

NAME

barcode – a library to create and print bar codes

SYNOPSIS

```
#include <barcode.h>
```

```
struct Barcode_Item *Barcode_Create(char *text);
int Barcode_Delete(struct Barcode_Item *bc);
int Barcode_Encode(struct Barcode_Item *bc, int flags);
int Barcode_Print(struct Barcode_Item *bc, FILE *f, int flags);
int Barcode_Position(struct Barcode_Item *bc, int wid, int hei, int xoff, int yoff, double scalef);
int Barcode_Encode_and_Print(char *text, FILE *f, int wid, int hei, int xoff, int yoff, int flags);
int Barcode_Version(char *versionname);
```

DESCRIPTION

The barcode family of library functions is meant to ease creation of bar-code printouts. The information below is extracted from the texinfo file, which is the preferred source of information.

The functions included in the barcode library are declared in the header file barcode.h. They perform the following tasks:

```
struct Barcode_Item *Barcode_Create(char *text);
```

The function creates a new barcode object to deal with a specified text string. It returns NULL in case of failure and a pointer to a barcode data structure in case of success.

```
int Barcode_Delete(struct Barcode_Item *bc);
```

Destroy a barcode object. Always returns 0 (success)

```
int Barcode_Encode(struct Barcode_Item *bc, int flags);
```

Encode the text included in the bc object. Valid flags are the encoding type (other flags are ignored) and BARCODE_NO_CHECKSUM (other flags are silently ignored); if the flag argument is zero, bc->flags will apply. The function returns 0 on success and -1 in case of error. After successful termination the data structure will host the description of the bar code and its textual representation, after a failure the error field will include the reason of the failure.

```
int Barcode_Print(struct Barcode_Item *bc, FILE *f, int flags);
```

Print the bar code described by bc to the specified file. Valid flags are the output type, BARCODE_NO_ASCII and BARCODE_OUT_NOHEADERS, other flags are ignored. If any of these flags is zero, it will be inherited from bc->flags which therefore takes precedence. The function returns 0 on success and -1 in case of error (with bc->error set accordingly). In case of success, the bar code is printed to the specified file, which won't be closed after use.

```
int Barcode_Position(struct Barcode_Item *bc, int wid, int hei, int xoff, int yoff, double scalef);
```

The function is a shortcut to assign values to the data structure.

```
int Barcode_Encode_and_Print(char *text, FILE *f, int wid, int hei, int xoff, int yoff, int flags);
```

The function deals with the whole life of the barcode object by calling the other functions; it uses all the specified flags.

```
int Barcode_Version(char *versionname);
```

Returns the current version as an integer number of the form major * 10000 + minor * 100 + release. Therefore, version 1.03.5 will be returned as 10305 and version 0.53 as 5300. If the argument is non-null, it will be used to return the version number as a string. Note that the same information is available from two preprocessor macros: BARCODE_VERSION (the string) and BARCODE_VERSION_INT (the integer number).