**Amazfit Bip WF Editor by Ilgruppotester**

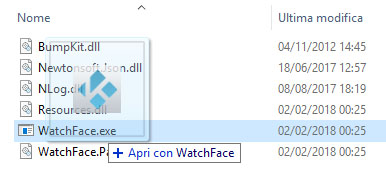
**Guide to Use**

**Prerequisites**

* VCRedist libraries 2008
* watchface.exe version 1.0.2.8  to extract the bin files and / or create them once the editing is finished
* <https://bitbucket.org/valeronm/amazfitbiptools/downloads/>

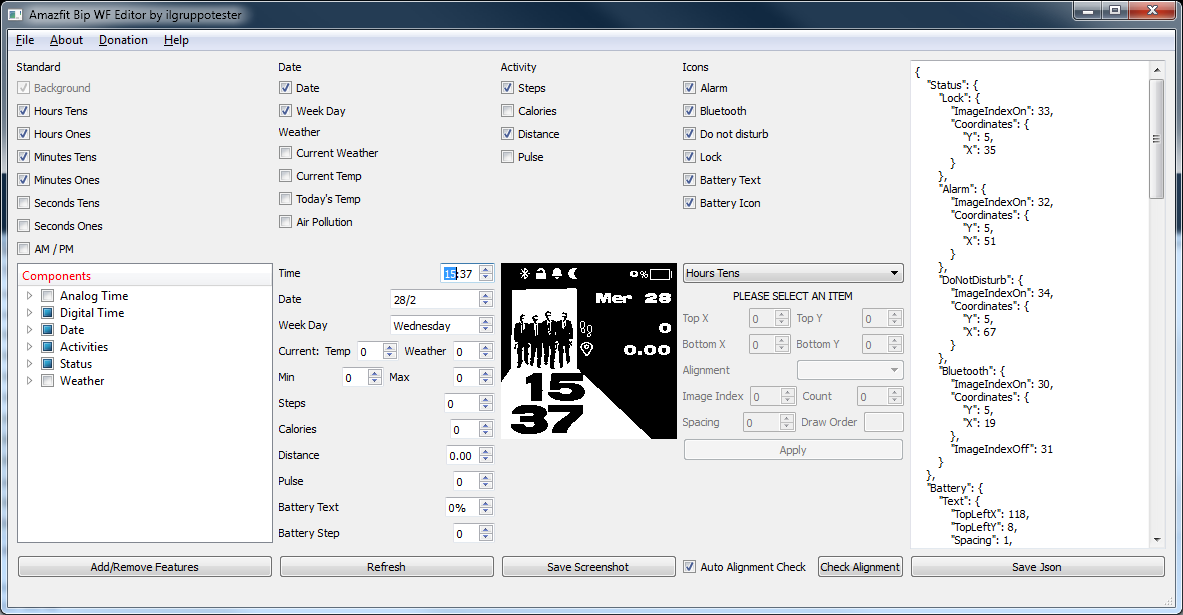
**Before starting**

- If you want to modify an existing watchface, you must first extract the contents of the .bin file. Once watchface is unpacked, simply drag the bin file over the executable icon



* - In case you want to create a new watchface instead, no special operations are required

**User Interface**



***Overview***

**

***Checkbox area*** *to enable / disable the display of an object. In the case of icons with On / Off properties, the checkbox has three states: object not displayed, status icon OFF and status icon ON*

**

***Components tree****: Makes it possible to add / remove objects from the json file. To make changes, select / deselect the desired objects and click on the Add / Remove Features button.*

**

***Preview Settings****:Area where the values shown in the preview can be changed.Warning: some objects are related to the set value, such as Step progress: as long as the number of steps entered does not allow reaching a level that can be drawn, there will be no visual feedback. Advances are calculated on the standard goal of 8,000 steps. Pressing the Refresh button, the new preview is generated*

**

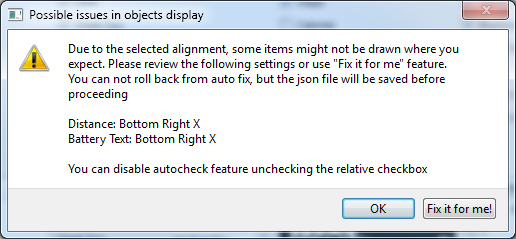
***Preview****: the preview of the result is displayed here.Clicking on the Save Screenshot button will save a preview jpg file in the same folder as the bin file*

**

***Edit Area****: by selecting an object from the dropbox, you can change the available properties. By clicking on theApply button, the new values will be inserted into the json and an updated preview will be generated.*



***Alignment Check Area****: When the Auto Alignment Check checkbox is selected, each refresh will be checked for alignment. If disabled, you can force a manual check by clicking on the "Check Alignment" button.*



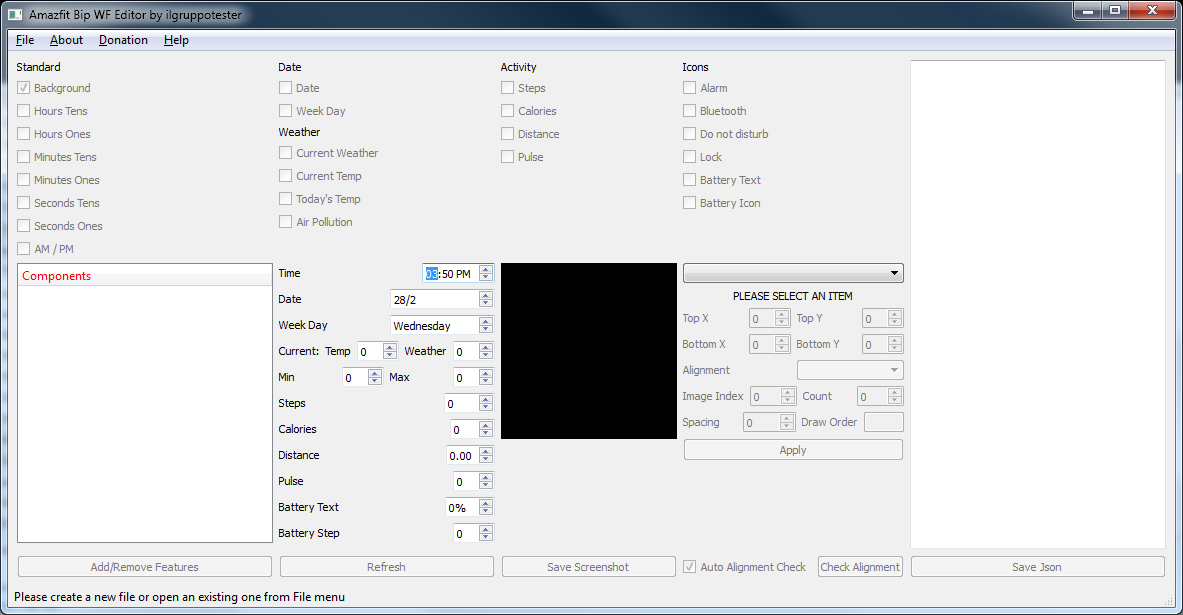
***Alignment Error*** *In the event that possible errors in the alignment are detected, a message similar to this is shown, reporting the objects for which there may be the problem.It is possible to solve them automatically by clicking on the "Fix it for me!" Button or click on OK to ignore the message. For more details, see Alignment in "Useful Information"*

**

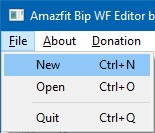
***Json Editor****: It is also possible to intervene manually on the Json file (this is not recommended unless you know what you are doing). The Save Json button is used to save the progress of the changes.* ***There is no automatic saving operation, so it will be necessary to manually save the file.***

**Instructions**

Unzip the editor zip file into any folder and run the executable "Amazfit\_Bip\_WF\_Editor\_ilgruppotester\_vxxx.exe"

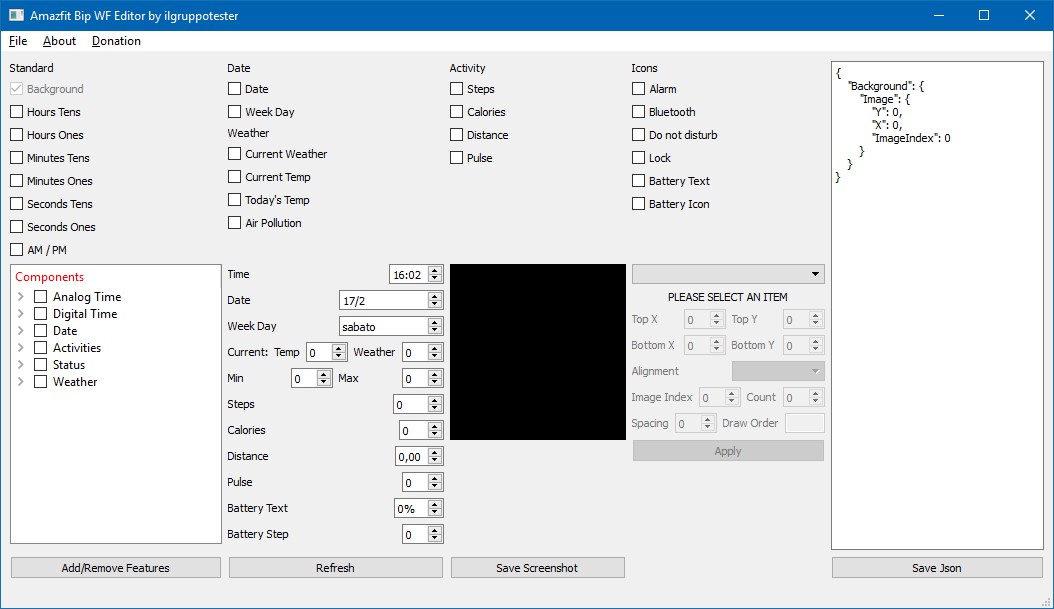


At startup, a clean interface will open, from which you will have to choose whether to create a new watchface or edit an existing one, via the "File New" menu or "File Open"

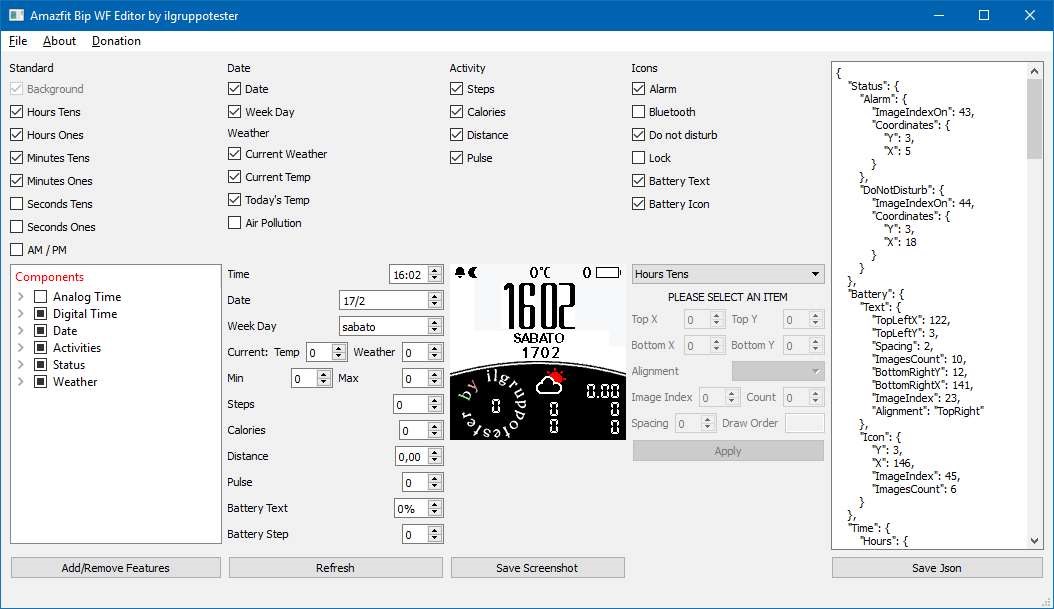


In case of new wf, you will be asked to select the folder where to create the json file, in case of opening an existing one, instead, you will have to select the desired json file.

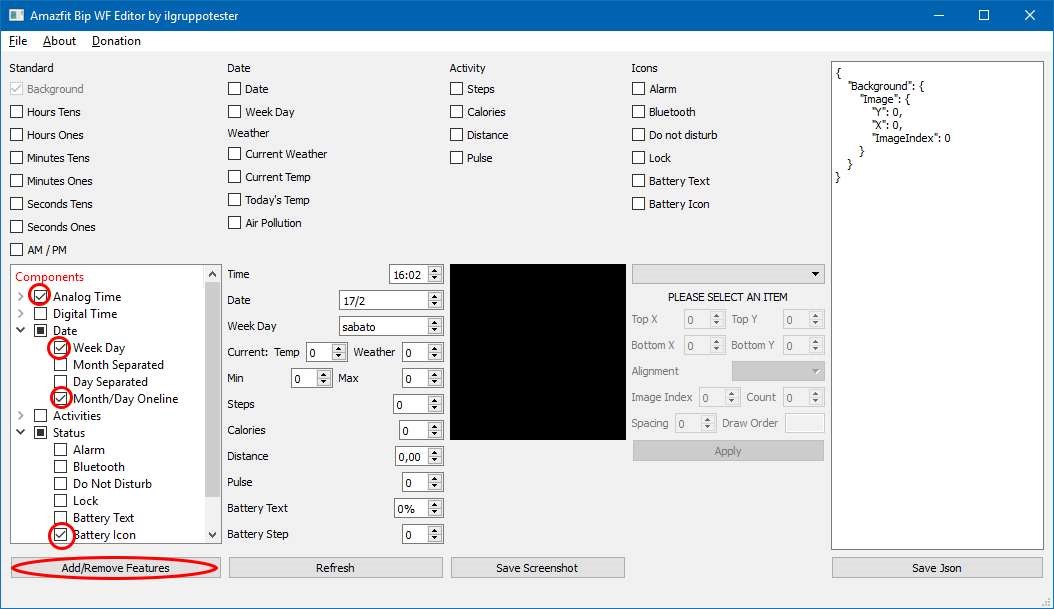
If a new wf is created, a background file will be copied to the folder and its part of the json will be populated.

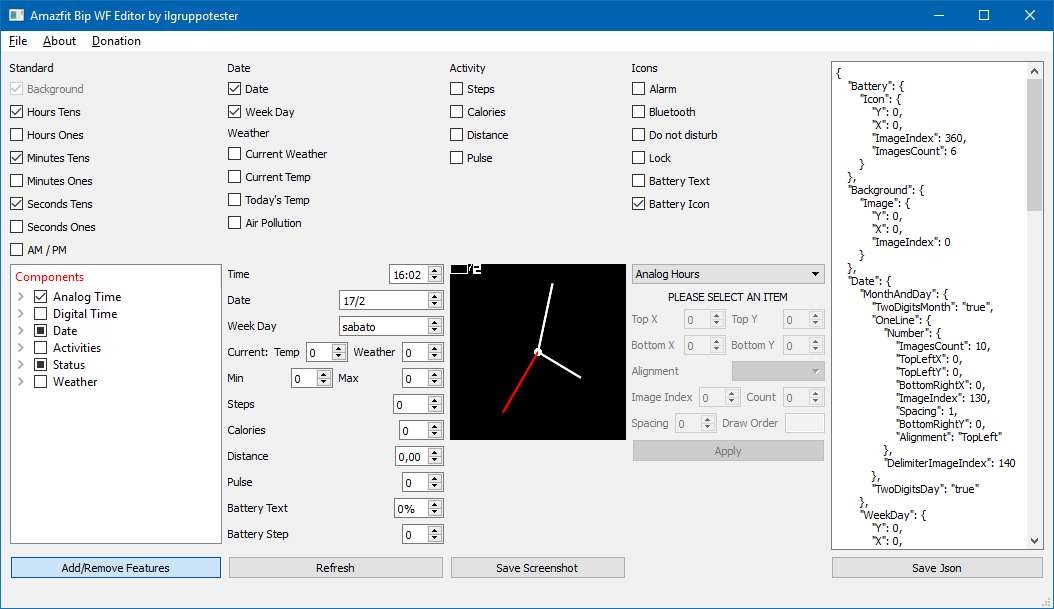


If instead an existing json is selected, all the elements present in the wf will be loaded and displayed, in addition the json section will be populated with the entire contents.

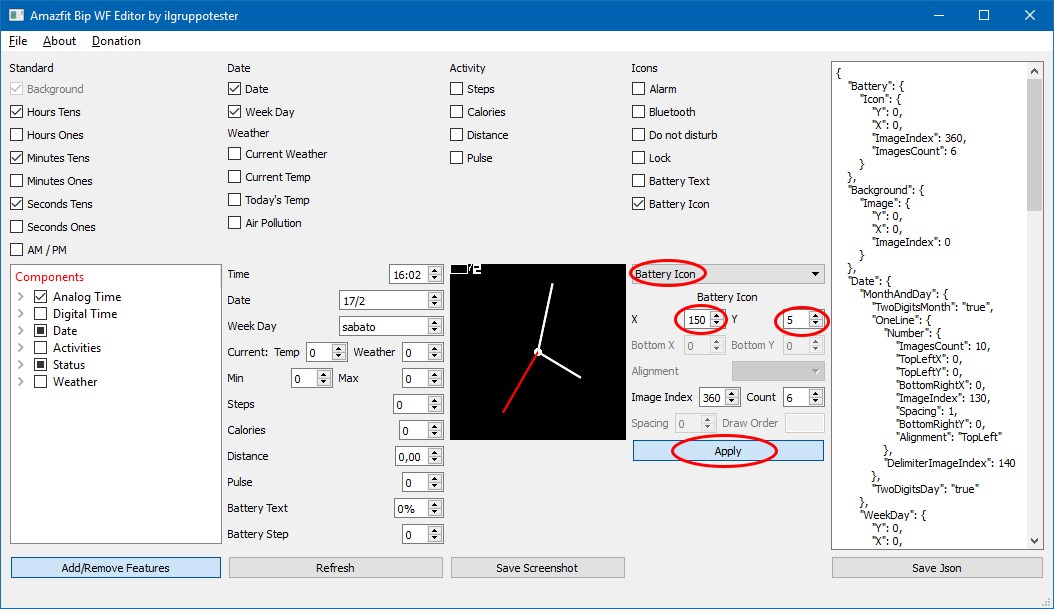


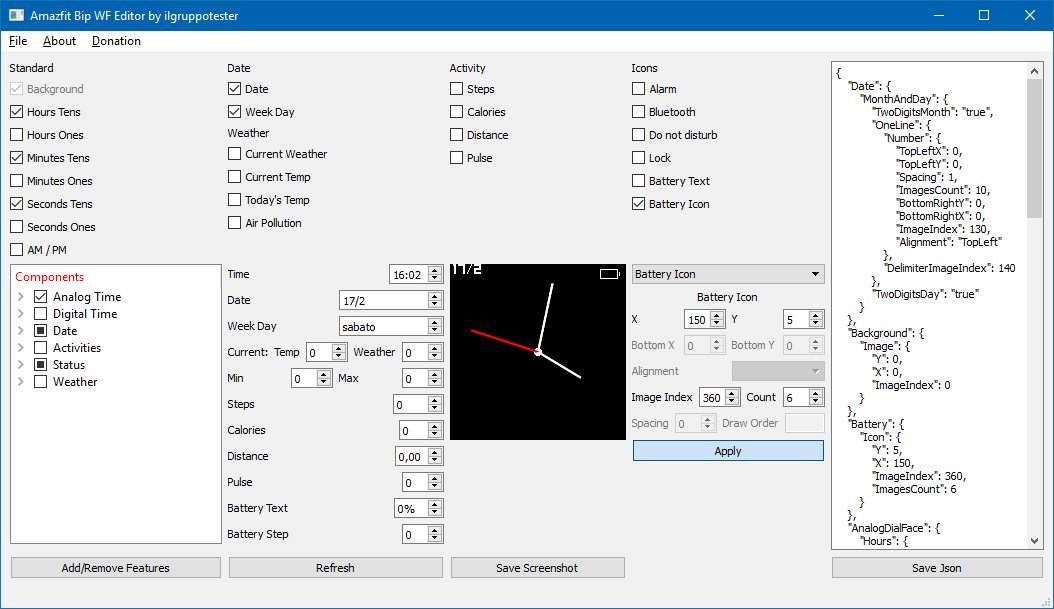
From this moment on, we proceed in the same way for both cases:

you can add and remove objects by simply selecting or de selecting them from the tree view "Components" and clicking on the "add / remove features". In case of addition, the default images will be copied to the json folder, so that you have immediately visual feedback. 

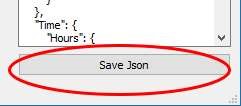


By selecting an object from the dropdown list instead, the related properties will be displayed and, once set, by clicking on the "Apply" button, the data will be entered in the json and the updated preview will be shown.

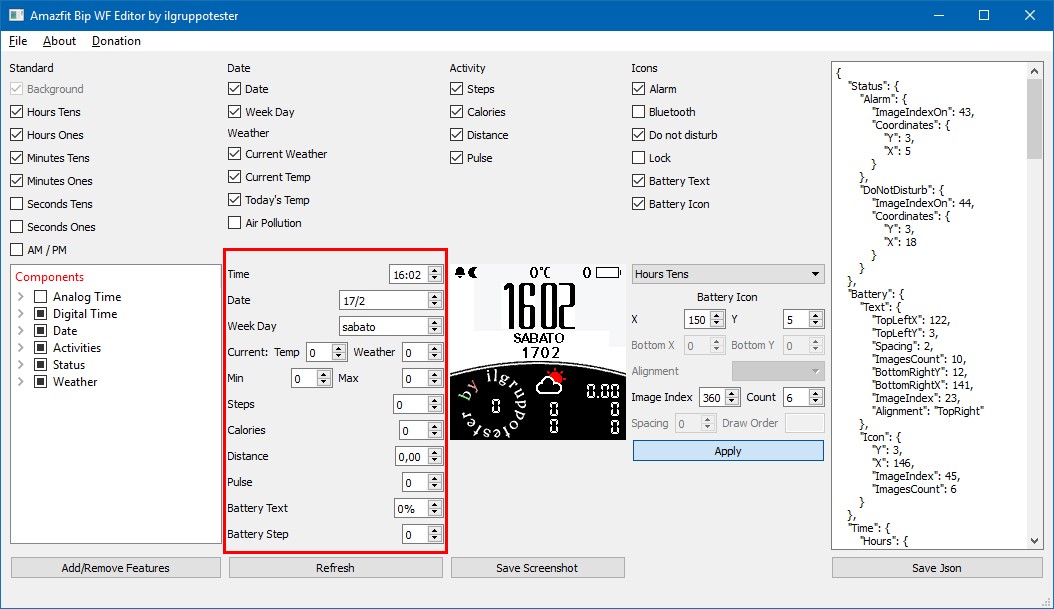


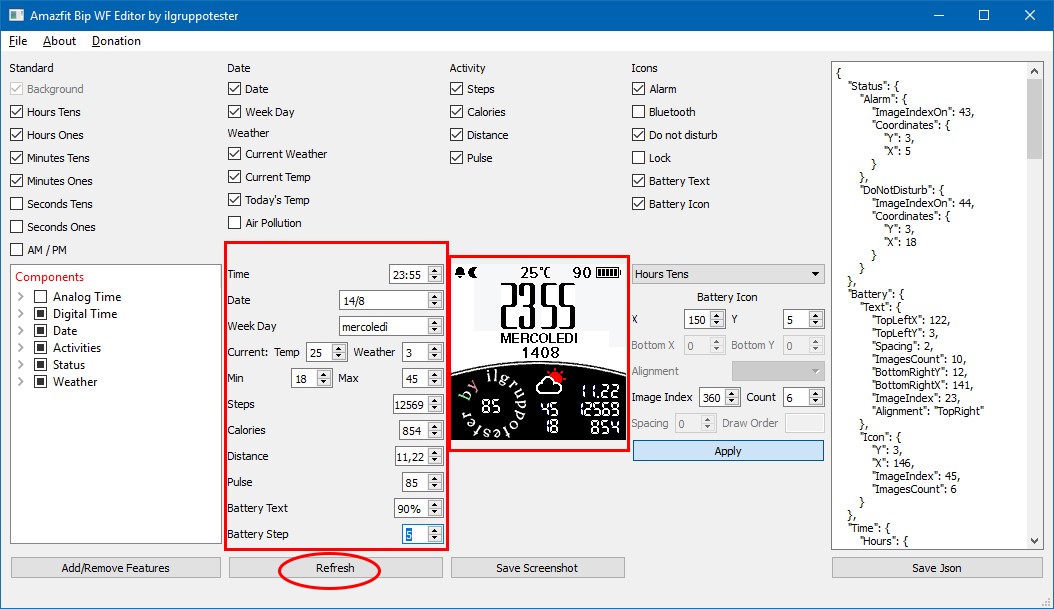


**To remember, the json is not saved automatically, but you will need to click on the "Save Json" button.**

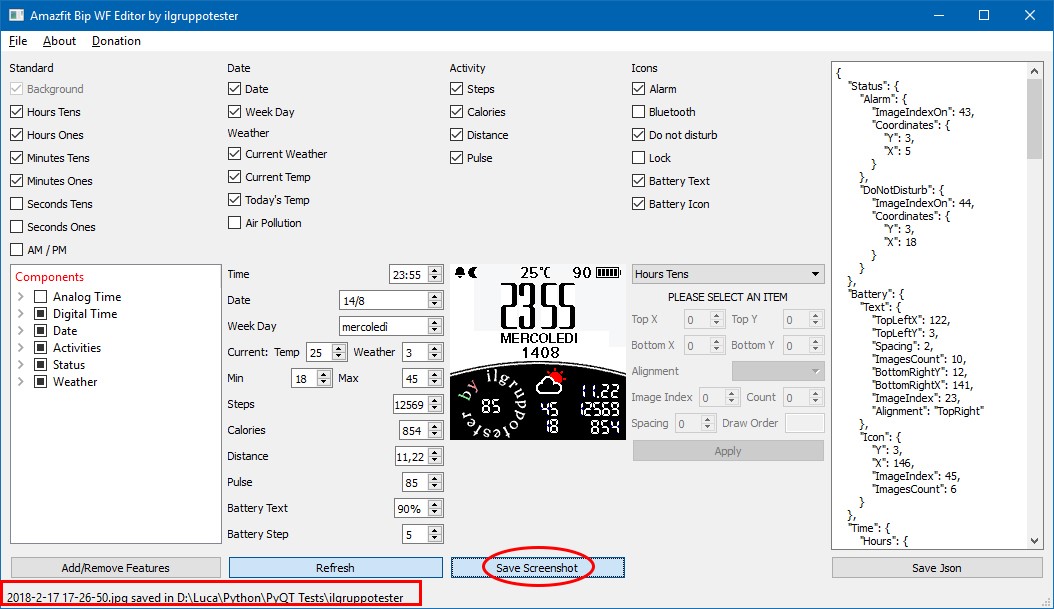


To get a detailed preview of the result, you can change the values ​​displayed through the appropriate area, so as not to have any nasty surprises when the values ​​change on the smartwatch.

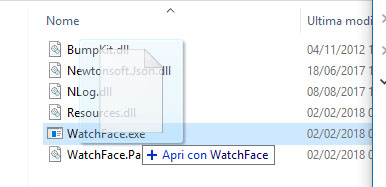




It is also possible to save a screenshot of the watchface by clicking on the "Save Screenshot" button: a jpg file will be created in the same folder as the json.



To create the .bin file to be loaded on the beep, just drag the json file onto the watchface executable.



**Useful Information**

Do not want to be an exhaustive guide, but just a smattering of how the most used items in watchfaces. The internet is full of complete guides, for example a more interesting one (in English) can be found here: <http://amazfitcentral.com/2018/02/05/amazfit-bip-watchface-reference-including-some-json-docs/>

**General**

**Useable Colours**

The palette of colors that can be used on the BIP is the following, if in the images there are other colors or other shades, watchface.exe will use the colors supported to return the visual effect of those not supported (dithering) when creating the bin file.



**Images**

It is possible, and recommended, to modify all the images provided as samples or extracted from existing bin files , or to create new ones, through an image editor such as GIMP or Photoshop. Also for this process there are many guides on the net, and the advice is to try, as the satisfaction of having a watchface with the fonts that best suit your taste is priceless.

**Background**

The background image must be a png of 176x176 pixel size



**File Json**

The Json file contains information related to all the elements present in the watch face, from the image to be used to the alignment, and are different for each object. The most common ones are:

**X, Y**

"X": 0, # X coordinates of the top left corner of the image

"Y": 0, # Y coordinates of the upper left corner of the image

In the case of single images, such as status icons, the hour digits ... the X, Y coordinates indicate the point on the screen from which the object will be drawn

**ImageIndex and ImagesCount**

"ImageIndex": 1, # Nome del file della prima cifra, in questo caso 001.png

"ImagesCount": 10 # Numero di immagini da utilizzare partendo dall’indice ( 001.png - 010.png)

Used for example in the hour and in the day of the week, it indicates the index of the first image and how many are the images to use starting from that

**TopLeft, BottomRight X,Y**

"TopLeftX": 102, #Coordinate X of the top left corner

"TopLeftY": 60, #Coordinate Y of the top left corner

"BottomRightX": 170, #Coordinate X of the lower right corner

"BottomRightY": 70, #Coordinate Y of the lower right corner

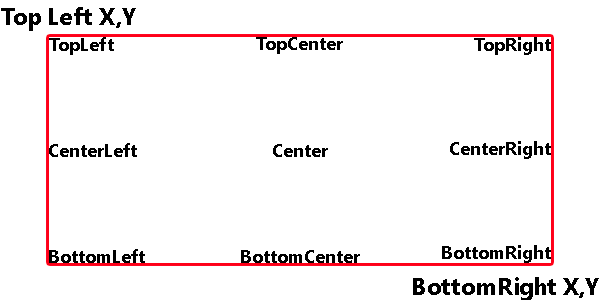
"Alignment": "BottomRight", #Alignment, See the next section

"Spacing": 2, # Space in pixels between each digit

Used in elements composed of several figures, such as battery percentage, number of steps ...Remember: BottomRight X must be greater than TopLeftX, same thing for BottomRightY and TopLeftY

**Alignment**

For objects consisting of several digits, such as the numerical percentage of the battery, the steps, the distance ... it is possible to choose how to align the text within a polygon having the coordinates from TopLeftX, TopLeftY to BottomRightX, BottomRightY. In case of different alignments from TopLeft, if the text does not enter the box, on the beep the text will be automatically aligned to TopLeft. For this reason, a warning message will be displayed when the information is incorrect



We will now analyse a watchface created when testing the editor:

**Background**

"Background": {

"Image": {

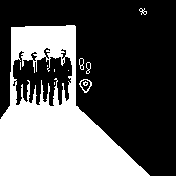
"X": 0, # X Coordinate of the upper left corner

"Y": 0, # Y Coordinate of the upper left corner

"ImageIndex": 0 # Name of the file, in this case 000.png

}

}



**Now**

"Time": {

"Hours": {

"Tens": { # Digit indicating tens of hours

"X": 10, #X Coordinate of the upper left corner

"Y": 110, # Y Coordinate of the upper left corner

"ImageIndex": 1, # Name of the first digit file ( the 0), in this case 001.png

"ImagesCount": 10 # Number oif images to be used starting from the index ( 001.png - 010.png)

},

"Ones": { # Digit indicating the units of hours

"X": 47,

"Y": 110,

"ImageIndex": 1,

"ImagesCount": 10

}

},

"Minutes": {

"Tens": { # Digit indicating the tens of minutes

"X": 10,

"Y": 142,

"ImageIndex": 1,

"ImagesCount": 10

},

"Ones": { # Digit indicating the units of minutes

"X": 47,

"Y": 142,

"ImageIndex": 1,

"ImagesCount": 10

}

}

},



**Activities**

**Steps**

"Steps": {

"TopLeftX": 102, #X Coordinate of the top left corner

"TopLeftY": 60, #Y Coordinate of the top left corner

"BottomRightX": 170, #X Coordinate of the lower right corner

"BottomRightY": 70, #Y Coordinate of the lower right corner

"Alignment": "BottomRight", #Alignment, see the section above

"Spacing": 2, # Spacing between the digits

"ImageIndex": 11, # Name of the first digit file

"ImagesCount": 10 # Number of images used starting from the image index

}

**Distance**

"Distance": {

"Number": {

"TopLeftX": 111,

"TopLeftY": 82,

"BottomRightX": 172,

"BottomRightY": 96,

"Alignment": "TopRight",

"Spacing": 2,

"ImageIndex": 11,

"ImagesCount": 10

},

"SuffixImageIndex": 21, # Index for the image of the unit of measure “Km”

"DecimalPointImageIndex": 22 # Index of the image to separate the units of distance from the decimals, generally “.” Or “,”

**Date**

**Day of the Week**

"WeekDay": {

"X": 92,

"Y": 30,

"ImageIndex": 23,

"ImagesCount": 7

}

**Day and/or month**

"Date": {

"MonthAndDay": {

"Separate": {

"Day": {

"TopLeftX": 130,

"TopLeftY": 30,

"BottomRightX": 170,

"BottomRightY": 40,

"Alignment": "TopRight",

"Spacing": 2,

"ImageIndex": 11,

"ImagesCount": 10

}

},

"TwoDigitsMonth": true, # If true, the month is always indicated with two digits. For example. January = 01

"TwoDigitsDay": true # If true, the day is always indicated with two digits. For example Friday = 05

**Status**

For objects in the Status category, we have two values that indicate the icon to be displayed if an object is active, and to display when it is not.

https://i1.wp.com/amazfitcentral.com/wp-content/uploads/2018/01/043-1.png?resize=12%2C14 https://i2.wp.com/amazfitcentral.com/wp-content/uploads/2018/01/044-1.png?resize=12%2C14

"Bluetooth": {

"Coordinates": {

"X": 19,

"Y": 5

},

"ImageIndexOn": 30, # Name of the file for the Bluetooth ON status icon

"ImageIndexOff": 31 # Name of the file for the Bluetooth OFF status icon

**Battery**

Regarding "Battery Text", the behavior is the same as the objects related to the activity

"Battery": {

"Text": {

"TopLeftX": 118,

"TopLeftY": 8,

"BottomRightX": 137,

"BottomRightY": 16,

"Alignment": "TopRight",

"Spacing": 1,

"ImageIndex": 35,

"ImagesCount": 10

},

"Battery": {

"Icon": {

"X": 149,

"Y": 5,

"ImageIndex": 45, # Initial image (Empty battery)

"ImagesCount": 5 # Number of images related to the various battery states

}

https://i0.wp.com/amazfitcentral.com/wp-content/uploads/2018/01/058-1.png?resize=25%2C11https://i2.wp.com/amazfitcentral.com/wp-content/uploads/2018/01/059-1.png?resize=25%2C11https://i2.wp.com/amazfitcentral.com/wp-content/uploads/2018/01/060-1.png?resize=25%2C11https://i0.wp.com/amazfitcentral.com/wp-content/uploads/2018/01/062-1.png?resize=25%2C11https://i2.wp.com/amazfitcentral.com/wp-content/uploads/2018/01/063-1.png?resize=25%2C11