Handling large maps with MAPC2MAPC and other tools

The capacities of various devices make it tempting to want to install bigger and bigger maps on them. But there are limits and different size files need different techniques.

Firstly, we need to understand the limits:

MAPC2MAPC usually loads a whole image into memory. If the image is palletised then that means one byte per pixel; if the image comes from a full-colour file (JPGs are always full colour) that means 3 bytes per pixel. If MAPC2MAPC is undertaking operations on the whole file – such as Warp, Sharpen – it needs a second full-colour copy. Each copy needs an area of contiguous storage.

32 bit Windows systems have a limit of around 3GB per process, even if the physical machine is much bigger. 64 bit systems are limited only by the physical memory installed. Realistically, images of much more than 200 million pixels are unlikely to be handled on a 32-bit machine.

MAPC2MAPC uses the Freeimage library which always loads a complete image. For TIF files, however, MAPC2MAPC offers a mode where the image is cut into pieces that are only loaded into memory when needed. This ‘Virtual Storage’ mode is slower and will not undertake operations on the whole image but it will still create output files in many formats including JNX.

The .NET library that MAPC2MAPC uses only recognises traditional TIF files - whose architecture limits the file size (independently of the number of pixels) to 4GB. This is important because proprietary file formats processed by GDAL – ECW, SID and many others) are converted to TIF files for processing by MAPC2MAPC.

For output files, JNX files are limited to 50,000 map tiles (up to 1024x1024 pixels) and a file size of 4GB.

KMZ files are unlimited but if they are used as Garmin Custom Maps then they are limited to 100 tiles of 1024x1024 pixels installed in the Garmin unit (NOT per map!). The Montana limit is 500 tiles.

Mobile Atlas (Slippy Map) formats use map tiles of 256x256 pixels. The number of these is limited only by the capacity of the device holding them (and it’s file system). If held as SQLITE the files are, again, limited only by the device and its file system. *[Note that large maps held as small map tiles may be very inefficient of space if the device cluster size is bigger than the tile size – on a large SD card half the space can easily be wasted. This does not apply to SQLITE and other formats such as MG Maps that join tiles into packages].*

Because of its unlimited size and simple structure, the Mobile Atlas format can be used as an intermediate storage format for large maps. These Atlases can also be defined as map sources to Mobile Atlas Creator so they can then be output in all the formats that MOBAC supports.

MAPC2MAPC contains some utility functions and there are two other programs to download.

So what to do to process a large file…

If the file is too big to load into MAPC2MAPC and is (or is converted to) a TIF less than 4GB in size then try the option 'Use VS for TIF'.

For file bigger than this, use the Utility Convert Huge File to Mobile Atlas which processes the file in sections to build up a store of seamless Mobile Atlas tiles. This may take several hours to run for a very big map.

For very large KMZ files use the Convert KMZ to Mobile Atlas.

If you want to use the content of a 'home made' JNX file then use the Convert JNX to Mobile Atlas Utility function

And to use the large file…

MOBAT2JNX will take a Mobile Atlas tile store and make multi-level JNX.

Mobile Atlas Creator will make many different formats of file for mobile devices. Define the tile store as a Mapsource (see the MOBAC documentation) and it then behaves just like an on-line source such as Open Street Map.

Mobile Atlas Creator : http://mobac.sourceforge.net
MOBAT2JNX : http://www.the-thorns.org.uk/net/mobat2jnx.zip