

It's so easy, we only need one page to tell you all about it. (We swear this is one page. Print double-sided and you'll agree.)

Installation (for Server Licenses)

When you put 6Barcode on your FileMaker server, it can install itself on every connecting client with just one script step. It's as easy as one, two, three:

1. Put the **6Barcode Helper.fp7** database on your FileMaker server. You'll need to use your FileMaker Server admin console to open it up.
2. Copy everything in the **Auto Update Files** folder to the **AutoUpdate** folder on your FileMaker Server. These are the actual plug-in files, packaged for FileMaker Server's Auto Update system
3. Set the **Install Plug-in** script from the **6Barcode Helper.fp7** database to run when your database opens. Choose File → File Options in FileMaker to tell FileMaker to run one of your scripts when the database opens. Your script can then perform the install script in the helper file.

Now, when someone opens your database, FileMaker will ensure that the plug-in is installed. If it isn't, FileMaker will download it from the server, install it, and enable it in just a few seconds, then your script will keep on truckin'. It's totally seamless.

Usage

Using 6Barcode is a breeze. Although there are lots of ways to generate and store your barcodes, this is probably the best:

1. **Add a new container field to your database.** This field will hold the actual barcode, so name it something like *Barcode*.
2. **Give this field an Auto-Enter calculation.** Use this formula: `sfrbc_Barcode (My Value)`

Instead of "My Value" put a field reference or formula that results in the value you want your barcode to have. Make sure you turn *off* "Do not replace existing value (if any)" in the Field Options dialog box.

3. **Put this field on a layout.** This field holds your barcode, so put it where you want the barcode to show up. For best results, choose **Format** → **Graphic**, set the first pop-up to "Reduce or Enlarge" and turn off "Maintain original proportions." You can now size and position the field as you see fit, and the barcode will fill your field perfectly.

You need to make your barcode big enough to be processed by a barcode reader. Test scanning with the largest value you expect, and make the barcode field bigger if it doesn't read reliably.

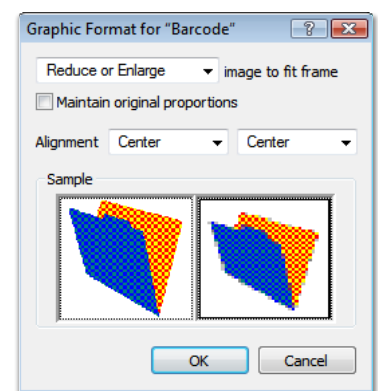
To install a **Demo** or **Client License**, simply drag the plug-in into the **Extensions** folder inside your FileMaker folder. Like this:



Then restart FileMaker.

Note

If your users have restricted access to their computers, and you're not using FileMaker Pro 9.0 or later, you may need to take extra steps to install. See sixfriedrice.com/autoupdate for more information.



That's about it. See the other side of this page for advanced options.

This is the back. Turn me over to get started.

Function Reference

sfrbc_Barcode (value { ; options })

This is the bread and butter. The bee's knees. It returns a Code-128 compliant barcode as a PNG graphic, so its result should go into a container field. You can use a calculation field (with a container result), an auto-enter calculation, or a Set Field script step targeting a container field.

The **value** parameter is the actual text value you want to store in the barcode. When you read this barcode (using a barcode scanner) this is the value it will send back to FileMaker.

Normally the barcode is black on a white background and has a reasonable size (which you'll want to reduce on the layout using the **Format** → **Graphic** options when you place your field). You can also customize your barcode by passing in options using the **sfrbc_Options** function, which is explained below.

sfrbc_LastError ({ as_text })

After creating a barcode (using **sfrbc_Barcode**), you can see if anything went wrong using the **sfrbc_LastError** function. Pass **True** to get back an explanatory message about the error. Pass **False** (or leave out the parameter) to get a numerical error code instead. Checking the error is entirely optional. In fact, it is usually only useful when you are testing.

sfrbc_Options (name ; value)

The **sfrbc_Barcode** function automatically uses settings that create the best bar code in most cases, but sometimes you may need a little more control. To customize your barcode, just pass in options using the **sfrbc_Options** function. Pass in the name of the option and the value you want. All available options are listed at the bottom of this page.

Passing options is easy:

```
sfrbc_Barcode (
  My Value ;
  sfrbc_Options ( "bar-color"; RGB(200,0,0) ) &
  sfrbc_Options ( "background-color"; "transparent" ) &
  sfrbc_Options ( "height"; 30 )
)
```

That formula will produce a 30 pixel tall barcode with red bars and a transparent background. As you can see, you can send in several options at once: just string them together with the & operator.

Options

bar-color: the color of the bars in the barcode; use FileMaker's built-in RGB function to specify the color, or use "transparent."

background-color: the color of the background in the barcode; use FileMaker's built-in RGB function to specify the color, or use "transparent."

height: the height of the barcode, in pixels

bar-width: the width, in pixels, of the thinnest possible bar; your barcodes overall width will grow as you increase this number; but the actual width of the barcode is determined by the length of the value being coded

compress: normally, barcodes are "compressed" to produce the smallest barcode possible; specify **False** if you don't want your barcodes compressed.