

## Application: gvSIG desktop - gvSIG bugs #2800

### Geoprocesos ignoran la selección

09/02/2014 04:38 AM - Álvaro Anguix

<b>Status:</b> Closed	<b>% Done:</b> 0%
<b>Priority:</b> Normal	<b>Spent time:</b> 0.00 hour
<b>Assignee:</b> Francisco Díaz Carsí	
<b>Category:</b> Geoprocess	
<b>Target version:</b> 2.1.0-2255-testing	
<b>Severity:</b> Major	<b>Add-on version:</b>
<b>gvSIG version:</b> 2.1.0	<b>Add-on build:</b>
<b>gvSIG build:</b> 2245	<b>Add-on resolve version:</b>
<b>Operative System:</b>	<b>Add-on resolve build:</b>
<b>Keywords:</b>	<b>Proyecto:</b>
<b>Has patch:</b>	<b>Hito:</b>
<b>Add-on name:</b> Unknown	

---

**Description**

Los geoprocesos heredados de la librería sextante ignoran los elementos seleccionados (teniendo activada la opción de usar sólo seleccionados) y ejecutan el geoproceso sobre toda la capa.  
Se puede comprobar, por ejemplo, con el buffer.

En los geoprocesos de gvSIG funciona bien (ej.el buffer de gvSIG)

---

**Related issues:**

Related to Application: gvSIG desktop - gvSIG bugs # 2849: Several SEXTANTE g...	<b>Closed</b>	<b>10/03/2014</b>
--	---------------	-------------------

#### Associated revisions

##### Revision 599 - 10/27/2014 12:25 PM - Francisco Díaz Carsí

refs #2800 Arreglado para que los geoprocesos tengan en cuenta la selección si existe.

#### History

##### #1 - 10/06/2014 12:17 PM - Antonio Falciano

I can reproduce the issue in build 2247. IMHO it could depend by the GEOMETRY field introduced in gvSIG 2.1, because it worked in gvSIG 1.12. See also #2849.

##### #2 - 10/06/2014 12:18 PM - Álvaro Anguix

- Related to gvSIG bugs #2849: Several SEXTANTE geoprocesses don't work fine due to the GEOMETRY field added

##### #3 - 10/21/2014 11:19 AM - Antonio Falciano

This issue doesn't depend by the GEOMETRY field, but the evolution of "bindings". Simply the SEXTANTE SelectionFilter is not applied like in the past. That's why each gvSIG geoprocess has a "Selected features" check box in its panel in order to pass those directly to algorithm process. Instead, the SEXTANTE geoprocesses need the application of a SelectionFilter (if we don't want to update all the algorithm classes...):

#### gvSIG 1.x

(gvVectorLayer.java, package es.unex.sextante.gvsig.core)

```
public void create(final FLyrVect flayer) {
```

```

m_Layer = flayer;
try {
    m_RV = m_Layer.getSource();
    m_Projection = m_Layer.getProjection();
    final FBitSet fbitset = m_RV.getRecordset().getSelection();
    final BitSet bitset = (BitSet) fbitset.clone();
    final SelectionFilter filter = new SelectionFilter(bitset);
    addFilter(filter);
}
catch (final Exception e) {
    e.printStackTrace();
}
}

```

#### gvSIG 2.1

(FlyrVectlVectorLayer.java, package org.gvsig.geoprocess.lib.sextante.dataObjects)

```

public void create(final FLyrVect layer) {
    m_Layer = layer;
    try {
        featureStore = layer.getFeatureStore();
        featureType = featureStore.getDefaultFeatureType();
        m_Projection = layer.getProjection();
    } catch (final Exception e) {
        Sextante.addErrorToLog(e);
    }
}
}

```

#### #4 - 10/21/2014 01:58 PM - Antonio Falciano

FYI I've tried with this in FlyrVectlVectorLayer.java and tested with SEXTANTE Buffer:

```

public void create(final FLyrVect layer) {
    m_Layer = layer;
    try {
        featureStore = layer.getFeatureStore();
        featureType = featureStore.getDefaultFeatureType();
        m_Projection = layer.getProjection();
        //new code from here
        FeatureSet featureSet = (FeatureSet)featureStore.getFeatureSet();
        FeatureSelection featureSelection = featureStore.getFeatureSelection();
        BitSet fBitSet = new BitSet((int) featureSet.getSize());
        DisposableIterator it = featureSet.fastIterator();
        int i = 0;
        while (it.hasNext()) {

```

```

        Feature feature = (Feature)it.next();
        if (featureSelection.isSelected(feature)) {
            fBitSet.set(i);
        }
        i++;
    }
    it.dispose();
    final BitSet bitset = (BitSet) fBitSet.clone();
    final SelectionFilter filter = new SelectionFilter(bitset);
    addFilter(filter);
} catch (final Exception e) {
    Sextante.addErrorToLog(e);
}
}

```

...but the selection filter works only when there aren't selected features or all the features are selected. Instead, if there are only some features selected, the Buffer returns an empty layer.

#### #5 - 10/22/2014 02:32 AM - Cesar Ordina

I'm not very used to the sextante APIs, but just in case, my proposal would be to implement a `IVectorLayerFilter` based on the `FeatureSelection` and to avoid the `BitSet` usage. Something like:

```

public class FeatureSelectionFilter implements IVectorLayerFilter {

    private final FeatureSelection selection;

    public FeatureSelectionFilter(FeatureSelection selection) {
        this.selection = selection;
    }

    public int getLast() {
        // Warning: long to int conversion
        return selection.getSize() - 1;
    }

    public boolean accept(final IFeature ifeature,
        final int iIndex) {

        // Replace with the right way to reach the gvSIG Feature from the sextante IFeature
        Feature feature = (Feature)ifeature;
        return selection.isSelected(feature);
    }
}

```

Even better would be to fully avoid sextante having to perform the filtering itself in the case of the selection, providing it a `FeatureSelection`, taking into account it implements `FeatureSet`. But don't really know if it would be feasible.

**#6 - 10/22/2014 03:44 AM - Álvaro Anguix**

- Target version set to 2.1.0-2259-rc3

**#7 - 10/23/2014 05:58 AM - Antonio Falciano**

Hi Cesar,

first of all thank you very much for your useful suggestions. Consider that I'm not a Java professional and I'm not very experienced with SEXTANTE and the new gvSIG APIs, so my question is... How to reach gvSIG Feature from SEXTANTE IFeature? From IFeature we can get Geometry and IRecord (and then its Object[]), but AFAIK there's not a gvSIG Feature constructor (if I remember well, Feature is an interface), but only iterable FeatureSets... Other suggestions?

**#8 - 10/23/2014 10:16 AM - Cesar Ordiñana**

Hi Antonio,

Sorry I've been taken a look at the current implementation, with the help of Francisco Díaz, and the first proposal about a custom filter is not feasible, there is not any direct relationship between a DAL Feature and the IFeature implementation being used.

By the other way, it seems the second proposal would be more simple to implement, and I think also more efficient. The idea would be to change the implementation of the DALIFeatureIterator class to use the data in the selection instead of the full datasource data, avoiding any filtering. Something like:

```
public DALIFeatureIterator(final FLyrVect layer,
    final List<IVectorLayerFilter> filters) {

    m_bIsNextConsumed = true;
    m_Filters = filters;

    FeatureSet set = null;
    try {
        /* CHANGE TO SUPPORT SELECTION */
        FeatureSelection selection = layer.getFeatureStore().getFeatureSelection();
        if (!selection.isEmpty()) {
            set = selection;
        } else {
            set = layer.getFeatureStore().getFeatureSet();
        }
        /* /CHANGE TO SUPPORT SELECTION */
        m_FeatureIterator = set.fastIterator();
    } catch (final DataException e) {
        if (m_FeatureIterator != null) {
            m_FeatureIterator.dispose();
        }
        if (set != null) {
            set.dispose();
        }
    }
}
```

The proposal checks if the selection is empty to use it or all the layer's data. But it is not correct, it must be changed to check instead if the user has selected the "Use selection" option in the algorithm form.

**#9 - 10/27/2014 11:11 AM - Francisco Díaz Carsí**

- Assignee set to *Francisco Díaz Carsí*
- Status changed from *New* to *In progress*

**#10 - 10/27/2014 12:56 PM - Francisco Díaz Carsí**

- Status changed from *In progress* to *Fixed*

Se ha arreglado de la forma en que dices Cesar. No ha sido necesario chequear si el usuario a marcado "usar la selección" porque los geoprosos de Sextante no lo tienen. Si hay selección en la capa la utilizan, si no, utilizan todas las features de la capa.

Por otra parte, he arreglado del mismo modo el método `getShapeCount` para que devuelva la cantidad de features de la selección, si la hay si las hay, y si no la hay, la cantidad de features de toda la capa.

**#11 - 10/27/2014 05:22 PM - Álvaro Anguix**

- Status changed from *Fixed* to *Closed*

Lo he probado en el 2254, instalando el nuevo paquete de geoprosos y funciona bien.

Lo paso a closed.

**#12 - 10/28/2014 04:41 AM - Antonio Falciano**

I can confirm that it works fine now (thanks Francisco and Cesar), but the update of the Geoprocesses addons is not very user-friendly (see #2944) at least on Windows.

**#13 - 11/14/2014 05:45 AM - Joaquín del Cerro Murciano**

- Target version changed from *2.1.0-2259-rc3* to *2.1.0-2255-testing*