Creating Custom Dashboards for your Nuvi GPS Review

Community contributed article **By Boyd Ostroff**

In 2012 Garmin added a new feature to the Nuvi line called “dashboards” — small files containing images and code that change the appearance of the main map screen. In the past, the Nuvi was criticized for its lack of user customization features, and the 2012 models sought to address this complaint with the additioin of dashboards and a couple other new features. But Garmin giveth and Garmin taketh away… the dashboard feature was removed from the 2013 Nuvi models. I believe the full list of dashboard-compatible devices would be:

Nuvi 24×5, Nuvi 25×5, Nuvi 34×0, Nuvi 35×0, Zumo 350, D?zl 760

I recently purchased a Nuvi 3550 and was anxious to explore the possibilities of changing the Nuvi interface by designing my own dashboards. The following tutorial draws on my study of the standard dashboard files included with the Nuvi plus lots of trial and error. I’ve never seen any official Garmin documentation for dashboards.

Which Nuvi models are compatible with the files in this tutorial?

This tutorial deals specifically with dashboards for the Nuvi 34×0 and 35×0 series (for example: Nuvi 3450, 3490, 3550, 3590). I would expect it to also apply to the D?zl 760 but can’t confirm this. Although the program code and files included here will not work on the Nuvi 24×5 or 25×5 series, the principles will be the same and I will include some notes about how to adapt them later.

Why would I want to make my own dashboards?

If you’re asking this question then you are probably in the wrong place! Some people might simply answer *“because I can.”*

But we spend a lot of time looking at the map screen. Maybe you just don’t like the way it looks? Perhaps you would like to see more of the map, with smaller buttons or less data? Maybe you want more information, such as arrival time, distance to the next via on the route, what direction you’re heading and the elevation of the mountain you’re climbing?

First, some words of caution before you proceed…

By modifying a dashboard file, you will be placing something on your Nuvi that Garmin has not approved or tested. Take a few minutes to think about whether you really want to do this. Do you depend on it for business? Did you have to work and save for months to afford it? Would you panic

if it didn't work when you turned it on tomorrow morning? Is this your only GPS? If you answered "yes" to any of these questions, this may not be for you.

In some cases a corrupt file can prevent the Nuvi from starting up (although I have never experienced this with a dashboard). If you create a home-made dashboard with the same name (id) as a standard Garmin dashboard, an error in your code may affect the standard Garmin dashboard in a way that isn't reversible.

And, of course, any time you start fooling with files on your Nuvi, there's a chance you'll delete the wrong file and cause a problem.

I have no "inside knowledge" about dashboards; everything in this article is just the result of trial and error combined with some educated guesses on my part. There's a lot here that I still don't understand.

Just be sure that you understand the risks before going down this road. If you consider your Nuvi "mission critical", this article may not be for you. If you decide to proceed, do a full backup of your nuvi now, before you make any changes. See this thread in the forums for instructions: http://forums.gpsreview.net/viewtopic.php?t=21019

To summarize, there's a chance – albeit a slim one – that this could damage your Nuvi. By proceeding with this tutorial, you take full responsibility for any problems that might occur as the result of using the files and techniques here. No warranties or representations of any kind are made regarding their suitability or stability. Caveat Emptor!

What you will need

If I didn't scare you away in the previous paragraph, then you've come to the right place! So let's get started with a short list of what you'll need. The good news is that you shouldn't have to buy anything. Editing dashboards is very straightforward and you only need three things:

(1) A text editor. It's important to always know the size of the dashboard file you're editing so I've been using the free Notepad++ editor.

(2) TurboCCC's free GIR_Editor program.

(3) A graphics program. This tutorial will be based on **Photoshop** since it's what I use myself. Actually, the screenshots included here are from **Photoshop Elements 6**. This is an old, simpler version of the full program but it has everything you'll need. **Photoshop Elements** has been included with scanners, printers and digital cameras for many years. You may even have a copy you don't know about on an old installation disk. If not, then have a look at the free gimp program.

Can I edit dashboards on my Mac?

Well... yes, and no. You can certainly edit the files in a text editor; TextWrangler is free and works well. The graphics aren't a problem either as we have Photoshop, Photoshop Elements, Gimp and other alternatives on the Mac. But there's no Mac equivalent to **GIR_Editor**, and that's what you

need to put your own images into a dashboard file. You can work through this tutorial on your Mac, but you will be limited to the images that are already included in the files.

So I'll assume you're using Windows for the tutorial. I'm a Mac user myself, going back to 1985. I'm actually writing this tutorial on my Mac. But for some things, you just have to use Windows.

What's a dashboard file?

There are two "flavors" of dashboards, corresponding with the screen resolution of the different Nuvi models: either 800×480 or 480×272. The file must match your device: an 800×480 dashboard file will not work on a Nuvi with a 480×272 screen. The structure of the files should be essentially the same, but the graphic elements will be differently sized and the coordinates specifying their location on the screen will also be different. Also note that 800×480 has an aspect ratio of 16:10 while 480×272 is a 16:9 screen. This is a relatively slight difference, but it does have implications for the proportions of objects between the two screen sizes.

Dashboard files contain three sections. The first is a header containing binary data. The second contains XML code as plain text – this is the part we will be modifying. The final, and largest section contains PNG images in binary format. These can be modified with TurboCCC's **GIR_Editor** program.

We don't really know how Garmin creates dashboard files, but I'm pretty sure they just use a text editor to write the XML code, gather all the images together and then process them with some simple program. Unfortunately, we don't have such a program. So any dashboards we create must be based on the skeleton of an existing Garmin dashboard. The end result will be completely different from the original Garmin file, but the structure will remain the same. This is a serious limitation, and it makes things more difficult. But there's still a lot of room to create new and different dashboards.

Since these are binary files, we must take care when making changes to the XML code in the second section. If our new code contains a different number of bytes than the original Garmin code, the file will be broken. A broken file can manifest itself in a couple different ways on your nuvi. Most of the time, it will just be ignored and the Nuvi menu will show the default dashboard in its place. Sometimes a broken file will be displayed but won't work properly, showing the wrong images or none at all. Part of the discipline of creating a new dashboard is carefully monitoring the file size before and after making any modification to the code and insuring that it doesn't change. This is done by either adding or deleting characters from a section of "filler" in the XML code.

More about this later.

Getting started: “Hello World”

dashboard_download and un-zip the file. You should now have a folder containing three files. We will start with the file **HelloWorld_800x480.dshb**. There’s an old computer tradition of writing a very simple first program when you learn a new language. Such a program usually just displays “**Hello World**” on the screen, then quits. Our first dashboard doesn’t quite do that, but it’s about the simplest functional example I could think of.

A dashboard must have a **back button** that allows you to close the map screen and return to the main menu. A dashboard also must have a **menu button** because the 2012 Nuvi’s have no way to cancel a route in progress without one. I believe that a background image is also a requirement, although I haven’t tried one without. In any case, it’s an important concept so I’ve included it here.

So **HelloWorld** will only provide these three elements. It does not show any data on the screen, such as your speed or arrival time.

Let’s start our tour of the dashboard file by opening it with the **Notepad++** editor. Since we’re going to be opening dashboard files frequently, you might want to set **Notepad++** to open .dshb files using **Default Programs > Associate a file type or protocol** in the Windows Control Panel. This will allow you to double-click a dashboard file and have it open in **Notepad++**.

Setting the dashboard id and screen resolution

Scroll past the binary data that looks like a bunch of gibberish to the beginning of the dashboard code where you’ll see:

```
        id="hello"
        software_version="1.0"
        version="1.0"
        screen_width="800"
        screen_height="480">
```

If you create your own dashboard it’s very important to create new id. Installing more than one

dashboard with the same id can cause serious problems. Dashboard preferences (data you choose to display in each field) are stored by the Nuvi internally. If two dashboards have the same id, these preferences can become corrupted beyond repair such that the dashboards will no longer work properly. Neither a hard reset nor reinstallation of the dashboard file will correct this kind of problem.

The version numbers are not significant, you can use anything you like. The screen size is important so that a Nuvi with a 480×272 screen knows this file won't work.

Specifying the boundaries of the dashboard

Now look at the following declaration, which states that your dashboard can place objects anywhere on the Nuvi screen. You could use smaller values if you only want the dashboard at the bottom of the screen, but I don't see any downside to defining the whole screen. If you define a smaller size but then place an object outside of those bounds, some "interesting" things will happen (and they aren't good)!

```
        <LandscapeRect x="0" y="0" w="800" h="480" />
        <PortraitRect x="0" y="0" w="480" h="800" />
```

Placing the vehicle on the screen

The values for Car specify how far from the bottom of the screen the "vehicle" will be displayed. This only affects track-up and 3d mode; north-up mode always centers the vehicle on the screen. But this setting may not do quite what you'd expect. It doesn't show any more of the road ahead when you move the vehicle closer to the bottom of the screen. You will still see the same distance between the vehicle and the top of the screen regardless of this setting, but the view zooms in closer as the value of this parameter decreases. Also note that you can set different values for Landscape (Land) and Portrait (Port) views – a common theme in most dashboard parameters.

```
        <Car Land="120" Port="185" />
```

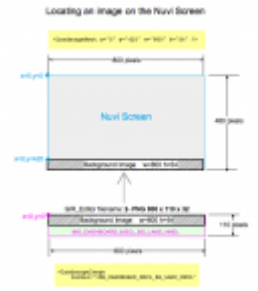
The background image

Now we come to our first dashboard component, the background image. The coordinate system of the Nuvi screen places 0,0 in the top left corner. Our landscape background image is 54 pixels high by 800 pixels wide. We will locate the top left corner at x=0, y=425 on the landscape screen. This places the bottom of the image at the bottom of the screen (y=479).

We'll do the same kind of thing with the portrait image which is 480 pixels wide x 54 pixels high, placing the top left corner at x=0, y=745.

```
<ImageComponent id="BackgroundImg" action="none">
        <Meta>
                <LandscapeRect x="0" y="425" w="800" h="54" />
                <PortraitRect x="0" y="745" w="480" h="54" />
```

```
        </Meta
```



It's important to note here that the width and height parameters here don't necessarily need to match the actual size of the image file you're using. This illustration shows the relationship between an image and its representation on the Nuvi screen. In this example, the image file is actually 800 x 110 pixels but we're only using and area 800 x 54 pixels as the background. Remember, 0,0 is the top left corner of the image, so when you create images, position them at the top left corner of an existing image in the dashboard file

Specifying which image files to use

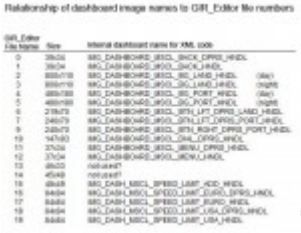
The relationship between the files you see listed in GIR_Editor and the image names shown in the dashboard code is rather bewildering.

```
<BackgroundDisplay>
        <LandscapeImage
                normal="IMG_DASHBOARD_MSCL_BG_LAND_HNDL"
```

Where does the name "**IMG_DASHBOARD_MSCL_BG_LAND_HNDL**" come from? Frankly, that's a mystery I haven't solved! This value is hard coded into the dashboard file somehow, but a search with a hex editor doesn't reveal where.

As stated earlier, we can't build our own dashboards from scratch, they must be based on an original Garmin file. All my examples are built from the **GarminMuscle_800x480** dashboard. Why? Because it has the largest XML code segment, giving us the most room to create our own dashboard code.

We also can't change the number of images contained in a dashboard file or the resolution of the individual images. The **GarminMuscle** dashboard has a good mix of different sized images and should provide lots of opportunity for customization. However we must always reference the images by their original names in the code.


Relationship of dashboard image names to GIR_Editor file numbers

To make it easier, I have prepared the following table that shows all available images and their corresponding names when viewed in **GIR_editor**. You might want to print this up as a future reference when creating a new dashboard.

Day and night images

Dashboard backgrounds use two sets of files: one for day and another for night. You cannot specify a day or night image explicitly, it seems to be hardwired into the background image. There are a total of four different images associated with the dashboard background: landscape day, landscape night, portrait day and portrait night.

Specifying how the image interacts with the user

All images can behave like “buttons” that have several modes. Normal is what you will see.... normally. Depressed will briefly show when you tap a screen button. I have not explored the use of the highlight or disabled attributes yet. I assume they would let you highlight a “default” button to make it stand out or “grey out” a button that is currently unavailable. I haven’t tried the tiling attribute either, but assume it lets you fill a large area with repeating tiles that consist of smaller images.

In the case of our example, most of this is irrelevant, because the background image isn’t a “button”

```
<ImageComponent id="BackgroundImg" action="none">
```

Action=”none” means that nothing will happen if you tap this image on the screen. We’ll explore other possible values for **action** later. Finally, notice that the values associated with **depressed**, **highlight** and **disabled** are all:

```
"BUILT_IN_NO_IMAGE_HNDL"
```

This is the equivalent of a null value for an image. Use it where you don’t want to have anything displayed but you still need to specify a value.

Placing a button on the screen

We now turn our attention to the two screen buttons in the **HelloWorld** dashboard. We’ll dissect the **Back Button**, and you can apply the same concepts to the **Menu Button**.

```
<ImageComponent id="BackBtn" action="back">
        <Meta>
                <LandscapeRect x="0" y="425" w="56" h="54" />
                <PortraitRect x="0" y="745" w="56" h="54" />
        </Meta>
```

Action=”back” tells the Nuvi to return to the main menu screen when this button is tapped. **Action=”menu”** takes you to the menu screen containing items you can use while navigating.

 The functions of the **LandscapeRect** and **PortraitRect** were covered above and work just like they did with our background image. But the images associated with these buttons are much smaller: 56 pixels wide x 54 pixels high. If the screen is tapped anywhere within this area, the **action** command is executed, taking you to the main menu. The same concepts mentioned earlier apply here as well. The image dimensions don't have to be the same as the full size image file. In fact, we are using an 84 x 84 pixel file to store our 56 x 54 image. Once again, you need to line your image up with the **top left corner** of the larger file. Here's what the **Back Button** looks like in Photoshop.

Providing user feedback when a button is pushed

Using the table provided above, you will see that **GIR_Editor** image number 16 is being used for the normal mode of the Back Button

```
normal="IMG_DASH_MSCL_SPEED_LIMIT_EURO_DPRS_HNDL"
```

But when you tap the button, you'll see a brief flash of another image. Consulting the table again you will note that this image come from GIR_Editor file number 18

```
depressed="IMG_DASH_MSCL_SPEED_LIMIT_USA_DPRS_HNDL"
```

 This is what is shown when the button is pressed. Compare it with the image of the **Back Button** above.

The **Menu Button** works similarly. Consult the table again to see which image files are actually being used. Since we can't change the size or names of the existing images, this unfortunately makes the code a bit cryptic. When designing a dashboard, you'll generally want to pick the original images that are as close as possible to the size of your new images. This is an unfortunate reality that I haven't found a way to avoid. It sure would be nice if we could change the names of the files and the resolution of the images, but we can't.

Exploring the graphic elements of the dashboard.

 This concludes the tour of the XML code used for **HelloWorld** dashboard. Now let's have a look at the images that are used and how they were created. Start up the **GIR_Editor** program and open the dashboard file. You should see a window like the one shown below. This is a very simple program with only a few functions.

Clicking once on an image shows a preview in the upper right corner of the window. Right-clicking an image allows you to extract it from the dashboard file, so that you can modify it with **Photoshop**. You can also select a group of images at the same time, or choose to extract all of them.

Preparing an image for use as a button.

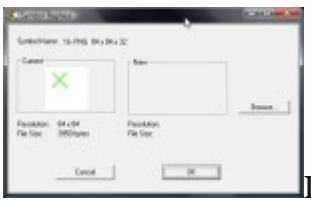
Let's examine image number 16 – the **Back Button**. Extract the image and open it in **Photoshop**. As discussed above, we're forced to use the same resolution image files as the original Garmin dashboard, but we don't have to use them in the same way. Image 16 is 84 x 84 pixels but our **Back Button** is only 56 x 54.

The "Info Palette" in **Photoshop** is your friend when you work on dashboards. It shows your cursor coordinates and also the width and height of the area you have selected. These numbers correspond with the x, y, w and h values we learned about in the XML code. If you look at the **Photoshop** screenshot posted earlier you can see that the back button was created in such a way as to be centered in a 56 x 54 rectangle.

Using transparency in dashboard buttons and images

Notice the checkerboard pattern in the screenshot. This indicates a transparent area of the image. Transparency is an extremely important concept to understand in regard to background images. It allows us to place irregularly shaped graphic elements on top of each other. In this case, the "X" symbol of the Back Button will be drawn on top of a rectangle in the dashboard background image. Explaining transparent images is beyond the scope of this article, so do a Google search if you aren't already familiar with the idea.

Replacing dashboard images

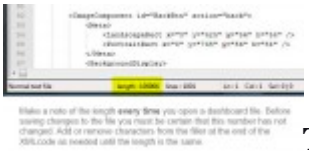
 Let's say we made a few changes to the back button in **Photoshop** and we now want to insert the new button into the dashboard file. This is done with **GIR_Editor**.

Double-click image #16 in the program. A Symbol Replace windows opens, showing a preview of the current image. Click the browse button to find your new file. The old image will be replaced with the new one when you save the dashboard file with **GIR_Editor**.

File size may change

If you replace some of the images in the dashboard file, you may notice that its size changes. This is to be expected because some images don't compress as well as others. If you replace a solid red square with a detailed photograph for example, the file will get larger.

Don't confuse this phenomenon with the earlier discussion of maintaining the same file size when changing XML code. The **GIR_Editor** program will adjust the dashboard file as needed to accommodate the new images.



This highlights the need to always record the size of the file before making any changes to the XML code. Let's say you write the file size down while using the text editor, quit, and then replace some images with GIR_Editor. The next time you open the file in **Notepad++**, it will be a different size and you need to make a note of this before changing any XML code.

Wrapping up the tour

Take a few minutes now and extract the other images from the **HelloWorld** dashboard. Notice the resolution of the image file itself and how much of the image is actually being used in the dashboard. Use the selection cursor in **Photoshop** to see the coordinates of the image elements, then find the corresponding section of the dashboard code in **Notepad++**. Do you understand the relationship between the two? If not, then study the images and code a little further. Eventually it will all make sense to you.

Putting it all together

First, let me again emphasize the importance of making sure you don't change the size of the dashboard file when you make changes.

Let me also stress the importance of assigning a new id to any new dashboards you create. Bearing this in mind, study the flowchart below. You can prepare the images in parallel with writing the dashboard code if you like. J**ust be sure to save and close all the image files and the Notepad++ file before assembling them in GIR_Editor.**

You'll then copy the file to the Nuvi and see how it works. If you only see the default Nuvi dashboard, there's either a syntax error in the XML code or the file size has been changed. You will then need to backtrack and fix the problem. This is why you want to keep saving the files with different names as you work, making it easier to step backwards and fix the problems.

The process can be slow and frustrating at times, but it will be a lot smoother if you work in small steps. Only change a few things at a time, then test the new file. If you make a lot of changes at the same time, it will be harder to find the bugs. As you gain experience and confidence, you can work in larger "chunks", but start small.

Adapting the HelloWorld dashboard for a Nuvi with a 480×272 screen

This will mainly involve changing all the x, y, z, w and h parameters and also re-sizing the images in **Photoshop**. The aspect ratio of an 800×480 screen is 16:10 while a 480×272 screen is 16:9 This will result in slightly different proportions (a square will become a rectangle). But you can start by multiplying all the x and w parameters by 0.60 and multiplying all the y and h parameters by 0.57.

Open the images in Photoshop and go to the Image Size dialog. Un-check the Constrain Proportions box. In the Pixel Dimensions area, use the drop-down menu to choose Percent for Width and Height, then enter 60 for Width and 57 for Height..

The **HelloWorld** dashboard should be relatively easy to adapt, since it doesn't have any data fields and you won't need to deal with font issues. I have never seen a 480×272 dashboard file myself, so I just don't know what else might be different. You will have to change the **screen_height** and **screen_width** parameters at the beginning of the file.

I have built my dashboards on the original **GarminMuscle** file, and I think you'll need to do the same thing. Make a copy of that file and note the size. Change the dashboard id to HelloWorld. Remove the original XML code and paste the HelloWorld code into its place. Adjust the filler section to make the size match the original file. Use **GIR_editor** to replace the images you have re-sized.

As you can see, this will not be a trivial project, but it should be possible. Let us know if you get it to work!

Exercises: fun with the HelloWorld dashboard

We've covered a lot of ground and now it's time to test your knowledge. Try a few of these:

Use **GIR_Editor** and **Photoshop** to change the background image. Try a solid background and

see if you like it better than transparency. Try different day and night colors. Can you take an irregular portion of photograph and use transparency to create a new background? Remember, it can't have any more pixels than the original file though.

Create a different symbol for the **Back Button** using **Photoshop**. Use **GIR_Editor** to open the original **GarminMuscle** dashboard and extract the arrow shaped Back Button image. Center it on the 56 x 54 area like the old button and see how you like it.

Can you figure out what changes would be needed to put both the **Back** and **Menu** buttons on the left side of the screen? How about at the top of the screen?

When you feel comfortable with the concepts we've covered so far, you'll be ready to take the next step and explore a more complex dashboard. But you're well along your way and the rest of the journey will be much easier!

Part 2

Located in Garmin, GPS on Jul 26, 2013

Comments are closed.