



# Barcode/13

## Chapter 0 / General information

### What is Barcode/13?



**Barcode/13** is a full software barcode scanner for **Treo 600, Treo 650, Treo 680, Treo 700p, Zire 71** and **Zire 72**. **Barcode/13** uses the camera to scan EAN-8, UPC-A, ISBN and EAN-13 barcodes and exports decoded values to the system clipboard: values can be retrieve from the clipboard within any application. **Barcode/13** can be easily integrated in third party applications to automate the scanning process.

**Barcode/13** is a free alternative to expensive hardware solutions. **Barcode/13** can't be seen as a business solution, but it is definitely the cheapest solution for non-business use, like cave, books, CDs or DVDs inventories.

### What are UPC and EAN symbols?



**UPC-A: the most common symbology in the US.** April 3, 1973: the grocery industry formally establishes Universal Product Code as the standard bar code symbology for product marking, and UPC is the first bar code symbology widely adopted. UPC-A is based on a proposal submitted by George J. Laurer, IBM. An UPC-A bar code is the bar code US customers will find on virtually every consumer good on the shelves of a local supermarket, as well as books, magazines, and newspapers. UPC-A bar code is often the bar code you will also find worldwide on CDs.



**EAN-13: a superset of UPC-A for international use.** December 1976: following the acceptance of the original UPC specification, George J. Laurer adds a thirteenth digit to allow for "country identification" and make the UPC world wide, EAN-13 is born. EAN-13 is a superset of UPC-A: an UPC-A symbol is an EAN-13 symbol with the first number system digit set to 0, for example the UPC-A code "012345678905" is the same than EAN-13 code "0012345678905".

**ISBN: a prefixed subset of EAN-13 for books.** Mid 1960s: a system is developed out of two systems in use at the time, that of J Whitaker and Sons, Ltd in the UK and R.R. Bowker's system in the U.S. The 10-digit code British SBN system is adopted as ISBN in 1972. Prefix 978 can be combined to create EAN-13 barcodes with a new checksum. The third revision of ISBN requires codes to be 13-digit by 2007 to conform EAN-13 standard, prefix 979 is introduced.

## Chapter 1 / Using Barcode/13

### Using Barcode/13 keys



In preview mode, use the 5-way navigator left and right keys. Left key will leave **Barcode/13** with the clipoard emptied, right key will scan the barcode in your target.



In decoder mode, use the 5-way navigator left and right keys. Left key will return to the preview mode to scan again a barcode, right key will leave **Barcode/13** with your decoded barcode value in your system clipboard.



**Barcode/13** recognizes EAN-13 and UPC-A subset of EAN-13 with a leading 0, but if you have some difficulties with a barcode you can tell **Barcode/13** the first digit:

- If your barcode is a UPC-A barcode (US barcodes, 12 digits, **x xxxxx xxxxx x**) you can start the scanner pressing the "0" key (or Graffiti)
- If your EAN-13 barcode (european barcodes, 13 digits, **x xxxxxx xxxxxx**) begins with a 3, you can press the "3" key (or Graffiti), and so on.

I recommend using this feature only if automatic recognition fails: usually, I press the digit keys only if **Barcode/13** fails twice to read a barcode (this happens more frequently when scanning bottles, you should not need this for standard barcodes in normal lighting conditions). *On Treo 650 and Treo 680: space key can zoom/unzoom, but you should use this only with very small barcodes.*

### Scanning a barcode properly

#### Barcode/13

- EAN-13 (13-digits worldwide)
- UPC-A (12-digits subset US only)
- ISBN (10-digits subset for books)
- Normalize subsets to EAN-13
- EAN-8 (8-digits on small items)

cancel



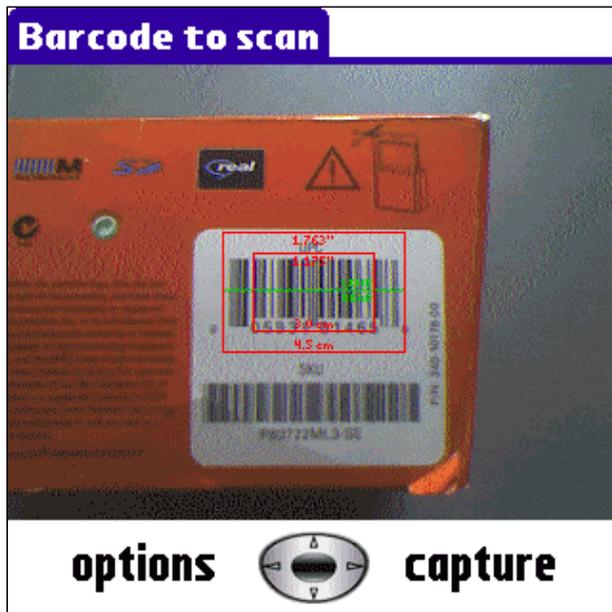
scanner

**Barcode/13** is compatible with **Treo 600, Treo 650, Treo 680, Treo 700p, Zire 71** and **Zire 72**. However, some limitations may apply depending on your device: **Treo 600**'s camera produces very saturated pictures and it can be difficult to get a sharp image, **Zire 71**'s camera gives good results but preview mode is pretty slow, **Zire 72**'s camera is very sensitive and an additional calibration has been added in preview mode. **Treo 680**'s camera (and probably **Treo 700p**'s) has a longer macro distance: it makes barcode recognition almost impossible except for large and flat UPC-A barcodes.

Nominal size of a UPC/EAN symbol is 1.469" wide x 1.02" high, the minimum recommended size is 80% of the nominal size or 1.175" wide x .816" high, the maximum recommended size is 200% of the nominal size or 2.938" wide x 2.04" high. Larger barcodes scan better, smaller barcodes do not scan as well or not at all.

During preview mode you can see two red targets and a green line. Green line is the position that will be used to analyse the barcode, targets are recommended sizes for large and small barcodes. Recent barcodes are often large barcodes, but some products have smaller barcodes.

- If you have a large barcode, point your



device to fill the large target with your barcode (keep the barcode inside, don't oversize it).

- If you have a small barcode, the small target will be a better idea.
- If your barcode is medium size, something between the 2 targets will be fine.

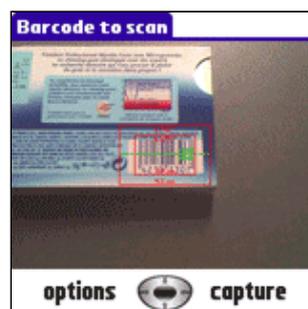
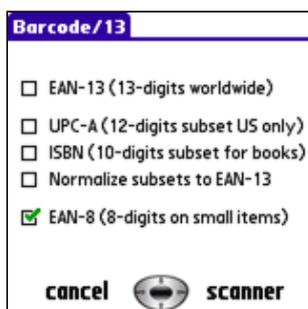


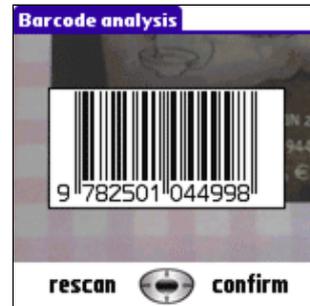
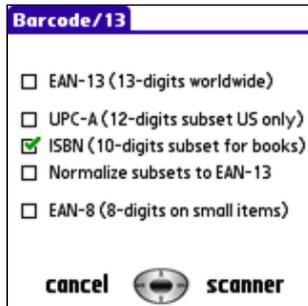
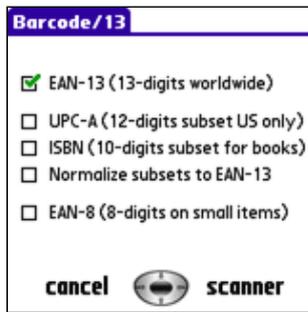
**Barcode/13** will give you best results if you follow some recommendations:

- Your barcode should be lighted without any shade, the easiest way and recommended way is to face the light source.
- Best distance is about 15cm with barcode parallel to your Treo.
- Barcode/13 works best with surfaces like CDs or DVDs, it is possible to use it with bottles but it may require a little training to get the good lighting conditions and to have a symmetric position.
- Don't move your Treo while scanning the barcode: a fuzzy image won't help!

If you don't have a proper contrast in preview mode, for example when scanning a black colored item, you can place a sheet of blank paper close to your barcode: this will force the camera to adjust the contrast level.

### Barcode/13 supported a barcodes





## Chapter 2 / Barcode/13 integration

### Checking if Barcode/13 is available

A feature is set to let applications know if **Barcode/13** is available, and barcode types that are supported. Currently exported value is `eanFtrNumBarcodeHasEAN8` | `eanFtrNumBarcodeHasUPCA` | `eanFtrNumBarcodeHasEAN13` | `eanFtrNumBarcodeHasISBN` | `eanFtrNumBarcodeHasSmall` | `eanFtrNumBarcodeHasConfirm` | `eanFtrNumBarcodeHasCallback`, meaning that EAN-8, ISBN, UPC-A and EAN-13 can be decoded.

```
#define eanAppCreator          'BcS2'
#define eanAppVersion         1

#define eanFtrCreator          eanAppCreator
#define eanFtrNumBarcode      0
#define eanFtrNumBarcodeDefault 0x00000000
#define eanFtrNumBarcodeHasUPCE 0x00000001 // Not available yet
#define eanFtrNumBarcodeHasEAN8 0x00000002
#define eanFtrNumBarcodeHasISBN 0x00000004
#define eanFtrNumBarcodeHasUPCA 0x00000008
#define eanFtrNumBarcodeHasEAN13 0x00000010

#define eanFtrNumBarcodeHasSmall 0x00010000
#define eanFtrNumBarcodeHasHuman 0x00020000 // Not available yet

#define eanFtrNumBarcodeHasConfirm 0x01000000
#define eanFtrNumBarcodeHasCallback 0x02000000

#define eanFtrNumBarcodeFormatMask 0x0000FFFF
#define eanFtrNumBarcodeDisplayMask 0x00FF0000
#define eanFtrNumBarcodeCommandMask 0xFF000000
```

### Launching Barcode/13 from your application

Launching **Barcode/13** within your application is very simple. Note: you're not supposed to copy/paste the following lines, this example is a short way to give all needed information and not a valid C syntax!

```
typedef struct
{
    UInt16 version;
    UInt32 creator;
    UInt16 key;
    UInt32 mode;
```

```

} barcodeKey;

barcodeKey *registration = MemPtrNew(sizeof(barcodeKey));
registration->version = eanAppVersion;
registration->creator = appFileCreator; /* YOUR application creator ID */
registration->key = eanRegistrationKey;
registration->mode = eanFtrNumBarcodeHasEAN13|eanFtrNumBarcodeHasCallback;
MemPtrSetOwner(registration, 0);

DmGetNextDatabaseByTypeCreator(true, (DmSearchStateType*)&searchState,
                                sysFileTApplication, eanAppCreator, true,
                                (UInt16*)&cardNo, (LocalID*)&dbID);
SysUIAppSwitch(cardNo, dbID, sysAppLaunchCmdPanelCalledFromApp, registration);

```

When launched with `eanFtrNumBarcodeHasCallback` **Barcode/13** will return to your application with `sysAppLaunchCmdReturnFromPanel`. This way, calling applications can make the difference between a normal launch and when returning from a **Barcode/13** call.

Developers wishing to use **Barcode/13** as a third party add-on to retrieve a barcode in their own application automatically must request a registration key `eanRegistrationKey`. Registration fees are linked to the price of the target application; freeware applications can get a free registration key to integrate with **Barcode/13**.

## Chapter 3 / Barcode/13 version history

---

### 2007-01-01: Barcode/13 1.0.2

- Treo 680 camera support added

### 2006-11-24: Barcode/13 1.0.1

- Zire 71 camera support added
- Zire 72 camera support added
- Treo 700p camera support added

### 2005-09-20: Barcode/13 0.9.8

- EAN-8 barcodes support added
- Treo 650 camera support added

### 2005-08-15: Barcode/13 0.9.6

- Minor fixes and improvements
- New applications integration features

### 2005-07-28: Barcode/13 0.9.4

- Improved capture

### 2005-07-22: Barcode/13 0.9.3

- Fix a compatibility issue with PiLoc

### 2005-06-15: Barcode/13 0.8.3

- First public release

*Last update 2007-01-01*

Copyright © 2005-2007 J.F.Morreeuw. All rights reserved.