

Application: gvSIG desktop - gvSIG feature requests #2592

Add a precision column in the Table properties

06/24/2014 05:20 AM - Antonio Falciano

Status: Closed	% Done: 0%
Priority: Normal	Spent time: 0.00 hour
Assignee: Francisco Díaz Carsí	
Category: Document table	
Target version: 2.1.0-2246-testing	
gvSIG version: 2.1.0	Add-on resolve version:
Keywords: precision, table, double, float	Add-on resolve build:
Has patch:	Proyecto:
Add-on name: Unknown	Hito:
Add-on version:	
Description	
The precision of double fields is not shown in the Table properties window, so a new column is needed.	
Related issues:	
Related to Application: gvSIG desktop - gvSIG bugs # 2591: Double fields appe...	Closed 06/24/2014
Related to Application: gvSIG desktop - gvSIG bugs # 2759: Comportamiento ext...	Closed 08/03/2014
Related to Application: gvSIG desktop - gvSIG feature requests # 2686: Option...	Closed 07/17/2014

Associated revisions

Revision 41699 - 09/06/2014 06:26 PM - Joaquín del Cerro Murciano

Preparacion del soporte para que el usuario pueda especificar una mascara de formateo y el locale en las propiedades de la tabla, refs #2592.

Revision 41707 - 09/14/2014 08:10 AM - Joaquín del Cerro Murciano

Modificaciones en propiedades del documento tabla asi como en la edicion de campos de tipo fecha en la tabla, refs #2592

Revision 41708 - 09/14/2014 08:11 AM - Joaquín del Cerro Murciano

refs #2592

History

#1 - 07/28/2014 10:41 AM - Joaquín del Cerro Murciano

- Target version set to 2.1.0-2259-rc3

- Assignee set to Joaquín del Cerro Murciano

#2 - 07/28/2014 10:44 AM - Joaquín del Cerro Murciano

- Related to gvSIG bugs #2591: Double fields appear always with precision 3 in tables added

#3 - 09/06/2014 06:08 PM - Joaquín del Cerro Murciano

- Related to gvSIG bugs #2759: Comportamiento extraño con fechas en el documento tabla. added

#4 - 09/06/2014 06:08 PM - Joaquín del Cerro Murciano

- Related to gvSIG feature requests #2686: Option to skip the decimal format inherited by the current locale added

#5 - 09/06/2014 06:23 PM - Joaquín del Cerro Murciano

- Assignee changed from Joaquín del Cerro Murciano to Francisco Díaz Carsí

- Category set to Document table

A ver...

Vamos a ver de añadir en el dialogo de propiedades de la tabla:

- Una columna mas en la tabla en la que indica el nombre, tipo y visibilidad con la mascara de formateo a usar para presentar el campo.
- Un campo mas con el "locale" a usar para formatear los campos.

La mascara de formatear junto con el locale se aplicaran a los campos de tipo: byte, int, long, float, double y date. Para otros tipos de datos no se aplicara ningun formateo presentandose con la conversion a cadena por defecto del dato.

Se usaran las clases de java [DecimalFormat](#) y [SimpleDateFormat](#) para realizar el formateo, consultar alli el formato de la mascara.

Ademas se intentara proveer de un valor por defecto adecuado para las mascaras, en principio seria:

- "#,###" para valores enteros
- "#,###.000" para valores decimales
- La mascara por defecto del locale para las fechas.

Habra que adaptar, no solo la presentacion de los datos en la tabla a estos formatos, si no tambien la edicion para que acepte esos mismos formatos por parte del usuario a la hora de introducir nuevos o modificar los datos de la tabla.

#6 - 09/08/2014 02:50 AM - Cesar Ordñana

As an alternative to a default mask for everyone, you can try to get the default mask for the current locale with the following code:

```
String format = "#,###.000";
NumberFormat f = NumberFormat.getInstance(loc);
if (f instanceof DecimalFormat) {
    format = ((DecimalFormat) f).toPattern();
}
```

Also, it would be nice to apply the user provided pattern to the locale default DecimalFormat, something like:

```
DecimalFormat decimalFormat = null;
NumberFormat f = NumberFormat.getInstance(loc);
if (f instanceof DecimalFormat) {
    decimalFormat = (DecimalFormat) f;
} else {
    decimalFormat = new DecimalFormat();
}
decimalFormat.apply(userFormatString);
```

This is because a DecimalFormat not only provides a formatting mask, but also the symbols to use for decimal, grouping and others. So don't think the

format "#,###" will use the "," character in the generated String, it will depend on the locale symbols. This is an example taken from the [Java tutorial](#) where the same pattern is applied with different locales:

Value: 123456.789
Pattern: ###,###.###

en_US: 123,456.789
de_DE: 123.456,789
fr_FR: 123 456,789

Finally, it would be nice if some kind of **preview** could be used for the user to see how the provided mask would apply to a sample number.

#7 - 09/08/2014 10:00 AM - Joaquín del Cerro Murciano

- *Target version changed from 2.1.0-2259-rc3 to 2.1.0-2246-testing*

#8 - 09/14/2014 09:55 AM - Joaquín del Cerro Murciano

- *Status changed from New to Fixed*

#9 - 09/15/2014 03:48 AM - Álvaro Anguix

- *Status changed from Fixed to Closed*