

## Introduce the cell renderer and editor in [JComponentPack 1.1.0](#)

June 16, 2008

1. [ObjectCellRenderer](#) and it's subclass such as [DateCellRenderer](#), [NumberCellRenderer](#) is used by [JDataGrid](#) or [JTable](#) component, if you want to write a cell renderer for [JDataGrid](#), you should extend [ObjectCellRenderer](#). The [ObjectCellEditor](#) and it's subclass is used by [JDataGrid](#) or [JTable](#) component.

You can view these cell renderers in the [com.zfjava.swing.cell](#) package before reinvent the wheel.

The [ObjectCellRenderer](#) have the following subclass in [com.zfjava.swing.cell](#) package:

[BigDecimalCellRenderer](#),

[BigIntegerCellRenderer](#),

[BooleanCellRenderer](#),

[DateCellRenderer](#),

[NumberCellRenderer](#),

[StreamCellRenderer](#),

[URLCellRenderer](#)

The [ObjectCellRenderer](#) and [ObjectCellEditor](#) and it's subclass are managed by CellManager object,

The [JDataGrid](#) use the CellManger object to mange all it's cell renderer and editor, so you can add the renderer and editor for CellManager, you can also remove or replace the cell renderer or editor in CellManger.

Write a [ObjectCellRenderer](#) subclass is very simple, generally you only need override it's "[setValue\(java.lang.Object value\)](#)" method.

Writer a [ObjectCellEditor](#) subclass, you can override it's three methods:

protected void	<a href="#">commitEdit()</a>
protected <a href="#">Object</a>	<a href="#">getValue()</a>
protected void	<a href="#">setValue(Object value)</a>

If the editing have illegal value, you should override the "[commitEdit\(\)](#)" method and throws a [ParseException](#).

You can use the method [setValue\(Object value\)](#) to save the edited value, and

Override the method [getValue\(\)](#) to return the new value.

You can use the protected field “label” to set the text, alignment, font, etc.

2. [DefaultCellRenderere](#) and its subclass such as [FileCellRenderere](#) is used by [JListView](#) component, [DefaultCellEditor](#) and its subclass such as [FileCellEditor](#) is used by [JListView](#) component, it works with the [JListView](#) component to provide the different view modes: small icon, large icon, list, thumbnails, details, so if you want to use [JListView](#) component, you can extend the [DefaultCellRenderere](#).

Write the [DefaultCellRenderere](#) and [DefaultCellEditor](#) is same as the [ObjectCellRenderere](#) and [ObjectCellEditor](#), the different is you should use the protected field “[iconLabel](#)” and “[textLabel](#)” to set the icon and text.