

## Application: gvSIG desktop - gvSIG bugs #3511

### The "Bounding box" geoprocess applied to a point layer returns an empty layer

05/22/2015 06:00 PM - Antonio Falciano

<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> 2.3.0-2437	
<b>Severity:</b> Minor	<b>Add-on version:</b>
<b>gvSIG version:</b> 2.2.0	<b>Add-on build:</b>
<b>gvSIG build:</b> 2310	<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**

If we load a point layer and execute the "Bounding box" SEXTANTE geoprocess on it, the result is an empty layer.

---

**Related issues:**

Related to Application: gvSIG desktop - gvSIG bugs # 3099: Exception writing ...	<b>Closed</b>	<b>01/08/2015</b>
--	---------------	-------------------

#### Associated revisions

##### Revision 202 - 07/14/2016 01:25 PM - Francisco Díaz Carsí

refs #3511:

- Fixed Bounding Box Algorithm Help
- Fixed Bounding Box Algorithm that to not accept point layers
- Added Mask shape types to AdditionalInfoVector
- Added isAvailableForShapeType and getShapeTypes methods to AdditionalInfoVector
- Added method addInputVectorLayer to ParametersSet that allow AdditionalInfoVectorLayer
- Fixed ParameterVectorLayer to use isAvailableForShapeType of AdditionalInfoVector
- Fixed DefaultParametersPanel to use getShapeTypes of AdditionalInfoVector

##### Revision 881 - 07/14/2016 01:25 PM - Francisco Díaz Carsí

refs #3511:

- Fixed Bounding Box Algorithm Help
- Fixed Bounding Box Algorithm that to not accept point layers
- Added Mask shape types to AdditionalInfoVector
- Added isAvailableForShapeType and getShapeTypes methods to AdditionalInfoVector
- Added method addInputVectorLayer to ParametersSet that allow AdditionalInfoVectorLayer
- Fixed ParameterVectorLayer to use isAvailableForShapeType of AdditionalInfoVector
- Fixed DefaultParametersPanel to use getShapeTypes of AdditionalInfoVector

#### History

##### #1 - 11/02/2015 04:39 PM - Álvaro Anguix

- Assignee set to Daniel Martinez

**#2 - 06/20/2016 01:19 PM - Joaquín del Cerro Murciano**

- Target version set to 2.3.0-2447-final (rev. org.gvsig.desktop-2.0.153)

**#3 - 06/23/2016 01:09 PM - Joaquín del Cerro Murciano**

- Assignee changed from Daniel Martinez to Francisco Díaz Carsí

**#4 - 06/24/2016 12:50 PM - Francisco Díaz Carsí**

Este algoritmo tiene varios problemas sobre los que deberíamos tomar algunas decisiones:

1. El algoritmo no hace lo que dice la descripción (al menos en español, he mirado en otros idiomas como inglés, italiano,... y allí la descripción en más vaga):

*En la descripción dice que va a generar el Bounding Box de un grupo de geometrías pero el algoritmo lo que crea es el bounding box de cada una de las geometrías (sin unirlos en una única feature) copiando todos los atributos de la capa original.*

*Deberíamos decidir si:* 1. Arreglamos el algoritmo para que haga lo que dice la descripción

2. Arreglamos la descripción para que describa lo que hace el algoritmo. Aunque debería parecer que tiene más sentido arreglar el algoritmo, me decanto por arreglar la descripción. Si optamos por 1, perderíamos la posibilidad de crear una capa de envelopes de geometrías, mientras que si optamos por 2 siempre podemos generar una unión de esos envelopes para generar un envelope global.

3. O, incluso, creamos otro algoritmo que haga lo que decidimos no hacer aquí, pero eso ya sería una Feature Request.

1. El algoritmo siempre crea como capa de salida una capa de polígonos, pero el envelope de un punto es un punto y al convertirlo a las geometrías de gvSIG se transforman también en un punto. Después, cuando se intenta guardar ese punto en la capa de polígonos se produce un error y no se añade la feature ==> capa vacía.

*Deberíamos decidir, cuando la capa de origen es de puntos, si:* 1. en lugar de crear una capa de polígonos, creamos una capa de puntos, con lo que prácticamente en la capa de salida tendríamos lo mismo que en la capa de entrada

2. o si restringimos este algoritmo para que no esté activo con una capa de puntos.

**#5 - 06/24/2016 12:56 PM - Francisco Díaz Carsí**

- Related to gvSIG bugs #3099: Exception writing 'Shape' caused by UnsupportedGeometryType: Point2D added

**#6 - 06/28/2016 12:56 PM - Antonio Falciano**

Hi Francisco,

I'm sorry but I didn't check the algorithm behaviour for lines and polygons, when I've opened this ticket. I have read its help and thought that it was not working fine only for point layers.

I think that a good solution could be to simply add a checkbox in the geoalgorithm gui, asking the user if he/she wants the global or local bbox. Maybe this is a feature request.

**#7 - 07/01/2016 10:19 AM - Antonio Falciano**

This how it could be implemented in a Jython script in gvSIG 2.3:

```
# encoding: utf-8
```

```
from gvsig.libs.toolbox import *
```

```
from es.unex.sextante.additionalInfo import AdditionalInfoVectorLayer
```

```
from es.unex.sextante.dataObjects.vectorFilters import BoundingBoxFilter
```

```
from com.vividsolutions.jts.operation.union import CascadedPolygonUnion
```

```
from java.lang import Integer
```

```
class BoundingBox(ToolboxProcess):
```

```
    def defineCharacteristics(self):
```

```
        self.setName("Bounding Box")
```

```
        self.setGroup("My algorithms")
```

```
        params = self.getParameters()
```

```
        params.addInputVectorLayer("INPUT",      # name  
                                   Sextante.getText("Vector_layer"), # description  
                                   AdditionalInfoVectorLayer.SHAPE_TYPE_ANY, # shape type  
                                   1)              # mandatory (0=false, 1=true)
```

```
        params.addBoolean("DISSOLVE", Sextante.getText("Dissolve_entities"), 0) # 0 = false
```

```
        self.addOutputVectorLayer("RESULT",      # name  
                                   Sextante.getText("Bounding_Box") # description  
                                   )
```

```
    def processAlgorithm(self):
```

```
        params = self.getParameters()
```

```
        layer = params.getParameterValueAsVectorLayer("INPUT")
```

```
        if not self.isAutoExtent():
```

```
            layer.addFilter(BoundingBoxFilter(self.getAnalysisExtent()))
```

```
        dissolve = params.getParameterValueAsBoolean("DISSOLVE")
```

```
        if dissolve:
```

```
            outputFieldTypes = []
```

```
            outputFieldTypes.append(Integer)
```

```
            outputFieldNames = []
```

```
            outputFieldNames.append("ID")
```

```
            output = self.getNewVectorLayer("RESULT",
```

```
                                             Sextante.getText("Bounding Box"),
```

```
                                             SHAPE_TYPE_POLYGON,
```

```
                                             outputFieldTypes,
```

```
                                             outputFieldNames)
```

```
            geometries = []
```

```
            outputValues = []
```

```
        else:
```

```
            output = self.getNewVectorLayer("RESULT",
```

```
                                             Sextante.getText("Bounding Box"),
```

```
                                             SHAPE_TYPE_POLYGON,
```

```
                                             layer.getFieldTypes(),
```

```
                                             layer.getFieldNames())
```

```
        iShapeCount = layer.getShapesCount()
```

```
        iterator = layer.iterator()
```

```
        self.setRangeOfValues(0, iShapeCount)
```

```

while iterator.hasNext():
    if self.isCanceled():
        break
    feature = iterator.next()
    geom = feature.getGeometry()

    if dissolve:
        geometries.append(geom)
    else:
        outputValues = feature.getRecord().getValues()
        output.addFeature(geom.getEnvelope(), outputValues)

    self.next()

if dissolve:
    cpu = CascadedPolygonUnion(geometries)
    geom = cpu.union()
    outputValues.append(Integer(0))
    output.addFeature(geom.getEnvelope(), outputValues)

return not(self.isCanceled())

def main(*args):

    # create the geoprocess
    process = BoundingBox()
    # register the geoprocess between those available in the "Scripting" group
    process.selfregister("Scripting")
    # update the Geoprocess toolbox
    process.updateToolbox()

    # inform the user that the geoprocess is added to the toolbox
    from org.gvsig.app import ApplicationLocator
    from javax.swing import JOptionPane
    application = ApplicationLocator.getManager()
    application.message("The '%s/%s/%s' geoprocess has been added to the Toolbox." % (
        "Scripting",
        process.getGroup(),
        process.getName()
    ),
        JOptionPane.INFORMATION_MESSAGE)

```

#### **#8 - 07/14/2016 01:34 PM - Francisco Díaz Carsí**

- Target version changed from 2.3.0-2447-final (rev. org.gvsig.desktop-2.0.153) to 2.3.0-2437
- Status changed from New to Fixed

Para arreglar este bug se ha hecho:

- Modificar la ayuda del algoritmo tanto en inglés como en español para que describa lo que hace el algoritmo (una capa de polígonos con tantas features como la capa de origen siendo las geometrías de estas features los bounding boxes de las geometrías de las features originales)

- No soportar capas de origen de tipo punto, (se ha arreglado para que no aparezcan si quiera en el combo box de capas de entrada)

**#9 - 07/22/2016 12:28 PM - Antonio Falciano**

- *Status changed from Fixed to Closed*

Hi Francisco,

I've just applied the "Bounding box" geoprocess on a point layer and the result is an empty layer according to what it should do described in its help. So it's ok form me. The related issue (#3099) is also solved. Thank you very much.