dovrà specificare tale campo come:

```
[SBC]<<Codice>>[EBC]
```

### 3) **per effettuare la stampa ottenendo i barcode sostituiti (in base alla versione di Word i passi potrebbero essere leggermente diversi):**

- scegliere dal menu *Strumenti* → *Stampa Unione*;
- premere il pulsante **Unisci**; selezionare *"Unisci a nuovo documento"* e premere nuovamente **Unisci**; si otterrà un nuovo documento contenente tutti i dati uniti;
- scegliere *Strumenti* → *Macro* ed eseguire la macro **BCFRender** dal modello **Normal**; si otterrà un nuovo documento con i barcode opportunamente sostituiti, che potrà finalmente essere stampato.



Ovviamente i passi 1-2 vanno fatti una sola volta; successivamente è sufficiente seguire il passo 3 per produrre le proprie stampe.

**Nota bene:** il modulo utilizza `BCFDLL.DLL` per tutte le operazioni di codifica; tale DLL deve, pertanto essere installata.

## Utilizzo con Microsoft Access®

La seguente procedura descrive come utilizzare BCF con Microsoft Access® per realizzare una query che aggiunga un campo calcolato con la rappresentazione intermedia a cui applicare il font di BCF per ottenere un barcode leggibile.

Procedere quindi in questo modo:

- 1) aprire il database, selezionare la scheda **Moduli** e premere il pulsante **Nuovo**;
- 2) incollare il seguente codice nel nuovo modulo (si trova nel database demo.mdb, cartella Examples\Access su CD):

```
Option Compare Database
Option Explicit

Declare Function SZPStringToBarcodeDLL Lib "BCFDLL" (ByVal codice As String, _
ByVal simbologia As String, ByVal ConCD As Integer, _
ByVal barcode As String) As Integer

Function BCFEncode(s As String, symb As String, checkdigit As Boolean) As String

Dim result As String * 255, length As Integer

length = SZPStringToBarcodeDLL(s, symb, IIf(checkdigit, 1, 0), result)
BCFEncode = Mid(result, 1, length)

End Function
```

- 3) salvare e (opzionalmente) rinominare il modulo in BCF;
- 4) selezionare la scheda **Query** e premere il pulsante **Nuovo**; confermare *Visualizzazione struttura* e premere **Ok**;
- 5) introdurre i campi necessari alla query; nella riga *Campo* di una colonna vuota digitare la seguente espressione:

```
Barcode: BCFEncode([Code];"C128";Vero)
```

L'interpretazione è la seguente:

- Barcode: nome dell'espressione, utilizzata come intestazione di colonna nella query;
- [Code]: "Code" è il nome del campo di cui si vuole calcolare il barcode; modificarlo opportunamente;
- "C128": simbologia da utilizzare; modificarla opportunamente;
- Vero/Falso: stato del check digit (Vero = abilitato, Falso = disabilitato).

A questo punto è possibile creare report che includono il campo calcolato, al quale applicare il font di BCF per ottenere il barcode desiderato.

**Nota bene:** il modulo utilizza BCFDLL.DLL per tutte le operazioni di codifica; tale DLL deve, pertanto essere installata.

## Utilizzo con Microsoft Excel®

La seguente procedura descrive come utilizzare BCF con Microsoft Excel® per aggiungere le formule di calcolo barcode.

Procedere quindi in questo modo:

- 1) se non fosse stato installato il supporto VBA (Visual Basic for Applications) eseguire il programma di installazione di Excel ed aggiungere tale opzione;
- 2) copiare il file `bcf.xla` (disponibile nella cartella `Examples\Excel` su CD) all'interno della directory `XLStart` delle aggiunte di Excel; dal menu di Excel scegliere *Strumenti* → *Aggiunte*, premere **Sfoglia** e selezionare `bcf.xla`; in questo modo le funzioni saranno disponibili in ogni nuovo foglio di Excel creato; in alternativa (e qualora il livello di protezione non permettesse il caricamento automatico dell'estensione), le funzioni possono essere importate all'interno del singolo file Excel, semplicemente seguendo questi passi:
  - dal menu di Excel selezionare *Strumenti*, quindi *Macro* → *Editor di Visual Basic*;
  - dall'editor di Visual Basic, verificare che sia visualizzata la finestra **Progetto - Project**; se fosse nascosta, dal menu scegliere *Visualizza* → *Gestione progetti*;
  - selezionare nella finestra **Progetto** la voce relativa al progetto corrente; premere il tasto destro e dal menu contestuale scegliere *Importa file*; alla richiesta del nome del file, selezionare `BCF.bas` (disponibile nella cartella `Examples\Excel` del CD di BCF);
  - salvare le modifiche effettuate e chiudere l'editor di Visual Basic;
- 3) creare un nuovo worksheet; supposto che il codice da trasformare in barcode si trovi nella colonna A, selezionare la prima cella della colonna B e digitare nel campo formula quanto segue:

```
=BCFEncode(A1;"C128";"True")
```

L'interpretazione è la seguente:

- `A1`: cella contenente il valore da trasformare in barcode; modificarla opportunamente;
- `"C128"`: simbologia da utilizzare; modificarla opportunamente;
- `Vero/Falso`: stato del check digit (Vero = abilitato, Falso = disabilitato).

A questo punto, la formula può essere propagata a tutta la colonna semplicemente facendo click sulla cella B1 e trascinando verso il basso la selezione facendo click sull'angolo inferiore destro (sul quale appare un piccolo quadratino nero).

Per ottenere il corrispondente barcode a questo punto sarà sufficiente agire sul formato celle, applicando il font di BCF corrispondente alla simbologia prescelta.

**Nota bene:** il modulo utilizza `BCFDLL.DLL` per tutte le operazioni di codifica; tale DLL deve, pertanto essere installata.



## Licenza d'uso

**Leggere attentamente prima di installare il software.**

Installando il software accluso confermate di aver letto, compreso e accettato i termini e le condizioni di questa Licenza d'uso del software.

### LICENZA SOFTWARE

Questo è un Contratto legale stipulato tra voi (un individuo o un'entità) e la Simone Zanella Productions (SZP) per stabilire i termini e le condizioni d'uso del Software allegato. Aggiornamenti del Software saranno anch'essi soggetti ai termini e alle condizioni di questo Contratto. Questo Contratto è in vigore finché terminato dalla distruzione del Software e di tutta la documentazione fornita con il pacchetto, assieme a tutte le copie, tangibili o intangibili. In questo Contratto, il termine "uso" significa il caricamento del Software nella RAM, come anche l'installazione su hard disk o altra periferica di memorizzazione.

Il Software è di proprietà della Simone Zanella Productions ed è protetto dalle leggi italiane sul diritto d'autore e dai trattati internazionali sul copyright. Dovete considerare il Software come qualsiasi altro materiale protetto dal diritto d'autore. Il tipo di licenza, indicato sul prodotto, determina i vincoli di seguito specificati.

#### Licenza SITE:

Il prezzo d'acquisto del Software vi concede una licenza non-esclusiva per l'uso alle seguenti pattuizioni: il software potrà essere installato in tutti i computer di una stessa società e nelle sue filiali entro 100 km; tale società potrà essere l'acquirente stesso oppure un suo cliente (ma non entrambi). L'acquirente dovrà registrare con SZP attraverso l'apposito modulo i dati della società utente finale e le località di installazione del Software. SZP mantiene titolo e proprietà sul Software. Potete fare una copia del Software solo a scopo di archivio.

Non potete noleggiare, vendere, prestare, sub-licenziare o condividere in qualsiasi modo il Software con una terza parte, e nemmeno trasferire questa Licenza senza un permesso scritto rilasciato da SZP. Non potete decompilare, disassemblare, modificare oppure applicare le tecniche di reverse-engineering al Software.

È strettamente proibita la redistribuzione del Software all'interno di un pacchetto strettamente finalizzato alla generazione di codici a barre e alla loro stampa, o comunque all'interno di un pacchetto che si possa prefigurare come un concorrente diretto del prodotto venduto da SZP.

#### Licenza CORPORATE:

Il prezzo d'acquisto del Software vi concede una licenza non-esclusiva per l'uso alle seguenti pattuizioni: il software potrà essere installato in tutti i computer di una stessa società e nelle sue filiali, ovunque esse si trovino; tale società potrà essere l'acquirente stesso oppure un suo cliente (ma non entrambi). L'acquirente dovrà registrare con SZP attraverso l'apposito modulo i dati della società utente finale e le località di installazione del Software. SZP mantiene titolo e proprietà sul Software. Potete fare una copia del Software solo a scopo di archivio.

Non potete noleggiare, vendere, prestare, sub-licenziare o condividere in qualsiasi modo il Software con una terza parte, e nemmeno trasferire questa Licenza senza un permesso scritto rilasciato da SZP. Non potete decompilare, disassemblare, modificare oppure applicare le tecniche di reverse-engineering al Software.

È strettamente proibita la redistribuzione del Software all'interno di un pacchetto strettamente finalizzato alla generazione di codici a barre e alla loro stampa, o comunque all'interno di un pacchetto che si possa prefigurare come un concorrente diretto del prodotto venduto da SZP.

### **Licenza DEVELOPER:**

Il prezzo d'acquisto del Software vi concede una licenza non-esclusiva per l'uso alle seguenti pattuizioni: il Software potrà essere distribuito come parte di un pacchetto sviluppato dalla società acquirente, purché le funzionalità dipendenti dal Software acquistato non superino il 10% delle funzionalità totali e comunque non rappresentino parte fondamentale del pacchetto rivenduto. Il pacchetto prodotto non dovrà essere un derivato del Software acquistato e non potrà includere più di 3 (tre) simbologie. Qualora tali limitazioni non possano essere rispettate, per ogni pacchetto rivenduto sarà necessario acquistare una licenza SITE oppure CORPORATE, in accordo con l'impiego previsto. L'utente finale del pacchetto distribuito dalla società acquirente sarà comunque legato dai vincoli di licenza di tipo SITE indicati più sopra. L'acquirente dovrà registrare con SZP attraverso l'apposito modulo i dati delle società utenti finali e le località di installazione del Software; SZP si riserva il diritto di negare l'autorizzazione alla re-distribuzione del Software se ritiene, a Suo insindacabile giudizio, che siano violate le limitazioni indicate più sopra. SZP mantiene titolo e proprietà sul Software. Potete fare una copia del Software solo a scopo di archivio. Non potete noleggiare, vendere, prestare, sub-licenziare o condividere in qualsiasi modo il Software con una terza parte, e nemmeno trasferire questa Licenza senza un permesso scritto rilasciato da SZP. Non potete decompilare, disassemblare, modificare oppure applicare le tecniche di reverse-engineering al Software.

È strettamente proibita la redistribuzione del Software all'interno di un pacchetto strettamente finalizzato alla generazione di codici a barre e alla loro stampa, o comunque all'interno di un pacchetto che si possa prefigurare come un concorrente diretto del prodotto venduto da SZP.

Per qualsiasi forma di licenza vale quanto segue.

Se non rispettate uno qualsiasi dei termini e delle condizioni di questo Contratto, la Licenza sarà immediatamente terminata e dovrete ritornare immediatamente alla SZP il Software, i dischetti e la documentazione di questo pacchetto, assieme a tutte le copie di back-up. Le provisions di questo Contratto che proteggono i diritti proprietari di SZP continueranno ad essere in forza dopo la terminazione.

### **RESPONSABILITÀ LIMITATA**

Il Software e la documentazione sono venduti COME SONO. Vi assumete la responsabilità della scelta di questo Software per raggiungere i risultati desiderati e per l'installazione, l'uso ed i risultati ottenuti dal Software. SZP non fornisce alcuna affermazione o garanzia riguardo al Software e alla documentazione, incluse, ma non solo, le garanzie implicite di commerciabilità o adattabilità ad un uso specifico. SZP non sarà responsabile per errori od omissioni contenute nel Software o nei manuali, per qualsiasi interruzione del servizio, perdita di affari o profitti e/o per i danni incidentali o conseguenti in connessione con la fornitura o il funzionamento di questi materiali.

### **GARANZIA LIMITATA**

Per un periodo di 24 mesi dalla data di acquisto, la SZP garantisce l'acquirente originale che i dischi sui quali il Software è registrato sono liberi da difetti nei materiali o nelle lavorazioni quando soggetti ad uso e servizio normali. Se durante questi 24 mesi dovesse evidenziarsi un difetto, il disco sarà rimpiazzato gratuitamente dopo la restituzione ad SZP. Se un difetto dovesse evidenziarsi scaduto il periodo di garanzia, la SZP si riserva il diritto di far pagare un costo per l'intervento. La SZP si riserva il diritto di rifiutare richieste di sostituzione reiterate. Questa Garanzia Limitata vi fornisce dei diritti specifici e potete anche avere altri diritti che variano da Paese a Paese. Alcuni Paesi non permettono la limitazione o l'esclusione di garanzie implicite o di danni conseguenti, per cui le limitazioni o esclusioni indicate potrebbero non riferirsi a voi.

Confermate che questo testo è il completo ed esclusivo Contratto tra voi e la SZP, il quale sostituisce qualsiasi proposta o patto precedente, orale o scritta, e ogni altra comunicazione tra noi riguardo la materia oggetto di questo Contratto. Questo Contratto sarà interpretato e governato dalle leggi dello Stato Italiano ed ogni contestazione sarà discussa al foro di Venezia. Se alcuni dei punti di questo contratto risultano inapplicabili, essi non inficeranno la validità dell'insieme di questo Contratto, che resterà valido ed applicabile in base ai suoi termini.