Obtención de información descriptiva de los datos. La clase Schema

Índice de contenido

1 Las fuentes de datos tabulares

Como ya hemos indicado hay 2 objetos que son considerados "fuentes de datos tabulares" y cuyo comportamiento en cuanto a la gestión de sus colecciones de datos es idéntica. Estos objetos son Layer y Table. La primera representa una *Capa* de la Vista y la segunda un documento de tipo Tabla. Ambos objetos contienen registros y la principal diferencia entre ellos respecto los datos que contienen es que un registro de una capa contiene un campo 'GEOMETRY'.

Hemos visto como acceder a los objetos Table y Layer tanto a través de las funciones currentTable() y currentLayer() y a través del objeto Project, mediante el método getTable, y del objeto Vista mediante el método getLayer(), respectivamente.

A continuación vamos a centrarnos en obtener la definición o estructura de los datos así como a obtener los registros de las colecciones.

2 La definición de los datos. El objeto Schema

La definición de datos de una fuente tabular se representa con el objeto *Schema*. Este objeto contiene la definición de la estructura de datos y los métodos necesarios para su gestión.

Los métodos principales son

```
- modify()
- append(name, type[, size][, default][, precision])
- get(name[, default])
- getAttrNames()
- getCopy()
```

- modify(): Establece el modo edición a la estructura de datos. Antes de añadir o modificar una propiedad es necesario activar el modo edición.
- append(name, type[, size][, default][, precision]): Añade una propiedad a la definición de los datos. Si la definición de datos no está en modo edición al invocar este método se activará.
    - name, string: Nombre de la propiedad
    - type, string: Tipo de dato que va a contener. Los valores admitidos más usuales son:
        - "BOOLEAN"
        - "DOUBLE"
        - "FLOAT"
        - "INTEGER"
        - "LONG"
        - "STRING"
        - "GEOMETRY"

      Existen más tipos de datos que pueden usarse y puedes consultarlos en la documentación de los DataTypes
    - size, int (opcional): Tamaño de la propiedad. Determina el tamaño máximo de los

datos de tipo string. También se utiliza en la representación del dato.

- default, object (opcional): Valor por defecto de la propiedad si no se indica ningún otro.
- precision, int (opcional): Establece el número de decimales del tipo de datos, si este es numérico y admite decimales.

- get(name, default): Devuelve el descriptor del atributo (FeatureAttributeDescriptor) cuyo nombre se haya pasado como parámetro si existe en la definición. En caso contrario devuelve el objeto que se haya pasado como parámetro (default) o None en caso de que se haya omitido.
  - name, string: nombre de la propiedad
  - default, object (opcional): valor por defecto que devuelve en caso de que no exista la propiedad en la definición de los datos.
- getAttrNames(): Devuelve una lista con los nombres de las propiedades
- getCopy(): Devuelve una copia de la definición de datos

Obtener la definición de datos de una fuente de datos

Si deseamos obtener la definición de datos de una fuente tabular basta con invocar al método *getSchema* de la fuente.

```
schemaTable = currentTable().getSchema()
schemaLayer = currentLayer().getSchema()
```

Estas líneas de código no pueden estar juntas porque sólo una de ellas será el elemento activo.

3 Modificar una definición de datos

Añadir campos a una definición de datos. El método append

El método *append* de una definición de datos nos permite añadir un atributo nuevo a la definición de los datos y definir sus propiedades. Si la definición de datos no está en modo edición al invocar este método se activará.

```
schema.append(name, type[, size][, default][, precision])
```

- name, string: Nombre de la propiedad
- type, string: Tipo de dato que va a contener. Los valores admitidos más usuales son:
  - "BOOLEAN"
  - "DOUBLE"
  - "FLOAT"
  - "INTEGER"
  - "LONG"
  - "STRING"
  - "GEOMETRY"

Existen más tipos de datos que pueden usarse y puedes consultarlos en la documentación de los

DataTypes

- size, int (opcional): Tamaño de la propiedad. Determina el tamaño máximo de los datos de tipo string. También se utiliza en la representación del dato.

  El valor por defecto del atributo size es 1 por lo que es recomendable indicar el tamaño del campo.

- default, object (opcional): Valor por defecto de la propiedad si no se indica ningún otro.
- precision, int (opcional): Establece el número de decimales del tipo de datos, si este es numérico y admite decimales.

  El valor por defecto del atributo precision es 4, por lo que si necesitamos una precisión mayor será necesario indicarlo.

Por ejemplo, queremos añadir a una definición de datos un campo llamado "ID" de tipo LONG

```
import  gvsig

def main():
    schema = gvsig.currentLayer().getSchema()
    schema.append("ID", "LONG", size=10)
    gvsig.currentLayer().edit()
    gvsig.currentLayer().updateSchema(schema)
    gvsig.currentLayer().commit()
```

El contenedor de datos debe estar en modo edición, y para que los cambios surtan efecto en el contenedor de datos hay que actualizar este usando el método updateSchema.

Veamos un ejemplo completo, vamos a añadir un campo nuevo a una capa que tiene 2 atributos llamados, *campo1* y *GEOMETRY*

```
Jython Completion Shell
Jython 2.5.2 (Release_2_5_2:7206, Mar 2 2011, 23:12:06)
[Java HotSpot(TM) Server VM (Sun Microsystems Inc.)] on java1.6.0_26
>>>
>>> from gvsig import *
>>> schema = currentLayer().getSchema()
>>> schema.getAttrNames()
[u'campo1', u'GEOMETRY']
>>> schema.append('ID','INTEGER')
org.gvsig.fmap.dal.feature.impl.DefaultEditableFeatureAttributeDescriptor
@4f4458
>>> #El campo que acabamos de añadir está en la definición de datos
>>> schema.getAttrNames()
[u'campo1', u'GEOMETRY', u'ID']
>>> #El campo que acabamos de añadir no está en la definición de datos de
>>> #la capa
>>> currentLayer().getSchema().getAttrNames()
[u'campo1', u'GEOMETRY']
```

```
>>> #Se produce un error porque la capa no está en edición
>>> currentLayer().updateSchema(schema)
Traceback (most recent call last):
  File "<input>", line 1, in <module>
  File "....
>>> currentLayer().edit()
>>> currentLayer().updateSchema(schema)
>>> currentLayer().getSchema().getAttrNames()
[u'campo1', u'GEOMETRY', u'ID']
>>> currentLayer().commit()
>>>
```

Eliminar campos de una definición de datos. El método remove.

Este método nos permite eliminar uno de los atributos de la fuente de datos. A diferencia del método *append* la definición de datos debe estar en modo edición antes de invocar al método remove.

El contenedor de datos debe estar en modo edición para realizar los cambios

La sintaxis es:

```
schema.remove(name)
```

- name, string: Nombre de la propiedad que queremos eliminar

Por ejemplo, queremos eliminar un campo llamado “campo1”.

```
import  gvsig

def main():
    schema = gvsig.currentLayer().getSchema()
    schema.modify()     #El schema debe estar en modo edicion
    schema.remove("campo1")
    gvsig.currentLayer().updateSchema(schema)
    gvsig.currentLayer().commit()
```

Siguiendo con el ejemplo anterior en la terminal ahora queremos eliminar el atributo llamado *campo1*.

```
>>> currentLayer().isEditing()
False
>>> currentLayer().getSchema().getAttrNames()
[u'campo1', u'ID', u'GEOMETRY']
>>> #No se puede eliminar el campo porque el schema no está en edición
>>> schema = currentLayer().getSchema()
>>> schema.remove('campo1')
False
>>> schema.modify()
>>> #Elimina el campo porque el schema está en edición
>>> schema.remove('campo1')
org.gvsig.fmap.dal.feature.impl.DefaultEditableFeatureAttributeDescriptor
@c863dd
>>> #La capa no refleja los cambios
>>> currentLayer().getSchema().getAttrNames()
```

```
[u'campo1', u'ID', u'GEOMETRY']
>>> currentLayer().edit()
>>> #Actualizamos la capa
>>> currentLayer().updateSchema(schema)
>>> currentLayer().getSchema().getAttrNames()
[u'ID', u'GEOMETRY']
>>> #Terminamos la edición de la capa
>>> currentLayer().commit()
```

4 El objeto FeatureAttributeDescriptor

Ya hemos visto que el método *get* del objeto Schema nos devuelve un objeto FeatureAttributeDescriptor, este objeto contiene información sobre uno de los atributos de los fenómenos, como el nombre, el tipo de dato que admite, o su precisión.

Los métodos más usados son

```
- getDataTypeName()
- getGeomType()
- getIndex()
- getPrecision()
- getDefaultValue()
```

- getDataTypeName(): Devuelve el nombre del tipo de dato del atributo.
- getGeomType(): Devuelve el tipo de geometría del atributo si es una geometría.
- getIndex(): Devuelve la posición relativa del atributo dentro del fenómeno.
- getPrecision(): Para atributos cuyo tipo de dato admita decimales, devuelve el máximo número de decimales.
- getDefaultValue(): Devuelve el valor por defecto de la propiedad.

gvSIG Association

Plaza Don Juan de Villarrasa 14-5,

46001, Valencia (Spain)

Registro Nacional de Asociaciones: 596206

e-mail : info@gvsig.com

Web: www.gvsig.com

Web del proyecto: http://www.gvsig.org

Documentación realizada por Víctor Acevedo.

Listas de Distribución

Existen tres listas de distribución con el objeto de facilitar la comunicación entre todos los interesados en el proyecto gvSIG. Las dos primeras, la de usuarios y la de desarrolladores, están principalmente orientadas a la comunidad de habla hispana, siendo el castellano el idioma preferente a utilizar en las mismas. La tercera de ellas, lista internacional, está orientada principalmente al resto de comunidades y la lengua preferente a utilizar es la inglesa.

- **Lista de usuarios**. Aquí podéis hacer llegar vuestra opinión sobre el funcionamiento: qué cosas os gustaría que se desarrollaran, dudas en el uso de gvSIG y todo aquello que penséis que tiene cabida en una lista de usuarios. El enlace para la suscripción a la lista de usuarios es:
  http://listserv.gva.es/mailman/listinfo/gvsig_usuarios

- **Lista de desarrolladores**. Está orientada para todos los interesados en conocer cómo está desarrollado el gvSIG. El enlace para la suscripción a esta lista esa:
  http://listserv.gva.es/mailman/listinfo/gvsig_desarrolladores

- **Lista internacional**. Está orientada tanto para usuarios como para desarrolladores de habla no hispana. El idioma a utilizar es preferentemente inglés. El enlace para la suscripción a esta lista es:
  http://listserv.gva.es/mailman/listinfo/gvsig_internacional