
Subject: FlyingObjects.dwar's picture and description fields

Posted by [etcddata](#) on Fri, 02 Feb 2018 02:45:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

The flyingobjects.dwar file which comes with DW has a picture and description associated with each row in the data table.

Can you point me to some documentation describing how such a visualization file is created?

Thank you.

Subject: Re: FlyingObjects.dwar's picture and description fields

Posted by [thomas](#) on Sun, 04 Feb 2018 18:54:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

There is not official explanation and there is no direct way from DataWarrior to import images or html content.

But you can do that with a little editing of the dwar file in a text editor.

Lets assume you intend to add html and jpeg files to a column named 'Name'. You start with:

- a dwar file without any associated html or jpeg content.
- a folder with html files and one with jpeg files, both to be associated with the rows of the dwar file.

First you need to define one of the dwar's columns to carry the detail information. This is done by defining column properties. Lets assume you want to add the detail links to a column called 'Name'. For that you need to extend the column properties section, if there is one. Otherwise you create a new one just after the </datawarrior-fileinfo> line. For that you add these lines:

```
<column properties>
<columnName="Name">
<columnProperty="detailCount 2">
<columnProperty="detailType0 image/jpeg">
<columnProperty="detailName0 Picture">
<columnProperty="detailSource0 relPath:images/">
<columnProperty="detailName1 Description">
<columnProperty="detailType1 text/html">
<columnProperty="detailSource1 relPath:html/">
</column properties>
```

These lines define for the column 'Name' two detail types (indicated by '0' and '1'):

- mime type 'image/jpeg' with the name 'Picture' to be found in a directory 'images'
- mime type 'text/html' with the name 'Description' to be found in a directory 'html'

Likewise you might use 'absPath' to define a complete detail directory path rather than a relative path from the dwar file.

Now you may concatenate detail file references to the respective cell content of the 'Name' column.

For instance you may add to a cell containing the name 'Aermacchi' the following:

'|#|0:aermacchi.jpg|#|1:aermacchi.html'.

'|#|' serves as separator. Then comes a number (here 0 or 1) referring to the detail number in the properties section.

After the columns comes the detail file name. It must reflect the defined mime type.

If you open the updated dwar file in DataWarrior, it will show you the detail information. You may save the file under a different name and DataWarrior will ask you, whether the detail shall be imported directly into the dwar file. If you do so, the directories with the individual files are not needed anymore.

I hope, I was clear and this solves your question...

Thomas

Subject: Re: FlyingObjects.dwar's picture and description fields

Posted by [etcddata](#) on Tue, 06 Feb 2018 10:34:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

Yes you have solved my question, thank you.

Inspection of the flyingobjects.dwar file shows that pictures are encoded to text for embedding and digging around in the source code revealed that the BinaryEncoder class is responsible for doing the encoding. I'll be able to generate dwar with embedded content now.

Subject: Re: FlyingObjects.dwar's picture and description fields

Posted by [helloWorld22](#) on Wed, 22 Nov 2023 12:19:20 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

Where can i find flyingobjects.dwar file please?

And is it possible to show embedded image from a link instead of referring to images from a file location?

Thanks,

Subject: Re: FlyingObjects.dwar's picture and description fields

Posted by [nbehrnd](#) on Wed, 22 Nov 2023 19:24:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello helloWorld22,

to get familiar with the program's working, the default installation of DW in Linux in path /opt/datawarrior/ provides the wanted file in the folder /opt/datawarrior/example. The same folder provides eight additional demo files, too.* For an installation in OS/X and Windows I assume the pattern to be a similar, if not identical.

Norwid

* boxplots.dwar, costOfLiving2018.dwar, healthAndWealth1950+.dwar, macroWithActivityCliffsAndScaffolds.dwar, polyeder.dwar, processorsAndTransistors.dwar, winterScenery.dwar, and worldfactbook.dwar

Subject: Re: FlyingObjects.dwar's picture and description fields

Posted by [helloWorld22](#) on Thu, 23 Nov 2023 10:24:01 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Norwid,

Thanks for pointing out the file in examples folder.

I can find the image in embedded in the flyingObjects.dwar file.

I can now render images from a link inside datawarrior without embedding it.

For anyone who wants to render images dynamically inside datawarrior:

And if your row has a variable, based on which you want to render the image, mention %s

So, for example, the row shows a link and the image in the link needs to be rendered inside datawarrior and also want to show an html page, following changes need to be done in the dwar file:

```
<columnName="Image URL">
<columnProperty="lookupName0 IC50 IMAGE">
<columnProperty="lookupURL0 <BASE_URL=?>%s">
<columnProperty="lookupCount 1">
<columnProperty="detailType0 image/jpeg">
<columnProperty="detailName0 Picture">
<columnProperty="detailType1 text/html">
<columnProperty="detailName1 Description">
<columnProperty="detailSource1 url/response:%s">
<columnProperty="detailSource0 url/response:<BASE_URL=?>%s">
<columnProperty="detailCount 2">
```

And the image url column will look like
70199_2_1|#|0:70199_2_1|#|1:<HTML_URL>

<BASE_URL=?> will have the base path and is the end point URI and final url after rendering will be
<BASE_URL=?>70199_2_1

<HTML_URL> is an html page that can be rendered inside DW(remember, javascript will not be rendered and only static html file will be rendered

inside SDK,

you can achieve this by mentioning like this

```
dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyLookupCount, "1");  
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyLookupName+"0", "IC50 Image");  
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyLookupURL+"0", BASE_IMAGE_URL+"%s");  
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyDetailCount, "2");
```

```
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyDetailType+"0", "image/jpeg");  
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyDetailName+"0", "Picture");  
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyDetailSource+"0",  
"url/response:"+BASE_IMAGE_URL+"%s");
```

```
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyDetailType+"1", "text/html");  
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyDetailName+"1", "Description");  
    dwInterface.setColumnProperty(collImage,  
CompoundTableConstants.cColumnPropertyDetailSource+"1", "url/response:%s");
```

FYI for others who want to render image inside datawarrior dynamically.

Awesome cheminformatics tool.
Cheers.