

JComponentPack 1.2.0 New Features

1. JDataGrid Spreadsheet New Features:

- a) LargeDataGridModel can convert the cell value to different type automatically, such as formula, date, number, boolean.

LargeDataGridModel is a [DataGridModel](#) subclass, it has the different with the [DefaultDataGridModel](#), if the row and column of the [DataGridModel](#) is very large or need not the Undo or Redo feature, you can use the LargeDataGridModel, it can reduce the memory footprint and improve the performance.

- b) [JDataGrid](#) have a PDF [WorkBook](#) writer, you can export the TableModel or [DataGridModel](#) to PDF file. This feature requires the third-party PDF library named “[iText](#)”.

- c) Adds new class named JTableWriteTableModelTask, it can write TableModel or [DataGridModel](#) to many format, such as Excel xls, csv, txt, xml in background thread. This task is different with the “[JTableReadTableModelTask](#)”, it can export the JTable or [JDataGrid](#) to Excel file, see the article “[How to import Excel file into JTable](#)” for read Excel file to JTable.

- d) Two problems in cell attributes dialog in [JDataGridBean](#) get fixed.

1. The format in format panel cannot match the selected cell.
2. The background panel can select and set the background of the cell.

- e) [JDataGridBean](#) also provide 4 new API, you can use the API to set the current file or URL of the bean, or get the current [WorkBook](#) or TableModel of this bean.

```
JDataGridBean.setData(Object); // set a Workbook or TableModel for this bean
JDataGridBean.getData(); // get the Workbook or TableModel of this bean
JDataGridBean.setDataLocation(Object); // set the current file or URL of this bean
JDataGridBean.getDataLocation(); // get the current file or URL of this bean
```

- f) Other features include the bug for supporting OpenOffice formula get fixed, [AbstractDataGridModel](#) adds new API to set the cell class:
AbstractDataGridModel.setCellClass(Class, Cell);

2. [JPrintPreview](#) component adds new API to get the current Frame after show the frame:

```
JPrintPreview.getFrame();
```

Learn how to use [JPrintPreview](#) component, you and read the article “[Use JPrintPreview and](#)

[PrintFactory in JComponentPack](#)".

3. [JWizardPane](#) component adds new helper class to make creating wizard simpler.

a) `com.zfjava.swing.WizardPage` // a interface to get the WizardAction notification

b) `com.zfjava.swing.WizardAction` // a enum to identify the wizard action

If your wizard panel implements WizardPage interface, it can receive the different

WizardAction, see the article "[Use JWizardPane in JComponentPack](#)" to learn how to use the

[JWizardPane](#) component in [JComponentPack 1.2.0](#).

4. [JListView](#) component make the drag and drop simpler, the [JListView](#) component support the "dragEnabled" property and "transferHandler" property, you can make the drag and drop work like other standard Java swing component, such as JList, JTree.

You can read this article "[Introduce the JListView component in JComponentPack 1.1.0](#)" to learn how to use the [JListView](#) component.

5. The JComponentPack 1.2.0 Release Notes give a change list in details.

View [JComponent Pack 1.2.0 Release Notes](#) online.