

| gvSIG desktop 1 - gvSIG bugs #583

Persistence of dbf tables in project

05/04/2012 10:55 AM - Antonio Falciano

Status: Won't fix	% Done: 0%
Priority: High	
Assignee: Francisco José Peñarrubia	
Category:	
Target version:	
Severity:	Add-on name: Unknown
gvSIG version: 1.12.0	Add-on version:
gvSIG build: 1406	Add-on build:
Operative System:	Add-on resolve version:
Keywords: dbf, table	Add-on resolve build:
Has patch: No	

Description

Steps in order to reproduce the bug:

- Execute gvSIG
- New table from dbf file
- Create and open a view
- Add event theme
- Save the project
- Re-open the project

Results: re-opening the project, gvSIG is not able to access to the dbf file(s) and so deletes it(them) from the project.

History

#1 - 05/31/2012 03:40 PM - Francisco José Peñarrubia

- Assignee set to Francisco José Peñarrubia

#2 - 05/31/2012 04:33 PM - Francisco José Peñarrubia

Hi Antonio.

This error was also in 1.11 release or only in 1.12 version?

Thanks.

#3 - 05/31/2012 04:45 PM - Antonio Falciano

Hi Francisco,

this issue did also occur in the 1.11 release. Don't mention it.

Cheers,

Antonio

#4 - 05/31/2012 04:54 PM - Francisco José Peñarrubia

- Status changed from New to Won't fix

Hi Antonio.

This error is hard to solve, and it would be necessary to touch some classes in core gvSIG, and maybe affect the stability of the 1.12 release that is about to come.

Right now, I think the only option is to let this bug as a "known bug".

Sorry, and my apologies :-)

Fran

#5 - 05/31/2012 05:13 PM - Francisco José Peñarrubia

Just in case someone want to do something with this bug in the future:

The problem is related to the name of class AddEventThemeDriver in extAddEventTheme. This driver is not registered y driverManager, and is stored in the project as "other".

Then, in FLayers.java, inside method addLayerFromXML, we found this code:

```
// Clases con algun driver genérico creado por otro
// programador
if (xml.contains("other")) {
    int classChild = 2;
    if(xml.contains("isLabeled") && xml.getBooleanProperty("isLabeled")){
        classChild++;
    }

    String driverName = xml.getStringProperty("other");
    VectorialDriver driver = null;
    try {
        driver = (VectorialDriver) LayerFactory.getDM().getDriver(driverName);
    } catch (DriverLoadException e) {
        // Si no existe ese driver, no pasa nada.
        // Puede que el desarrollador no quiera que
        // aparezca en el cuadro de diálogo y ha metido
        // el jar con sus clases en nuestro directorio lib.
        // Intentamos cargar esa clase "a pelo".
        if (xml.getChild(classChild).contains("className"))
        {
            String className2 = xml.getChild(classChild).getStringProperty("className");
            try {
                driver = (VectorialDriver) Class.forName(className2).newInstance();
            } catch (Exception e1) {
                throw new DriverLayerException(name,e);
            }
        }
    } catch (NullPointerException npe) {
        // Si no existe ese driver, no pasa nada.
        // Puede que el desarrollador no quiera que
        // aparezca en el cuadro de diálogo y ha metido
        // el jar con sus clases en nuestro directorio lib.
        // Intentamos cargar esa clase "a pelo".
        if (xml.getChild(2).contains("className"))
        {
            String className2 = xml.getChild(classChild).getStringProperty("className");
```

```

        try {
            driver = (VectorialDriver) Class.forName(className2).newInstance();
        } catch (Exception e1) {
            throw new DriverLayerException(name,e1);
        }
    }
}
if (driver instanceof IPersistence)
{
    IPersistence persist = (IPersistence) driver;
    persist.setXMLEntity(xml.getChild(classChild));
}
layer = LayerFactory.createLayer(name, driver, proj);
}

```

One possible solution would be to add a method to driverManager to register manually one driver, or do something with the code that captures the exception.

For example, it may be useful if a driver is not found in the usual drivers lists, search in other drivers list.

The problem is with the generated XML. First tag (selectionSupport should appear AFTER tag of AddEventThemDriver:

```

<xml-tag>
  <property key="className" value="com.iver.cit.gvsig.fmap.layers.SelectionSupport"/>
  <property key="numBitSet" value="0"/>
</xml-tag>
<xml-tag>
  <property key="className" value="com.iver.gvsig.addeventtheme.AddEventThemeDriver"/>
  <property key="xFieldIndex" value="0"/>
  <property key="yFieldIndex" value="0"/>
  <property key="tableName" value="0_test_label.dbf"/>
  <property key="hasDSInfo" value="true"/>
  <xml-tag>
    <property key="type" value="otherDriverFile"/>
    <property key="gdbmsname" value="0_test_label.dbf"/>
    <property key="file" value="C:\Users\Fjp\Documents\0_test_label.dbf"/>
    <property key="driverName" value="gdbms dbf driver"/>
  </xml-tag>
</xml-tag>

```

Hope it helps.

Fran.

#6 - 05/31/2012 05:59 PM - Francisco José Peñarrubia

- File libFMap_FLayers_AddEventTheme_reading_patch added

This is a patch that works (but is a very dirty patch, I will not put this in the repository). It's just in case of someone needs this functionality.

#7 - 02/21/2014 09:36 AM - Álvaro Anguix

- Project changed from Application: gvSIG desktop to | gvSIG desktop 1

Files

libFMap_FLayers_AddEventTheme_reading.patch	2.37 KB	05/31/2012	Francisco José Peñarrubia
---	---------	------------	---------------------------