

OCR-A and OCR-B fonts version 0.2

Matthew Skala

September 27, 2012

1 Introduction

In the mid-2000s I had a reason to need the fonts OCR A and OCR B, and I found it much more difficult than it should have been to obtain copies and get them working with my software. There are commercial versions on the market; but these fonts, being defined in and required by public standards documents, really ought to be available in the free world. There were free versions available but only in the semi-obsolete MetaFont format, designed to produce bitmap output for use with TeX. I ended up getting a bunch of conversion and tracing software and semi-manually converting the fonts to TrueType format. The results were just barely good enough for my own purposes; but I posted them on my Web site anyway just in case they might be of use to others.

It turns out that in the years since then, those fonts have been among the most-requested resources on my Web site. Dozens if not hundreds of people have downloaded them and used them - and suffered the consequences of the crummy conversion job, and complained about the bad encoding, and so on. In the time since then I've also learned a lot more about how fonts work, and have designed a few of my own. I'm no longer thrilled to have my name on the old, poor-quality converted OCR fonts; and with my improved knowledge, I'm now in a position to write better ones.

This package contains those better fonts. More work still needs to be done, but it's already a significant improvement on the older packages.

Each font is provided in three ready-made forms: PostScript (.pfb and .afm files), TrueType (.ttf files), and OpenType (.otf files). Almost any computer typesetting or word processing system should be able to read at least one of these formats. OpenType is probably best if you

have a choice.

Please note that by definition these fonts have limited glyph sets - and if you actually want to use these fonts for OCR applications (their original purpose) then you should be very careful about which glyphs you use, because both typefaces have been extended by others (not me) to contain nonstandard glyphs beyond the official ones. Similarly, this package contains some nonstandard styles for OCR B (italic, reverse video, and outline) which may be visually appealing but are probably not appropriate for actual OCR use.

Please note that although the Metafont definitions for both OCR A and OCR B purport to support optical size, they actually just scale the outlines linearly for the different sizes, so when converting to a scaleable format there is no point treating the different sizes separately.

My email address is <mailto:mskala@ansuz.sooke.bc.ca>. As of this version, this package has become part of the Tsukurimashou Project at <http://tsukurimashou.sourceforge.jp/>. That is a bilingual page, English and Japanese; you can select the other one in the upper right corner if your browser's language preferences are misconfigured. All bug reports and feature and support requests for these OCR fonts should be filed in the Tsukurimashou Project's ticket tracker, with the component set to Parasite font packages.

2 OCR A

Text in this section of this document is set in OCR A; the other sections are OCR B.

OCR A is the standard font for the human-readable ISBN printed above the bar code on most books. It has an old-fashioned computerish feel, evoking the mythos of big iron data processing for which it was originally designed.

The version in this package originates with ANSI Standard X3.17-1977, approved January 20, 1977. Tor Lillqvist of the Technical Research Centre of Finland created a Metafont definition and added some semi-official characters for writing Nordic languages, based on an appendix of the standard. There is a bang path email address (no longer routable on the vast majority of Internet email systems) for Lillqvist in the source code comments, but little other information about this stage of the font's development is available to me. Richard B. Wales of UCLA picked up the project in 1988 and his copyright assertion and notes are

in the code comments as well. It is not clear to me how much of the original content of the font actually belongs to Wales. In his copyright notice he states that the font may be used freely, but cannot be distributed for profit (see the notices at the start of msk-ocra.mp for details). I released an earlier version in 2006 but that is now obsolete; the present version is newly derived in 2011 from the Wales version. I make no copyright claim on it myself.

The Metafont package I worked from is available at <http://www.ctan.org/tex-archive/fonts/ocr-a> .

This font includes alternate versions for some characters. The alternates are available in the OpenType version via the stylistic set and all alternates features. In each case I have let the default versions of the characters be the ones that were default in the Wales version; in some cases those are actually the more recently revised versions of the glyphs. See the source code or experiment with the OpenType features for more information. The alternates should also be available through the TrueType and PostScript versions, but I can't comment on exactly how to access them.

3 OCR B

OCR B is the standard font for the human-readable number printed along the bottom edge of a UPC/EAN bar code. That means that a standard book, if it's really using the correct fonts, needs both OCR A and OCR B in its bar code block. Many UPC codes do not actually use OCR B, however; often they fudge it with Courier, Helvetica, or even Arial. Unlike the ISBN (which is human-readable but meant to also be machine-readable), the number along the bottom of the bar code is only for human beings, and a computer would only scan the bars themselves.

The version in this package descends from a set of Metafont definitions by Norbert Schwarz of Ruhr-Universitaet Bochum, bearing dates ranging from 1986 to 2010. He originally distributed it under a non-commercial use only restriction but has since released it for unrestricted use and distribution. See the README file for more details.

The Metafont definitions include a number of variants for things like sharp ends and reverse video. It's not clear how valuable those alternate fonts are; they aren't suitable for actual OCR use, and there are some problems in the outlines (for instance, with poor overlapping of sharp-ended strokes) that make them less than optimal for human use too. Making the alternate versions work with

MetaType1 is going to require a fair bit of effort working around bugs in MetaType1 and Fontforge, as well as correcting the problems in the originals; I plan to do that eventually, but it s not done in this version.

Just as with the OCR A font, there are alternate glyphs for a few characters. In the OpenType version, these are available through the stylistic set and all alternates features.

The current version of the Schwarz package is available from CTAN at this address:
<http://www.ctan.org/tex-archive/fonts/ocr-b>

There is also a package by Zdeněk Wagner that is similar in general nature to this one:
<http://www.ctan.org/tex-archive/fonts/ocr-b-outline>

Like my earlier package, Wagner s was derived by tracing the outlines from the Metafont originals in a semi-automated way. The fonts in my current package are probably cleaner.

4 Compiling the fonts

Note that the binary font files are included in the package. Most users will have no reason to recompile them, and can safely ignore this section.

As of this version, this package builds using a stripped-down version of MetaType1 with some bugs fixed, inherited from the Tsukurimashou Project. The relevant code is bundled with this package, and (where applicable) relicensed to public domain. MetaType1 is no longer a dependency; Perl, Metapost (which should be included in a standard TeX distribution) and t1asm (part of the t1utils package) are now dependencies. FontForge is a dependency. Recompiling this document will also require XeLaTeX (which should be included in a standard TeX distribution); and there are additional considerations relevant to the test suite, for which see the Testing the fonts section below. Having `expect` is recommended but not required.

This package includes a standard GNU Autotools build system; if you have the prerequisites, it should work by running `./configure` followed by `make`. If you turn off the default feature that hides them, the compilation process will produce a large number of error messages; most of these are associated with bugs in Fontforge s spline geometry code, and are unavoidable.

The build system supports a `make install` target; however,

you might not want to use it, because it will install all the different styles and formats of the fonts and quite possibly install them in places other than where you expect. Most users will probably only want one format and a limited selection of styles, and would be better served by manually copying the files they want after building. Note that all the finished fonts intended for installation and use have filenames starting with capitalized OCR ; there are intermediate Postscript files created during the build under names that start with lowercase msk-ocr, but those are lacking important metadata and should not be installed and used directly.

Most of the Metapost source code files in this package have had their names changed by prefixing msk- ; that is to prevent a collision with the filenames used by the original TeX packages. If, like me, you try to compile these fonts on a system that also has the original TeX packages installed, there would otherwise be a danger of getting the original MetaFont files mixed with these Metapost files in a way that would cause it to fail. I couldn't figure out how to force Metapost to really use a specified pathname instead of going through TeX's filename search; it appears to strip off all specified path information.

5 Testing the fonts

The build system supports a `make check` target, which will run FontForge's `fontlint` program on all the installable fonts. This is a very demanding test. It will report a failure on anything that FontForge's developers (and even though I have sometimes been credited as one of these, I do not take responsibility for this point) think is against the rules or even vaguely questionable. Most `fontlint` validation errors are harmless in actual practice; so if you run `make check` and see nothing but red, Don't Panic.

At the very least, `make check` will almost certainly fail if your fonts were built with a version of FontForge that did not support my proposed optional argument to `AddExtrema()`; and as of this writing the only version of FontForge that supports that is the one in my Github fork at <https://github.com/mskala/fontforge>. The issue, for those interested, is that FontForge's `add extrema` operation has several different operating modes, including one that adds all possible extrema and one that only adds them in cases where it's considered safe to do so. The scripting language by default can only invoke the safe version; but `fontlint` demands all extrema whether safe or not, so without a patched version that makes the other modes available to the scripting language, it's not

possible for scripts to generate fonts that can pass fontlint.

Please do not report validation errors as bugs if you are not using the version of FontForge from my Github fork.