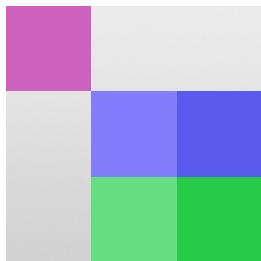


# MSX Tiles devtool

## User Guide



# MSX Tiles devtool

## User Guide

version: 0.9b

Copyleft 303bcn 2014

This program is distributed under the terms of the GNU General PublicLicense

WEB: <http://code.google.com/p/msx-tiles-tools>

mail: [aorante@gmail.com](mailto:aorante@gmail.com)

nMSXtiles is a Pentacour's tool: <http://pentacour.com/nmsxtiles>

Dedicated to Karoshi MSX Community <http://karoshi.auic.es>

# Contents

- 1 What's MSX Tiles devtool?.....4
- 2 Main Window.....5
- 3 Load.....6
  - 3.1 Load a project.....6
  - 3.2 Load a tileset.....7
  - 3.3 Load a Map.....7
- 4 Project Info.....8
- 5 Screen viewer.....9
- 6 Tools.....10
  - 6.1 Fill order screen map.....10
  - 6.2 Fill the screen map with the specified tile..10
  - 6.3 Changing a tile to other.....11
  - 6.4 Change one color to other.....11
  - 6.5 Optimize tool.....12
  - 6.6 Invert graphic data.....12
- 7 Color palette tool.....13
  - 7.1 Project toolbox.....14
  - 7.2 Confirmation color buttons .....14
  - 7.3 Colors list.....14
  - 7.4 Palette map.....14
  - 7.5 Sliders.....15
  - 7.6 Confirmation palette buttons.....15
- 8 Code output.....16
  - 8.1 Basic settings.....17
  - 8.2 Compress.....18
  - 8.3 Tilesets.....19
  - 8.4 Map.....20
  - 8.5 Color palette.....21
  - 8.6 Save Output.....21
- 9 Save.....22
  - 9.1 Save a project.....22
  - 9.2 Save a Tileset.....22
  - 9.3 Save a Map .....22

# 1 What's MSX Tiles devtool?

This application to work with images in graphics mode 2 of VDP TMS9918 and get a dump in different code formats (ASM, C and Basic).

Born from the need to convert nMSXtiles projects to C code, plus some utilities to develop GUI for MSX application PSGed.

This application is designed for agile and intuitive handling, but always can be improved. We are working on it. We are waiting your suggestions.

This software was developed in Microsoft Visual Basic 2008 Express.

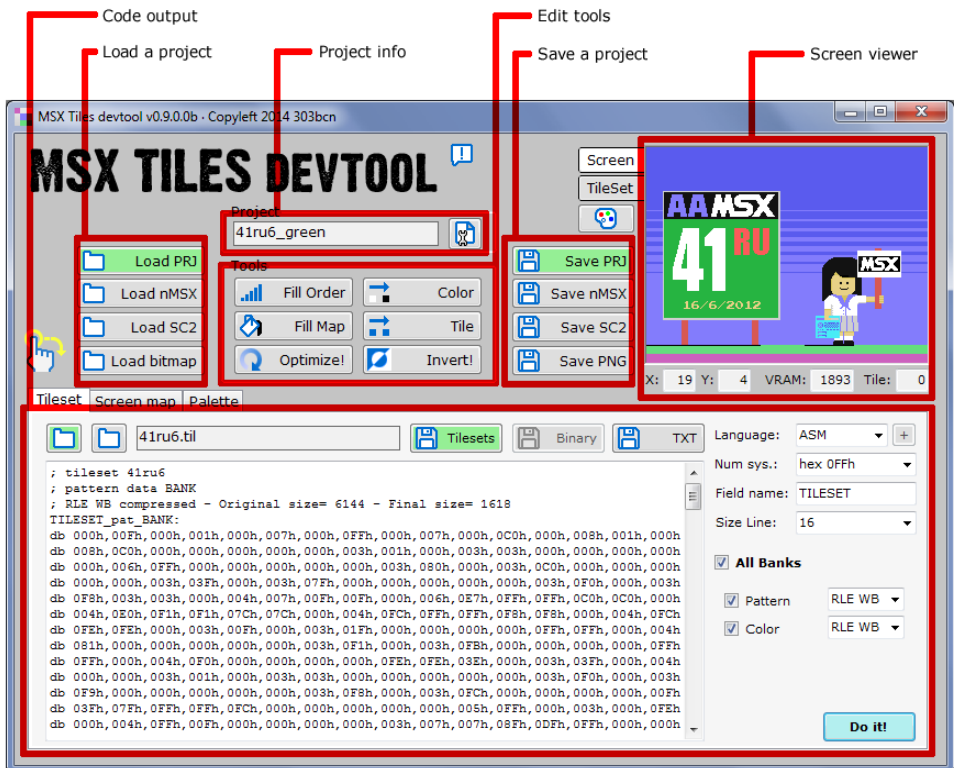
To obtain the source code, go to the WEB project:

<http://code.google.com/p/msx-tiles-tools>

## 2 Main Window

The main window consists of several areas with flow from left to right and top to bottom:

- Project info
- Screen viewer
- Load a project
- Edit tools
- Save a project
- Code Output



## 3 Load

First, you'll need to load a project (screen) from a file with any of the different supported formats.

The native file format is the same for the total amount for parts, so you can load the Tilesets, palette or map independently from any file XSCP project.

### 3.1 Load a project

To get started, first you have to upload a screen. MSX Tiles devtool accepts these formats:

- Native project. Includes palette (XSCP).
- nMSXtiles project.
- SC2. It is the VRAM dump a binary file from the MSX Basic. It may contain the palette for V9938 VDP.
- PNG o GIF. Convert images to TMS9918's graphic mode 2. This created so you can paint your pictures from your favorite graphics application.

For the result is good, it must meet the following conditions:

- The image must have a resolution of 256x192. If the image has a different resolution, the program will adapt, but produce unexpected effects.
- The palette has to be similar to that defined by Sean Young:  
<http://bifi.msxnet.org/msxnet/tech/tms9918a.txt>  
<http://aorante.blogspot.com.es/2011/10/paleta-vdp-de-los-msx-1.html>

You can drag files from OS explorer win and drop in the MSXTilesdevtool main window. Accept PNG, GIF and SC2 formats.

## 3.2 Load a tileset

In Tileset tab, you can load the three banks tilesets in formats XTIL (native) and TIL from nMSXtiles.

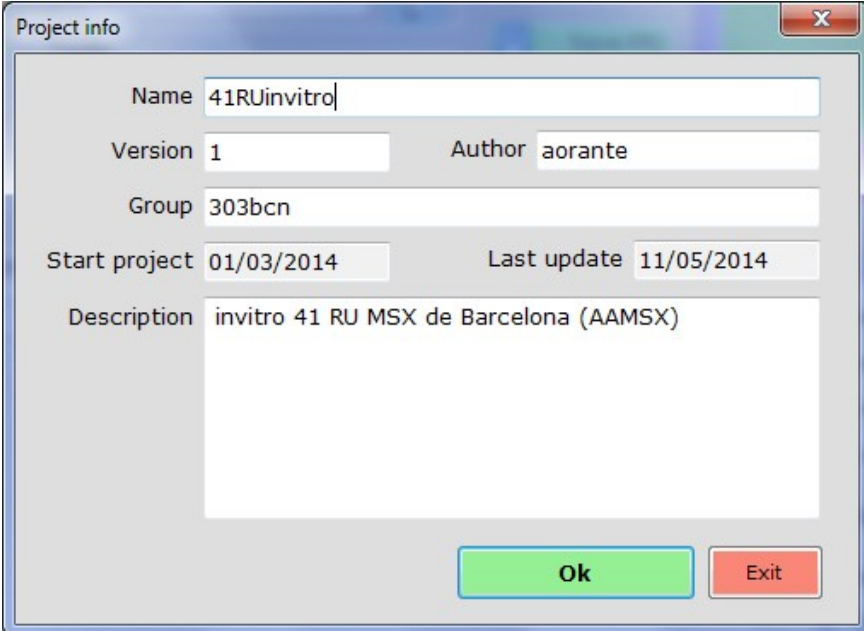
## 3.3 Load a Map

If you want to change the map screen, you must select the corresponding tab, where you can load a map in formats XMAP (internal) and SCR nMSXtiles.

## 4 Project Info

To help you organize your project, we added a form for you to add information that is stored within the MSX Tiles devtool project file.

Includes: project name, version, author, group, description, start date and date of last update.



The image shows a 'Project info' dialog box with a title bar and a close button (X). The form contains the following fields:

Field	Value
Name	41RUinvitro
Version	1
Author	aorante
Group	303bcn
Start project	01/03/2014
Last update	11/05/2014
Description	invitro 41 RU MSX de Barcelona (AAMSX)

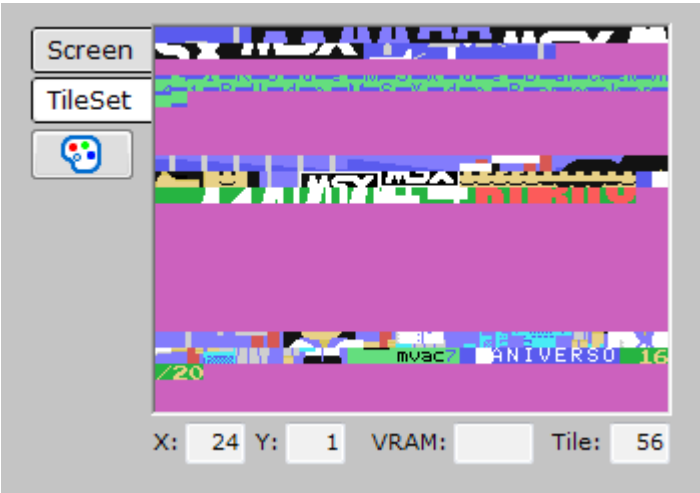
At the bottom right, there are two buttons: 'Ok' (green) and 'Exit' (red).



# 5 Screen viewer

Display the map, just like on a real MSX or tilesets.

In screen mode, at the bottom shows: the cursor position, direction of VRAM (Table Name), and the value of tile. You can also select an area to determine the output code ([8.4](#)).



## 6 Tools

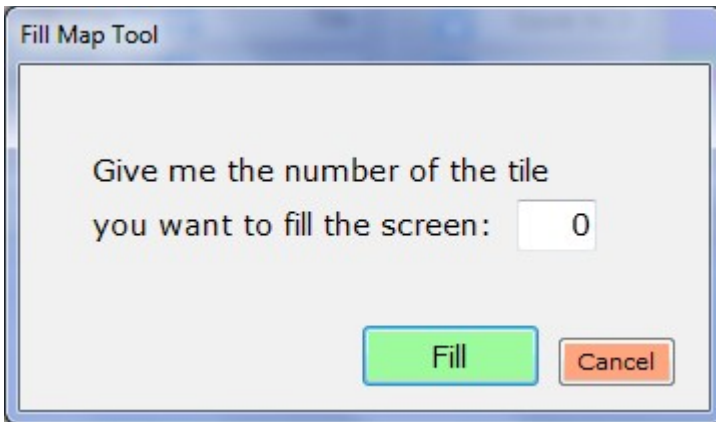
The application includes tools to make some small modifications to our screens.

### 6.1 Fill order screen map

This tool organizes all map tiles as screen2 initialization.

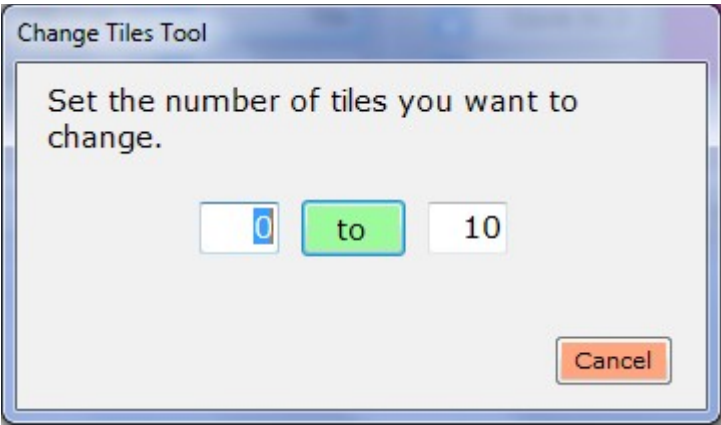
### 6.2 Fill the screen map with the specified tile

This tool fills the entire map with the number of tile.



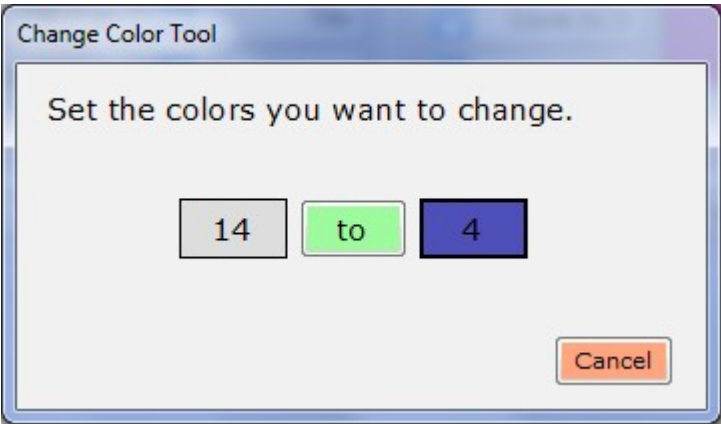
## 6.3 Changing a tile to other

From this tool you can change one tile to another, on the map.



## 6.4 Change one color to other

Allows you to change one color with another. Color is specified, from a selector that displays the color palette.



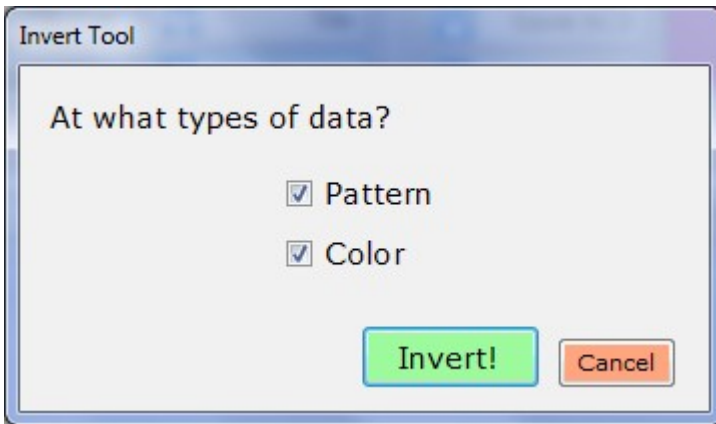
## 6.5 Optimize tool

Sort pattern and color data to enhance compression.

**Note:** When a bitmap is loaded, the optimization is applied automatically.

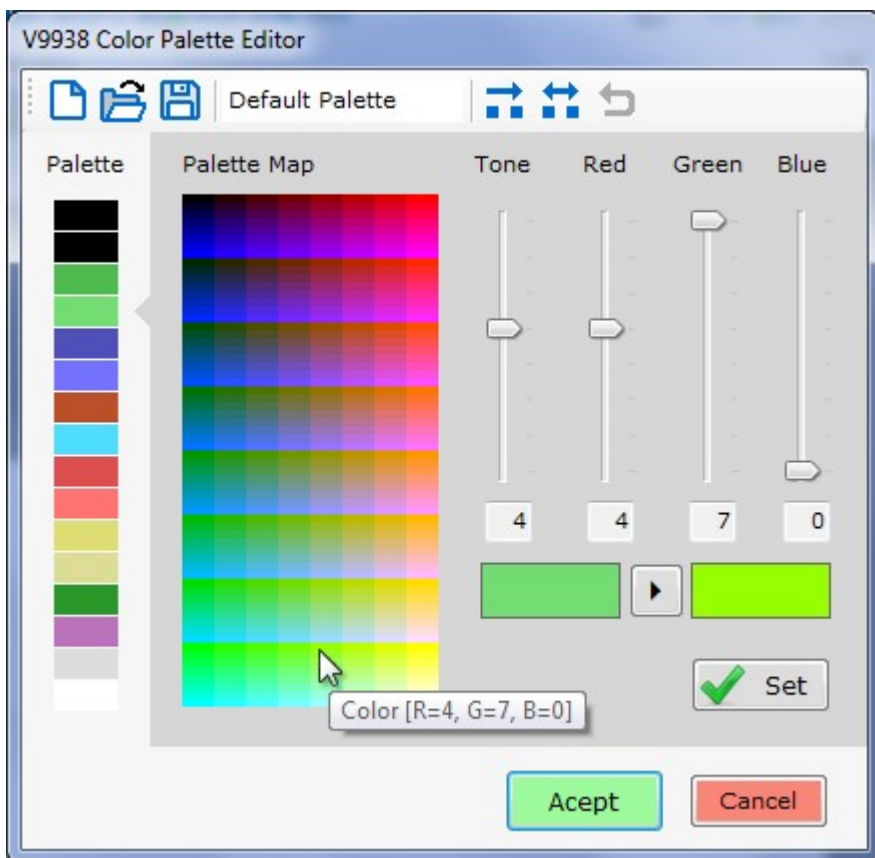
## 6.6 Invert graphic data

This tool is not very useful, because the graphics information is usually unordered.



## 7 Color palette tool

From this window we can design a color palette for computers that have the V9938 or higher graphics processor.



## 7.1 Project toolbox

Contains these tools:

- **New.** Sets the default palette..
- **Load.** Displays a dialog for loading a color palette.
- **Save.** Displays a dialog to save the palette to disk.
- **Palette name.** Displays and lets you change the name of the palette.
- **Copy color.** Copy a color to the specified number.
- **Swap colors.** Swap two colors.
- **Undo.** Removes the last change.

## 7.2 Confirmation color buttons

**Set.** Sets the color in the palette.

**Right Arrow.** Retrieves the previous color (left) to edit.

## 7.3 Colors list

Displays the list of the 16 colors from the palette.

The color 0 is not allowed to edit.

## 7.4 Palette map

Show in map form, the 512 colors from the palette of V9938.

By placing the cursor over a color shows a tooltip with the color values and if you make a click, is assigned to the color you are editing.

## 7.5 Sliders

There are four sliders to adjust a color:

- **Tone slider.** A color tone slider to change the current color.
- **Red slider.** Slider to modify the red component.
- **Green slider.** Slider to modify the green component.
- **Blue slider.** Slider to modify the blue component.

## 7.6 Confirmation palette buttons

**Accept.** Return to the main window.

**Cancel.** Return to the main window without making changes to the palette.

## 8 Code output

MSX Tiles devtool, allows you to take any of the three types (tileset, map or palette) independently, in a list of data for different programming languages.

The output includes comments with data that can be useful when you want to access them from our program.

The screenshot shows the MSX Tiles devtool interface. At the top, there are three tabs: "Tileset", "Screen map", and "Palette". The "Screen map" tab is selected. Below the tabs, there are four icons: a folder icon, a file icon, a "Map" button (highlighted in green), a "Binary" button, and a "TXT" button. The main area displays the source code for a map file named "41ru6\_init.scr". The code includes comments for VRAM address, start/end coordinates, and RLE compression. The map data is represented as an array of unsigned char values. On the right side, there are configuration options: Language (C), Num sys. (hex 0xFF), Field name (SCR00), Size Line (16), and Compress (RLE WB). At the bottom right, there are input fields for StartTile (x,y) (0, 0) and EndTile (x,y) (31, 23), and a "Do it!" button.

```
// map 41ru6_init
// VRAM address= 1800h
// start x=0 y=0
// end x=31 y=23
// RLE WB compressed - Original size= 768 - Final size= 348
unsigned char SCR00[]={
0x00,0x81,0x00,0x01,0x00,0x0D,0x02,0x03,0x00,0x11,0x00,0x04,0x06,0x07,0x08,0x09,
0x0A,0x15,0x16,0x17,0x18,0x19,0x1A,0x1B,0x1C,0x05,0x00,0x11,0x00,0x04,0x0B,0x0C,
0x0D,0x0E,0x0F,0x1D,0x1E,0x1F,0x20,0x21,0x22,0x23,0x24,0x05,0x00,0x10,0x00,0x36,
0x37,0x10,0x11,0x12,0x13,0x14,0x25,0x26,0x27,0x28,0x29,0x2A,0x2B,0x2C,0x38,0x00,
0x10,0x36,0x0C,0x00,0x00,0x00,0x0D,0x40,0x01,0x00,0x10,0x0C,0x0D,0x02,0x00,0x0D,
0x40,0x03,0x00,0x10,0x0D,0x0E,0x04,0x00,0x0D,0x40,0x05,0x00,0x10,0x0E,0x0F,0x06,
0x00,0x0D,0x40,0x07,0x00,0x08,0x0F,0x13,0x13,0x14,0x13,0x13,0x00,0x03,0x0F,0x10,
0x08,0x00,0x0D,0x40,0x09,0x00,0x04,0x10,0x19,0x1A,0x1B,0x1C,0x29,0x2A,0x2B,0x2C,
0x2D,0x00,0x03,0x10,0x11,0x0A,0x00,0x0D,0x40,0x0B,0x00,0x04,0x11,0x1D,0x1F,0x1E,
0x1E,0x2E,0x2F,0x30,0x31,0x32,0x00,0x03,0x11,0x12,0x28,0x00,0x0D,0x40,0x27,0x00,
0x04,0x12,0x1E,0x20,0x21,0x22,0x15,0x16,0x17,0x16,0x16,0x00,0x04,0x12,0x28,0x00,
```



## 8.1 Basic settings

This output can be configured in the panel on the right.

First find the programming language selector, with these possibilities:

- Basic (MSX)
- C
- Assembler
- Assembler from SDCC

If you select Basic, on your right, enable a button that will open a window where you can set the start line, the interval and if you want us to delete the zeros (provided that the list takes up less memory).

Below you will find the numeral system selector with a list of different formats to support different interpreters and compilers.

When you select a language, this selector will be changed to a default value.

These are the different supported formats:

- decimal nnn (default Basic)
- decimal nnnd
- hexadecimal FF
- hexadecimal 0xFF (default C & ASM C)
- hexadecimal \$FF
- hexadecimal #FF
- hexadecimal 0FFh (default Assembler)
- hexadecimal &HFF

The next field is used as a label for assembler or variable name for C.

Then you will see a selector for you select the amount of data per line. The values depend on the data type you have selected.

The following fields are specific to each type, to specify the data you want to obtain. The detail in the next points.

Finally you will find the "Do It!" button, which generates the required code.

## 8.2 Compress

The Map and the tilesets, you can take in Raw format (uncompressed), or two types of RLE compression.

Have a basic RLE, which uses two values: the first as a control digit: indicates the amount to be repeated or if a value is zero marks the end; and the second contains the value. If it contains many values that are not repeated, the result may be greater than without compression. It is often useful to compress the color table. (More info: [http://en.wikipedia.org/wiki/Run-length\\_encoding](http://en.wikipedia.org/wiki/Run-length_encoding) )

The next is the WB RLE (RLE WonderBoy SMS). Is optimal because a byte is not lost through non-repeating value. (More info: <http://aorante.blogspot.com.es/2014/06/compresion-rle-sms-wonder-boy.html>)

**Tip:** To make more effective compression, try the "Optimize" tool.

**Note:** In the case of using a compression mode in the map, is effective only if you select an area with maximum width (32 tiles), because the unpacker works in consecutive blocks of data.

## 8.3 Tilesets

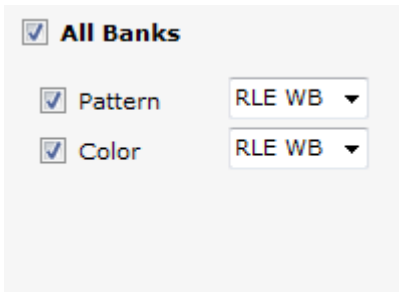
It has two modes: All or configurable.

You can change the mode using the checkbox "All Banks".

### 8.3.1 All Banks Mode

From here you can enable / disable the data type you want to make: patterns and/ or colors; and indicate the compression mode you want for each.

It is useful for screens to obtain a good degree of compression. In programming you just have to dump one or two blocks (patterns and / or color) to the VRAM.



### 8.3.2 Configurable Mode

In this mode, you can indicate that banks want to go out. Default enabled is first. You'll find a checkbox to enable the bank in each tab.

You can define the range of tiles of each bank from two numeric fields.

You can also specify which types of data you want (patterns and colors) and their compression. Default is Raw.

☐ **All Banks**

Bank0 Bank1 Bank2

☒ Bank0 0 to 255


☒ Pattern RAW ▼

☒ Color RLE WB ▼

## 8.4 Map

You can extract the entire screen (by default), or you can specify a smaller area. You have four fields where you can enter the coordinates of the corners.

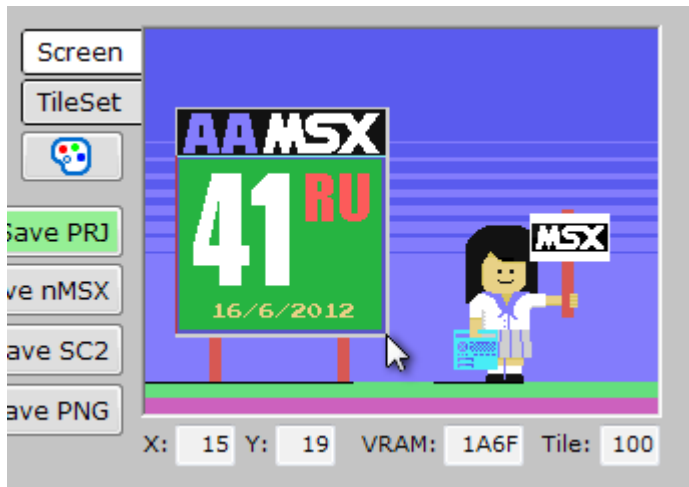
StartTile (x,y) 2 8

EndTile (x,y) 14 18 

You can also do this directly by selecting the area with the cursor on the screen viewer.

### How to?

From viewer in "Screen" mode by clicking on the upper left corner and dragging the cursor to the bottom right corner. Displayed a box in the viewer and update the values of the fields.



To initialize values (32x24), you can press the reload button or click on viewer in "Screen" mode.

## 8.5 Color palette

If you select Basic, you get three values per color (RGB), so that you can assign using the COLOR function.

In other languages you get two values per color (RB, G), so you can set from the VDP registers.

## 8.6 Save Output

The output can select and copy/paste over your editing application (context menu) or can save it to a text file by clicking on the save to TXT button.

## 9 Save

Finally, you can save your work in whole or in different parts, in any of the supported formats.

### 9.1 Save a project

If you want to save screen you can do it in the following formats:

- Native project. Includes palette (XSCP).
- nMSXtiles project.
- MSX Basic binary (SC2). VRAM dump from 0 to h1FFF. Include V9938 palette (VRAM: h1B80).
- PNG bitmap from display selected (tileset or Map).

### 9.2 Save a Tileset

You can save the whole tileset banks in XTIL native format.

### 9.3 Save a Map

You can also save the map screen in XMAP format.