

# m9editor v3.0.3

---

## What

m9editor is your little helper when it comes to editing manila files. It lets you edit nearly all aspects of a mode9 file. Additionally it will assist you with graphics, allowing import, export and CFC compression. And It includes a directory viewer with information specific to manila files.

## Who and why

From the first day I owned a HTC touch HD I was fascinated from this shiny little gadget. On xda-developers, a discussion about reverse-engineering manila caught my eye, and so I begin to write some hacking stuff to have a deeper look inside. Meanwhile it has grown in functionality - time to write a documentation. I've been a software developer since my school days (which is now back several decades), and though some people think that's none of a girls business, I always enjoyed it and still do.

## Requirements

m9editor is written in Visual Basic .net. You'll need to install dotnet framework 3.5 (currently SP1) - usually its already on your PC. For Lua decompilation you'll need sztupy's decompiler, plus the necessary C++ redistributables.

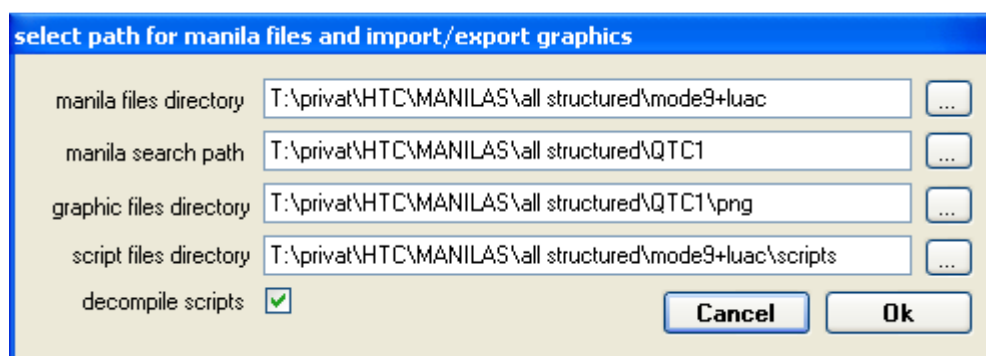
dotnet framework: <http://msdn.microsoft.com/en-us/netframework/default.aspx>

sztupy's Lua decompiler: <http://forum.xda-developers.com/showthread.php?t=479910>

C++ redistributables: <http://www.microsoft.com/downloads/details.aspx?FamilyID=9b2da534-3e03-4391-8a4d-074b9f2bc1bf>

## How

When you start m9editor the first time, you will be asked for different directories:



**MANILA FILES DIRECTORY:** the main path where m9editor looks for files to view and load, shown in the directory overview (lower left panel)

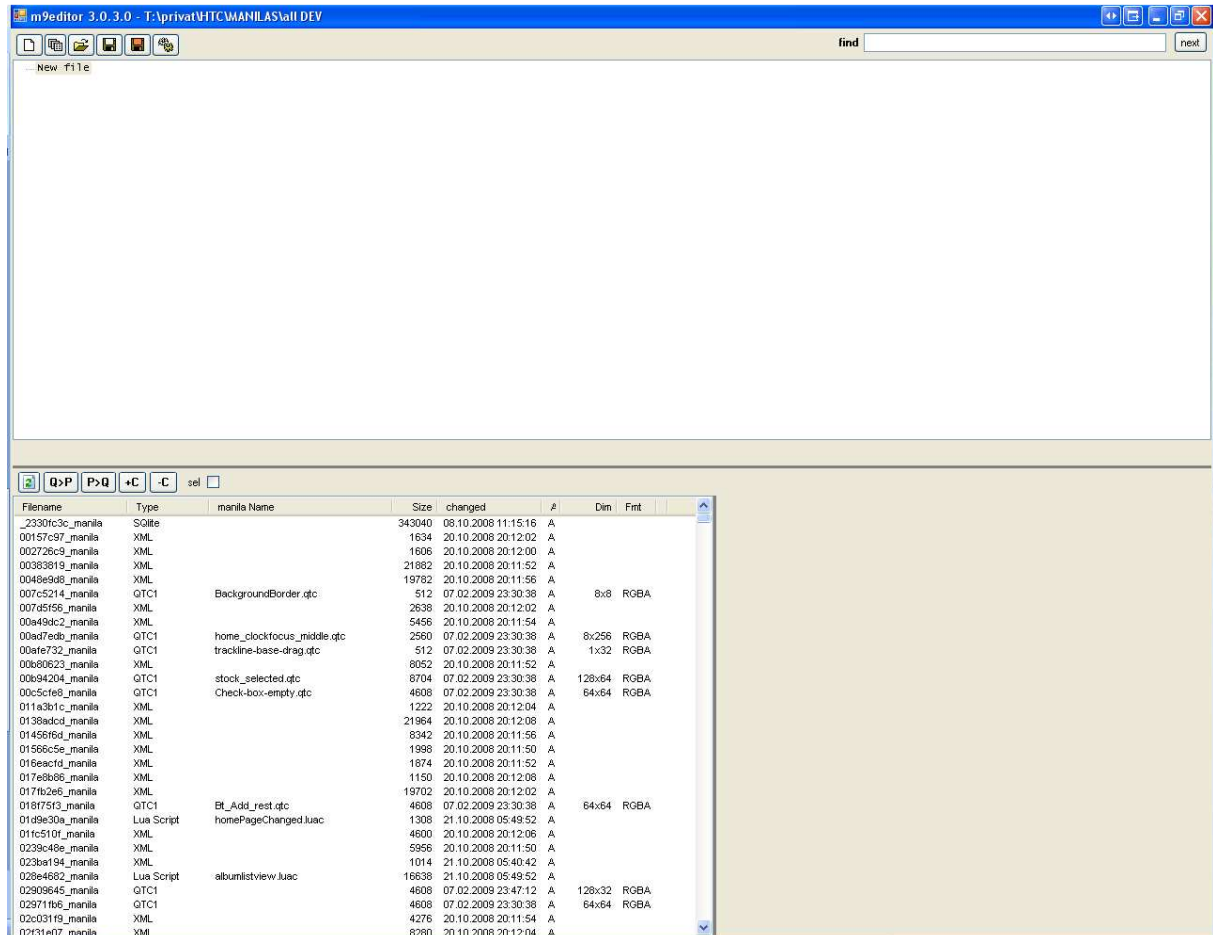
**MANILA SEARCH PATH:** additional path where m9editor searches for files. Imagine you separated scripts and graphics (\*\_manila) in different dirs. Your manila files directory will thus be the path to the scripts+mode9's. Now you open a mode9 file and encounter a link to a graphic, which should be displayed in the lower right corner. m9editor searches for that graphic in the manila files directory, if its not found it searches the manila search path. If still not found it simply will not show.

**GRAPHIC FILES DIRECTORY:** this is where the png files go when you save a QTC1 as png. Vice versa, if you import a png to a QTC1 its taken from there.

**SCRIPT FILES DIRECTORY:** this is used by all script import/export/decompile functions

**DECOMPILE SCRIPTS:** if checked, m9editor will try to decompile lua scripts from directory viewer, or embedded scripts within a mode9 file.

After that, m9editor will read your main directory and show its contents. After quick-loading filenames, files will be analyzed in background for in-depth information, you'll notice a progressbar during that phase.



There are three areas of interest:

**UPPER PANEL:** here's the structure of a mode9 file, if one is loaded

**LOWER LEFT PANEL:** the directory view of your main directory. Columns may be sorted ascending/descending by clicking on the column names (as with windows explorer), resize and autofit works too. Right click on a filename offers various options depending on filetype, described later.

**LOWER RIGHT PANEL:** shows additional things depending on selection, e.g. picture preview, script sourcecode

Panels are resizable by dragging the splitterbar between upper and lower panels, and between the two lower panels.

## Directory viewer - columns

**TYPE:** derived from file content, no matter what extension the file has

**MANILA NAME:** will be shown if known. As the filenames cant be reverse-calculated to real manila names, they are taken from a table inside m9editor.

**SIZE:** filesize, from filesystem

**CHANGED:** change date, from filesystem

**ATTRIBUTES:** filesystem attributes

**DIM:** graphics dimension, only shown for QTC1 graphic format.

**FMT:** graphic format, analyzed from file content. Shown for QTC files only. Usually RGBA, sometimes RGB

**CPR:** shows CFC if compressed accordingly

## Directory viewer - mouse actions (click on filename)

**LEFTCLICK:** various actions depending on filetype:

- Lua Scripts: shows sourcecode in lower right panel (depending on settings)
- graphic files (QTC, BMP, PNG, JPG): displays preview in lower right panel

**DOUBLECLICK:** for mode9 files only, opens it in upper panel

**RIGHTCLICK:** various actions, depending on filetype:

- **Lua Script:**
  - **convert to ANSI:** converts the script to ANSI and saves it to the scripts directory. File extension will be \*.luaA
  - **decompile and save sourcecode:** decompiles the script and saves its sourcecode to the scripts directory. File extension is \*.src
- **mode9:**
  - **open:** same as doubleclick, opens it in upper panel for editing
  - **write debug file:** writes a tag analysis file to the main directory, extension is \*.tags.txt
- **QTC1:**
  - **CFC compress:** for uncompressed files, compress graphic content with CFC. That's done in-place, file will be overwritten. Notice the CFC marker in the Cpr column
  - **CFC uncompress:** for CFC-compressed files, uncompresses graphic content. That's done in-place, file will be overwritten.
  - **save as PNG:** saves graphic in PNG format to the graphic files directory
  - **load from PNG:** loads graphic from PNG file contained in graphic files directory
- all files:
  - **open with:** depending on m9editor.cfg, see detailed description

## Directory viewer - menubar

**REFRESH DIRECTORY:** re-reads directory content and runs file analysis in background

**Q>P:** convert all QTC1 graphic files in main directory and save them as PNG to graphics directory

**P>Q:** convert all PNG graphic files in graphics directory and save them in QTC1 format to your main directory. This will overwrite all existing manila graphics.

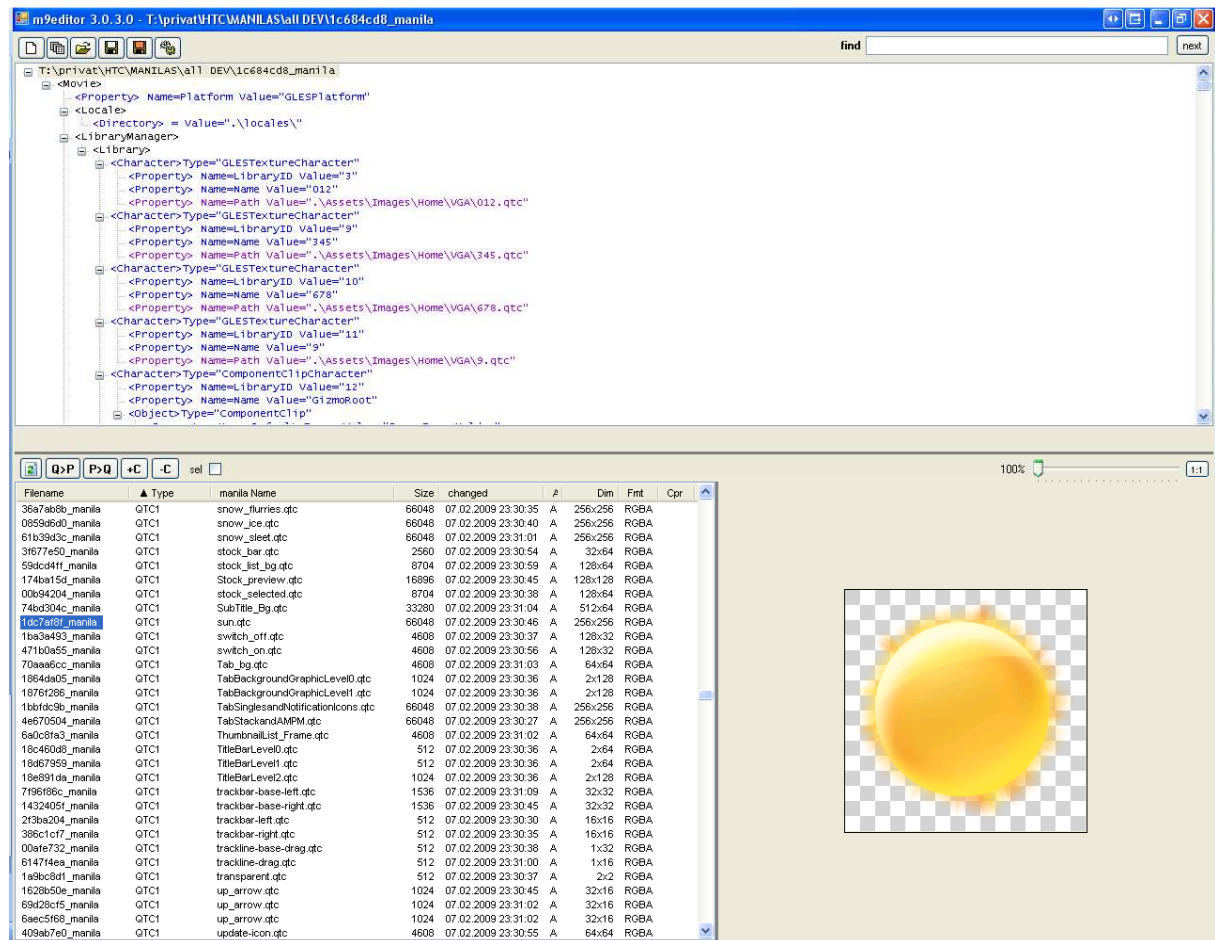
**+C:** CFC-compress all QTC1 manila graphics in main directory. Done in-place, original files will be replaced

**-C:** CFC-uncompress all manila graphics in main directory: Done in-place, original files will be replaced

**SEL CHECKBOX:** enable additional file manipulation options:

- enables a checkbox on all files, select/deselect files for action
- right-click on any file to:
  - select all
  - select none
  - copy selection to a path of your choice: choose destination in directory picker

**EXAMPLE:** m9editor with home.mode9 opened in editor, while viewing sun.qtc in preview



## Editor overview

After loading a mode9 file via doubleclick in directory viewer, or using the open button in the menu, the tree structure of the file is displayed. Nodes can have various foreground and background colors, and various actions are offered depending on the nodetype. Here's an overview:

**Black:** normal node, not editable

**Blue:** node with editable values

**DarkViolet:** in Texture, TexturePath properties, if a QTC picture is referenced. Click it to display the graphic in lower right panel

**Green:** a reference to another mode9 file, can be loaded via context menu

**Light Red:** a reference to an external script (.luac). sourcecode will be shown in lower right panel (if enabled and decompilable)

**Dark Red:** embedded script, sourcecode will be shown in lower right panel (if enabled and decompilable)

**Annotation:** node has annotation, displayed in tooltip, edit via contextmenu

**node modified:** if you modified values, background color will change to orange, so you'll what you've modified

You'll notice several nodetypes which don't have a value to change (black), others with values, and properties where name and value can be modified. m9editor will show you whats changeable in the editline below the tree. Doubleclick a node to get there quickly, and use <enter> to hop to the next field, until you reach the "Ok" button. Hitting <enter> on "Ok" will save your changes.

For property nodes, name and values of the property can be modified. Depending on a property's name, possible values can be different (number of values and datatypes), so if you change a propertyname the editline might change to reflect possible fields. If a propertyname is generic (has no special meaning) m9editor will assume a text property.

New nodes can be inserted via contextmenu "insert":



select the desired nodetype, and fill in values via editline.

You may move a node within the tree structure through holding Shift+up/down/left/right keys. Up/down will move it correspondingly under the current parent node, left/right will change the hierarchy level. It can be quite tricky to move a misplaced node to the desired position, so better choose the right position before inserting.

## Editor - context menu

m9editor offers several actions via context menu:

- **expand**: expand tree starting at selected node
- **collapse**: collapse tree under selected node
- **annotation**: write an annotation in the annotation line under the treeview, press <enter> when done. To delete an annotation, simply delete the text and press <enter>
- **insert**: insert a new node (see editor overview)
- **delete**: delete node and all subnodes
- **copy**: copy node including structure to the clipboard
- **paste**: paste node from clipboard. Only available if you copied one to the clipboard before. The selected node will be the parent of the pasted node.
- **paste below**: same as paste, but places the new node as a sibling under the selected node (new and selected node has same parent node)
- **toggle mark**: set/reset the color mark used for tagging modifications. Only available on editable nodes
- **open mode9 file**: available on the reference to another mode9 file, this can be used to quick-open the referenced file
- **export script (native Unicode)**: on embedded scripts, this will save the script exactly as its contained in the mode9 file. Saved files go to the scripts directory, and are named as follows: <manilafilename>\_<hexoffset>.luaA
- **export script (ANSI converted)**: on a script property, will convert the script to ANSI and save in to the scripts directory. Filename will be: <manilafilename>\_<hexoffset>.luaA
- **decompile and save sourcecode**: on a script property, will convert the script to ANSI and send it to luadec (if enabled) for decompilation. The output of luadec (aka sourcecode) goes to the script directory and is named <manilafilename>\_<hexoffset>.src
- **import script**: **WARNING: EXPERIMENTAL**: on a script property, a filepicker shows up. Direct it to the script you want to import. The script must be compiled with Lua compiler 5.1. The

signature “LuaQ” will be checked and used to recognize ANSI or Unicode format. If Unicode, the file is used without modification. If ANSI it will be converted to Unicode first, then written to the property.

### Editor - menubar

**NEW FILE:** clears the tree

**SELECT WORKING DIRECTORY:** select manila files directory. Will be saved to m9editor.cfg on exit

**OPEN MODE9 FILE:** open a file which is not in your current working path

**SAVE:** save file, will be overwritten (you'll be warned)

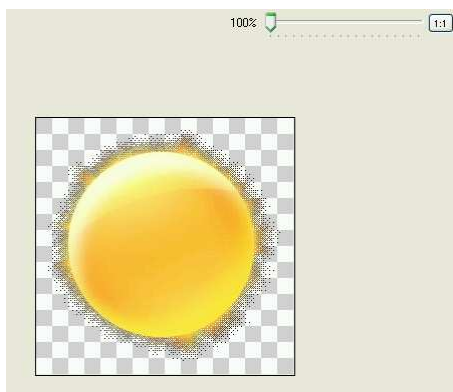
**SAVE AS:** save file to different path and/or with different name

**CHANGE SETTINGS:** show/edit settings

### lower right panel

This panel is used for displaying additional informations depending on a selected file or node:

- 1) show a referenced graphic in directory viewer or while browsing a mode9 file and clicking a referenced graphic. Use the slider to adjust zoomlevel.



- 2) show the sourcecode of a .luac file in directory viewer, or an embedded script within a mode9 file. Use the searchbox to search for text within the sourcecode.

```
find  next

require("playcontrol")
require("weather")
require("scrollphysics")
require("weathereasteregg")
animatingUpOrDown = false
forecastMinutes = -1
allowUnlimitedForcedUpdates = _application.Store:GetIntValue(Lifetime_Permanent,
"weather.AllowUnlimitedForcedUpdates")
declinedRetry = false
functionForAfterOnNavigateIn = nil
navigatingIn = true
forceRoaming = false
resetSoftkeysOnDialogExit = false
loadedCity = false
navigatingOut = false
skiperrorbox = false
drawForecastTimeInterval = 2
drawForecastData = nil
drawForecastTimer = nil
updateStatus = 1
if CityNumberSeparator == nil then
    CityNumberSeparator = "/"
end
OnShowUpdateNowConfirmationDialogReturned = function(loc_0, loc_1)
    loc_0.OnExit:disconnect(OnShowUpdateNowConfirmationDialogReturned)
    if loc_0.Result.ExitType == DialogExitType_Cancel then
        if currentMode == "Gizmo" then
            SwitchMode("Gizmo")
        end
        forceRoaming = true
        loc_1()
    else
        -- Warning: missing end command somewhere! Added here
    end
end
end
ShowUpdateNowConfirmationDialog = function(loc_0)
    if EnterKeyPressed == true then
```

## m9editor config file

If you made it 'till here without falling asleep ... congrats ☺

So, I guess you deserved some bonus material, here it is:

Usually, you'll expect the following content in the m9editor.cfg, which is (I guess) self-explanatory:

```
DataPath=T:\privat\HTC\MANILAS\all DEV
SearchPath=T:\privat\HTC\MANILAS\all structured\QTC1
GraphicPath=T:\privat\HTC\MANILAS\all structured\QTC1\png
ScriptPath=T:\privat\HTC\MANILAS\all structured\mode9+luac\scripts
DecompileScripts=Y
```

Additionally, you may extend m9editor's features with that:

```
SourceCodeViewer=<path to your favorite text editor >
OpenWith=<path to your favorite hex editor>
OpenWith=<path to your favorite text editor>
OpenWith=<path to your favorite ... insert own ideas here>
```

### ***What it does:***

**SourceCodeViewer=<...>** enables a "view sourcecode" contextmenu when you rightclick a Lua-script in directory viewer. It will run the program you specified and send the selected item as a parameter

**OpenWith=<...>** enables a "open with" contextmenu on every file in directory viewer. As you might suspect, it runs the specified program and pass the selected filename as a parameter.

### **Example:**

```
SourceCodeViewer=C:\Program Files (x86)\Notepad2\Notepad2.exe
OpenWith=XVI32,C:\Program Files (x86)\XVI32\XVI32.exe
OpenWith=Notepad2,C:\Program Files (x86)\Notepad2\Notepad2.exe
```