

FUJITSU Software Interstage Studio

A horizontal decorative band with a dark red background. It features several glowing, overlapping red lines and curves that create a sense of motion and depth, with bright white highlights at the intersections.

J Business Kit GUI Library Sample Guide

B1WD-3189-01ENZ0(00)
November 2013

Preface

Purpose of this Document

The sample of GUI library and UI Screen Control Library is being provided.

The manual provides samples of each JBK library and explains how to work with these samples.

Structure of This Document

The structure of this manual is as follows:

- [Chapter 1 Explanation of GUI Library Samples](#)

This section explains how to start the GUI library samples.

- [Chapter 2 Explanation of UI Screen Control Library Samples](#)

This section explains how to start the UI Screen Control Library samples.

Trademarks

Microsoft, Active Directory, ActiveX, Excel, Internet Explorer, MS-DOS, MSDN, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Win32 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Other company and product names in this documentation are trademarks or registered trademarks of their respective owners.

The symbols (R) and TM are omitted throughout this document.

October 2013

Revision history

First Edition (November 2012)

Second Edition (November 2013)

Contents

Chapter 1 Explanation of GUI Library Samples.....	1
1.1 Compilation.....	1
1.2 Start Method.....	1
1.3 GUI Library Samples.....	2
1.3.1 JFComponent(Component).....	3
1.3.2 JFCalendarView(Calendar View).....	4
1.3.3 JFCheckList(Checklist).....	6
1.3.4 JFChoice(Choice).....	8
1.3.5 JFFieldDate(Date View).....	9
1.3.6 JFFieldDouble(Real View).....	10
1.3.7 JFFieldLong(Integer View).....	11
1.3.8 JFFieldString(Character String View).....	12
1.3.9 JFFieldFilled(Pad Character String View).....	14
1.3.10 JFTextArea(Text Area).....	16
1.3.11 JFImageButton(Image Button).....	18
1.3.12 JFListView(List View).....	20
1.3.13 JFMultiLineLabel(Multi-Line Label).....	21
1.3.14 JFProgressbar(Progress Bar).....	22
1.3.15 JFSlider(Slider).....	24
1.3.16 JFSpinButton(Spin Button).....	25
1.3.17 JFStatusBar(Status Bar).....	26
1.3.18 JFTableView(Table View Basic Function).....	27
1.3.19 Reversion(Color Reversion).....	28
1.3.20 JFCell(Cell).....	29
1.3.21 JFTimer(Timer).....	30
1.3.22 JFToggleButton(Toggle Button).....	32
1.3.23 JFToolbar(Toolbar).....	32
1.3.24 JFToolTip(Tooltip View).....	33
1.3.25 JFTree(Tree View Basic Function).....	34
1.3.26 FolderTree(Folder Tree View).....	36
1.3.27 ColorSelector(Color Selection).....	37
1.3.28 FileListViewer(File List View).....	38
1.3.29 FontSelector(Font Selection).....	40
Chapter 2 Explanation of UI Screen Control Library Samples.....	42
2.1 Compilation.....	42
2.2 Start Method.....	42
2.3 UI Screen Control Library Samples.....	42
2.3.1 UI Screen Control Panel Sample.....	42
2.3.2 UI Screen Control Panel Sample(Light Weight Version).....	45
2.3.3 UI Screen Control Tabbed Panel.....	48
2.3.4 Scroll Panel Sample.....	50
2.3.5 Scroll Panel Sample(Light Weight Version).....	51
2.3.6 Window Control Facility Sample.....	52
2.3.7 Object Control Facility Sample.....	55

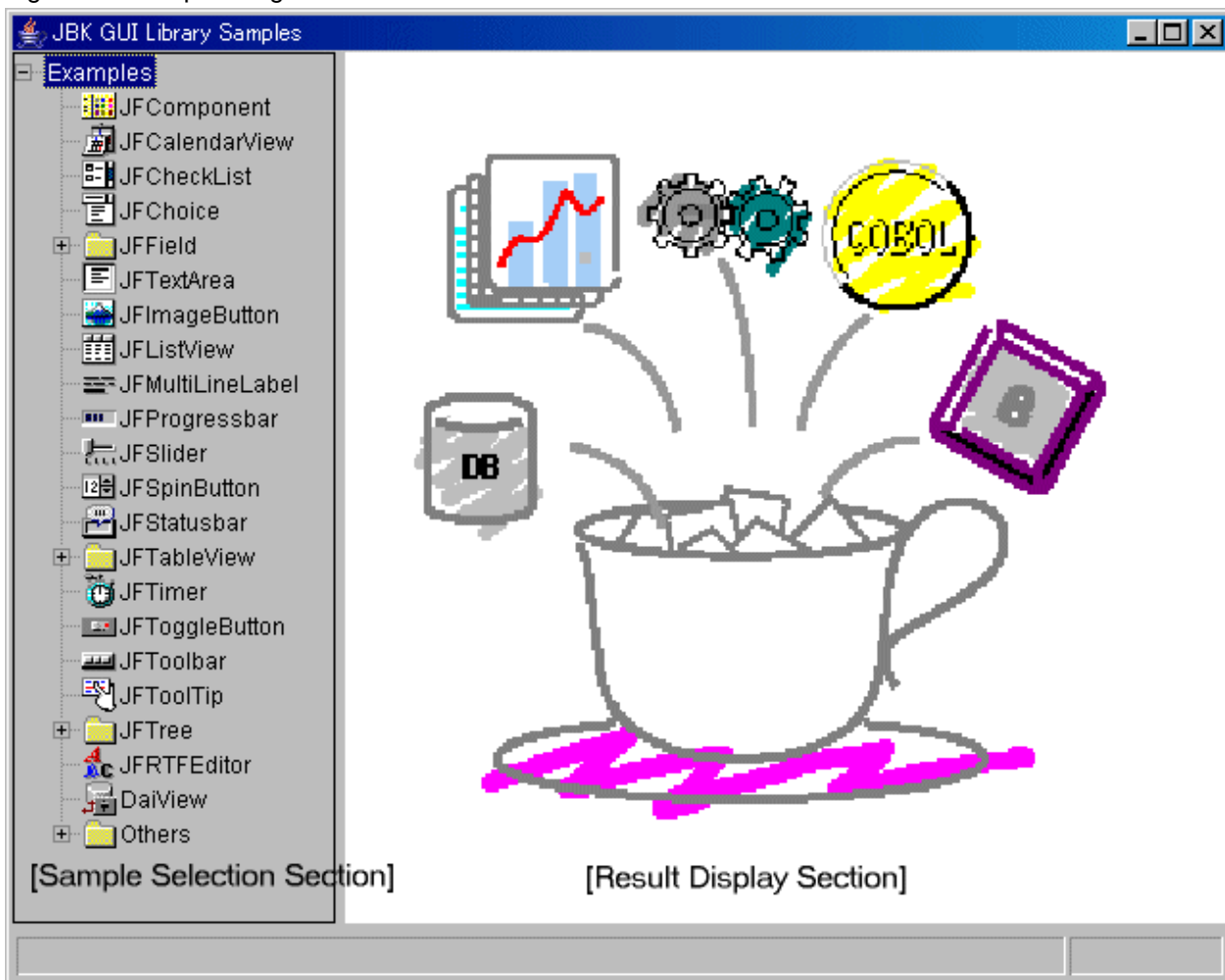
Chapter 1 Explanation of GUI Library Samples

This section explains the application samples of GUI Library.

Each sample can be executed from the "Sample Integration screen".

The "Sample Integration screen" consists of the "Sample Selection Section" and "Result Display Section". Select (double-click) the sample to be executed in the "Sample Selection Section". The execution result of the sample is displayed in the "Result Display Section".

Figure 1.1 Sample Integration screen



Sample Storage Destinations

"JBK installation folder"\examples\gui\classes

1.1 Compilation

The compiled file (.class file: Byte code) is included with the sample so the sample can be executed without change. However, if the user corrected the sample source code, compilation is required.

In this case, run compile.bat(in sample storage destinations) to start compilation.

1.2 Start Method

Run example.bat (in sample storage destinations) to start the sample.

1.3 GUI Library Samples

GUI library samples include the following:

- JFComponent (Component)
- JFCalendarView (Calendar view)
- JFCheckList (Checklist)
- JFChoice (Choice)
- JFField (I/O field)
 - JFFieldDate (Date view)
 - JFFieldDouble (Real view)
 - JFFieldLong (Integer view)
 - JFFieldString (Character string view)
 - JFFieldFilled (Pad character string view)
- JFTextArea (Text area)
- JFImageButton (Image button)
- JFListView (List view)
- JFMultilineLabel (Multiline label)
- JFProgressbar (Progress bar)
- JFSlider (Slider)
- JFSpinButton (Spin button)
- JFStatusbar (Status bar)
- JFTableView (Table format)
 - JFTableView (Table view basic function)
 - Reversion (Color reversion)
 - JFCell (Cell)
- JFTimer (Timer)
- JFToggleButton (Toggle button)
- JFToolbar (Toolbar)
- JFToolTip (Tooltip view)
- JFTree (Tree format)
 - JFTree (Tree view basic function)
 - FolderTree (Folder tree view)
- Others
 - ColorSelector (Color selection)
 - FileListViewer (File list view)
 - FontSelector (Font selection)

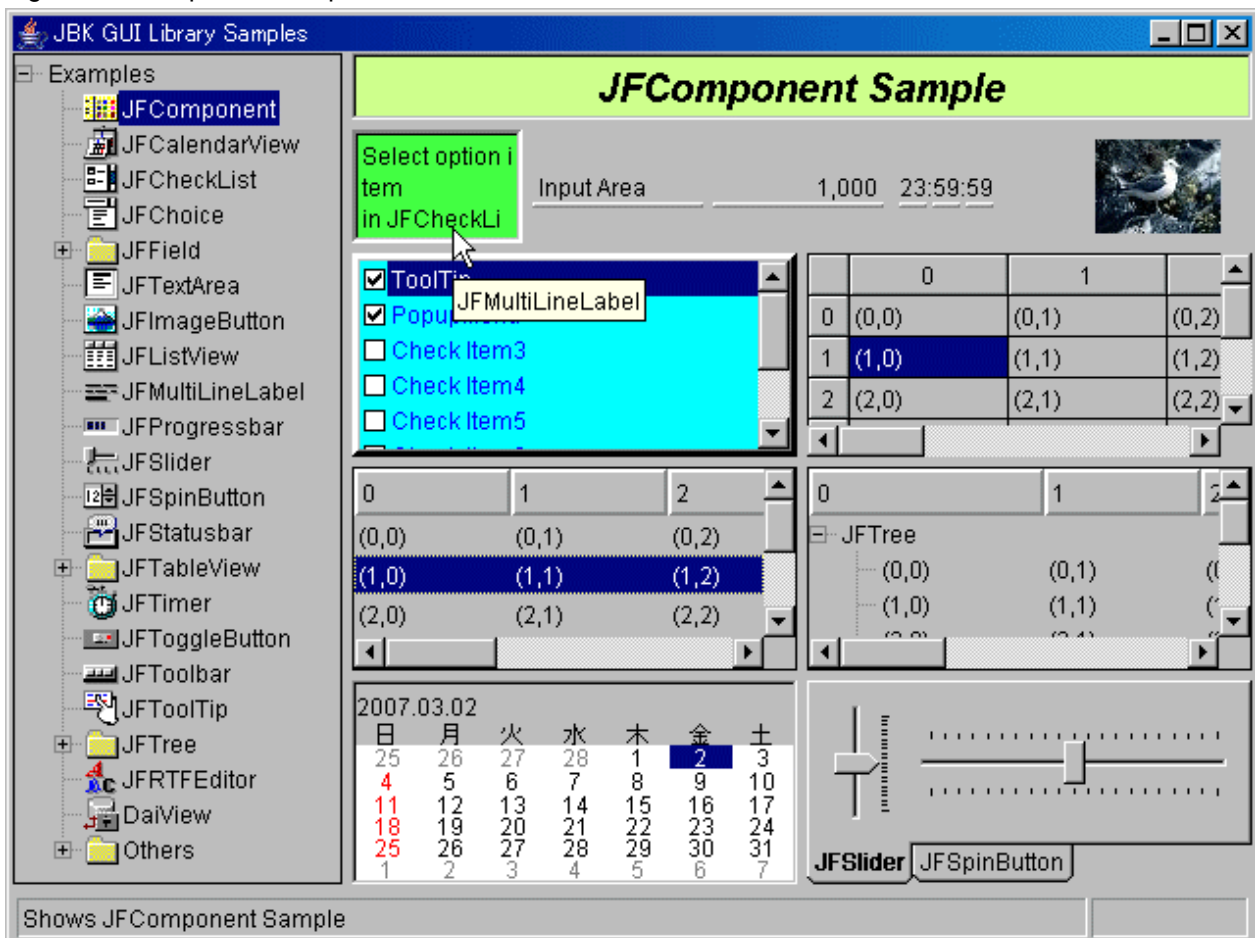
1.3.1 JFComponent(Component)

The component sample enables common component functions such as tool tip display, pop-up menu display, and gridline specification using the JFComponent class and JFContainer class.

The component sample uses the following classes:

- JFMultiLineLabel
- JFFieldString
- JFFieldLong
- JFFieldFilled
- JFImageButton
- JFCheckList
- JFTableView
- JFListView
- JFTree
- JFCalendarView
- JFTabPanel
- JFSlider
- JFSpinButton
- JFPanel

Figure 1.2 Component sample



Operation method

Items checked with the checklist (tool tip/pop-up menu) are enabled.

- Tooltip

Moving the mouse cursor to the component displays the tool tip.

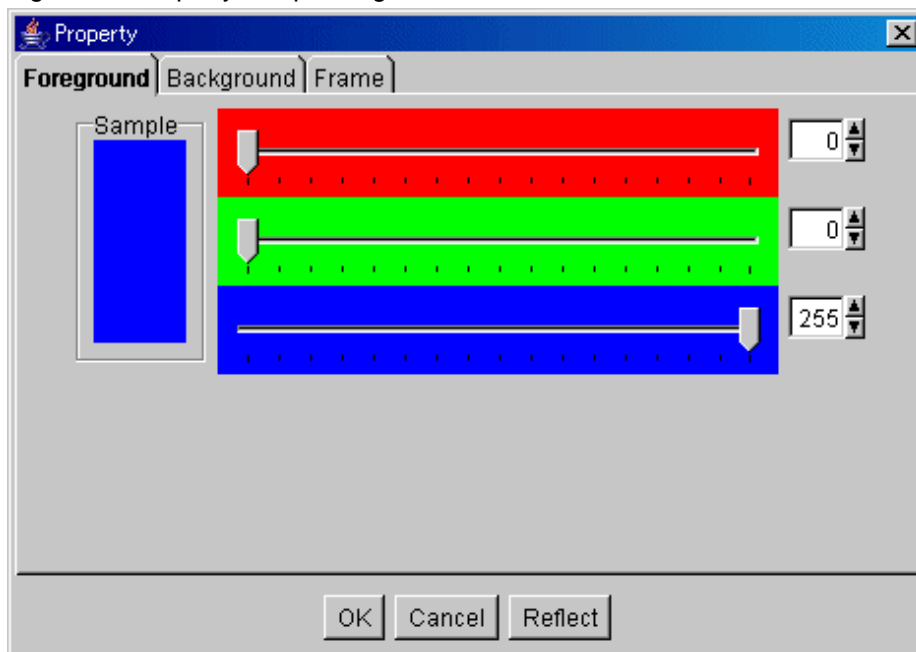
- Pop-up menu

Clicking the right mouse button on the component displays the pop-up menu (Properties can be set).

- Setting properties

Selecting the command (property) in the pop-up menu changes the foreground color, background color, and gridlines.

Figure 1.3 Property setup dialog



The foreground color/background color of gridlines appears only when flatbed ("FLAT" is selected) is selected as the type of gridlines in the [Property Setup] dialog.

With the JFCalendarView, changing the foreground color/background color changes the foreground color/background color of the selected date.

Source codes

- Component sample:"Sample Storage Destinations"/jfcomponent/JFComponentSample.java
- Property setup dialog:"Sample Storage Destinations"/jfcomponent/PropertyDialog.java

1.3.2 JFCalendarView(Calendar View)

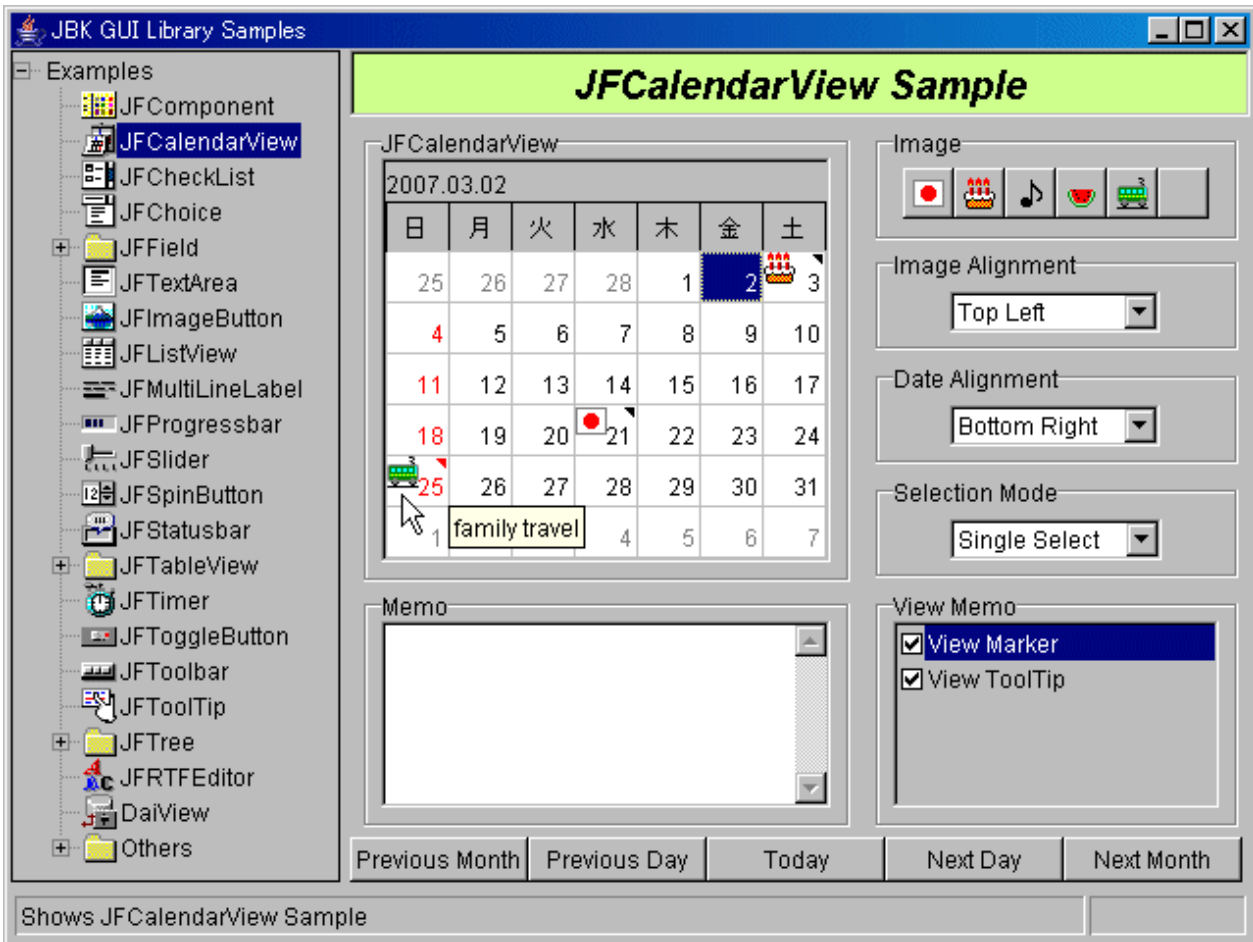
The calendar view sample displays the monthly calendar, moves for each month, and moves for each day using the JFCalendarView class.

The calendar view sample uses the following classes:

- JFCalendarView
- JFCalendar
- JFCheckList
- JFFieldString
- JFGroupbox

- JFImageButton
- JFPanel

Figure 1.4 Calendar view sample



Operation method

Sets the memo on the specified date and displays the image.

- JFCalendarView

Displays the calendar and selects the date.

- Memo

Select the date and specify the memo to be set.

- Image

Select the date and specify the image to be set. Use the right-end button to cancel the image setup.

- Image Alignment

Specify the position for displaying the image with Choice. Specifiable positions are as follows:

- View positions: All Area/Top Left/Top/Top Right/Left/Center/Right/Bottom Left/Bottom/Bottom Right

- Date Alignment

Specify the position for displaying the date with Choice. Specifiable positions are as follows:

- View positions: Top Left/Top/Top Right/Left/Center/Right/Bottom Left/Bottom/Bottom Right

- Selection Mode

Specify the mode for selecting the date with Choice. The selection modes are as follows:

- Single select: Only one day can be selected.
- Multi select: Multiple selections are enabled.
- Bounds select: Selection with the range is enabled.

- View Memo

Specify the view method when the memo has been set. The item containing a check in the check box is enabled.

- View Marker: The marker appears in the upper-right corner of the date position to which the memo has been set.
- View ToolTips: Moving the mouse cursor to the position of the date to which the memo has been set displays the tooltip.

- Last Month/Last Day/Today/Next Day/Next Month

The buttons for operating the calendar are explained as follows:

- Last Month: Returns to the same day of the previous month.
- Last Day: Returns to the previous day.
- Today: Returns to the date the sample was executed.
- Next Day: Goes to the next day.
- Next Month: Goes to the same day of the next month.

Source codes

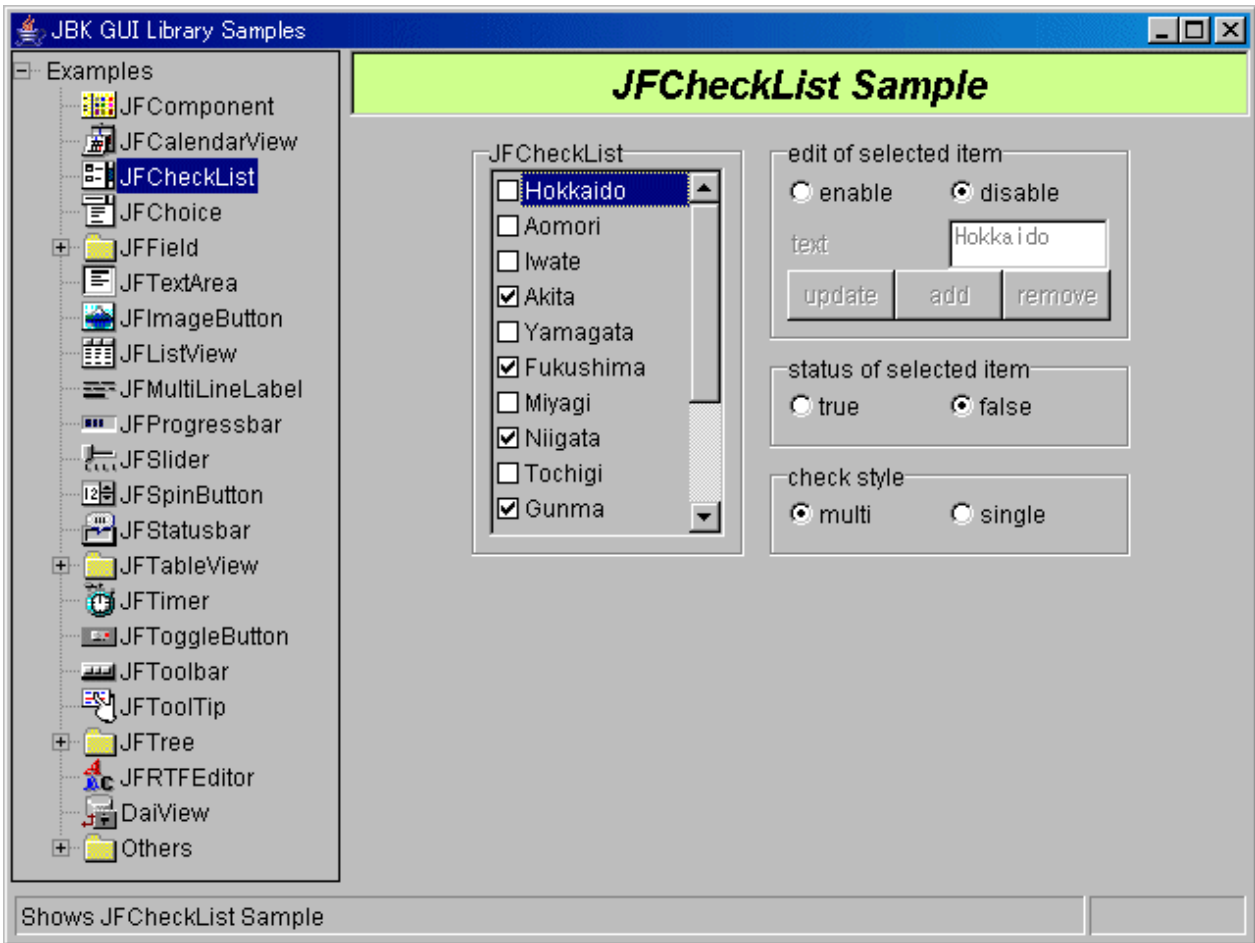
- Calendar view sample:"Sample Storage Destinations"/jfcalendarview/JFCalendarViewSample.java

1.3.3 JFCheckList(Checklist)

The checklist sample is the checklist using the JFCheckList class. The checklist sample uses the following classes:

- JFCheckList
- JFGroupbox
- JFPanel

Figure 1.5 Checklist sample



Operation method

- JFCheckList

Click the check boxes of the required items.

- Edit of selected item

Edit the item selected in the checklist.

- enable/disable: Specify whether to edit the selected item.
- text: Specify the item in the checklist.
- update: Updates the selected item to the contents specified in the "text".
- add: Adds the contents specified in the "text" as the item.
- remove: Deletes the selected item.

- Status of selected item

The status of the item selected in the checklist is displayed.

- true: Indicates that the selected item has been checked.
- false: Indicates that the selected item has not been checked.

- Check style

The item containing a check in the check box is enabled.

- multi: Multiple items can be checked.

- single: Only single item can be checked.

Source codes

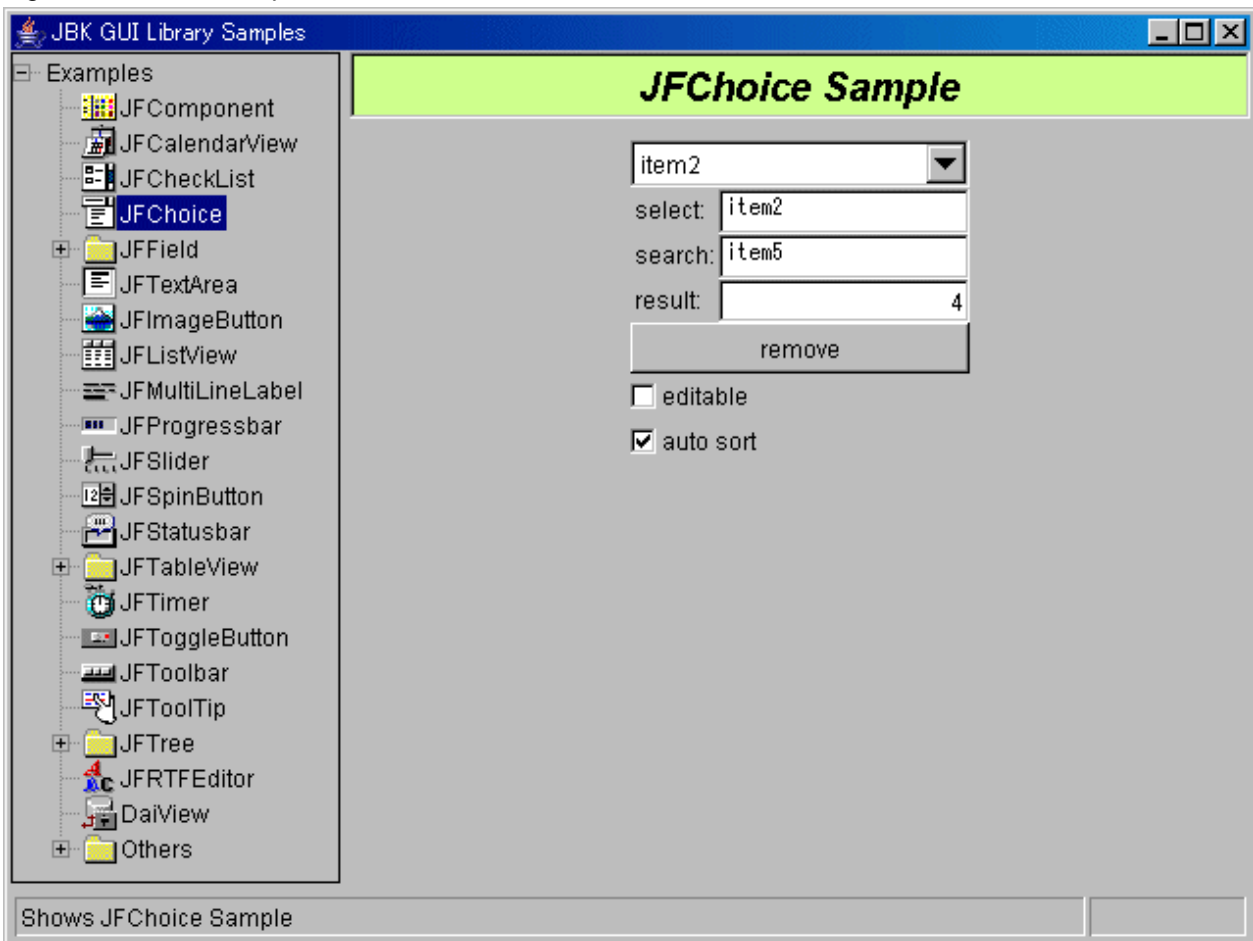
- Checklist sample:"Sample Storage Destinations"/jfchecklist/JFCheckListSample.java

1.3.4 JFChoice(Choice)

The choice sample is a list box using the JFChoice class. The choice sample uses the following classes:

- JFChoice
- JFFieldLong

Figure 1.6 Choice sample



Operation method

- List box section (Choice)

"item1" to "item5" have been registered.

- Select

Entering the item to be selected (such as "item2") in the input field on the right side and pressing the [Enter] key selects and displays "item2" in the list box.

- Search/Result

Entering the item to be searched (such as "item5") in the input field on the right side and pressing the [Enter] key searches "item5" in the list box and displays index number "4" on the right side of the searched result. Only complete matching items are searched.

Index numbers to be displayed in the searched result starts from 0.

- Remove

The selected item (such as "item2") is deleted.

- Editable

Clicking the check box enables Choice to be edited. Entering the item (character string) in the edit field of Choice and pressing the [Enter] key adds the item. However, the item that has already been added or null character string cannot be added.

- Auto sort

Clicking the check box sorts the items so that the selected item is positioned at the beginning of the list box.

Source codes

- Choice sample: "Sample Storage Destinations"/jfchoice/JFChoiceSample.java

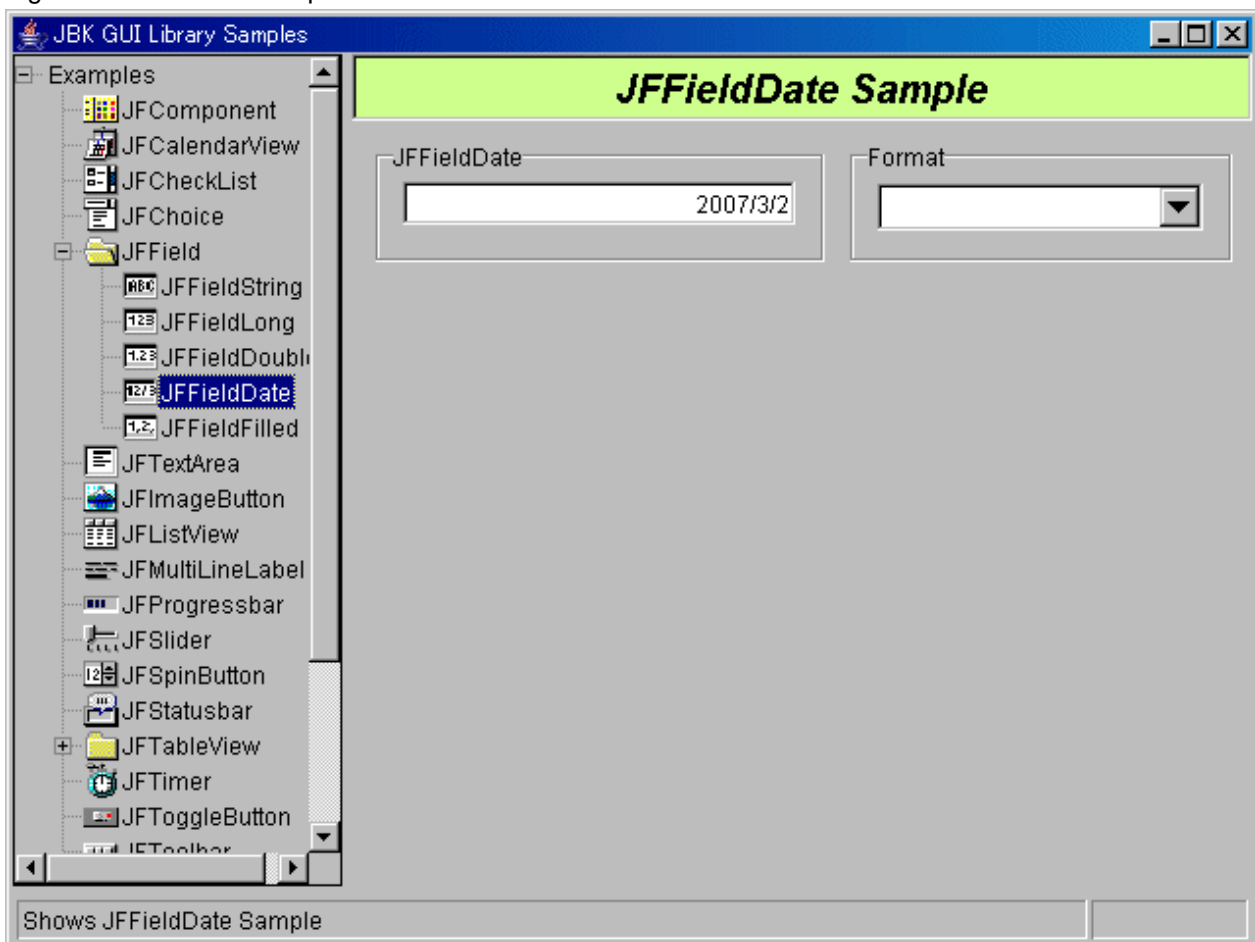
1.3.5 JFFieldDate(Date View)

The date view sample displays the date (time) of the result edited conforming to the specified edit format using the JFFieldDate class.

The date view sample uses the following classes:

- JFFieldDate
- JFGroupbox

Figure 1.7 Date view sample



Operation method

- JFFieldDate

Displays the edited date (time) result.

- Format

Selecting the edit format with Choice displays the result edited conforming to the edit format.

Switching the edit format returns the items non-included in the previous edit format to the initial value and displays them.



Example

.....
When "1997 to 1998/07/18/ 13:44:18" is displayed:

- yyyy(Year)M(Month)d(Day): 1997 to 1998.7.18

At this point, the time (Hours:Minutes:Seconds) returns to the initial value (00:00:00).

- H:mm: 0:00

At this point, the date (Year/Month/Day) returns to the initial value (1970/1/1).

- yyyy/M/d: 1970/1/1
-

Source codes

- Date view sample:"Sample Storage Destinations"/jffielddate/JFFieldDateSample.java
- Date edit format pattern:"Sample Storage Destinations"/common/DateFormatPatternChoice.java

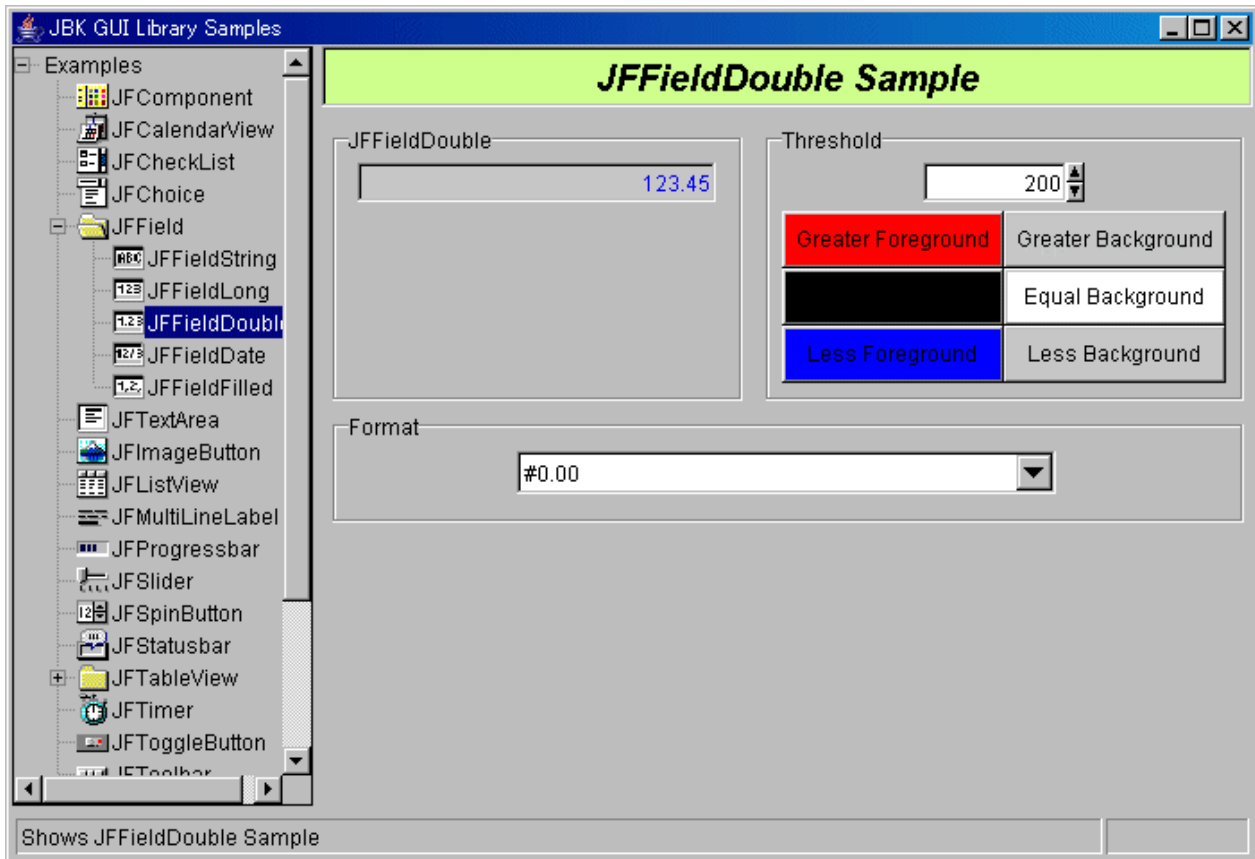
1.3.6 JFFieldDouble(Real View)

The real view sample displays the numeric value (real) of the result edited conforming to the specified edit format using the JFFieldDouble class.

The real view sample uses the following classes:

- JFFieldDouble
- JFGroupbox

Figure 1.8 Real view sample



Operation method

- JFFieldDouble

Entering the value (real) to be edited and pressing the [Enter] key displays the edited result.

- Format

Selecting the edit format with Choice displays the result edited conforming to the edit format.

- Boundary value

Specify the threshold. The threshold can also be entered by clicking the spin button arrow using the mouse. The six buttons under the "Boundary value" are used to compare the entered numeric value with the threshold and to specify the foreground color and background color of the edited result. Clicking each button displays the [Figure 1.29 Color selection sample](#).

Source codes

- Real view sample:"Sample Storage Destinations"/ffielddouble/JFFieldDoubleSample.java

- Edit format pattern:"Sample Storage Destinations"/common/DecimalFormatPatternChoice.java

1.3.7 JFFieldLong(Integer View)

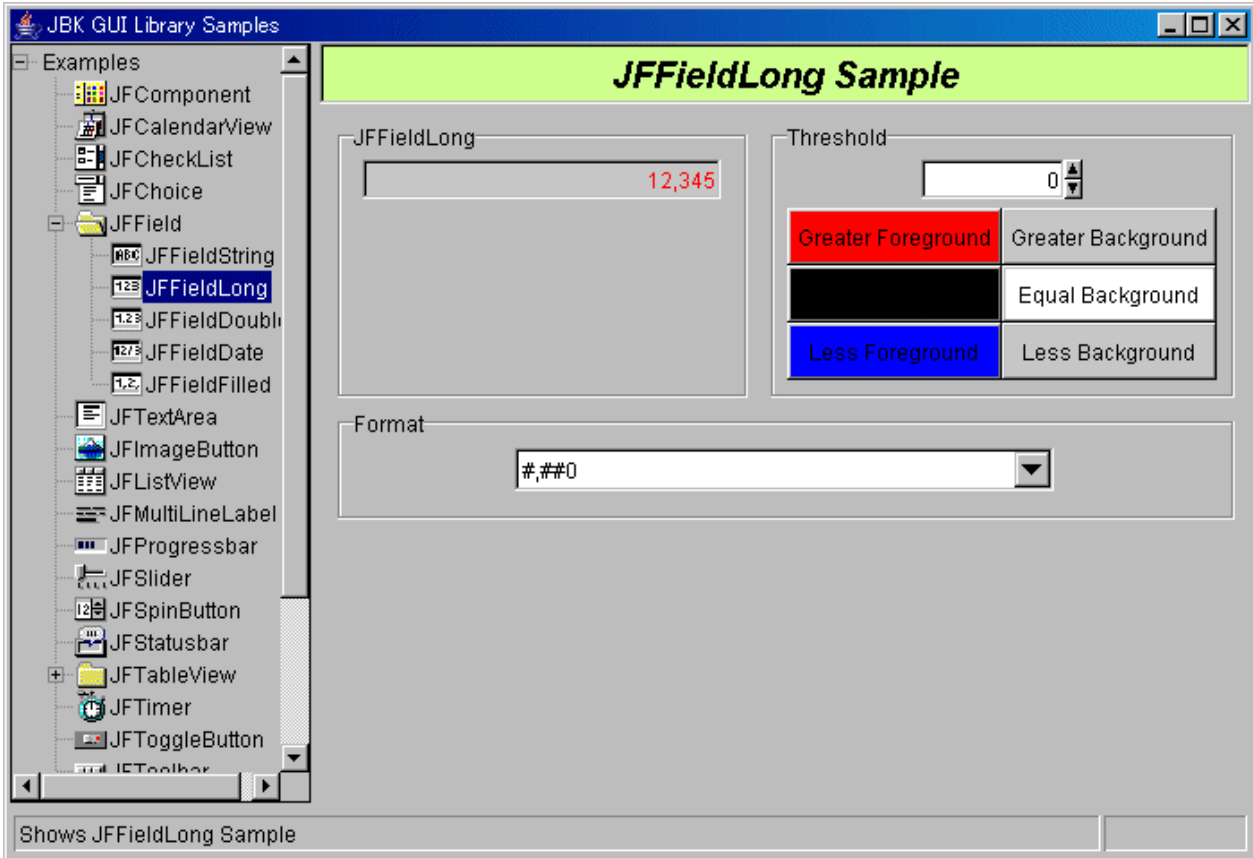
The integer view sample displays the numeric value (integer) of the result edited conforming to the specified edit format using the JFFieldLong class.

The integer view sample uses the following classes:

- JFFieldLong

- JFGroupbox

Figure 1.9 Integer view sample



Operation method

- JFFieldLong
Entering the numeric value to be edited (integer) and pressing the [Enter] key displays the edited result.
- Format
Selecting the edit format with Choice displays the result edited conforming to the edit format.
- Boundary value
Specify the threshold. The threshold can also be entered by clicking the spin button arrow using the mouse. The six buttons under the "Boundary value" are used to compare the entered numeric value with the threshold and to specify the foreground color and background color of the edited result. Clicking each button displays the [Figure 1.29 Color selection sample](#).

Source codes

- Integer view sample:"Sample Storage Destinations"/jffieldlong/JFFieldLongSample.java
- Edit format pattern:"Sample Storage Destinations"/common/DecimalFormatPatternChoice.java

1.3.8 JFFieldString(Character String View)

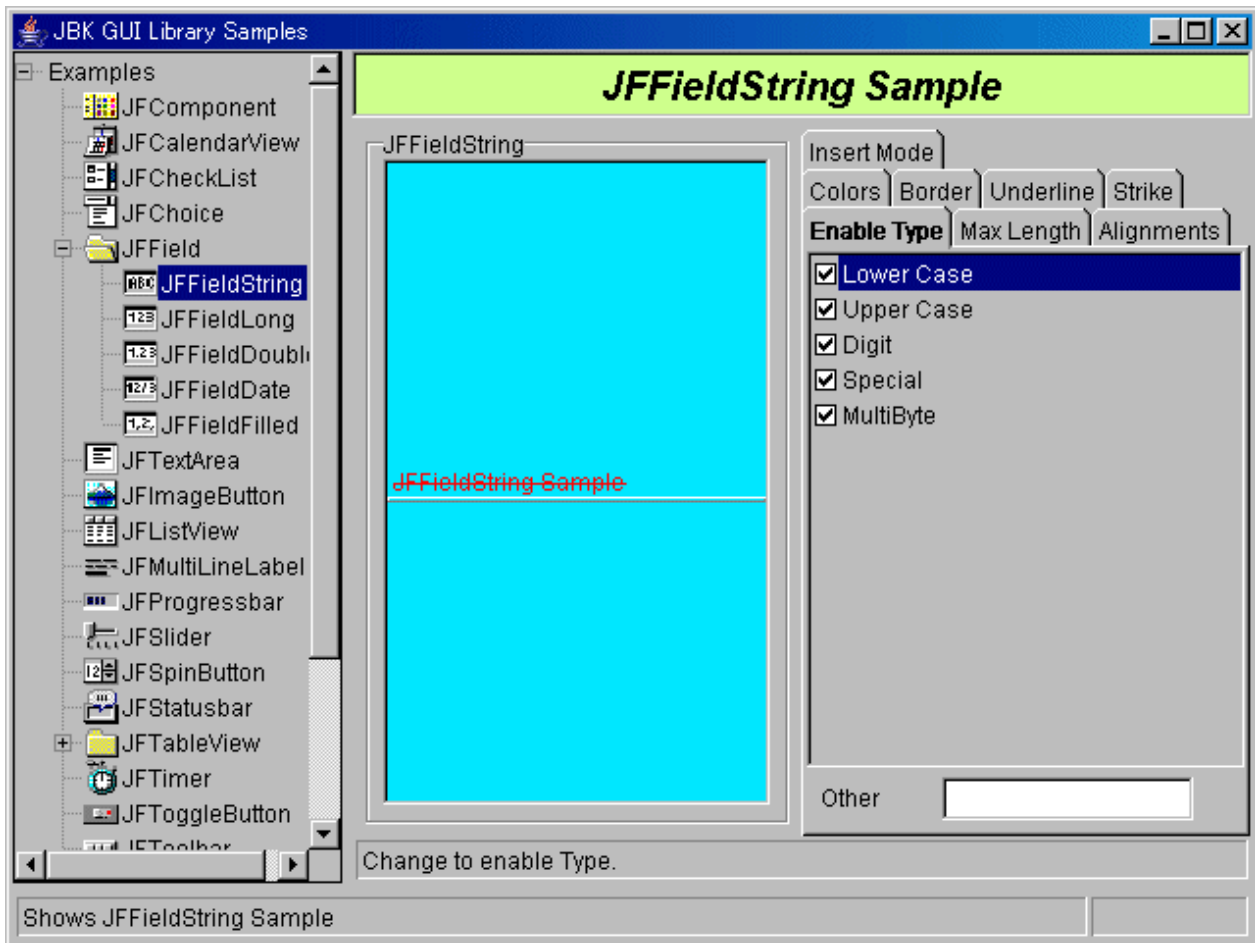
The character string view sample displays the character string of the result edited conforming to the specified edit format using the JFFieldString class.

The character string view sample uses the following classes:

- JFFieldString
- JFGroupbox
- JFTabPanel

- JFPanel

Figure 1.10 Character string view sample



Operation method

Specifies the view character string option with the tab panel.

- JFFieldString

Entering the character string and pressing the [Enter] key displays the result conforming to the specified option.

- Inputting enabled characters

Specify the enterable character types by clicking the check boxes. Specifiable character types are as follows; however, character types without check marks in their check boxes cannot be entered.

- Character types: Lower case/Upper case/Digit/Special/MultiByte

Changing the enterable character types clears the displayed character strings.

Specify "Others" under "Inputting enabled characters" only if a character type from the character types (non-enterable character types) whose check boxes are not to be clicked is to be entered. If "A" only is to be entered, even though uppercase letters are to be non-enterable, specify "A" in "Others".

- Max length of inputted characters

Specify the number of enterable characters.

- String alignment

Specify the align positions in horizontal and vertical directions of the input character string. Specifiable positions are as follows:

- Horizontal direction: Left justification/Center justification/Right justification

- Vertical direction: Top justification/Center justification/Bottom justification
- Foreground/Background color

Specify the foreground and background colors of the input character string.
How to specify the color is the same as the [Figure 1.29 Color selection sample](#) specification method.
- Border

Specify the type and color of gridlines. Specifiable types are as follows:

 - Types of gridlines: Nothing/Convex/Flat/Concave

Specifying the color is the same as the [Figure 1.29 Color selection sample](#) specification method.
- Underline

Specify the type and color of the underline. Specifiable types are as follows:

 - Underline types: Nothing/Sunken/Flat/Raised

Specifying the color is the same as the [Figure 1.29 Color selection sample](#) specification method.
- Strike line

Specify the type and color of the strikethrough. Specifiable types are as follows:

 - Strikethrough types: Nothing/Single Line/Double Line

Specifying the color is the same as the [Figure 1.29 Color selection sample](#) specification method.
- Insert mode

Specify the input method. Specifiable types are as follows:

 - Input method: Insert/Overwrite

Source codes

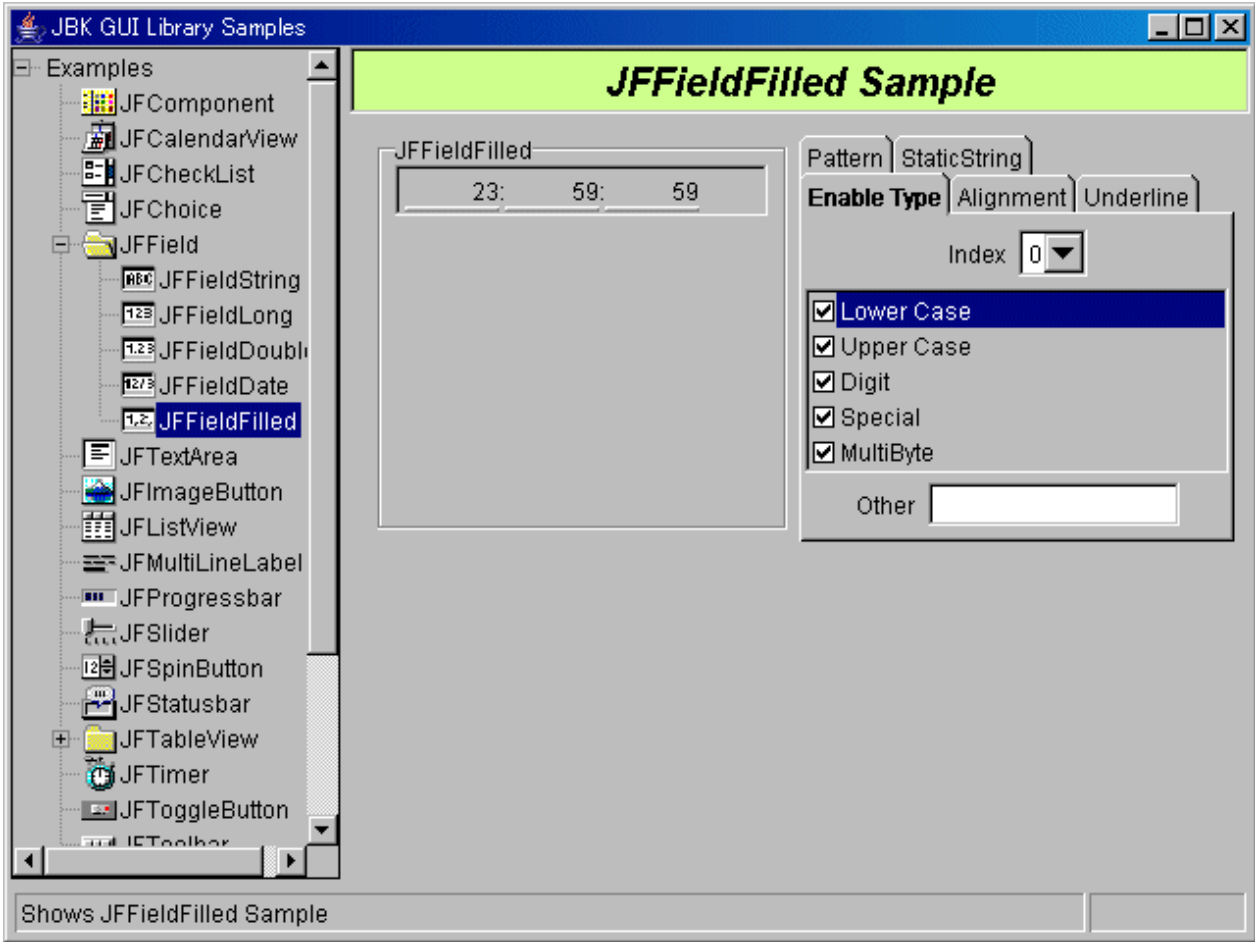
- Character string view sample:"Sample Storage Destinations"/jffieldstring/JFFieldStringSample.java

1.3.9 JFFieldFilled(Pad Character String View)

The pad character string view sample displays the character string using the JFFieldFilled class.

- JFFieldFilled
- JFGroupbox
- JFTabPanel
- JFPanel

Figure 1.11 Pad character string view sample



Operation method

Specifies the pad character string view option with the tab panel.

- JFFieldFilled

The fixed section and variable section of the pad character string are displayed conforming to the specified option.

The variable section (input field) of the pad character string is called the first field, second field, and so on, in sequence from the left. The "Index" corresponding to the field starts from "0".

- Inputting enabled characters

Specify the enterable character types by clicking the check box. Specifiable character types are as follows; however, character types without check marks in their check boxes cannot be entered.

- Index: Specify the index corresponding to the number of the field in which the character type is to be specified.
- Character types: Lower case/Upper case/Digit/Special/MultiByte

Changing the enterable character types clears the displayed character strings.

Specify "Others" under "Inputting enabled characters" only if a character type from the character types (non-enterable character types) whose check boxes are not to be clicked is to be entered. If "A" only is to be entered, even though uppercase letters are to be non-enterable, specify "A" in "Others".

- String alignment

Specify the align positions of the input character string. Specifiable positions are as follows:

- Index: Specify the index corresponding to the number of the field in which the align position is to be specified.
- Align: Left justification/Center justification/Right justification

- Underline

Specify the underline type. Specifiable types are as follows:

- Type: Nothing/Sunken/Flat/Raised

- Pattern

Specify the pad character string format with Choice.

- Separator of string

Change the delimiter string corresponding to the input field.

- Index: Specify the index corresponding to the number of the field in which the delimiter string is to be changed.
- Delimiter string: Specify the delimiter string.

Source codes

- Pad character string view sample sample:"Sample Storage Destinations"/jffieldfilled/JFFieldFilledSample.java

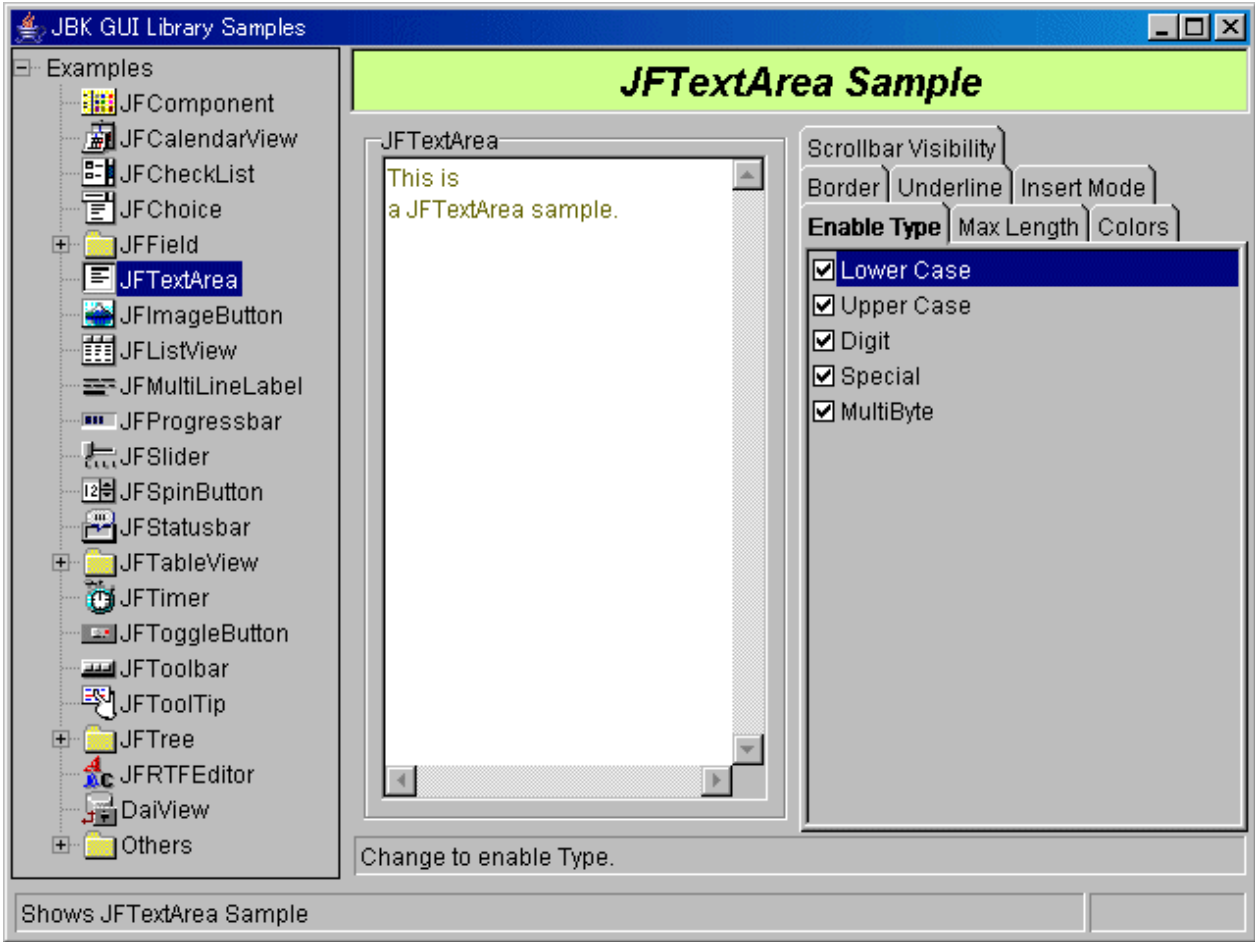
1.3.10 JFTextArea(Text Area)

The text area sample displays the character string of the result edited conforming to the specified edit format using the JFTextArea class.

The text sample uses the following classes:

- JFTextArea
- JFGroupbox
- JFTabPanel
- JFPanel

Figure 1.12 Text area sample



Operation method

Specifies the display character string option with the tab panel.

- JFTextArea

Entering the character string displays the result conforming to the specified option.

- Inputting enabled characters

Specify the enterable character types by clicking the check box. Specifiable character types are as follows; however, character types without check marks in their check boxes cannot be entered.

- Character types: Lower case/Upper case/Digit/Special/MultiByte

Changing the enterable character types clears the displayed character strings.

- Max length of inputted characters

Specify the number of enterable characters.

- Foreground/Background color

Specify the foreground and background colors of the input character string.

Specifying the color is the same as the [Figure 1.29 Color selection sample](#) specification method.

- Border

Specify the type and color of gridlines. Specifiable types are as follows:

- Types of gridlines: Nothing/Convex/Flat/Concave

Specifying the color is the same as the [Figure 1.29 Color selection sample](#) specification method.

- Underline

Specify the type and color of the underline. Specifiable types are as follows:

- Underline types: Nothing/Sunken/Flat/Raised

Specifying the color is the same as the [Figure 1.29 Color selection sample](#) specification method.

- Insert mode

Specify the input method. Specifiable types are as follows:

- Input method: Insert/Overwrite

- Scrollbar Visibility

Specify how to display the scroll bar. Specifiable types are as follows:

- Scroll bar display: Both/Horizontal Only/Vertical Only(Linewrap)/None(Linewrap)

Source codes

- Text area sample:"Sample Storage Destinations"/jftextarea/JFTextAreaSample.java

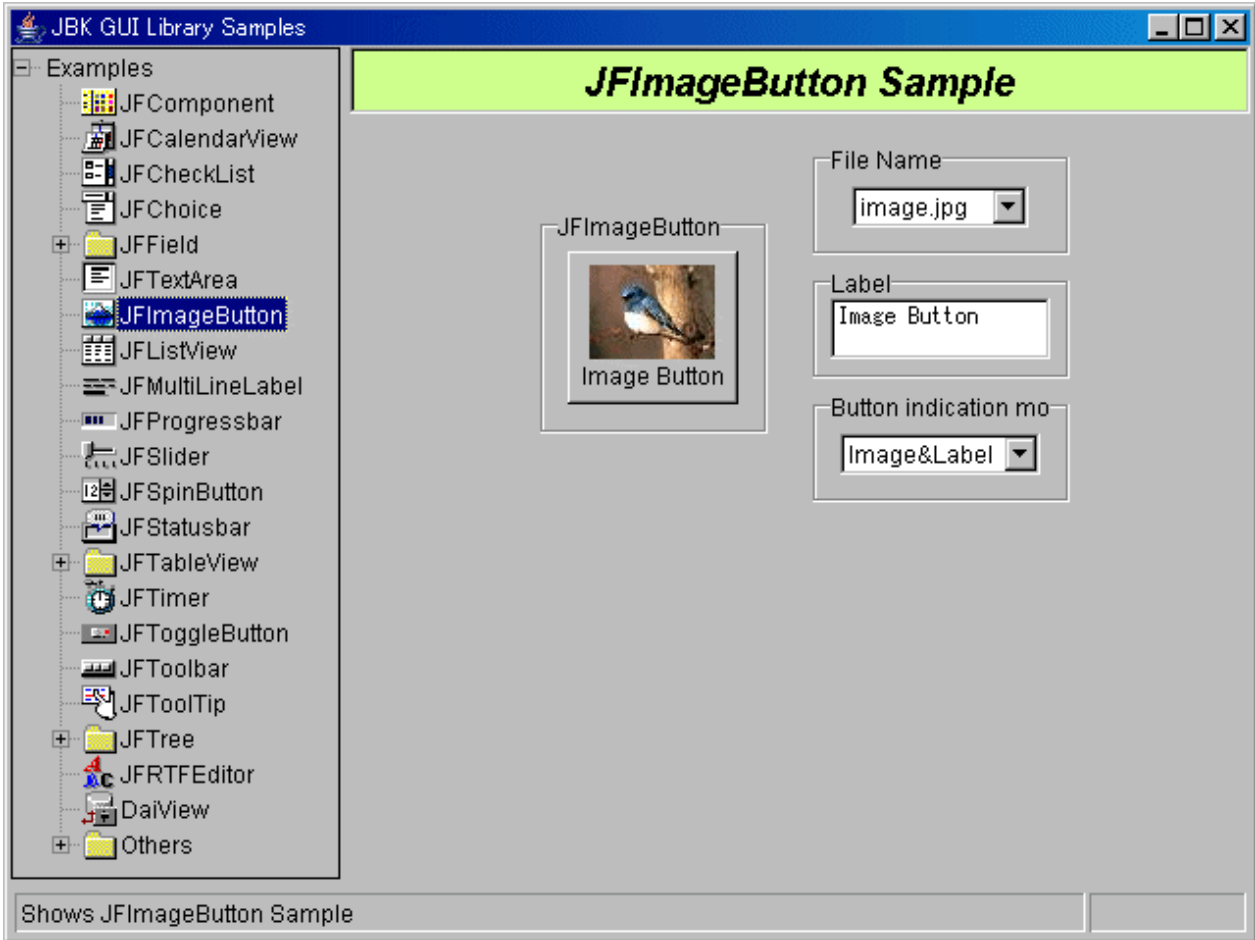
1.3.11 JFImageButton(Image Button)

The image button sample displays only the image, only the label, or both the image and the label using the JFImageButton class.

The image button sample uses the following classes:

- JFImageButton
- JFFieldString
- JFChoice

Figure 1.13 Image button sample



Operation method

- JFImageButton
Displays the image button.
- File name
Specify the file name of the image to be displayed with Choice. If the "Button indication mode" is a "Label," the file name is not displayed.
- Label
Specify the character string to be displayed as a label. If the "Button indication mode" is an "Image," the label is not displayed.
- Button indication mode
Specify how to display the image button with Choice.
 - Image&Label: The image and label are displayed.
 - Image: Only the image is displayed.
 - Label: Only the label is displayed.

Source codes

- Image button sample:"Sample Storage Destinations"/jfimagebutton/JFImageButtonSample.java

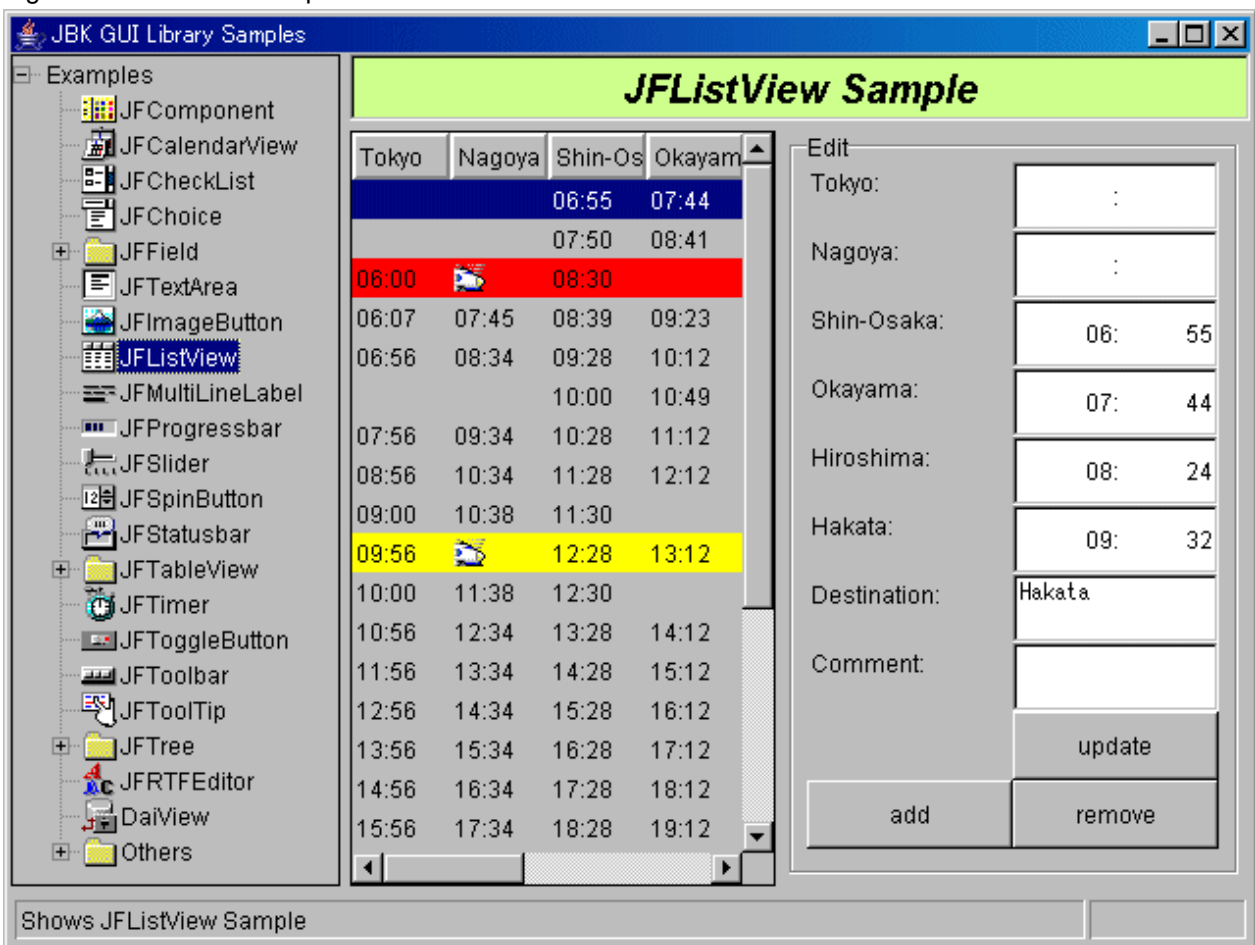
1.3.12 JFListView(List View)

The list view sample displays data in the list format and updates data using the JFListView class.

The list view sample uses the following classes:

- JFListView
- JFFieldString
- JFFieldFilled
- JFGroupbox
- JFPanel

Figure 1.14 List view sample



Operation method

- List view
 - Reads and displays the sample data in the list format.
- Edit
 - Displays the data of the line selected with list view. Use the following buttons to update data.
 - update: Updates and displays the data in the input field corresponding to the column name.
 - add: Displays the data in the input field corresponding to the column name.
 - remove: Deletes the selected line.

The input field of Tokyo/Nagoya/Shin Osaka/Okayama/Hiroshima/Hakata of "Edit" uses the JFFieldFilled class.

Source codes

- List view sample:"Sample Storage Destinations"/jflistview/JFListViewSample.java

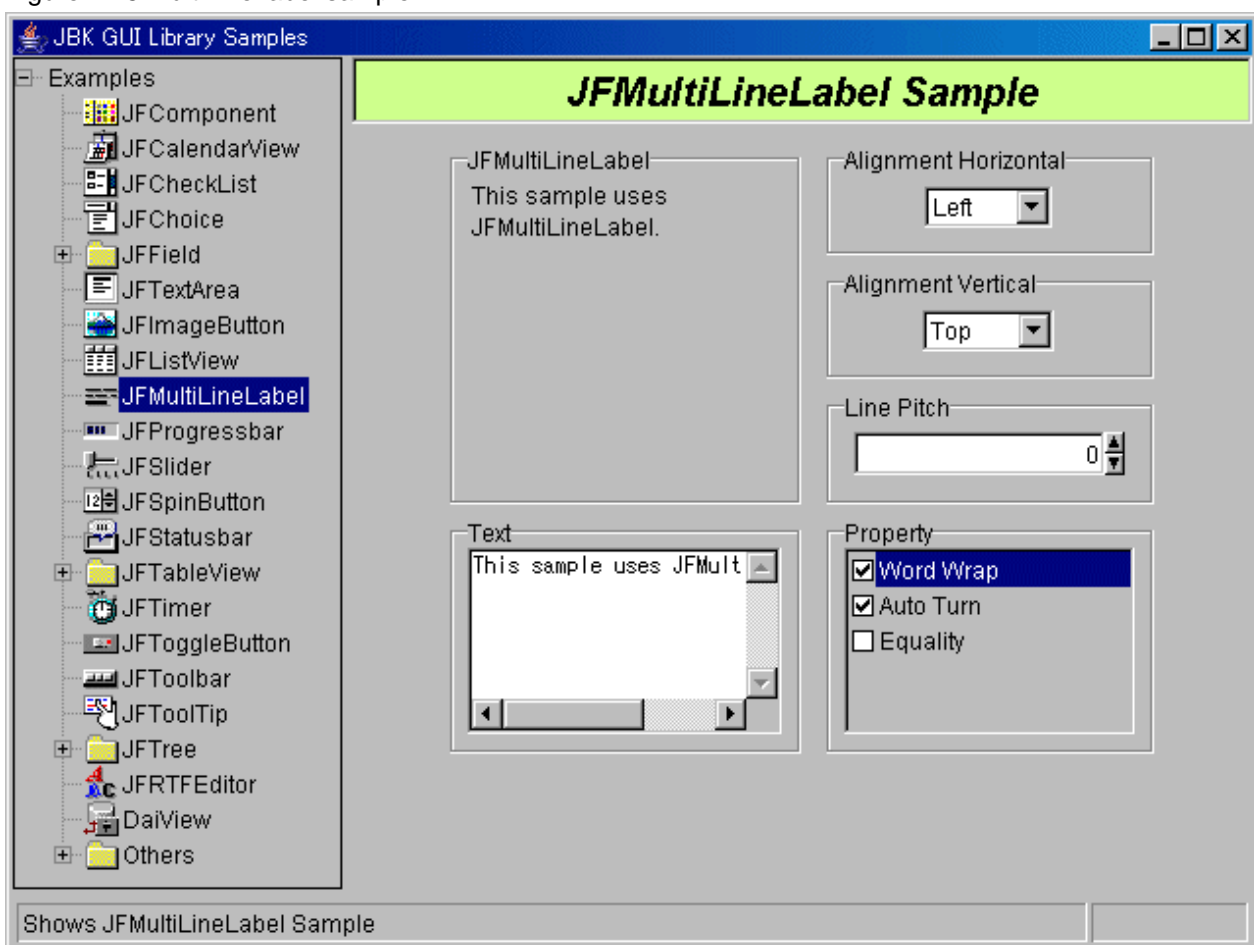
1.3.13 JFMultiLineLabel(Multi-Line Label)

The multi-line label sample displays the multi-line label using the JFMultiLineLabel class.

The multi-line label sample uses the following classes:

- JFMultiLineLabel
- JFFieldString
- JFCheckList
- JFGroupbox

Figure 1.15 Multi-line label sample



Operation method

- JFMultiLineLabel

Multi-line label display field

- Text

Specify the contents of multi-line label.

- Horizontal alignment

Specify the horizontal-direction position when displaying the multi-line label. Specifiable positions are as follows:

- Display positions: Left/Center/Right
- Vertical alignment
Specify the vertical-direction position when displaying the multi-line label. Specifiable positions are as follows:
 - Display positions: Top/Center/Bottom
- Line pitch
Specify the line spacing when displaying the multi-line label.
- Property
The item containing a check in the check box is enabled.
 - Word Wrap: Wraps words.
 - Auto Run: Wraps data automatically.
 - Equality: Displays data equally.

Source codes

- Multi-line label sample: "Sample Storage Destinations"/jfmultilinelabel/JFMultiLineLabelSample.java

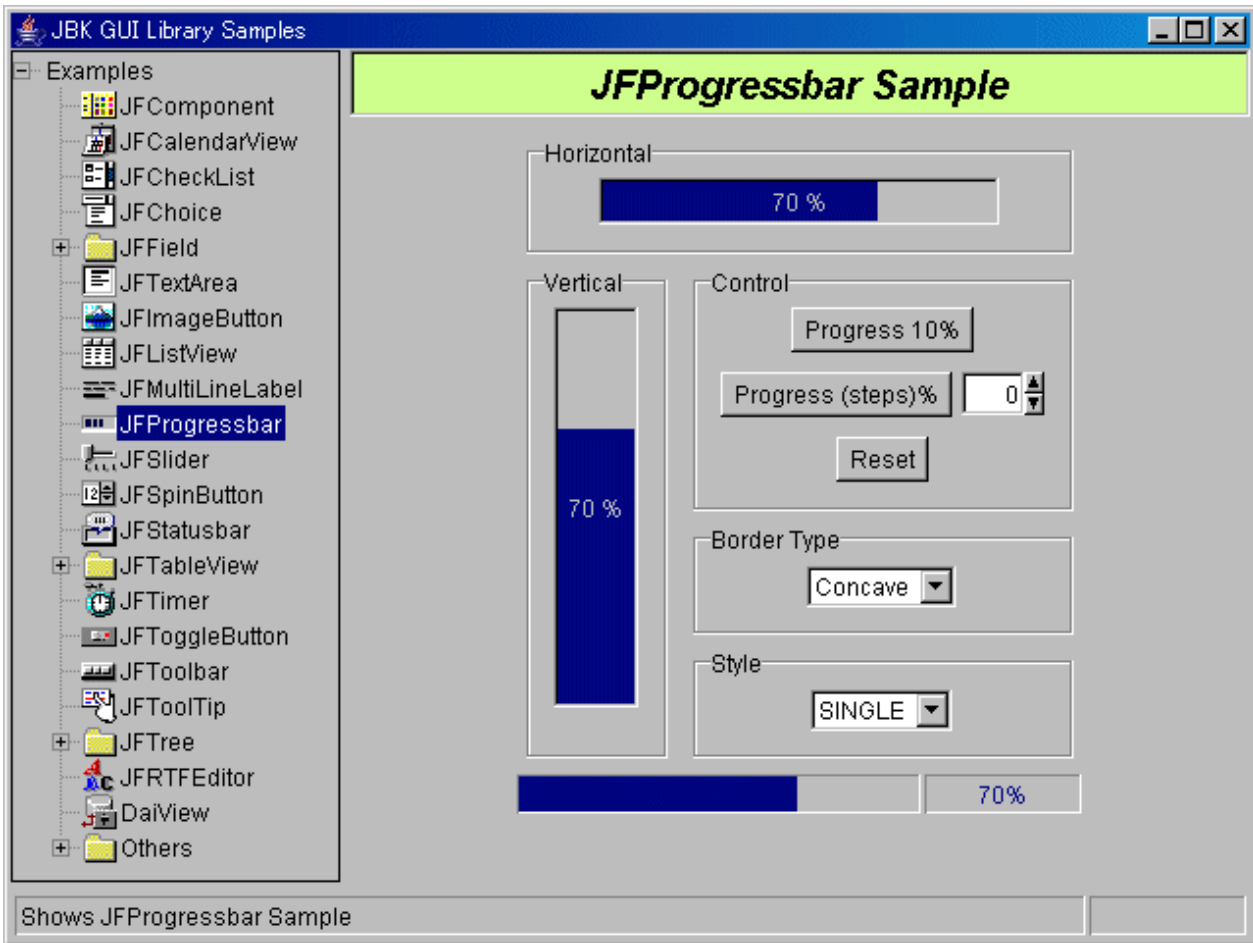
1.3.14 JFProgressbar(Progress Bar)

The progress bar sample displays the progress status using the JFProgressbar class. The progress bar installed (linked) in the status bar is also displayed.

The progress bar sample uses the following classes:

- JFProgressbar
- JFFieldLong
- JFStatusbar
- JFGroupbox

Figure 1.16 Progress bar sample



Operation method

- Horizontal/Vertical

Displays the horizontal-type/vertical-type progress bar.

- Control

- Progress 10%: Pressing this button ensures 10-% progress.

- Progress (step)%: Pressing this button ensures progress for the specified numeric value in the numeric value input field on the right side.

- Reset: Pressing this button returns to 0-% display.

- Border type

Specify the type of gridlines. Specifiable types are as follows:

- Types of gridlines: Concave/Convex/Flat

- Bar style

Specify the form indicating the progress status of the progress bar.

- Bar forms: SINGLE/MULTI

Source codes

- Progress bar sample:"Sample Storage Destinations"/jfprogressbar/JFProgressbarSample.java

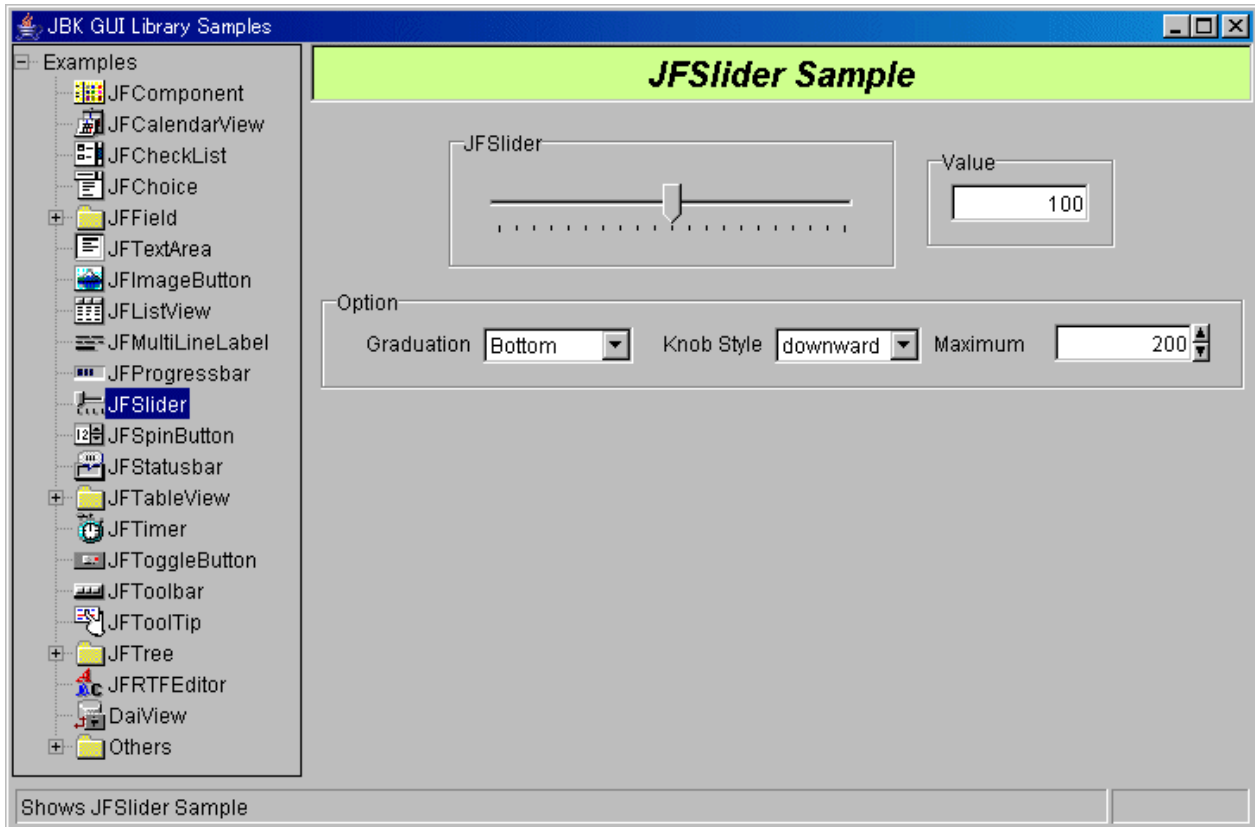
1.3.15 JFSlider(Slider)

The slider sample sets the type of control, type of scale, and maximum value using the JFSlider class.

The slider sample uses the following classes:

- JFSlider
- JFChoice
- JFFieldLong
- JFSpinButton

Figure 1.17 Slider sample



Operation method

- JFSlider
 - Displays and operates the slider.
- Value
 - Displays the value of the slider control position.
- Option
 - Graduation
 - Specify how to display the scale with Choice.
 - Knob
 - Specify the type of control with Choice.
 - Maximum
 - The left edge of the slider is set to "0" and the right edge of the slider is set to "200" as the default.

Pressing the upper/lower spin button sets the maximum value of the slider from "20" to "200" every 20 steps.

Source codes

- Slider sample:"Sample Storage Destinations"/jfslder/JFSliderSample.java

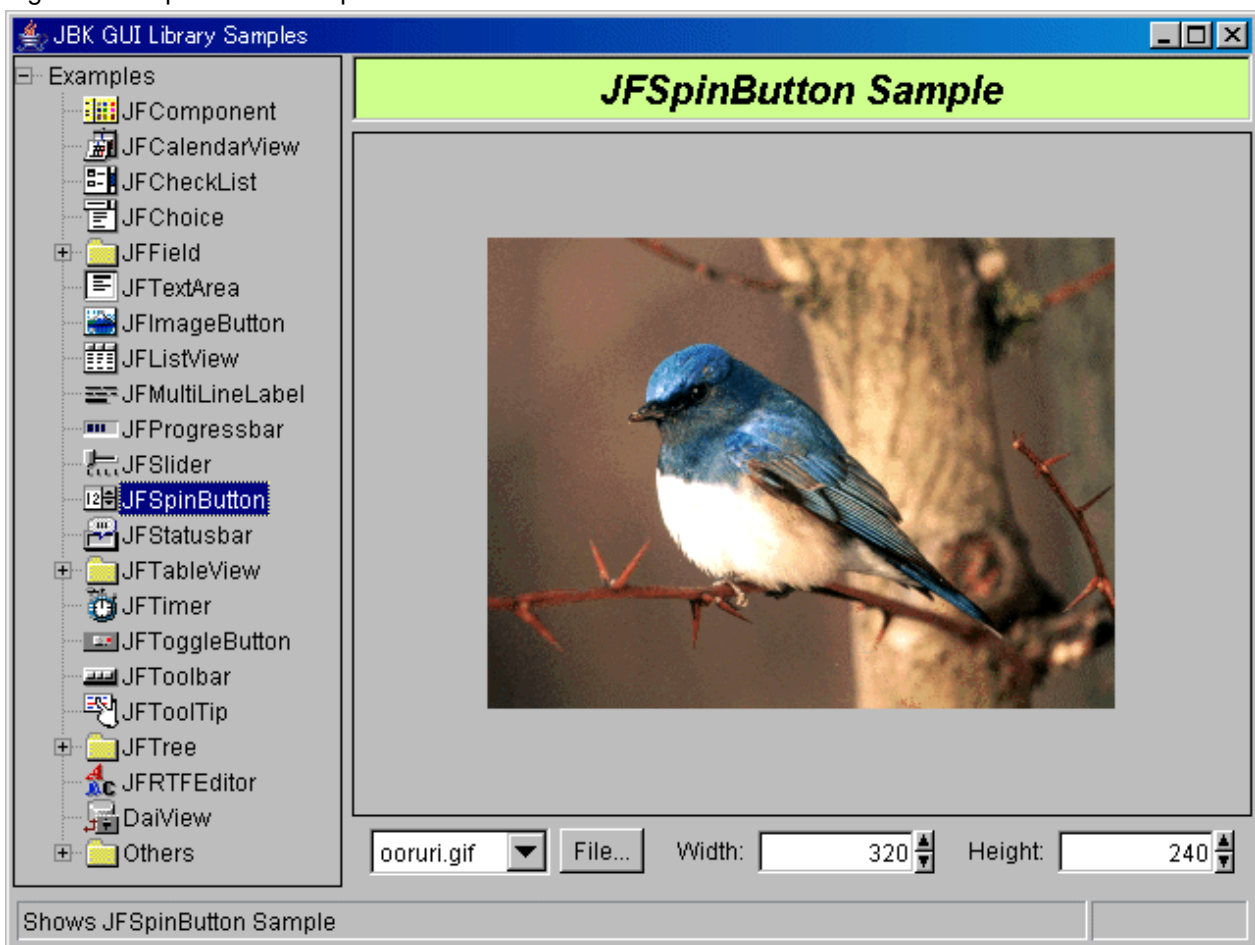
1.3.16 JFSpinButton(Spin Button)

The spin button sample enlarges or reduces image data using the component (called SpinButtonLong) combining the JFSpinButton class with JFFieldLong class.

The spin button sample uses the following classes:

- JFSpinButton
- JFFieldLong

Figure 1.18 Spin button sample



Operation method

- List box section (Choice)

Select the image data to be displayed with Choice.

Sample data uguisu.gif, umineko.gif, ooruri.gif, and tobi.gif are provided.

- File

Pressing this button opens the [File Selection] dialog; therefore, the user image data can also be displayed.

Specifiable image data types are GIF and JPEG formats.

- Width and Height

Operating the spin button arrow using the mouse or keyboard enlarges or reduces the image data.

The input field of the spin button changes as follows, depending on the image data size.

- The image data is displayed in white for the actual image size.
- The image data is displayed in light blue when the image data is reduced.
- The image data is displayed in red when the image data is enlarged.

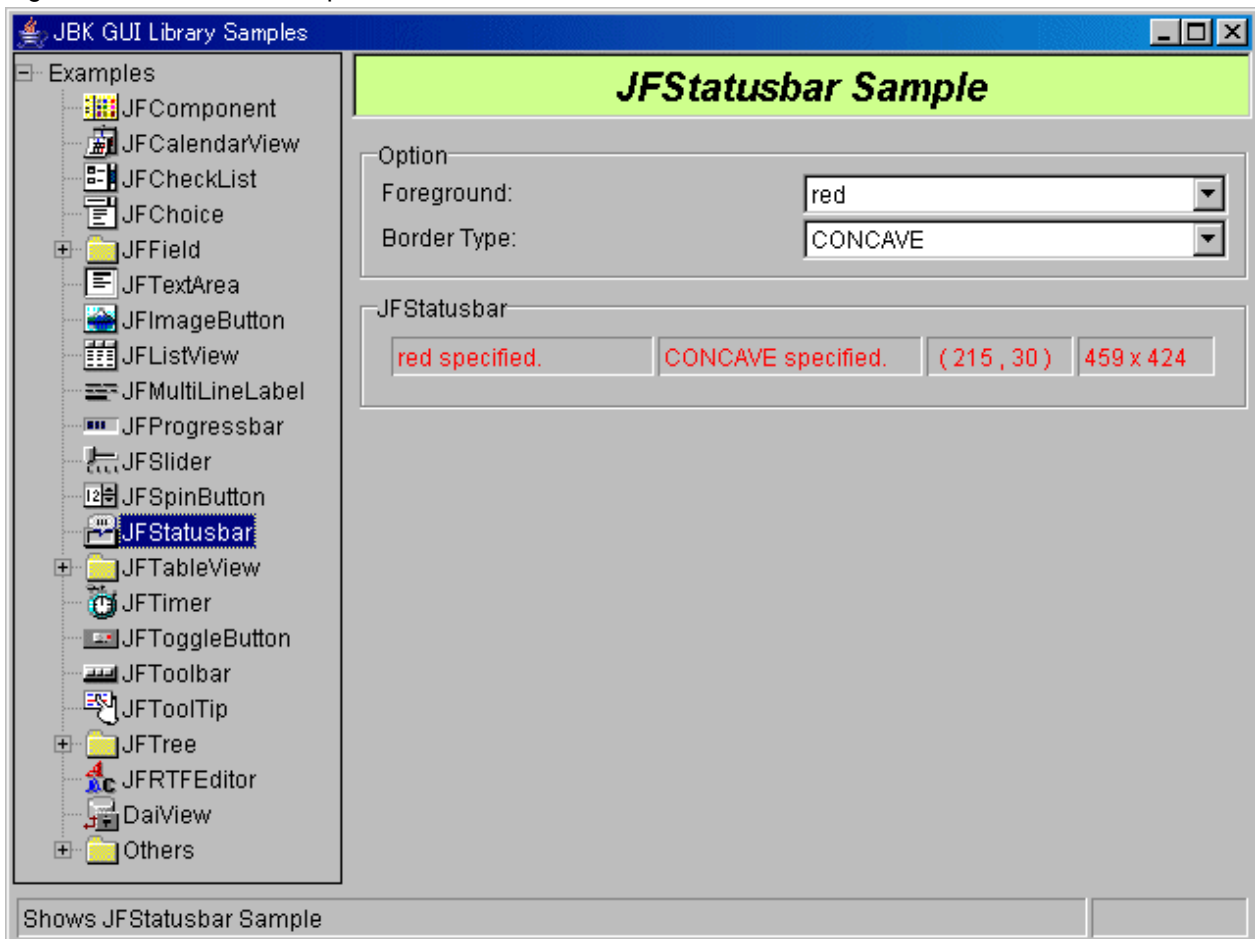
Source codes

- Spin button sample: "Sample Storage Destinations"/jfspinbutton/JFSpinButtonSample.java
- Spin button with the input field: "Sample Storage Destinations"/common/SpinButtonLong.java
- Image panel: "Sample Storage Destinations"/jfspinbutton/ImagePanel.java

1.3.17 JFStatusBar(Status Bar)

The status bar sample uses the JFStatusBar class.

Figure 1.19 Status bar sample



Operation method

- Options
 - Foreground color
 - Select the foreground color of the status bar with Choice.

- Pane line types

Select the type of the status bar pane with Choice. Specifiable types are as follows:

- Types of gridlines: concave/convex/plane

- JFStatusBar

Displays the following items in sequence from the left on the status bar pane.

- Foreground color
- Pane line types
- Position of mouse coordinates (Initially set to the upper-left of the "Result Display Section" on the [Sample Integration] screen.
- Window size (Initially set to the size of the "Result Display Section" on the [Sample Integration] screen.

Source codes

- Status bar sample:"Sample Storage Destinations"/jfstatusbar/JFStatusBarSample.java

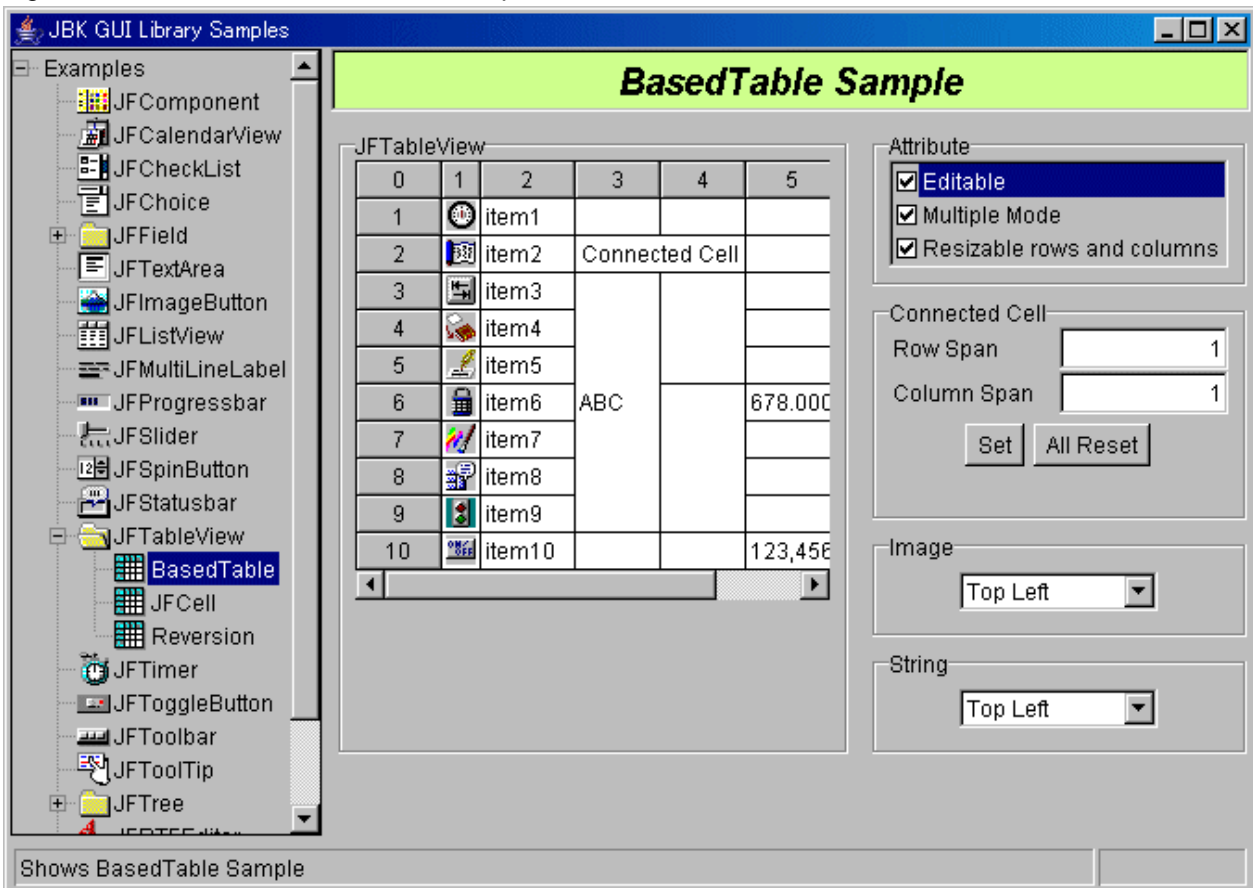
1.3.18 JFTableView(Table View Basic Function)

The table view sample displays and updates data in the table format using the JFTableView class.

The table view sample uses the following classes:

- JFTableView
- JFCheckList
- JFGroupbox
- JFPanel

Figure 1.20 Table view basic function sample



Operation method

- JFTableView

Displays and operates the table view.

- Properties

The item containing a check in the check box is enabled.

- Edit: Updates the table view.
- Multi selections: Multiple selections are enabled for each cell.
- Size change of row and column: The size of the lines and columns can be changed.

- Combined cells

Displays the selected cell information and specifies the option for connecting cells.

- Vertical Multiple/Horizontal Multiple: Displays the selected cell information. To connect the cell, specify the cell scale factor.
- Settings: Sets the connecting cell with the specified vertical and horizontal scale factors.
- Clear all: Releases the connecting cell.

- Location of image show

Specify the position for displaying the image. Specifiable positions are as follows:

- View positions: top left/top/top right/left/center/right/bottom left/bottom/bottom right/all/icon status

- Location of text show

Specify the position for displaying the character string. Specifiable positions are as follows:

- View positions: top left/top/top right/left/center/right/bottom left/bottom/bottom right

Source codes

- Table view basic function sample:"Sample Storage Destinations"/jftableview/simple/JFTableViewSample.java

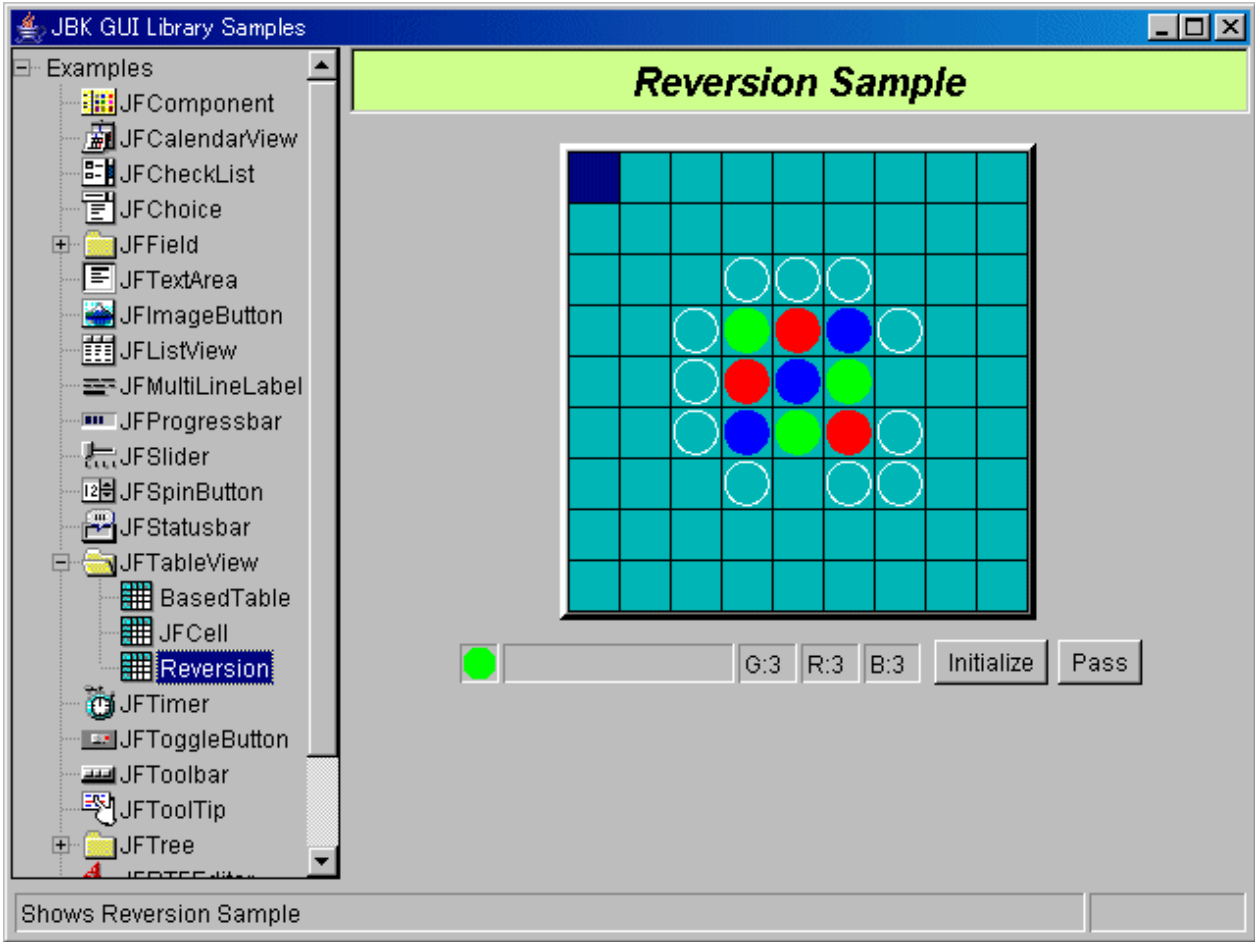
1.3.19 Reversion(Color Reversion)

The color reversion sample places the remote cell between the local cells and reverses the remote cell using the application of the JFTableView class. The most number of cells in the local color wins the game.

The color reversion sample uses the following classes:

- JFTableView
- JFStatusBar

Figure 1.21 Color reversion sample



Operation method

Double-click the cell so that the remote cell (red/blue) is placed between local cells (green). White circles indicate the positions the local cells can be positioned.

- Status bar
Displays the game progress and the number of cells gained thus far.
- Initialization
Pressing the button initializes the game.
- Pass
Press the button if there is no position for reversing the remote color. Pressing this button transfers the right to the remote system.

Source codes

- Color reversion sample: "Sample Storage Destinations"/jftableView/reversion/Reversion.java

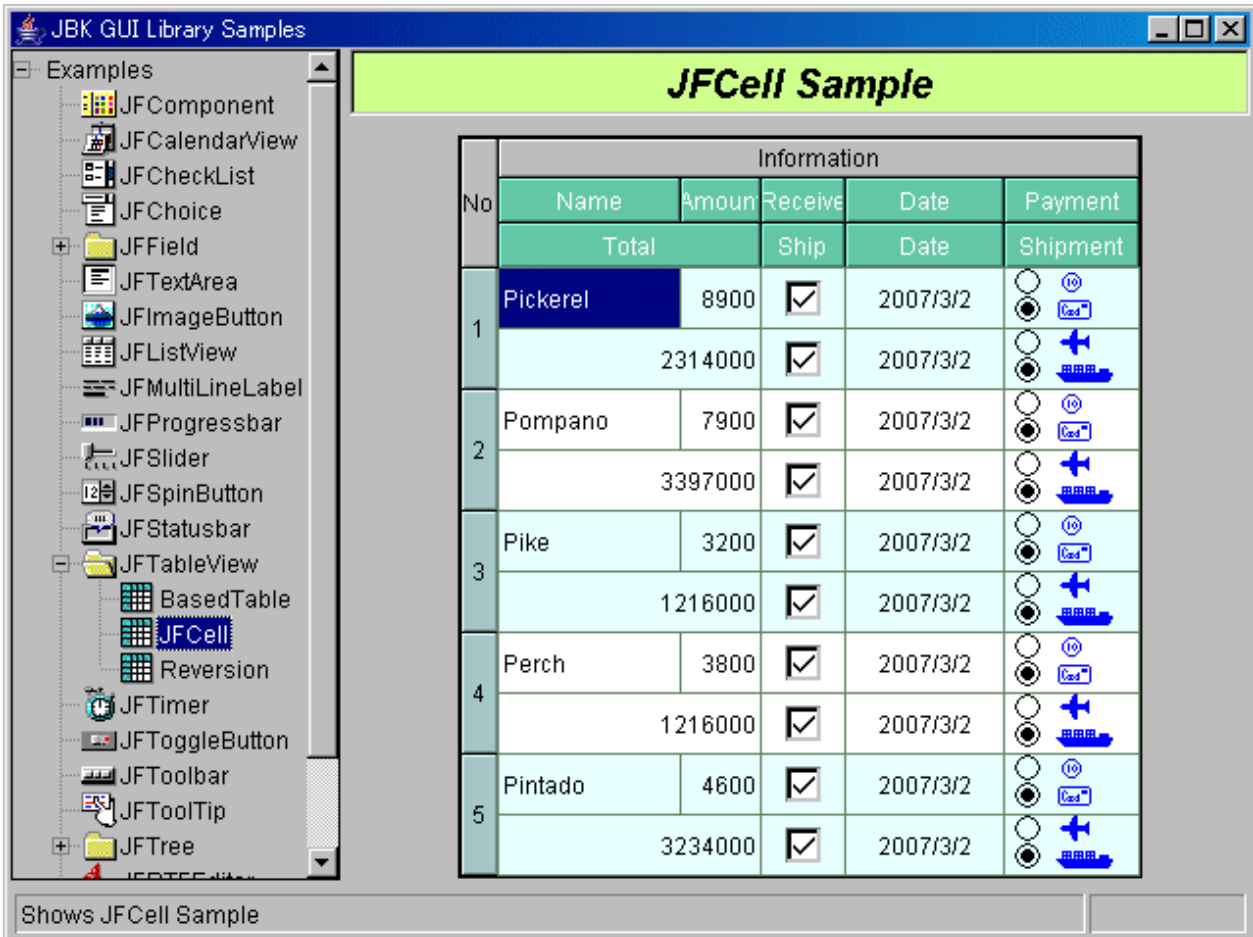
1.3.20 JFCell(Cell)

The cell sample uses the application of the JFCell class. The cell sample uses the following classes:

- JFTableView
- JFCalendarView
- JFCell

- JFInputCellObject
- JFPaintCellObject
- JFStringCellObject
- JFPanel

Figure 1.22 Cell sample



Operation method

- Income/Shipment
 - The check box image is pasted on the cell. The ON/OFF status of the check can be switched.
- Date
 - Double-clicking the date displays the calendar to select the date.
- Payment/Sending out
 - The image can be selected.

Source codes

- Cell sample: "Sample Storage Destinations"/jftableView/jfcell/JFTableViewCellSample.java

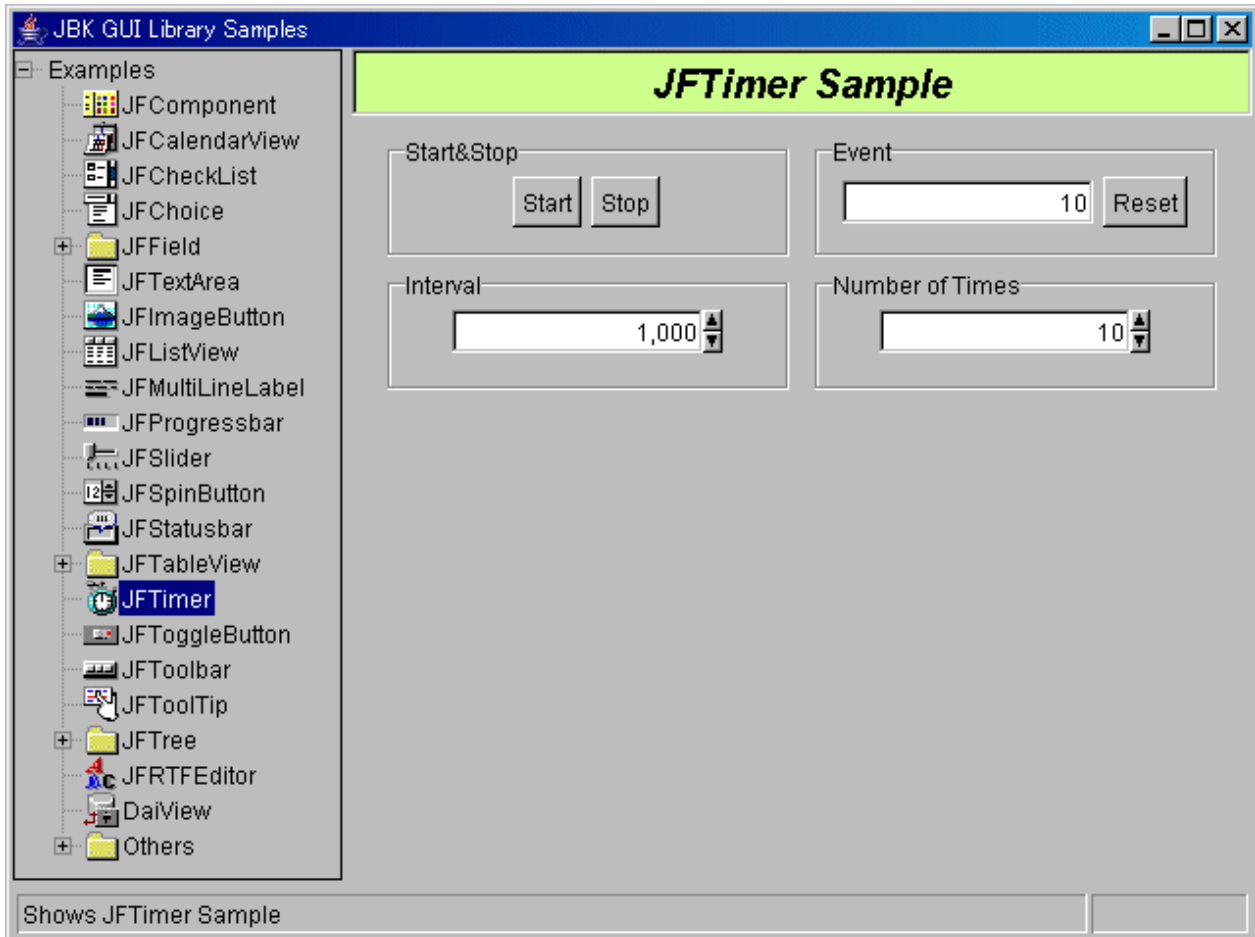
1.3.21 JFTimer(Timer)

The timer sample generates an event using the JFTimer class.

The timer sample uses the following classes:

- JFTimer
- JFSpinButton
- JFImageButton
- JFGroupbox

Figure 1.23 Timer sample



Operation method

- Start/End

Pressing the [Start] button generates events and pressing the [End] button terminates the timer.

Pressing the [Start] button generates events for the number of event occurrences with the specified occurrence interval and then terminates the timer.

- Event

Displays the number of generated events.

- Interval

Specify the event occurrence interval in milliseconds.

- Count

Specify the number of event occurrences.

Source codes

- Timer sample:"Sample Storage Destinations"/jftimer/JFTimerSample.java

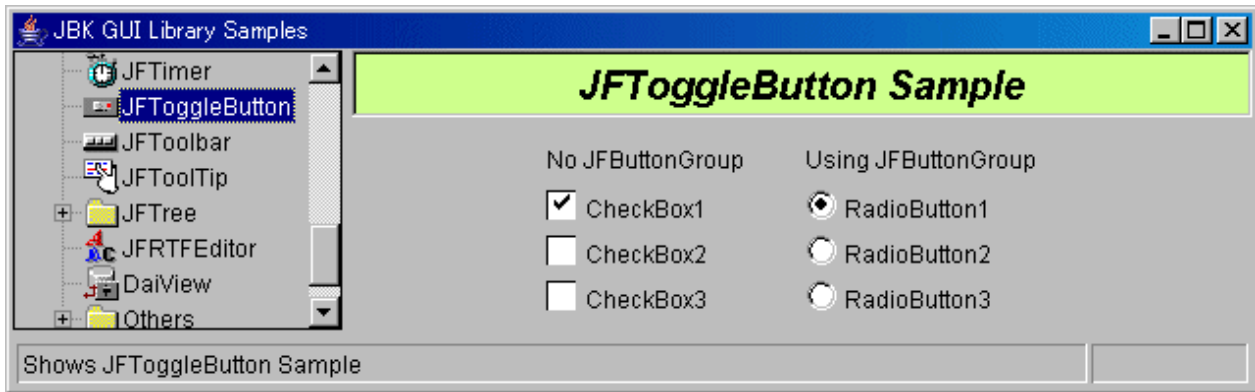
1.3.22 JFToggleButton(Toggle Button)

The toggle button sample uses the JFToggleButton class.

The toggle button sample uses the following classes:

- JFToggleButton
- JFButtonGroup

Figure 1.24 Toggle button sample



Operation method

- No JFButtonGroup

Clicking the check box switches the ON/OFF status.

- CheckBox1
- CheckBox2
- CheckBox3

- Using JFButtonGroup

Clicking one radio button switches the other two radio buttons to OFF status and switches the clicked radio button to ON status.

Only one radio button within the same group can be switched to ON status.

- RadioButton1
- RadioButton2
- RadioButton3

Source codes

- Toggle button sample:"Sample Storage Destinations"/jftogglebutton/JFToggleButtonSample.java

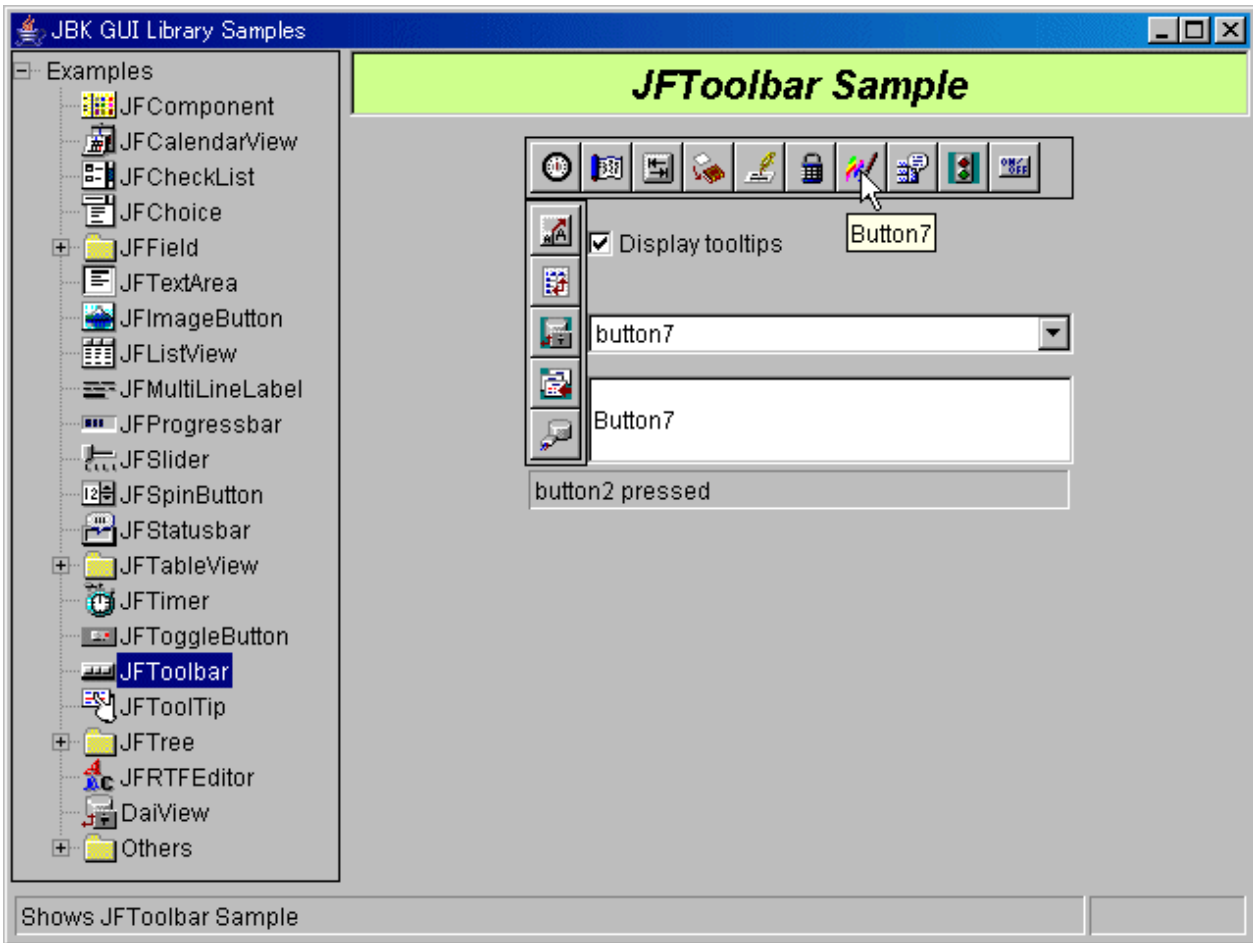
1.3.23 JFToolBar(Toolbar)

The toolbar sample aligns the tool bar vertically and horizontally using the JFToolBar class.

The toolbar sample uses the following classes:

- JFToolBar
- JFImageButton
- JFChoice

Figure 1.25 Toolbar sample



Operation method

- Buttons

Clicking each button displays the type of button clicked on the status bar.

- Tool tip show

Clicking a check box displays the tool tip when moving the mouse on the toolbar.

- Changing the tool tip contents

The 15 items corresponding to each button have been defined in the list box of Choice. Changing the character string displayed in the input field under the selected item displays the tool tip with the changed character string. Specify the character string using up to 10 characters.

Source codes

- Toolbar sample:"Sample Storage Destinations"/jftoolbar/JFToolbarSample.java

1.3.24 JFToolTip(Tooltip View)

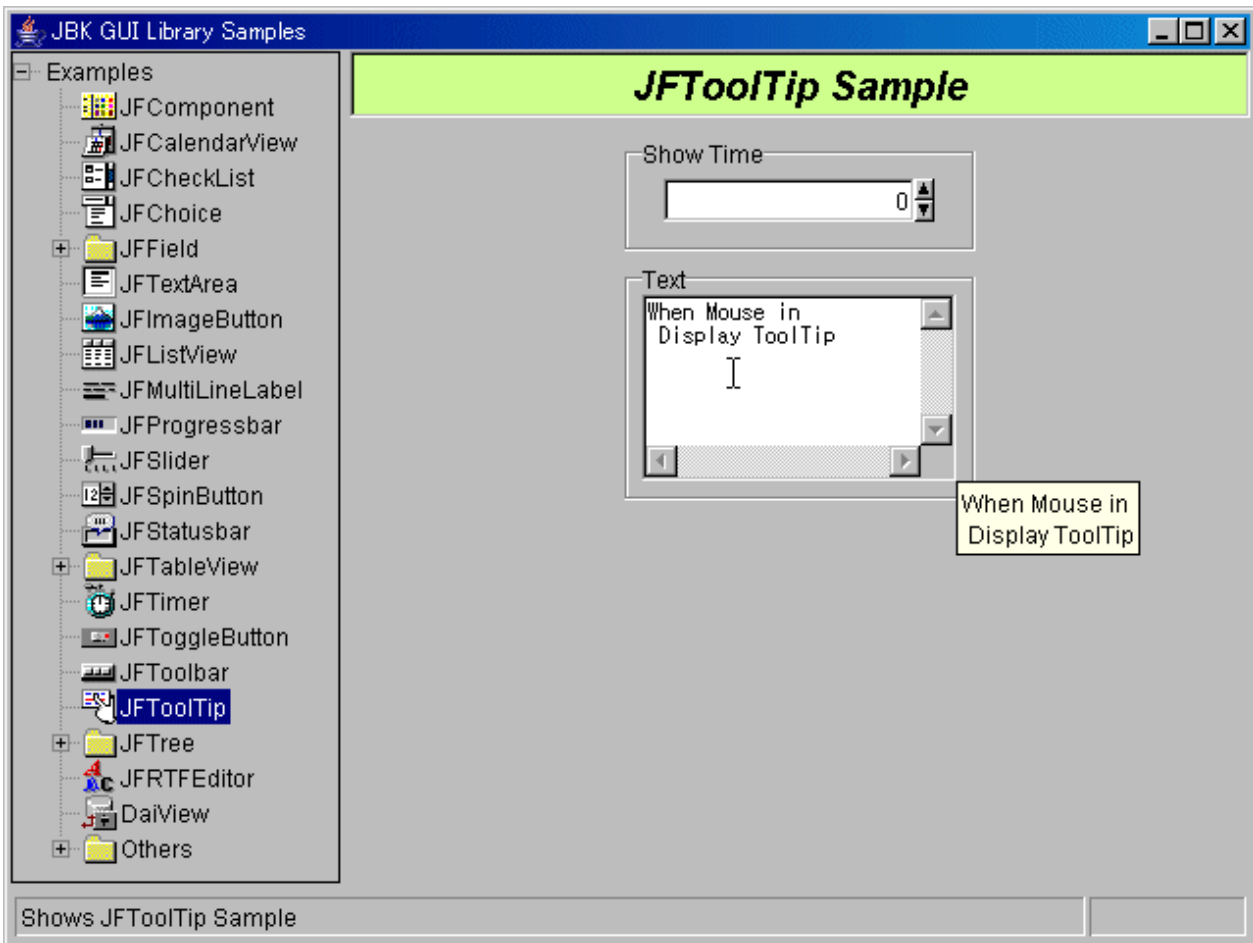
The tool tip view sample displays the tool tip using the JFToolTip class.

The tool tip view sample uses the following classes:

- JFToolTip
- JFSpinButton
- JFFieldLong

- JFGroupbox

Figure 1.26 Tooltip view sample



Operation method

When the mouse cursor is positioned in the text, the tool tip is displayed.

- Displaying time
Specify the tool tip view time in milliseconds.
- Text
Specify the character string (text) to be displayed on the tool tip.

Source codes

- Tooltip view sample:"Sample Storage Destinations"/jftooltip/JFToolTipSample.java

1.3.25 JFTree(Tree View Basic Function)

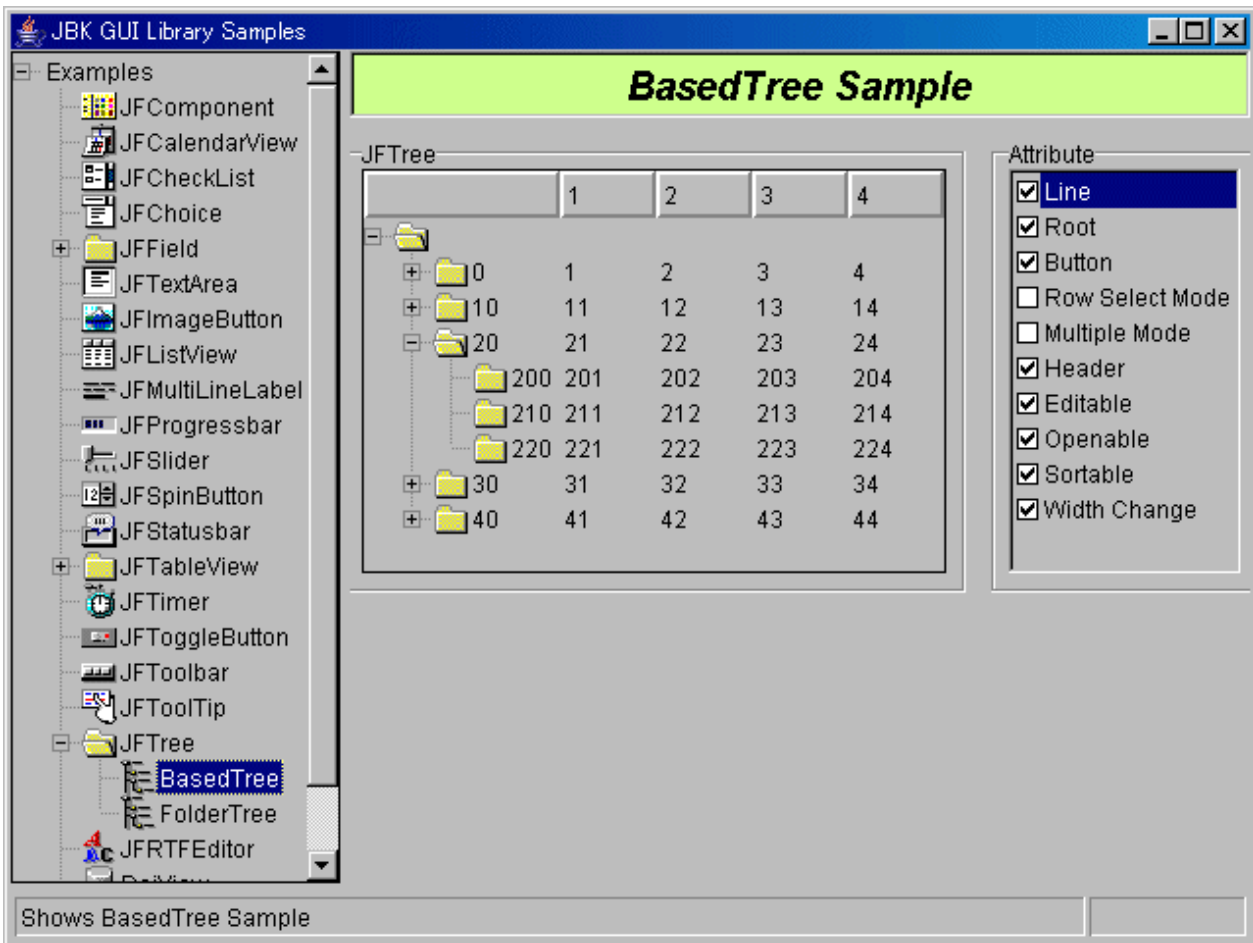
The sample creates and displays the basic tree view (multi-column tree) using the JFTree class and JFSimpleTreeNode class.

The basic tree view sample uses the following classes:

- JFTree
- JFSimpleTreeNode
- JFCheckList
- JFGroupbox

- JFPanel

Figure 1.27 Tree view sample



Operation method

- JFTree

Displays the data (multi-column data) having column information in the tree format.

- Properties

The item containing a check in the check box is enabled.

- Order line display: Displays the node for each hierarchy.
- Root display: Displays the root.
- Button display: Displays the button (image) at the beginning of each node.
- Row selection: The line can be selected.
- Multiple selection: Multiple nodes can be selected.
- Head display: Displays the header.
- Editable: The first column can be edited.
- Openable and closeable: Each node can be opened or closed.
- Sortable: Data can be sorted for each column.
- Width changeable: The cell width can be changed.

Source codes

- Tree view sample:"Sample Storage Destinations"/jftree/simple/JFTreeSample.java

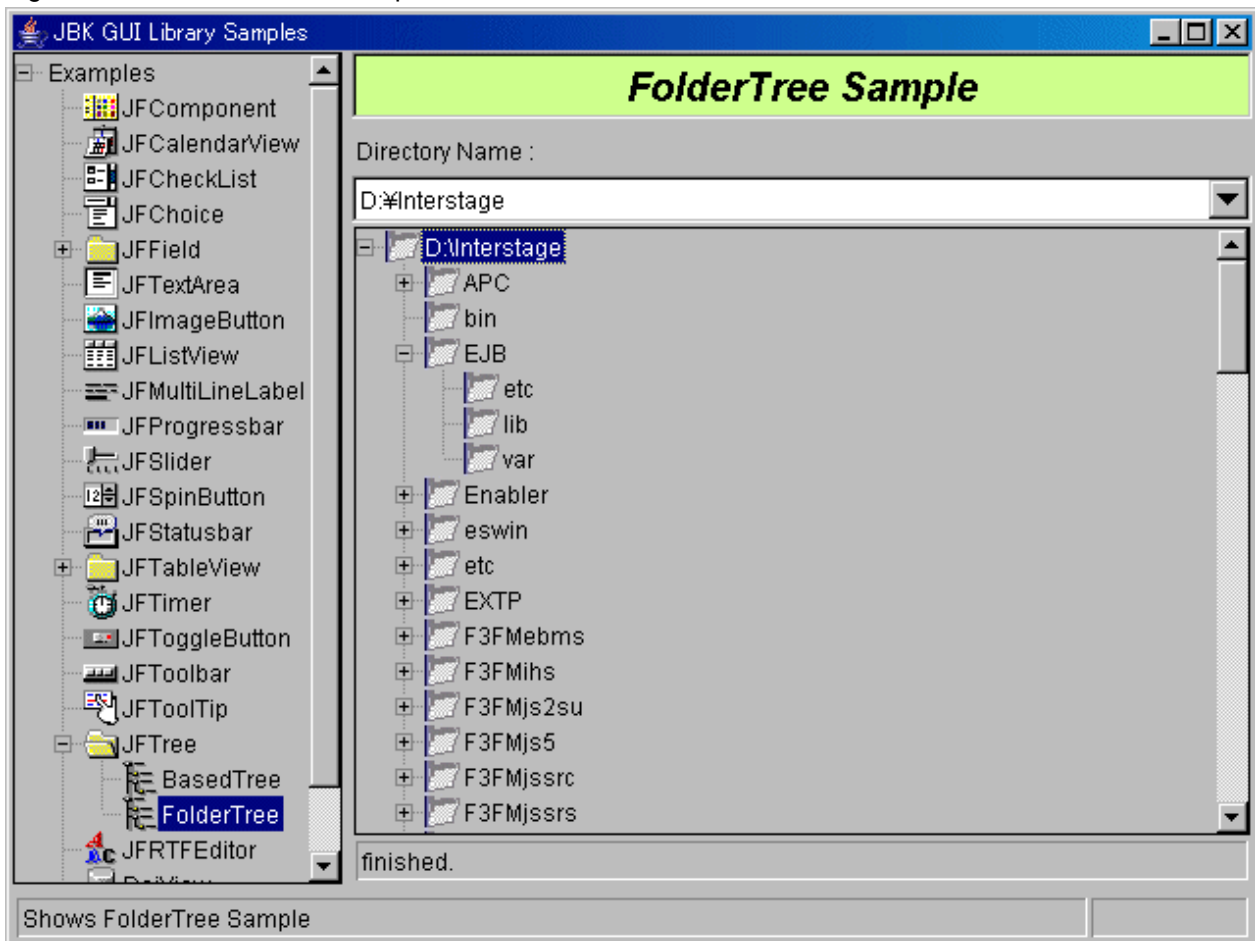
1.3.26 FolderTree(Folder Tree View)

The folder tree view sample creates and displays the folder tree under an optional folder using the JFTree class and JFSimpleTreeNode class.

The folder tree view sample uses the following classes:

- JFTree
- JFSimpleTreeNode
- JFStatusBar
- JFChoice
- JFGroupbox
- JFPanel

Figure 1.28 Folder tree view sample



Operation method

- Entering the folder to be displayed with a full path in the folder path input field (Choice under "folder path:") and pressing the [Enter] key displays the folder tree using the specified path as the root node.

The processing status is displayed on the lower status bar.

- Selecting an optional folder displayed on the folder tree using the mouse or keyboard displays the selected folder on the status bar.

- The path data entered can be stored in the folder path input field (Choice under "folder path:") and the path can be selected therefrom.

Source codes

- Folder tree view sample:"Sample Storage Destinations"/jftree/filefolder/JFTreeSample.java
- Folder tree view:"Sample Storage Destinations"/jftree/filefolder/FolderTreeView.java
- Folder tree view listener:"Sample Storage Destinations"/jftree/filefolder/FolderTreeViewListener.java

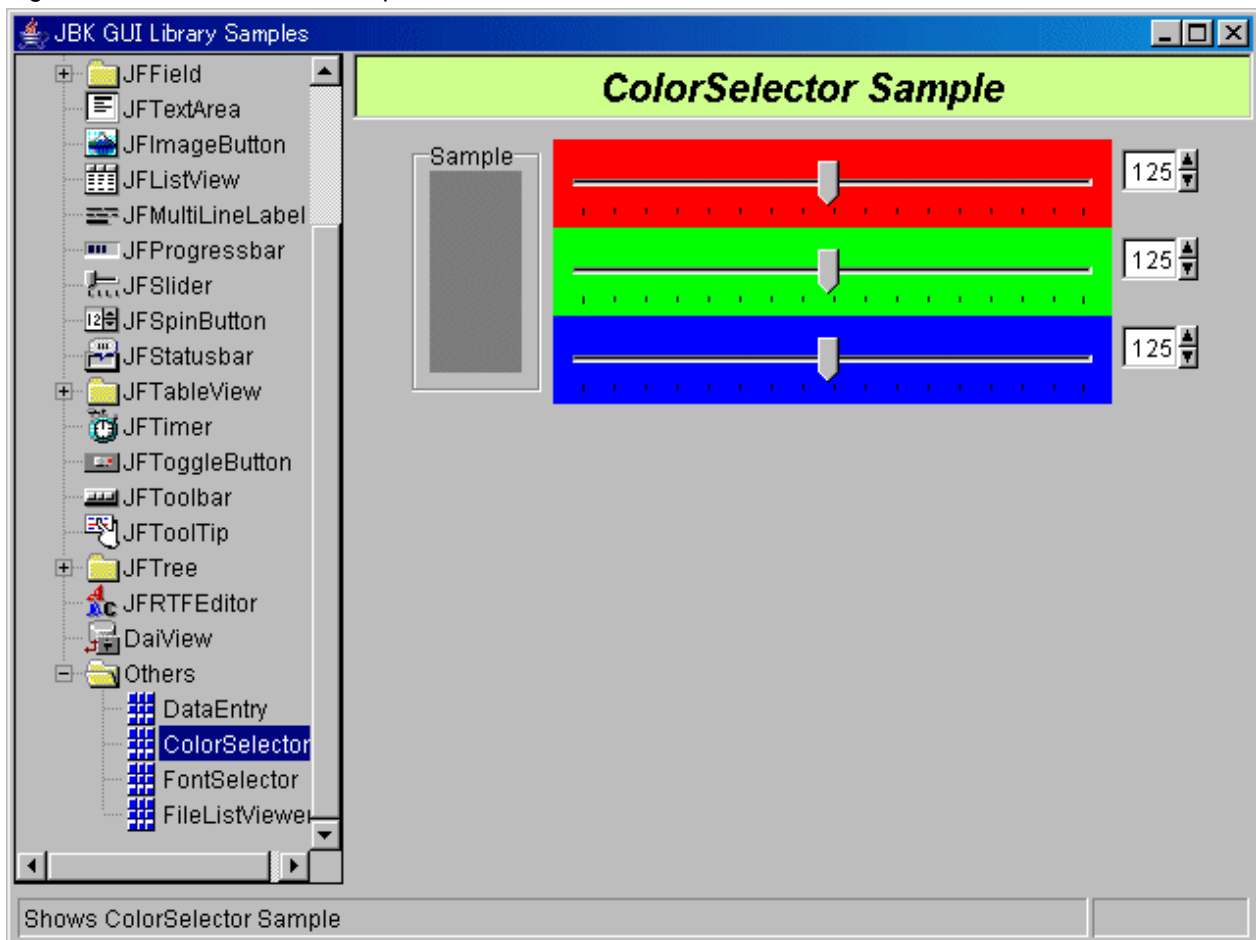
1.3.27 ColorSelector(Color Selection)

The color selection sample changes the RGB value and selects the color using the slider and spin button.

The color selection sample uses the following classes:

- JFSlider
- JFFieldLong
- JFSpinButton
- JFGroupbox
- JFPanel

Figure 1.29 Color selection sample



Operation method

- Sample
 - Displays the color specified with the slider or spin button.

- Slider operation panel

Operating the slider control using the mouse or keyboard displays the color corresponding to the slider position (value) and displays the value on the spin button operation panel.

- Spin button operation panel

Operating the spin button arrow using the mouse or keyboard changes the value and displays the color corresponding to the changed value on the sample. The operations of the spin button and slider are linked; therefore, the slider control changes in accordance with the spin button operation. Data can be entered in the section displaying the numeric value. The value can also be specified directly. The specifiable range is between 0 to 255.

Source codes

- Color selection sample: "Sample Storage Destinations"/common/ColorSelector.java

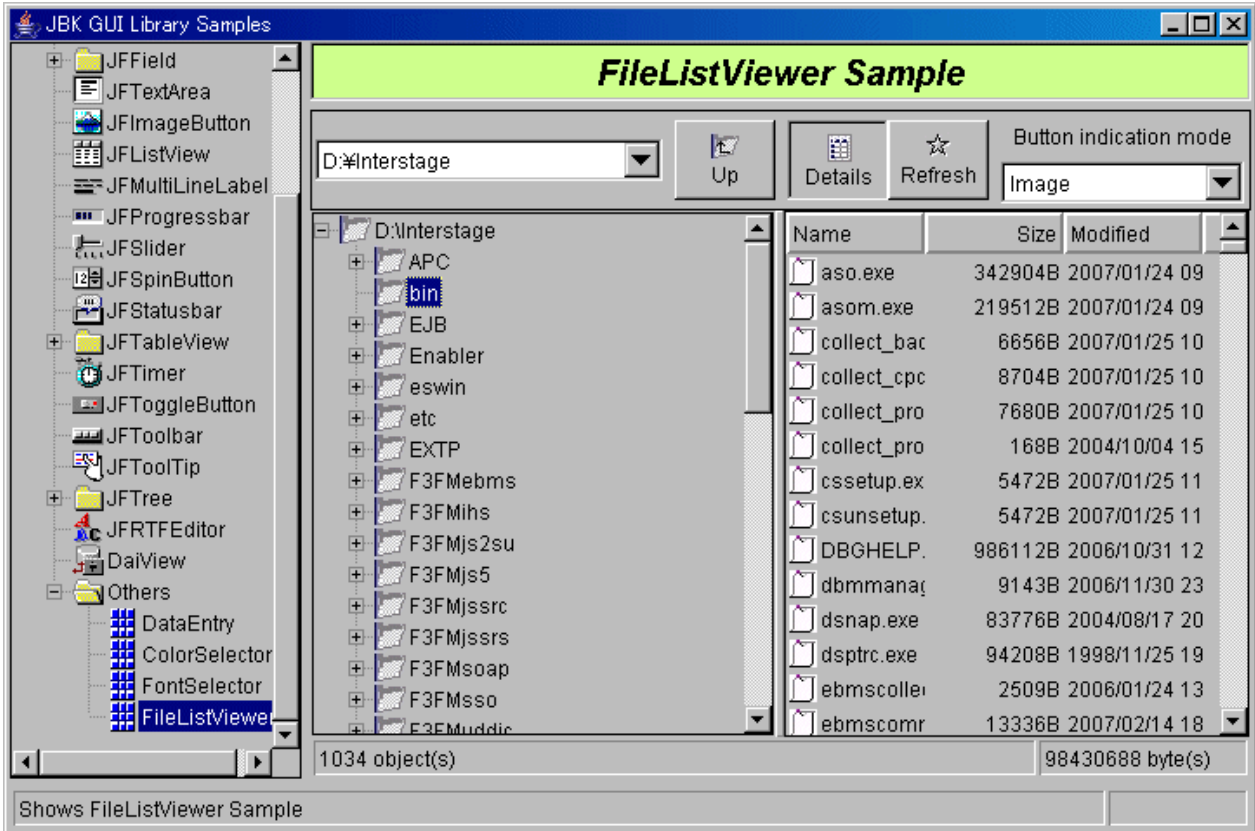
1.3.28 FileListViewer(File List View)

The file list view sample displays the folder tree using the specified path as the root item. Selecting an optional folder displays the files under the folder.

The file list view sample uses the following classes:

- JFTree
- JFListView
- JFChoice
- JFToolbar
- JFImageButton
- JFStatusbar
- JFPanel

Figure 1.30 File list view sample



Operation method

- Toolbar

The toolbar consists of the folder path input field and command buttons.

- Folder path input field

Specifying the folder to be displayed with a full path and pressing the [Enter] key displays the folder tree using the path as the root node. Path data entered can be stored in the folder path input field and the path can be selected from the list box.

- Command button

The command buttons include the following:

- Up: Moves the folder to the upper hierarchy and moves the selection position to the parent item.
- Detail: Displays files in detail. The size and date are displayed in addition to the file names.
- Update: Displays the latest information of the folder tree of the specified path.

- Button display mode

The view format of the toolbar buttons can be selected from Label, Image, and Image & Label. Select the view format and press the [Enter] key.

- Tree view section

The folder tree specified in the folder path input field appears in the tree view section. Selecting an optional folder displayed in the folder tree using the mouse or keyboard displays a list of files under the folder in the file list view section on the right side. The components used in the tree view section are the same as the components used in the [Figure 1.28 Folder tree view sample](#).

- File list view section

In the file list view section, selecting the folder displayed in the tree view section displays a list of files under the folder.

If the column header is clicked when the details are displayed, the column is sorted and displayed.

- Status section

The status of the file list view sample is displayed in the status section.

Source codes

- File list view sample:"Sample Storage Destinations"/filelistviewer/FileListViewer.java
- File list view:"Sample Storage Destinations"/filelistviewer/FileList.java
- Toolvar view:"Sample Storage Destinations"/filelistviewer/FileListViewerToolBar.java
- Folder tree view:"Sample Storage Destinations"/jftree/filefolder/FolderTreeView.java
- Folder tree view listener:"Sample Storage Destinations"/jftree/filefolder/FolderTreeViewListener.java

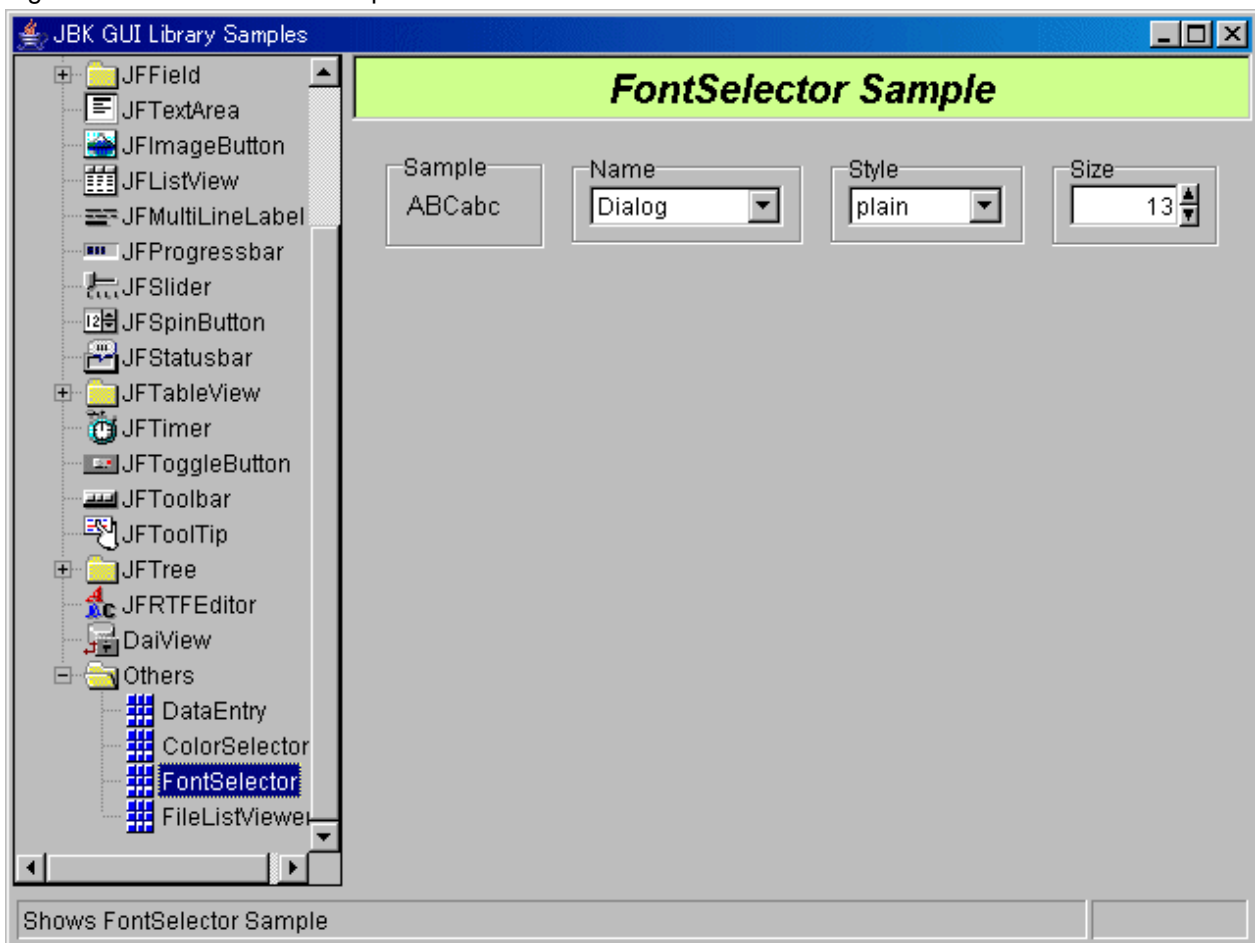
1.3.29 FontSelector(Font Selection)

The font selection sample specifies the character string font to be displayed.

The font selection sample uses the following classes:

- JFGroupbox
- JFFieldLong
- JFSpinButton

Figure 1.31 Font selection sample



Operation method

- Sample

Displays the character string with the specified font.

- Name

Selecting the font type with Choice displays the font type.

- Style

Selecting the font style with Choice displays the font style.

- Size

Operating the spin button arrow using the mouse or keyboard and specifying the font size displays the character string with the specified font size.

Source codes

- Font selection sample:"Sample Storage Destinations"/common/FontSelector.java

Chapter 2 Explanation of UI Screen Control Library Samples

The UI Screen Control Library consists the UI Screen Control Facility the Window Control Facility and the Object Control Facility.

This section explains the application samples of UI Screen Control Library.

Sample Storage Destinations

```
"JBK installation folder"\examples\gui\ctrl
```

2.1 Compilation

The compiled file (.class file: Byte code) is included with the sample so the sample can be executed without change. However, if the user corrected the sample source code, compilation is required.

In this case, run compile.bat(in sample storage destinations) to start compilation.

2.2 Start Method

Run example.bat (in sample storage destinations) to start the sample.

2.3 UI Screen Control Library Samples

Screen operation library samples include the following:

- UI Screen Control Panel Sample
- UI Screen Control Panel Sample (Light Weight Version)
- UI Screen Control Tabbed Panel Sample
- Scroll Panel Sample
- Scroll Panel Sample (Light Weight Version)
- Window Control Facility Sample
- Object Control Facility Sample

2.3.1 UI Screen Control Panel Sample

The UI Screen Control Panel Sample transits the UI Screen and transmits screen data using UI Screen Control Panel(JFCCardPanel).

This sample can be referenced as a simple sample of the program handling mass screens.

The UI Screen Control Panel Sample uses the following classes and interface:

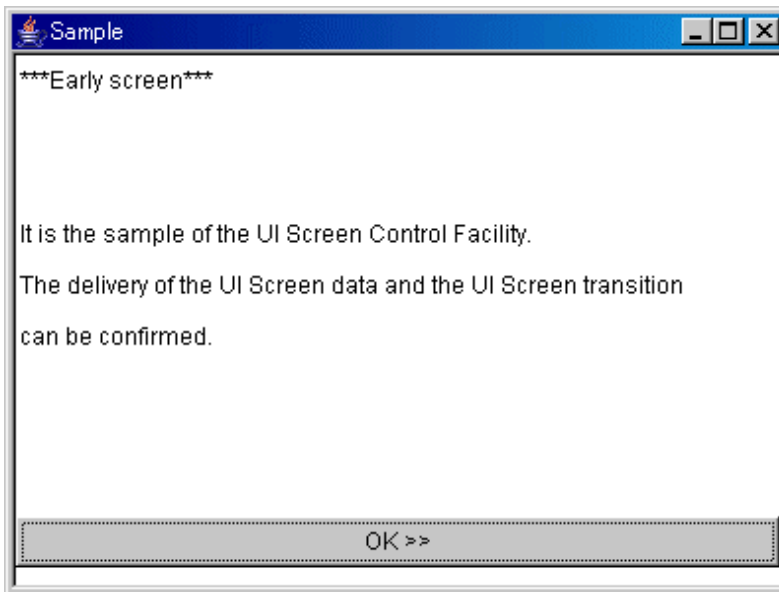
- JFCEntryComponent
- JFCCardPanel
- JFCEntryPanel
- JFCEntryInterfacePanel
- JFCMultiPagePanel

Operation method

Proceed as follows to operate this sample.

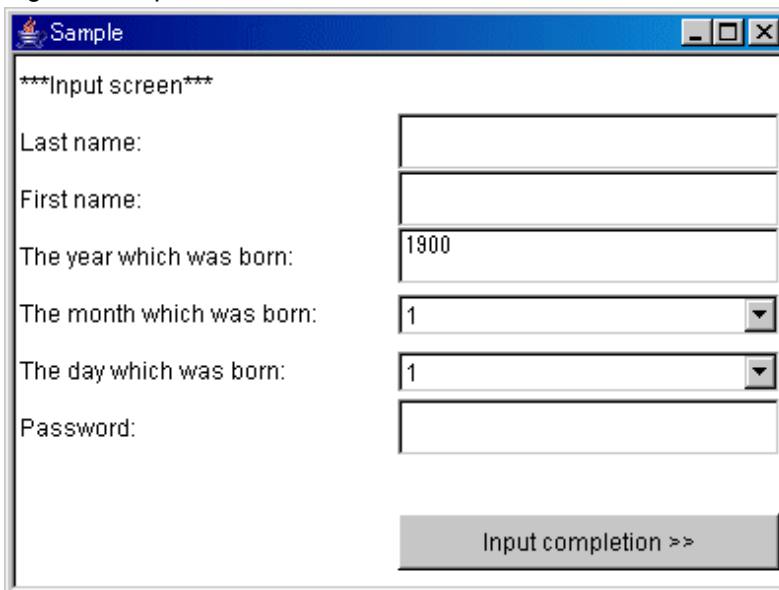
1. Start the application.

Figure 2.1 Application screen



2. Pressing the [OK >>] button transits to the next screen and displays the screen for entering the name and the date (Year/Month/Day).

Figure 2.2 Input screen



3. Entering an appropriate value and pressing the [Input completion >>] button transits to the next screen and displays the screen for confirming the name and the date (Year/Month/Day).

Figure 2.3 Confirmation screen

Confirmation screen	
Last name:	Fujitsu
First name:	Taro
The year which was born:	1950
The month which was born:	4
The day which was born:	1
Password:	test

<< Correction Confirmation >>

4. Checking the displayed value and pressing the [Confirmation >>] button transits to the next screen and displays the screen for verifying the name and the password. Pressing the [<< Correction] button here returns to the previous screen, on which the value can be reentered.

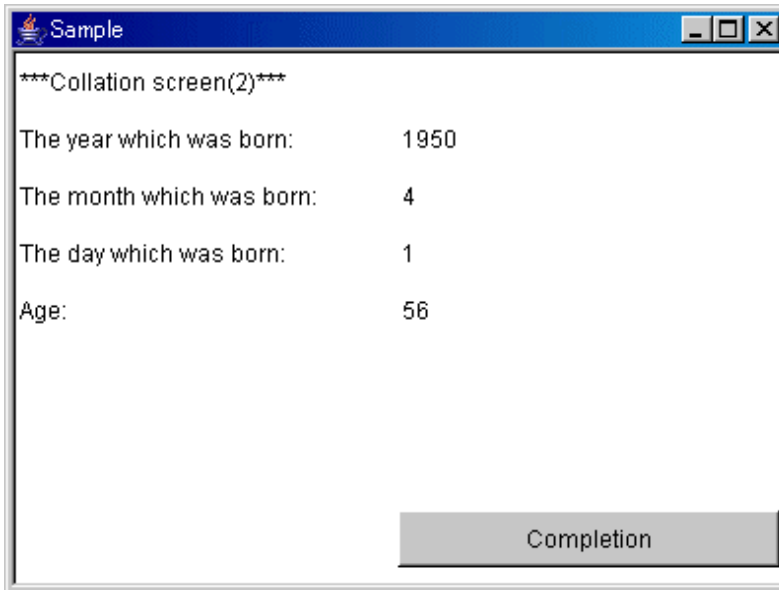
Figure 2.4 Collation screen (1)

Collation screen(1)	
Last name:	Fujitsu
First name:	Taro
Password:	<input type="text"/>

Collation

5. Entering the password displayed on the previous screen and pressing the [Collation] button transits to the next page and displays the screen for displaying the age and the birth date.

Figure 2.5 Collation screen (2)



6. Pressing the [Completion] button returns the sample to the [Input screen].

Source codes

- Data Entry Field sample:"Sample Storage Destinations"/JFCCardPanel/EntryButton.java
- Data Entry Field sample:"Sample Storage Destinations"/JFCCardPanel/EntryField.java
- Data Entry Field sample:"Sample Storage Destinations"/JFCCardPanel/EntryLabel.java
- UI Screen Control Panel sample:"Sample Storage Destinations"/JFCCardPanel/Sample.java
- UI Screen Control Panel sample:"Sample Storage Destinations"/JFCCardPanel/SampleCard.java
- UI Screen Panel sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel0.java
- UI Screen Panel sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel1.java
- UI Screen Panel for operating on Data Entry Fields sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel2.java
- UI Screen Panel for operating on Data Entry Fields across multiple pages sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel3.java

2.3.2 UI Screen Control Panel Sample(Light Weight Version)

The Light Weight UI Screen Control Panel Sample transits the UI Screen and transmits screen data using Light Weight UI Screen Control Panel(JFLCardPanel).

This sample can be referenced as a simple sample of the program handling mass screens.

The Light Weight UI Screen Control Panel Sample uses the following classes and interface:

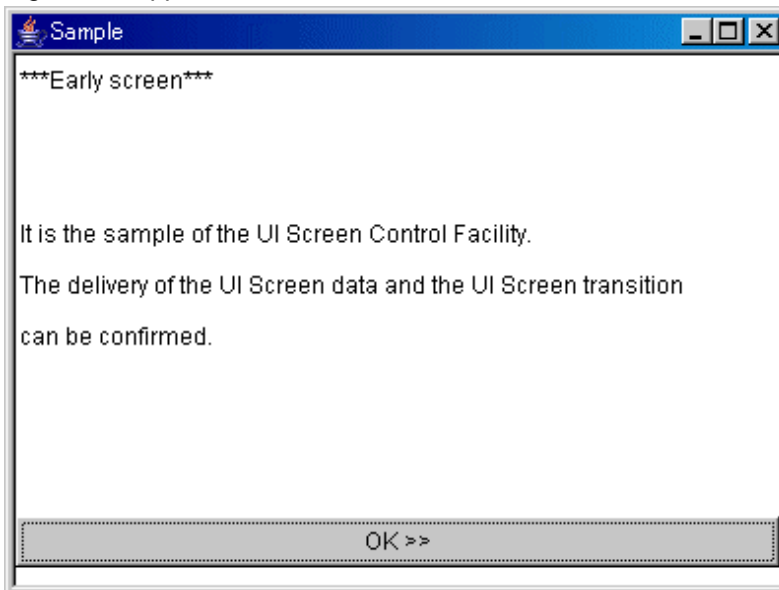
- JFCEntryComponent
- JFLCardPanel
- JFLEntryPanel
- JFLEntryInterfacePanel
- JFLMultiPagePanel

Operation method

Proceed as follows to operate this sample.

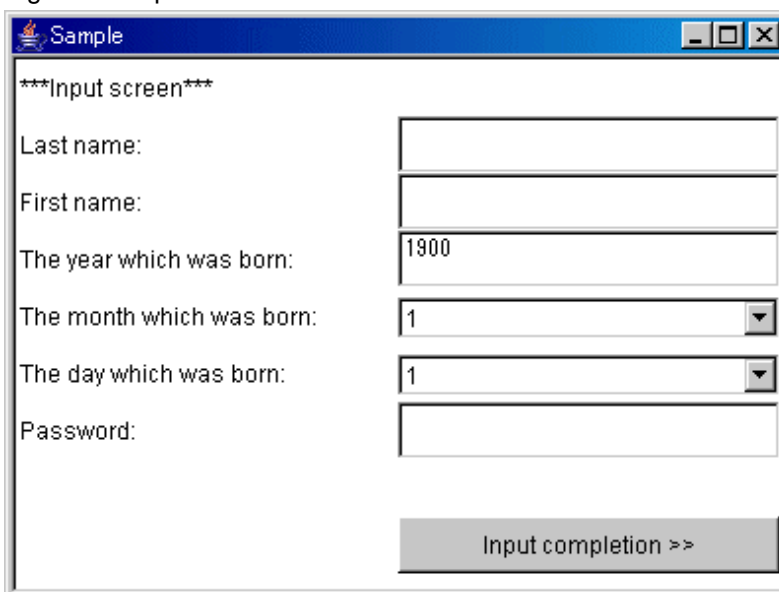
1. Start the application.

Figure 2.6 Application screen



2. Pressing the [OK >>] button transits to the next screen and displays the screen for entering the name and the date (Year/Month/Day).

Figure 2.7 Input screen



The screenshot shows a window titled "Sample" with a blue title bar. The main content area contains the text: "***Input screen***". Below this, there are several input fields: "Last name:" with an empty text box, "First name:" with an empty text box, "The year which was born:" with a text box containing "1900", "The month which was born:" with a dropdown menu showing "1", "The day which was born:" with a dropdown menu showing "1", and "Password:" with an empty text box. At the bottom of the window, there is a grey button labeled "Input completion >>".

3. Entering an appropriate value and pressing the [Input completion >>] button transits to the next screen and displays the screen for confirming the name and the date (Year/Month/Day).

Figure 2.8 Confirmation screen

Confirmation screen	
Last name:	Fujitsu
First name:	Taro
The year which was born:	1950
The month which was born:	4
The day which was born:	1
Password:	test

<< Correction Confirmation >>

4. Checking the displayed value and pressing the [Confirmation >>] button transits to the next screen and displays the screen for verifying the name and the password. Pressing the [<< Correction] button here returns to the previous screen, on which the value can be reentered.

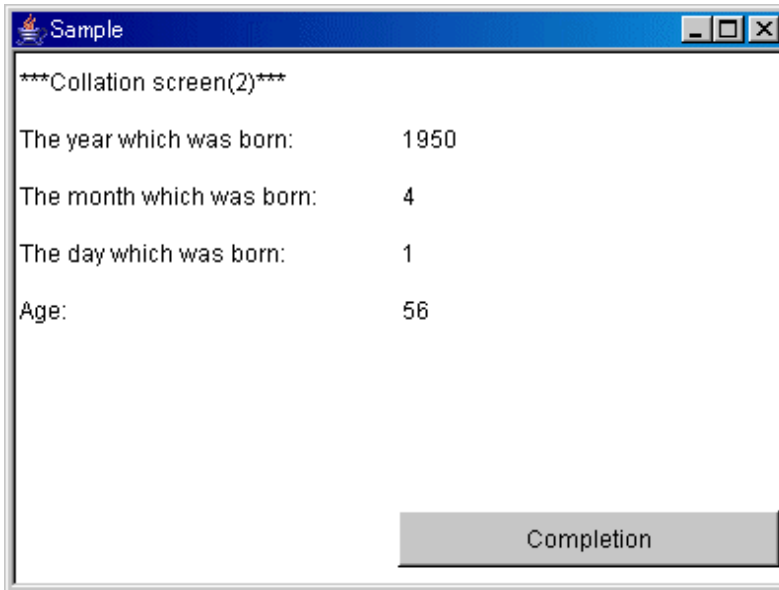
Figure 2.9 Collation screen (1)

Collation screen(1)	
Last name:	Fujitsu
First name:	Taro
Password:	<input type="text"/>

Collation

5. Entering the password displayed on the previous screen and pressing the [Collation] button transits to the next page and displays the screen for displaying the age and the birth date.

Figure 2.10 Collation screen (2)



6. Pressing the [Completion] button returns the sample to the [Input screen].

Source codes

- Data Entry Field sample:"Sample Storage Destinations"/JFCCardPanel/EntryButton.java
- Data Entry Field sample:"Sample Storage Destinations"/JFCCardPanel/EntryField.java
- Data Entry Field sample:"Sample Storage Destinations"/JFCCardPanel/EntryLabel.java
- UI Screen Control Panel sample:"Sample Storage Destinations"/JFCCardPanel/Sample.java
- UI Screen Control Panel sample:"Sample Storage Destinations"/JFCCardPanel/SampleCard.java
- UI Screen Panel sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel0.java
- UI Screen Panel sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel1.java
- UI Screen Panel for operating on Data Entry Fields sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel2.java
- UI Screen Panel for operating on Data Entry Fields across multiple pages sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel3.java

2.3.3 UI Screen Control Tabbed Panel

The UI Screen Control Tabbed Panel Sample transits the UI Screen and transmits screen data using UI Screen Control Tabbed Panel(JFCTabbedPanel).

This sample can be referenced as a simple sample of the program handling mass screens.

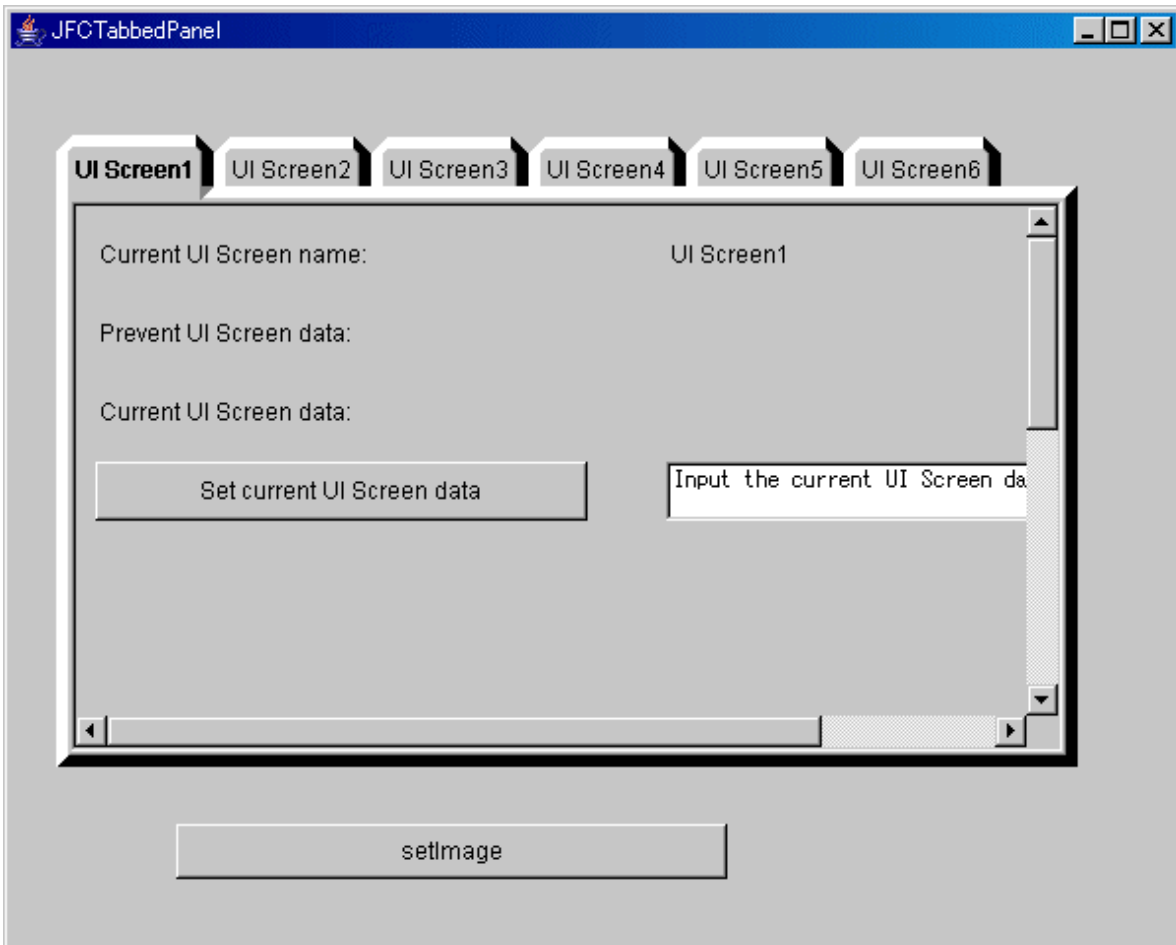
The UI Screen Control Tabbed Panel Sample uses the following classes and interface:

- JFCTabbedPanel
- JFCEntryPanel

Screen structure

The following screen is just after started this sample.

Figure 2.11 Screen just after started



There are components on this screen.

- [Current UI Screen name] label
Displays the current UI Screen name.
- [Prevent UI Screen data] label
Displays the previous UI Screen name.
After this sample started and there is no data in the previous UI Screen, the UI Screen name is not displayed.
- [Current UI Screen data] label
Displays the current UI Screen name.
If the current UI Screen has no data, the UI Screen data is not displayed.
- [Set current UI Screen data] button and [Input the current UI Screen data] text field.
To set the data of the current UI Screen.
To set the data, input any string data on the text field and press the button.
- [Screen Transition] button and [Input the UI Screen name to be transited] text field.
To transit a UI Screen.
To transit a UI Screen, input a UI Screen name on the text field and press the button.
- [setImage] button
The Button to set image on the tab of the current UI Screen.

Operation method

The following is how to operate this sample.

- Start this sample, and set the image on the tab.

Press the [SetImage] button, then the image is displayed on the tab.

- Set the current UI Screen data.

Input any string data on the [Input the current UI Screen data] textfield, and press the [Set current UI Screen data] button.

Then the current data is displayed on the [Current UI Screen data] label.

- Transit a UI Screen.

The following three way is to transit a UI Screen.

- Input a UI Screen name on the [Input the UI Screen name to be transited] text field, and press the [Screen Transition] button.
- Click on a tab by mouse.
- Type [CTRL]+[TAB] or [CTRL]+[SHIFT]+[TAB] by keyboard.

Source codes

- UI Screen Control Tabbed Panel Applet sample:"Sample Storage Destinations"/JFCTabbedPanel/Sample.java
- UI Screen Control Tabbed Panel sample:"Sample Storage Destinations"/JFCTabbedPanel/TabPanel.java
- UI Screen Transitions Panel sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel0.java
- UI Screen Transitions Panel sample:"Sample Storage Destinations"/JFCCardPanel/SamplePanel1.java

2.3.4 Scroll Panel Sample

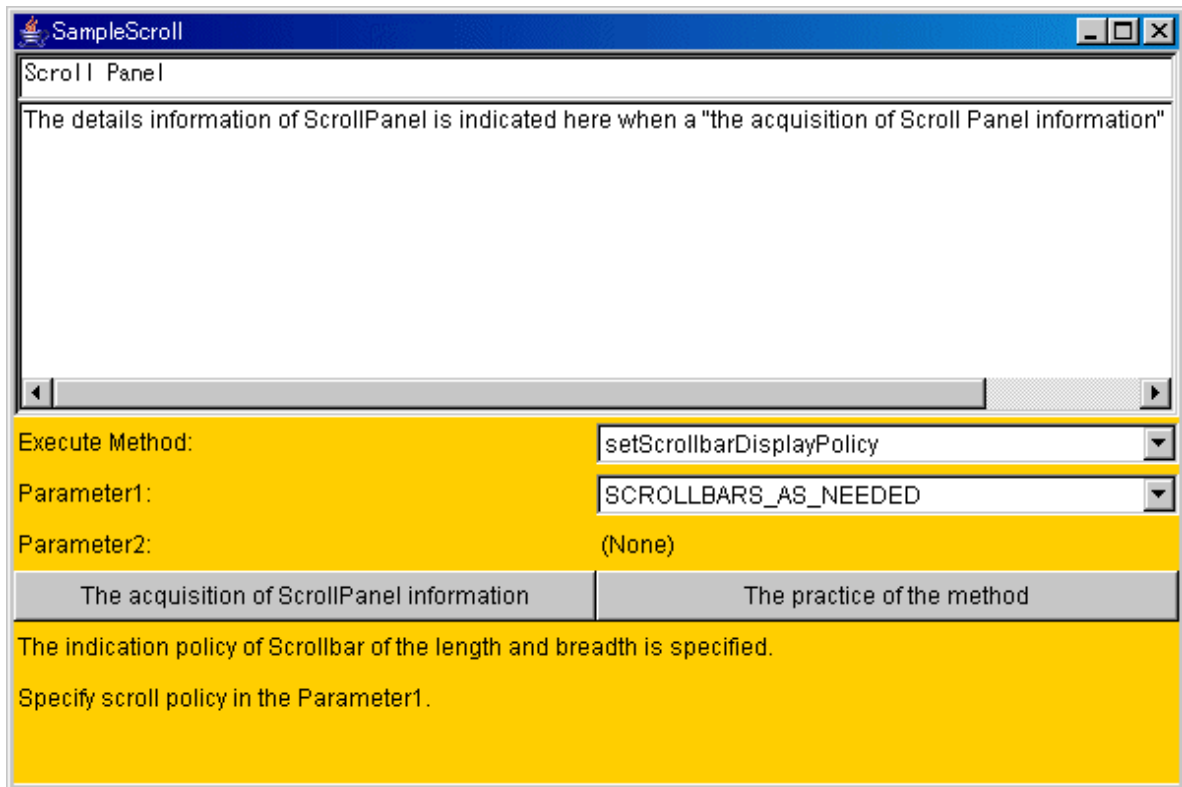
The Scroll Panel sample calls some methods for the JFCScrollablePanel-class Scroll Panel and monitors the operation of the Scroll Panel.

The Scroll Panel sample uses the following classes:

- JFAlignment

- JFCScrollablePanel

Figure 2.12 Scroll panel sample



Operation method

1. Select the method of the Scroll Panel from "Execute Method:".
2. An explanation of the function and parameter of the selected method is displayed at the bottom. Enter an appropriate value in "Parameter 1:" and "Parameter 2:" in accordance with the explanation.
3. Pressing the [The practice of the method] button reflects the execution result of the method in the upper-half of the scroll panel.
4. Pressing the [The acquisition of ScrollPanel information] button displays the properties of the scroll panel in the scroll panel.
5. Henceforth, the Scroll Panel operation status can be monitored by repeating steps 1 to 4.

Source codes

- Scroll panel sample:"Sample Storage Destinations"/JFCScrollablePanel/SampleScroll.java

2.3.5 Scroll Panel Sample(Light Weight Version)

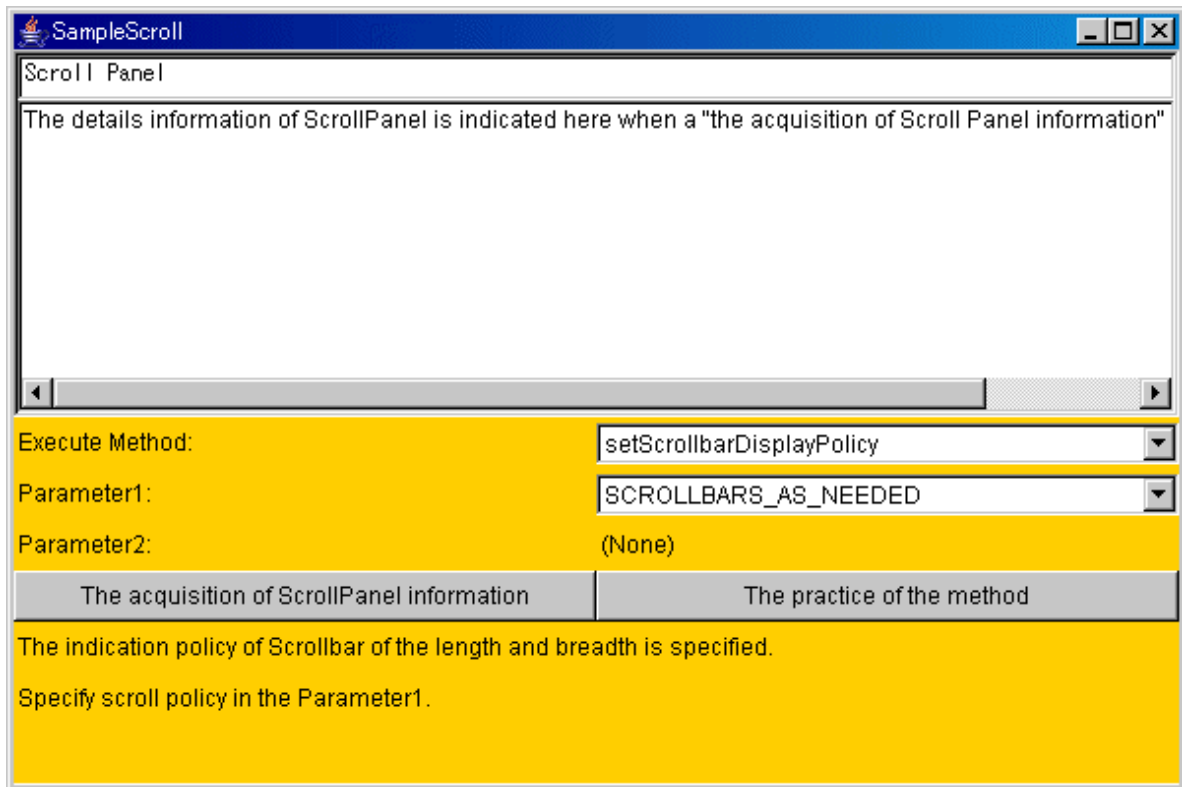
The Light Weight Scroll Panel sample calls some methods for the JFLScrollablePanel-class Scroll Panel and monitors the operation of the Scroll Panel.

The Scroll Panel sample uses the following classes:

- JFAlignment

- JFLScrollablePanel

Figure 2.13 Scroll panel sample



Operation method

1. Select the method of the Scroll Panel from "Execute Method:".
2. An explanation of the function and parameter of the selected method is displayed at the bottom. Enter an appropriate value in "Parameter 1:" and "Parameter 2:" in accordance with the explanation.
3. Pressing the [The practice of the method] button reflects the execution result of the method in the upper-half of the scroll panel.
4. Pressing the [The acquisition of ScrollPanel information] button displays the properties of the scroll panel in the scroll panel.
5. Henceforth, the Scroll Panel operation status can be monitored by repeating steps 1 to 4.

Source codes

- Scroll panel sample:"Sample Storage Destinations"/JFCScrollablePanel/SampleScroll.java

2.3.6 Window Control Facility Sample

The Window Control Facility Sample transits the Transition Window using Window Control Facility.

This sample can be referenced as a simple sample of the program handling mass windows.

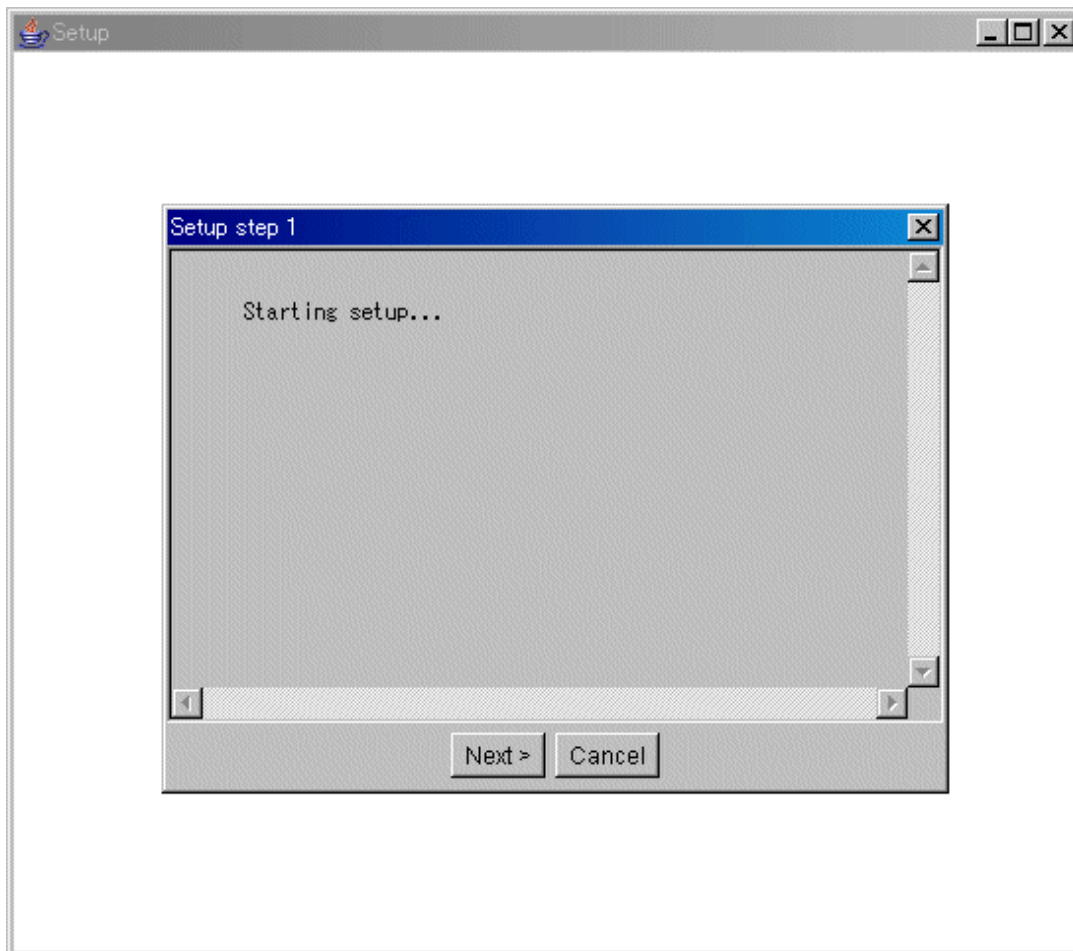
The Window Control Facility Sample uses the following classes.

- JFCWindowLoaderManager
- JFCFrame
- JFCDialog
- JFCWindowAdapter

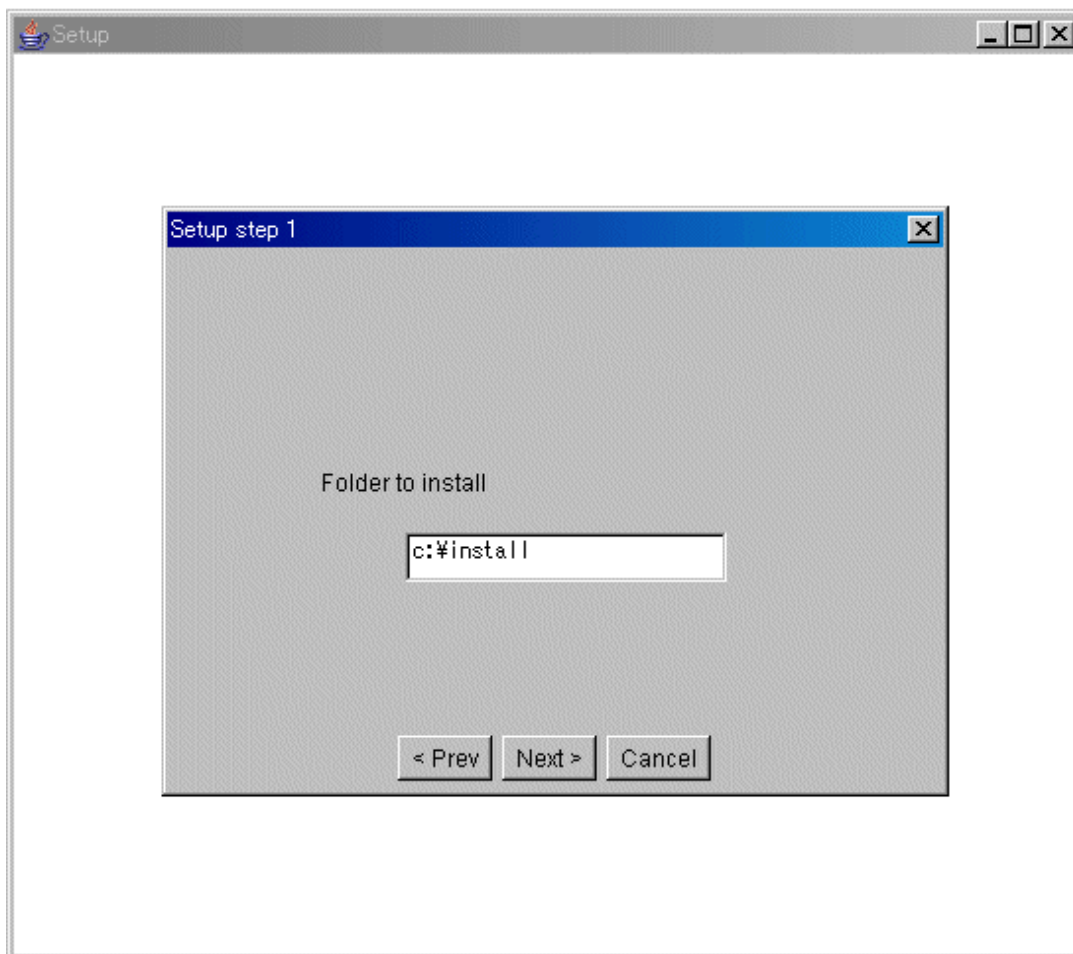
Operation method

Proceed as follows to operate this sample.

1. Start the application.

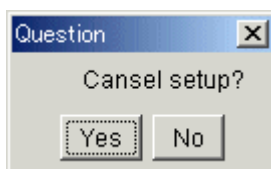


2. Pressing the [Next>] button transits to the next window and displays the window for entering the folder to install.

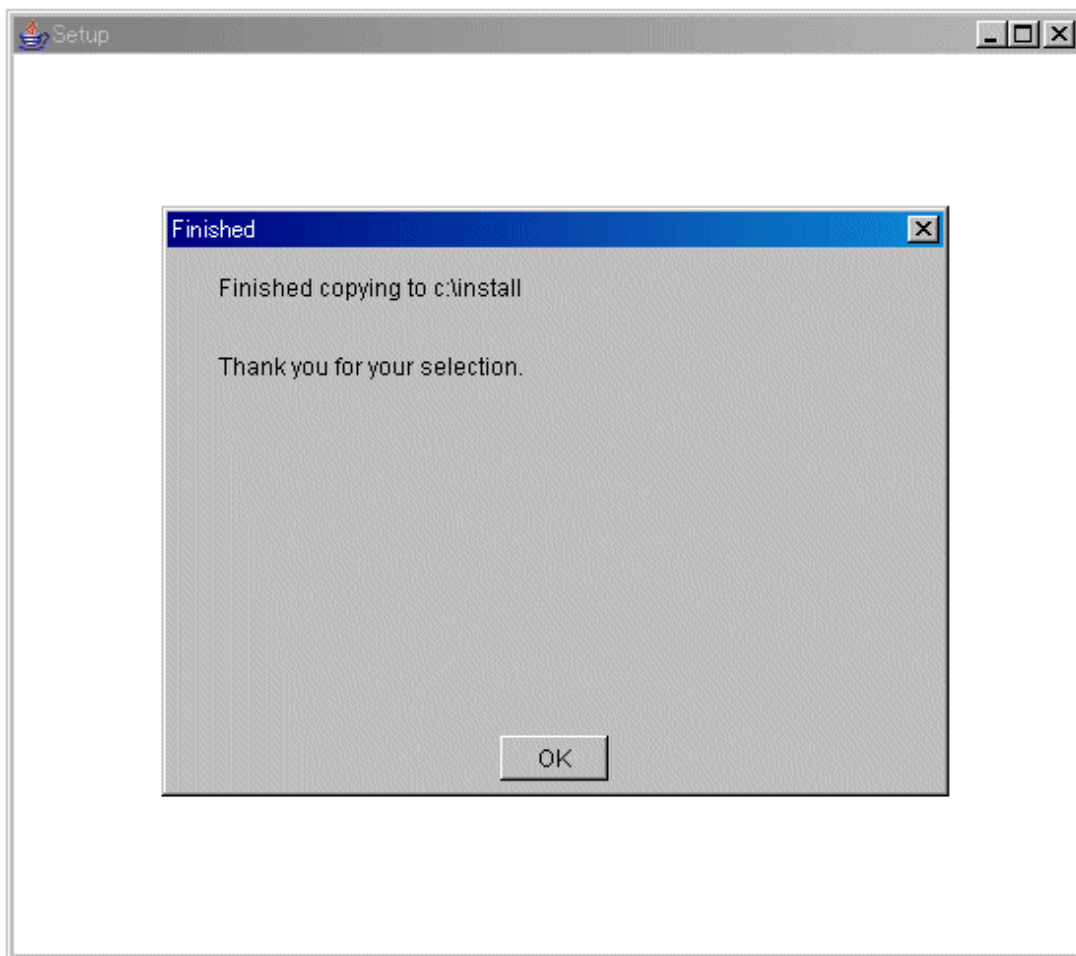


This sample does not write to the disk, however any string specified for the folder.

3. Pressing the [Cancel] button displays the window for confirm whether finish the setup.



4. Pressing the [Next>] button transits to the next window and displays the window for show finish the copying.



Source codes

- Window Control Facility sample:"Sample Storage Destinations"/JFCWindow/Install.java
- Subclass of WindowLoaderManager:"Sample Storage Destinations"/JFCWindow/WindowManager.java
- Transition Frame sample:"Sample Storage Destinations"/JFCWindow/Frame1.java
- Transition Dialig sample:"Sample Storage Destinations"/JFCWindow/Dialog1.java
- Transition Dialig sample:"Sample Storage Destinations"/JFCWindow/Dialog2.java
- Transition Dialig sample:"Sample Storage Destinations"/JFCWindow/Dialog3.java
- Transition Dialig sample:"Sample Storage Destinations"/JFCWindow/QuestDialog.java

2.3.7 Object Control Facility Sample

The Object Control Facility Sample manages objects using Object Control Facility.

This sample can be referenced as a simple sample of the program handling mass objects.

The Object Control Facility Sample uses the following classes.

- JFCObjectLoaderManager
- JFCObjectLoader

Operation method

Proceed as follows to operate this sample.

1. Start the application.

Figure 2.14 Screen after started



2. Pressing the [Add Label] displays the label on the screen.
At the same time, [Label data] is set as the label's data internally.

Figure 2.15 Screen after added the label

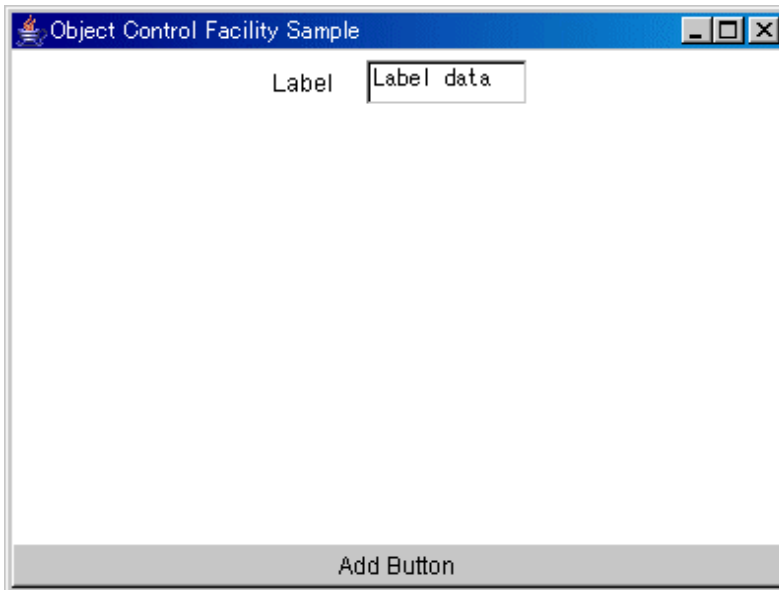


3. Pressing the [Add TextField] displays the text field on the screen.

The text field displays [Label data] that is the label's data.

Also, at the same time, [TextField data] is set as the text field's data internally.

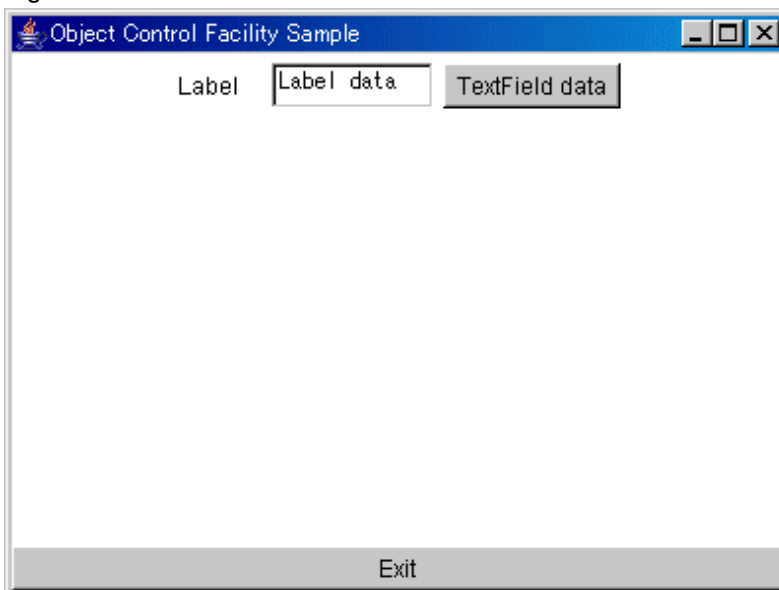
Figure 2.16 Screen after added the text field



4. Pressing [Add Button] displays the button on the screen.

The button displays [TextField data] that is the text field's data.

Figure 2.17 Screen after added the button



5. Pressing [Exit] button to finish the sample.

Source codes

- Object Control Facility sample:"Sample Storage Destinations"/JFCObject/Sample.java
- Subclass of JFCObjectLoaderManager:"Sample Storage Destinations"/JFCObject/ObjectManager.java