Mongol Swing 技术设计书

# 前言

蒙古文的书写特点与其他文种不同，是左竖式书写的文字。蒙古文的书写，每个行是从上到下竖写，换行是从左侧向右侧前进。这个竖写方式的特殊性，当前的很多计算机软件都不支持这一书写方式。在Java应用程序软件包Swing是众多领域广泛使用的应用程序开发框架。利用Java Swing开发的应用程序具有跨平台运行的能力，为此很多通用性软件都选择Java的Swing开发框架开发。如：文本编辑器，工程管理，ERP等。

OpenType字体技术和AAT字体技术的出现，能够为国际标准蒙古文的实现提供了根本性的方法。但是在Java的Swing环境中还不能支持这些字体技术。

为此我们开发了达英国际标准编码蒙古文Java Swing应用程序软件开发包。该软件包中的Swing控件不仅完全支持了国际标准编码蒙古文的名义字符到显现字符的转换显示与编辑功能，而且也能够用竖写显示和编辑国际标准编码蒙古文的能力。

该编辑器把把Java的Swing控件一一改造为支持国际标准蒙古文的名义字符到显现字符的转换以及竖向显示功能。该软件包中支持国际标准编码蒙古文的竖写显示与编辑的控件有：

1. MLabel– 继承和扩展Swing的控件JLabel。
2. MTextField – 继承和扩展Swing的控件JTextField。
3. MButton– 继承和扩展Swing的控件JButton。
4. MCheckBox– 继承和扩展Swing的控件JCheckBox。
5. MComboBox – 继承和扩展Swing的控件JComboBox。
6. MList– 继承和扩展Swing的控件JList。
7. MToolTip – 继承和扩展Swing的控件JToolTip。
8. MTextArea–继承和扩展Swing的控件JTextArea。
9. MTextPane – 继承和扩展Swing的控件JTextPane。
10. MEditorPane – 继承和扩展Swing的控件JEditorPane。

本软件开发包为利用Java Swing开发跨平台国际标准编码蒙古文应用程序的开发人员提供了便利的开发工具和环境，为蒙古文信息化的推进提供了有力的基础性工具环境。

# 设计思路

为了利用Java开发国际标准蒙古文的应用程序，首先把Java的Swing控件改造为支持国际标准蒙古文的名义字符到显现字符的转换功能。其次，继承和扩展Swing的控件（JTextArea、JTextField、JTextPane、JEditorPane、JLabel、JList、JButton、JCheckBox、JComboBox、JToolTip等）并改造成（MTextArea、MTextField、MTextPane、MEditorPane、MLabel、MList、MButton、MCheckBox、MComboBox、MToolTip）让它们支持国际标准编码蒙古文的左竖式竖写格式。

# Swing控件的改造设计（MTextArea为例）

* 1. MTextArea要继承JTextArea并把JTextArea扩展成能够支持国际标准蒙古文竖写编辑功能的控件。
  2. MTextArea要实装MRotation Interface。

MRotation是控件横写，竖写的模式设定Interface。它具有下列选择值。

RotateHint – (*ROTATE\_DEFAULT，ROTATE\_NONE，ROTATE\_LEFTTORIGHT，ROTATE\_RIGHTTOLEFT)*

RotateDirection – (*ROTATE\_HORIZANTAL, ROTATE\_VERTICAL)*

蒙古文的左竖式竖写需要

RotateHint = *ROTATE\_LEFTTORIGHT*

RotateDirection = *ROTATE\_VERTICAL*

* 1. MTextArea的竖写实现是用MTeatAreaUI来完成的。
  2. MTeatAreaUI类继承BasicTextAreaUI并实装MRotation Interface。

MTeatAreaUI在蒙古文的左竖式竖写需要

RotateHint = ROTATE\_LEFTTORIGHT

RotateDirection = ROTATE\_VERTICAL

的设定下完成和实现显示和光标定位的功能。

* 1. 因为Java还没有支持OpenType或AAT字体，所以我们自己编写了国际标准蒙古文名义字符和显现字符的双向转换器MongolianConverter。

# Mongol Swing控件的使用方法

* 1. MLabel的使用方法

MLabel 控件是继承Java Swing JLabel的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠠᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ ABCD 中文 日本語" ;

MLabel lblTxt = **new** MLabel( context );

lblTxt.setToolTipText( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.***ROTATE\_HORIZANTAL***);

lblTxt.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

lblTxt.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 32));

如上所示:

* + 1. 第一步要构建MLabel Object

MLabel lblTxt = **new** MLabel( context );

* + 1. 如果需要的话，设定标签的提示文本

lblTxt.setToolTipText( context );

* + 1. 指定标签的文本显示方向

lblTxt.setRotateDirection(MSwingRotateUtilities.***ROTATE\_HORIZANTAL***);

* + 1. 指定标签的文本的进行方向

lblTxt.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

* + 1. 指定标签的显示字体

lblTxt.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 32));

下面的程序是，MLabel的各种方向显示的完整示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Font;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

import com.mongol.encode.MongolianConverter;

import com.mongol.swing.MLabel;

import com.mongol.swing.MSwingRotateUtilities;

public class MLabelDemo extends JFrame {

/\*\* serialVersionUID \*/

private static final long serialVersionUID = 1L;

/\*\*

\* @param args

\*/

public static void main(String[] args) {

new MLabelDemo();

}

MLabelDemo() {

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠠᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ ABCD 中文 日本語" ;

JFrame frame = new JFrame("Mongolian Encoding example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

MLabel lblTxt = new MLabel( context );

lblTxt.setToolTipText( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_HORIZANTAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt.setFont(new Font("Mongolian White Pua", Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.NORTH);

lblTxt = new MLabel( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt.setFont(new Font("Mongolian White Pua", Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.WEST);

context = "中文正确吗？日本語正しいですか。" ;

lblTxt = new MLabel( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_NONE);

lblTxt.setFont(new Font("MS明朝", Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.CENTER);

context = "This is English Sentence." ;

lblTxt = new MLabel( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_DEFAULT);

lblTxt.setFont(new Font(null, Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.EAST);

context = "中文正确吗？日本語正しいですか。" ;

lblTxt = new MLabel( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_HORIZANTAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_RIGHTTOLEFT);

lblTxt.setFont(new Font("MS明朝", Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.SOUTH);

frame.setSize(800, 600);

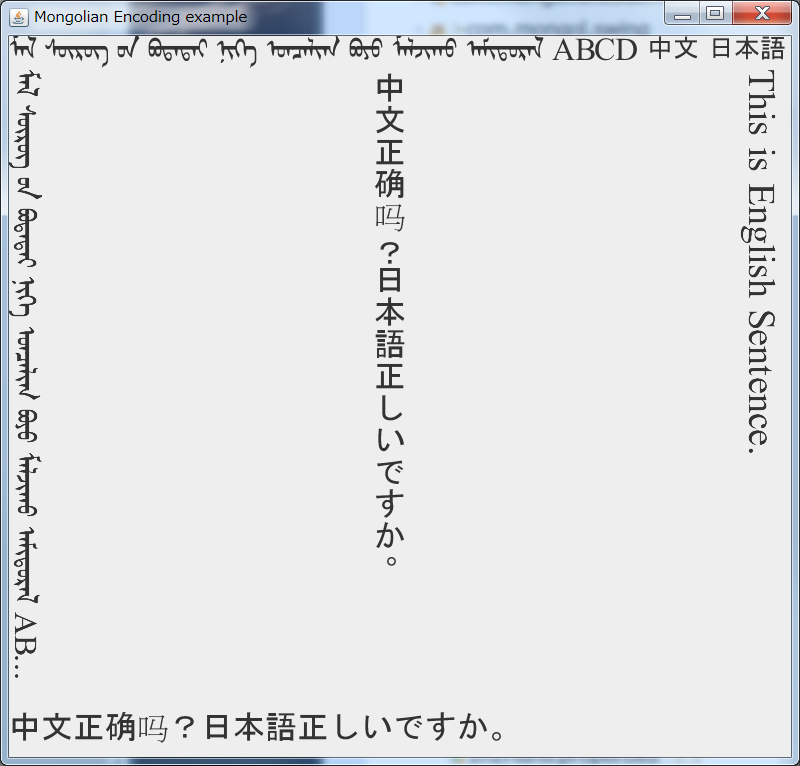
frame.setLocationRelativeTo(null);

frame.setVisible(true);

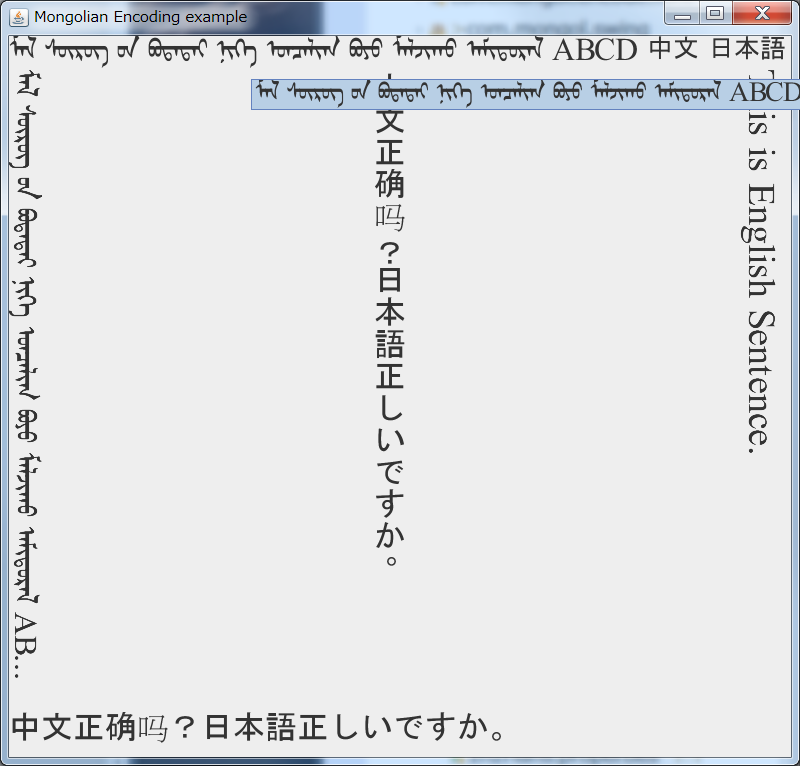
}

}

此程序的运行结果如下图所示。



鼠标移动到标签上停留一会儿，如果标签有提示（ToolTips）设定就显示提示文本。



* 1. MTextField 的使用方法。

MTextField控件是继承Java Swing JTextField的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

String context = "ᠡᠷᠡᠨ ᠵᠠᠭᠣᠨ ᠪ ᠨᠠᠷᠠᠨ ᠳᠧᠩᠯᠦ ᠳᠠᠳᠭᠠᠯ ᠳᠠᠰᠭᠠᠯ  日本語です。" ;

MTextField txtField2 = **new** MTextField( context, 40 );

txtField2.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

txtField2.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

txtField2.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

如上所示:

1. 第一步要构建文本输入框MTextField Object

MTextField txtField2 = **new** MTextField( context, 40 );

1. 指定文本输入框的文本显示方向

txtField2.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

1. 指定文本输入框的文本的进行方向

txtField2.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

1. 指定文本输入框的边框

txtField2.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

下面的程序是，MTextField使用的完整示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Font;

import java.awt.GridBagConstraints;

import java.awt.GridBagLayout;

import java.awt.Insets;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EtchedBorder;

import com.mongol.encode.MongolianFontUtil;

import com.mongol.swing.MLabel;

import com.mongol.swing.MSwingRotateUtilities;

import com.mongol.swing.MTextField;

public class MTextFieldDemo extends JFrame {

/\*\* serialVersionUID \*/

private static final long serialVersionUID = 1L;

/\*\*

\* @param args

\*/

public static void main(String[] args) {

MTextFieldDemo frame = new MTextFieldDemo();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// frame.setBounds(10, 10, 300, 200);

frame.setTitle("Mongolian Swing Component example");

frame.setSize(800, 600);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

public MTextFieldDemo() {

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

GridBagLayout layout = new GridBagLayout();

JPanel p = new JPanel();

p.setLayout(layout);

p.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(5, 5, 5, 5);

int line = 0;

MLabel lblTxt7 = new MLabel( "Mongolian" );

lblTxt7.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt7.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt7.setFont(new Font("MS明朝", Font.PLAIN, 16));

gbc.gridx = line;

gbc.gridy = 0;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(lblTxt7, gbc);

p.add(lblTxt7);

context = "ᠡᠷᠡᠨ ᠵᠠᠭᠣᠨ ᠪ ᠨᠠᠷᠠᠨ ᠳᠧᠩᠯᠦ ᠳᠠᠳᠭᠠᠯ ᠳᠠᠰᠭᠠᠯ  日本語です。" ;

MTextField txtField2 = new MTextField( context, 40 );

txtField2.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

txtField2.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

// Font font = MongolianFontUtil.getFont("Dialog");

// Font mfont = font.deriveFont(Font.PLAIN, 24);

// txtField2.setFont(mfont);

txtField2.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

gbc.gridx = line;

gbc.gridy = 1;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(txtField2, gbc);

p.add(txtField2);

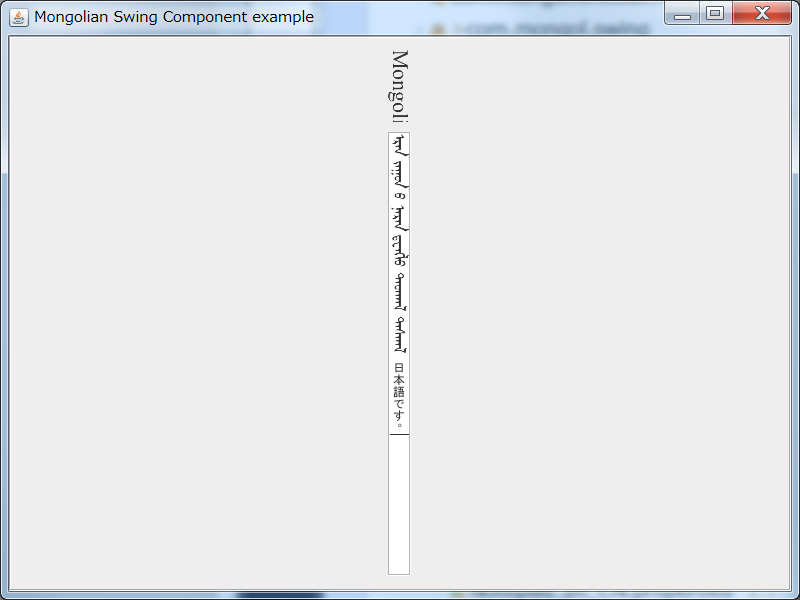
getContentPane().add(p, BorderLayout.CENTER);

p.setSize(780, 580);

}

}

此程序的运行结果如下图所示。



* 1. MButton的使用方法

MButton控件是继承Java Swing JButton的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

String context = "ᠡᠷᠡᠨ ᠵᠠᠭᠣᠨ ᠪ ᠨᠠᠷᠠᠨ ᠳᠧᠩᠯᠦ ᠳᠠᠳᠭᠠᠯ ᠳᠠᠰᠭᠠᠯ  " ;

MButton button2 = **new** MButton( context );

button2.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

button2.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

button2.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

button2.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

size = **new** Dimension(60, 300);

button2.setPreferredSize(size);

如上所示:

1. 第一步要构建按钮MButton Object

MButton button2 = **new** MButton( context );

1. 指定按钮的文本显示方向

button2.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

1. 指定按钮的文本的进行方向

button2.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

1. 指定按钮的字体字号

button2.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

1. 指定按钮的边框

button2.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

1. 指定按钮的大小

size = **new** Dimension(60, 300);

button2.setPreferredSize(size);

下面的程序是，MButton使用的完整示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.GridBagConstraints;

import java.awt.GridBagLayout;

import java.awt.Insets;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EtchedBorder;

import com.mongol.swing.MButton;

import com.mongol.swing.MLabel;

import com.mongol.swing.MSwingRotateUtilities;

public class MButtonDemo extends JFrame {

/\*\* serialVersionUID \*/

private static final long serialVersionUID = 1L;

/\*\*

\* @param args

\*/

public static void main(String[] args) {

MButtonDemo frame = new MButtonDemo();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// frame.setBounds(10, 10, 300, 200);

frame.setTitle("Mongolian Swing Component example");

frame.setSize(800, 600);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

public MButtonDemo() {

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

GridBagLayout layout = new GridBagLayout();

JPanel p = new JPanel();

p.setLayout(layout);

p.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(5, 5, 5, 5);

int line = 0;

MLabel lblTxt7 = new MLabel( "Mongolian" );

lblTxt7.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt7.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt7.setFont(new Font("MS明朝", Font.PLAIN, 16));

gbc.gridx = line;

gbc.gridy = 0;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(lblTxt7, gbc);

p.add(lblTxt7);

context = "ᠡᠷᠡᠨ ᠵᠠᠭᠣᠨ ᠪ ᠨᠠᠷᠠᠨ ᠳᠧᠩᠯᠦ ᠳᠠᠳᠭᠠᠯ ᠳᠠᠰᠭᠠᠯ  " ;

MButton button1 = new MButton( context );

// button2.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

// button2.setRotateHint(MSwingRotateUtilities.ROTATE\_RIGHTTOLEFT);

button1.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

button1.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

Dimension size = new Dimension(300, 100);

button1.setPreferredSize(size);

gbc.gridx = line;

gbc.gridy = 1;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(button1, gbc);

p.add(button1);

MButton button2 = new MButton( context );

button2.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

button2.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

button2.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

button2.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

size = new Dimension(60, 300);

button2.setPreferredSize(size);

gbc.gridx = line;

gbc.gridy = 2;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(button2, gbc);

p.add(button2);

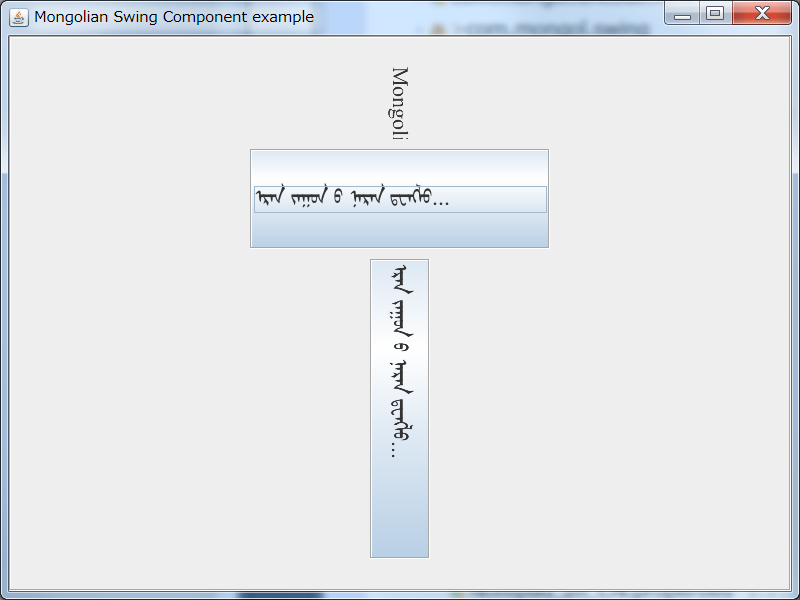
getContentPane().add(p, BorderLayout.CENTER);

p.setSize(780, 580);

}

}

此程序的运行结果如下图所示。



* 1. MCheckBox的使用方法

MCheckBox控件是继承Java Swing JCheckBox的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

MCheckBox jcbChin = **new** MCheckBox("ᠮᠠᠯ ᠰᠦᠷᠦᠭ᠌");

jcbChin.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 32));

jcbChin.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

jcbChin.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

jcbChin.setMnemonic(KeyEvent.***VK\_C***); //Alt+C Checks/Unchecks the check Box

jcbChin.setSelected(**true**);

如上所示:

1. 第一步要构建选择框MCheckBox Object

MCheckBox jcbChin = **new** MCheckBox("ᠮᠠᠯ ᠰᠦᠷᠦᠭ᠌");

1. 指定选择框的字体字号

jcbChin.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 32));

1. 指定选择框的文本显示方向

jcbChin.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

1. 指定选择框的文本的进行方向

jcbChin.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

1. 指定选择框的快捷键

jcbChin.setMnemonic(KeyEvent.***VK\_C***); //Alt+C Checks/Unchecks the check Box

1. 指定选择框的选择/非选择状态

jcbChin.setSelected(**true**);

下面的程序是，MCheckBox使用的完整示例程序。

package com.mongol.demo;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import com.mongol.swing.MCheckBox;

import com.mongol.swing.MSwingRotateUtilities;

public class MCheckBoxDemo extends JPanel {

//Four accessory choices provide for 16 different combinations

MCheckBox jcbChin;

MCheckBox jcbGlasses;

MCheckBox jcbHair;

MCheckBox jcbTeeth;

/\* The image for each combination is contained in a

separate image file whose name indicates the accessories.

The filenames are "geek-XXXX.gif" where XXXX can be one

\* of the following 16 choices.

\*/

StringBuffer choices;

JLabel jlbPicture;

CheckBoxListener myListener = null;

public MCheckBoxDemo() {

// Add an item listener for each of the check boxes.

// This is the listener class which contains business logic

myListener = new CheckBoxListener();

// Create check boxes with default selection true

jcbChin = new MCheckBox("ᠮᠠᠯ ᠰᠦᠷᠦᠭ᠌");

jcbChin.setFont(new Font("Mongolian White Pua", Font.PLAIN, 32));

jcbChin.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

jcbChin.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

jcbChin.setMnemonic(KeyEvent.VK\_C); //Alt+C Checks/Unchecks the check Box

jcbChin.setSelected(true);

jcbChin.addItemListener(myListener);

jcbGlasses = new MCheckBox("ᠪᠣᠳᠠᠲᠠᠢ");

jcbGlasses.setFont(new Font("Mongolian White Pua", Font.PLAIN, 32));

jcbGlasses.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

jcbGlasses.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

jcbGlasses.setMnemonic(KeyEvent.VK\_G); //Alt+G Checks/Unchecks the check Box

jcbGlasses.setSelected(true);

jcbGlasses.addItemListener(myListener);

jcbHair = new MCheckBox("ᠣᠨᠴᠠᠯᠢᠭ");

jcbHair.setFont(new Font("Mongolian White Pua", Font.PLAIN, 32));

jcbHair.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

jcbHair.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

jcbHair.setMnemonic(KeyEvent.VK\_H); //Alt+H Checks/Unchecks the check Box

jcbHair.setSelected(true);

jcbHair.addItemListener(myListener);

jcbTeeth = new MCheckBox("ᠠᠮᠢᠳᠤᠷᠠᠯ");

jcbTeeth.setFont(new Font("Mongolian White Pua", Font.PLAIN, 32));

jcbTeeth.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

jcbTeeth.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

jcbTeeth.setMnemonic(KeyEvent.VK\_T); //Alt+T Checks/Unchecks the check Box

jcbTeeth.setSelected(true);

jcbTeeth.addItemListener(myListener);

// Indicates what's on the geek.

choices = new StringBuffer("cght");//Default Image has all the parts.

// Set up the picture label

jlbPicture = new JLabel(new ImageIcon("geek-" +

choices.toString().trim() + ".gif"));

jlbPicture.setToolTipText(choices.toString().trim());

// Put the check boxes in a column in a panel

JPanel jplCheckBox = new JPanel();

jplCheckBox.setLayout(new GridLayout(1, 0)); //0 rows, 1 Column

jplCheckBox.add(jcbChin);

jplCheckBox.add(jcbGlasses);

jplCheckBox.add(jcbHair);

jplCheckBox.add(jcbTeeth);

setLayout(new BorderLayout());

add(jplCheckBox, BorderLayout.WEST);

add(jlbPicture, BorderLayout.CENTER);

setBorder(BorderFactory.createEmptyBorder(20,20,20,20));

}

//Listens to the check boxes events

class CheckBoxListener implements ItemListener {

public void itemStateChanged(ItemEvent e) {

int index = 0;

char c = '-';

Object source = e.getSource();

if (source == jcbChin) {

index = 0;

c = 'c';

} else if (source == jcbGlasses) {

index = 1;

c = 'g';

} else if (source == jcbHair) {

index = 2;

c = 'h';

} else if (source == jcbTeeth) {

index = 3;

c = 't';

}

if (e.getStateChange() == ItemEvent.DESELECTED)

c = '-';

choices.setCharAt(index, c);

jlbPicture.setIcon(new ImageIcon("geek-"

+ choices.toString().trim() + ".gif"));

jlbPicture.setToolTipText(choices.toString());

}

}

public static void main(String s[]) {

JFrame frame = new JFrame("JCheckBox Usage Demo");

frame.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e) {

System.exit(0);

}

});

frame.setContentPane(new MCheckBoxDemo());

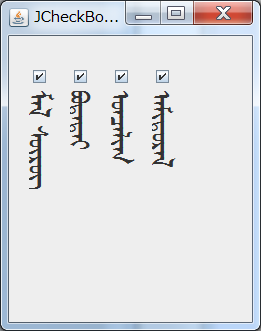
frame.pack();

frame.setVisible(true);

}

}

此程序的运行结果如下图所示。



* 1. MComboBox的使用方法

MComboBox控件是继承Java Swing JComboBox的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

MComboBox comboBox = **new** MComboBox();

comboBox.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

comboBox.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

comboBox.setRotateHint(MSwingRotateUtilities.***ROTATE\_RIGHTTOLEFT***);

comboBox.addItem("ItemA ᠨᠢᠭᠡᠳᠦᠭᠡᠷ");

comboBox.addItem("ItemB ᠬᠣᠶᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemC ᠭᠤᠷᠪᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemD ᠳᠥᠷᠪᠡᠳᠦᠭᠡᠷ");

comboBox.addItem("ItemE ᠲᠠᠪᠤᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemF ᠵᠢᠷᠭᠤᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemG ᠳᠣᠯᠤᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemH ᠨᠠᠢ᠍ᠮᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemI ᠶᠢᠰᠦᠳᠦᠭᠡᠷ");

comboBox.addItem("ItemJ ᠠᠷᠪᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemK ᠠᠷᠪᠠᠨ ᠨᠢᠭᠡᠳᠦᠭᠡᠷ");

comboBox.addItem("ItemL ᠠᠷᠪᠠᠨ ᠬᠣᠶᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemM ᠠᠷᠪᠠᠨ ᠭᠤᠷᠪᠠᠳᠤᠭᠠᠷ");

Dimension size = **new** Dimension(24, 360);

comboBox.setPreferredSize(size);

如上所示:

1. 第一步要构建下拉框MComboBox Object

MComboBox comboBox = **new** MComboBox();

1. 指定下拉框的字体字号

comboBox.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

1. 指定下拉框的文本显示方向

comboBox.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

1. 指定下拉框的文本的进行方向

comboBox.setRotateHint(MSwingRotateUtilities.***ROTATE\_RIGHTTOLEFT***);

1. 指定下拉框的可选项

comboBox.addItem("ItemM ᠭᠤᠷᠪᠠᠳᠤᠭᠠᠷ");

1. 指定下拉框的快尺寸

Dimension size = **new** Dimension(24, 360);

comboBox.setPreferredSize(size);

下面的程序是，MComboBox使用的完整示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.Font;

import javax.swing.\*;

import com.mongol.swing.MComboBox;

import com.mongol.swing.MLabel;

import com.mongol.swing.MSwingRotateUtilities;

public class MComboBoxDemo extends JFrame {

private MComboBox comboBox;

private MLabel lbl;

public MComboBoxDemo() {

comboBox = new MComboBox();

comboBox.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

comboBox.setRotateHint(MSwingRotateUtilities.ROTATE\_RIGHTTOLEFT);

comboBox.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

comboBox.addItem("ItemA ᠨᠢᠭᠡᠳᠦᠭᠡᠷ");

comboBox.addItem("ItemB ᠬᠣᠶᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemC ᠭᠤᠷᠪᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemD ᠳᠥᠷᠪᠡᠳᠦᠭᠡᠷ");

comboBox.addItem("ItemE ᠲᠠᠪᠤᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemF ᠵᠢᠷᠭᠤᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemG ᠳᠣᠯᠤᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemH ᠨᠠᠢ᠍ᠮᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemI ᠶᠢᠰᠦᠳᠦᠭᠡᠷ");

comboBox.addItem("ItemJ ᠠᠷᠪᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemK ᠠᠷᠪᠠᠨ ᠨᠢᠭᠡᠳᠦᠭᠡᠷ");

comboBox.addItem("ItemL ᠠᠷᠪᠠᠨ ᠬᠣᠶᠠᠳᠤᠭᠠᠷ");

comboBox.addItem("ItemM ᠠᠷᠪᠠᠨ ᠭᠤᠷᠪᠠᠳᠤᠭᠠᠷ");

Dimension size = new Dimension(24, 360);

comboBox.setPreferredSize(size);

lbl = new MLabel();

lbl.setText("");

lbl.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

comboBox.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

comboBoxMouseClicked(evt);

}

});

getContentPane().add(comboBox, BorderLayout.NORTH);

getContentPane().add(lbl, BorderLayout.SOUTH);

getContentPane().setLayout(new java.awt.FlowLayout());

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setTitle("ComboBox");

setSize(560, 400);

}

private void comboBoxMouseClicked(java.awt.event.ActionEvent evt) {

lbl.setText(String.valueOf(comboBox.getSelectedItem()) + " was selected");

}

public static void main(String[] args) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new MComboBoxDemo().setVisible(true);

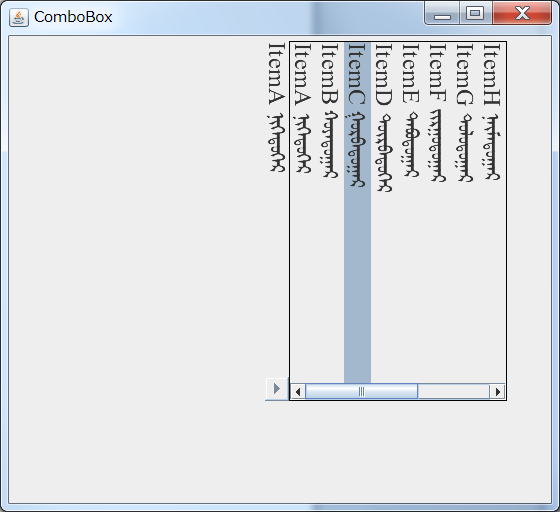
}

});

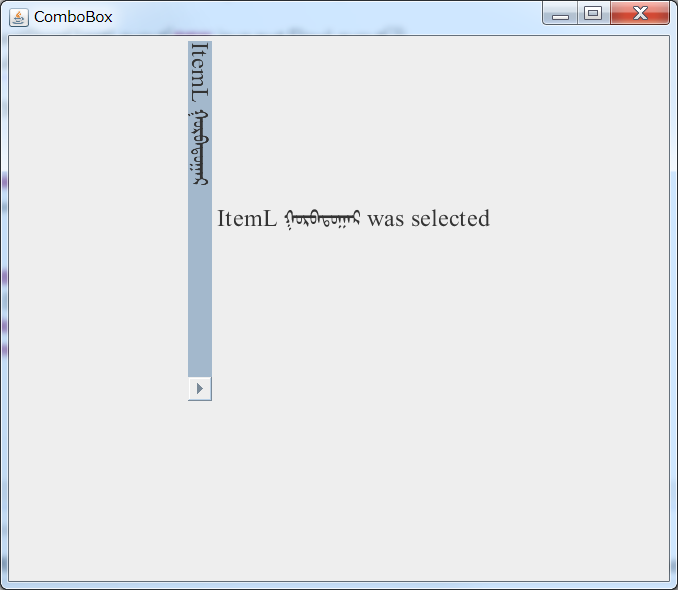
}

}

此程序的运行结果如下图所示。



选择其中的一个选项之后，显示其结果到下拉框旁边的标签控件中。



* 1. MList的使用方法

MList控件是继承Java Swing JList的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

MList list = **new** MList(model);

list.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

list.setRotateHint(MSwingRotateUtilities.***ROTATE\_RIGHTTOLEFT***);

list.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

Color fg = **new** Color( 255, 128, 128);

list.setSelectionForeground(fg);

**for** (**int** i = 0; i < 15; i++)

model.addElement("ᠮᠤᠩᠭᠤᠯ Element 番号：" + i);

如上所示:

1. 第一步要构建列表框MList Object

DefaultListModel model = **new** DefaultListModel();

MList list = **new** MList(model);

1. 指定列表框的文本显示方向

list.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

1. 指定列表框的文本的进行方向

list.setRotateHint(MSwingRotateUtilities.***ROTATE\_RIGHTTOLEFT***);

1. 指定列表框的字体字号

list.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

1. 指定列表框的选择颜色

Color fg = **new** Color( 255, 128, 128);

list.setSelectionForeground(fg);

1. 指定列表框的选项列表

model.addElement("ᠮᠤᠩᠭᠤᠯ Element 番号：" + i);

下面的程序是，MList使用的完整示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.DefaultListModel;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import com.mongol.swing.MList;

import com.mongol.swing.MSwingRotateUtilities;

public class MListDemo extends JPanel {

MList list;

DefaultListModel model;

int counter = 15;

public MListDemo() {

setLayout(new BorderLayout());

model = new DefaultListModel();

list = new MList(model);

list.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

list.setRotateHint(MSwingRotateUtilities.ROTATE\_RIGHTTOLEFT);

list.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

Color fg = new Color( 255, 128, 128);

list.setSelectionForeground(fg);

JScrollPane pane = new JScrollPane(list);

JButton addButton = new JButton("Add Element");

JButton removeButton = new JButton("Remove Element");

for (int i = 0; i < 15; i++)

model.addElement("ᠮᠤᠩᠭᠤᠯ Element 番号：" + i);

addButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

model.addElement("ᠮᠤᠩᠭᠤᠯ Element 番号：" + counter);

counter++;

}

});

removeButton.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if (model.getSize() > 0)

model.removeElementAt(0);

}

});

add(pane, BorderLayout.NORTH);

add(addButton, BorderLayout.WEST);

add(removeButton, BorderLayout.EAST);

}

public static void main(String s[]) {

JFrame frame = new JFrame("List Model Example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setContentPane(new MListDemo());

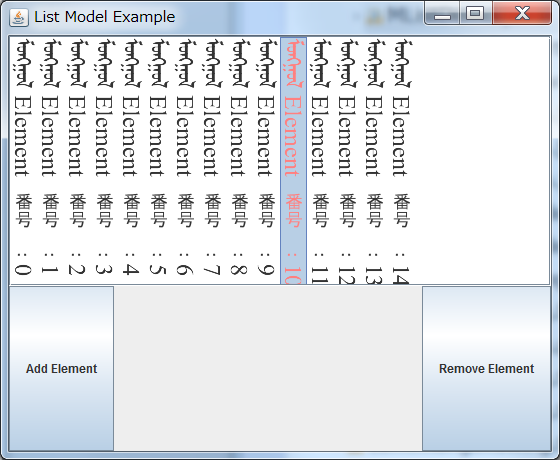
frame.setSize(560, 460);

frame.setVisible(true);

}

}

此程序的运行结果如下图所示。



* 1. MToolTip 的使用方法

MToolTip控件是继承Java Swing JToolTip的支持国际标准编码蒙古文的标签控件。MtoolTip还没有实装Mrotation Interface。所以只能横向显示提示文本。

其使用方法主要为MLabel的提示文本，在Mlabel中Override下边的`createToolTip() 函数。

**public** JToolTip createToolTip() {

MToolTip tip = **new** MToolTip();

Font font = **this**.getFont();

font = font.deriveFont(Font.***PLAIN***, font.getSize()-4 );

tip.setFont(font);

tip.setComponent(**this**);

**return** tip;

}

如上所示:

1. 第一步要构建列表框MToolTip Object

MToolTip tip = **new** MToolTip();

1. 指定列表框的字体字号

tip.setFont(**new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

下面的程序是，MLabel中MToolTip使用的示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Font;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

import com.mongol.encode.MongolianConverter;

import com.mongol.swing.MLabel;

import com.mongol.swing.MSwingRotateUtilities;

public class MLabelDemo extends JFrame {

/\*\* serialVersionUID \*/

private static final long serialVersionUID = 1L;

/\*\*

\* @param args

\*/

public static void main(String[] args) {

new MLabelDemo();

}

MLabelDemo() {

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠠᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ ABCD 中文 日本語" ;

JFrame frame = new JFrame("Mongolian Encoding example");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

MLabel lblTxt = new MLabel( context );

lblTxt.setToolTipText( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_HORIZANTAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt.setFont(new Font("Mongolian White Pua", Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.NORTH);

lblTxt = new MLabel( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt.setFont(new Font("Mongolian White Pua", Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.WEST);

context = "中文正确吗？日本語正しいですか。" ;

lblTxt = new MLabel( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_NONE);

lblTxt.setFont(new Font("MS明朝", Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.CENTER);

context = "This is English Sentence." ;

lblTxt = new MLabel( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_DEFAULT);

lblTxt.setFont(new Font(null, Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.EAST);

context = "中文正确吗？日本語正しいですか。" ;

lblTxt = new MLabel( context );

lblTxt.setRotateDirection(MSwingRotateUtilities.ROTATE\_HORIZANTAL);

lblTxt.setRotateHint(MSwingRotateUtilities.ROTATE\_RIGHTTOLEFT);

lblTxt.setFont(new Font("MS明朝", Font.PLAIN, 32));

frame.getContentPane().add(lblTxt, BorderLayout.SOUTH);

frame.setSize(800, 600);

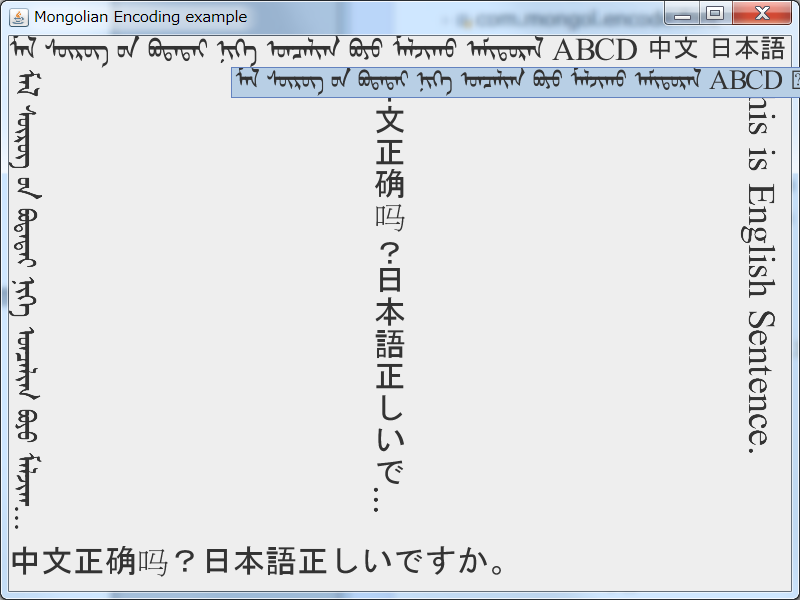
frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

}

此程序的运行结果如下图所示。提示文字是在鼠标移到标签上停留一段时间后出现。



* 1. MTextArea的使用方法

MTextArea控件是继承Java Swing JTextArea的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ \nᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

MTextArea txtArea0 = **new** MTextArea( context, 10, 40 );

txtArea0.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

txtArea0.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

txtArea0.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

txtArea0.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

JScrollPane scrollingArea = **new** JScrollPane(txtArea0);

scrollingArea.getHorizontalScrollBar().setUnitIncrement(26);

如上所示:

1. 第一步要构建文本编辑框MTextArea Object

MTextArea txtArea0 = **new** MTextArea( context, 10, 40 );

1. 指定文本编辑框的文本显示方向

txtArea0.setRotateDirection(MSwingRotateUtilities.***ROTATE\_VERTICAL***);

1. 指定文本编辑框的文本的进行方向

txtArea0.setRotateHint(MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

1. 指定文本编辑框的字体字号

txtArea0.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

1. 指定文本编辑框的边框

txtArea0.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

1. 指定文本编辑框的滚动条

JScrollPane scrollingArea = **new** JScrollPane(txtArea0);

scrollingArea.getHorizontalScrollBar().setUnitIncrement(26);

下面的程序是，MTextArea使用的完整示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.Font;

import java.awt.GridBagConstraints;

import java.awt.GridBagLayout;

import java.awt.Insets;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EtchedBorder;

import com.mongol.swing.MLabel;

import com.mongol.swing.MSwingRotateUtilities;

import com.mongol.swing.MTextArea;

public class MTextAreaDemo extends JFrame {

/\*\* serialVersionUID \*/

private static final long serialVersionUID = 1L;

/\*\*

\* @param args

\*/

public static void main(String[] args) {

MTextAreaDemo frame = new MTextAreaDemo();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// frame.setBounds(10, 10, 300, 200);

frame.setTitle("Mongolian Swing Component example");

frame.setSize(1000, 600);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

public MTextAreaDemo() {

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

GridBagLayout layout = new GridBagLayout();

JPanel p = new JPanel();

p.setLayout(layout);

p.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

GridBagConstraints gbc = new GridBagConstraints();

gbc.insets = new Insets(5, 5, 5, 5);

int line = 0;

MLabel lblTxt0 = new MLabel( "Mongolian" );

lblTxt0.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt0.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt0.setFont(new Font("MS明朝", Font.PLAIN, 16));

gbc.gridx = line;

gbc.gridy = 0;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(lblTxt0, gbc);

p.add(lblTxt0);

context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ \nᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

MTextArea txtArea0 = new MTextArea( context, 10, 40 );

txtArea0.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

txtArea0.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

// txtArea0.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

txtArea0.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

JScrollPane scrollingArea = new JScrollPane(txtArea0);

scrollingArea.getHorizontalScrollBar().setUnitIncrement(26);

gbc.gridx = line;

gbc.gridy = 1;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(scrollingArea, gbc);

p.add(scrollingArea);

// line++;

// MLabel lblTxt1 = new MLabel( "Mongolian" );

// lblTxt1.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

// lblTxt1.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

// lblTxt1.setFont(new Font("MS明朝", Font.PLAIN, 16));

// gbc.gridx = line;

// gbc.gridy = 0;

// gbc.gridwidth = 1;

// gbc.gridheight = 1;

// layout.setConstraints(lblTxt1, gbc);

// p.add(lblTxt1);

//

// context = "ᠮᠠᠯ \nᠰᠦᠷᠦᠭ ᠦᠨ\n ᠪᠣᠳᠠᠲᠠᠢ \nᠨᠢᠭᠡ \nᠣᠨᠴᠠᠯᠢᠭ \nᠪᠣᠶᠣ \nᠮᠣᠯᠵᠢᠬᠤ \nᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

// MTextArea txtArea1 = new MTextArea( context, 5, 20 );

// txtArea1.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

// txtArea1.setRotateHint(MSwingRotateUtilities.ROTATE\_RIGHTTOLEFT);

// txtArea1.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

// txtArea1.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

// JScrollPane scrollingArea1 = new JScrollPane(txtArea1);

// scrollingArea1.getHorizontalScrollBar().setUnitIncrement(26);

//

// gbc.gridx = line;

// gbc.gridy = 1;

// gbc.gridwidth = 1;

// gbc.gridheight = 1;

// layout.setConstraints(scrollingArea1, gbc);

// p.add(scrollingArea1);

line++;

MLabel lblTxt8 = new MLabel( "Mongolian" );

lblTxt8.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt8.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt8.setFont(new Font("MS明朝", Font.PLAIN, 16));

gbc.gridx = line;

gbc.gridy = 0;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(lblTxt8, gbc);

p.add(lblTxt8);

context = "ᠲᠤᠰ ᠦᠭᠦᠯᠡᠯ ᠦᠨ ᠤᠳᠤᠷᠢᠳᠭᠠᠯ ᠤᠨ ᠬᠡᠰᠡᠭ ᠲᠥ ᠰᠡᠳᠦᠪ ᠰᠤᠩᠭ᠋ᠤᠭᠰᠠᠨ ᠰᠢᠯᠲᠠᠭᠠᠨ᠂ " +

"ᠠᠴᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠯ᠂ ᠵᠣᠷᠢᠯᠭ᠎ᠠ᠂ ᠠᠷᠭ᠎ᠠ ᠪᠠ ᠰᠣᠳᠤᠯᠤᠯ ᠤᠨ ᠲᠣᠢ᠌ᠮᠣ ᠶᠢ ᠲᠠᠨᠢᠯᠴᠠᠭᠤᠯᠪᠠ᠃ " +

"ᠨᠢᠭᠡᠳᠦᠭᠡᠷ ᠪᠦᠯᠦᠭ ᠲᠥ᠂ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠡᠭᠦᠰᠦᠯ ᠦᠨ ᠱᠠᠲᠤ ᠦᠶ᠎ᠡ ᠨᠡᠯᠢᠶᠡᠳ ᠡᠷᠲᠡ ᠦᠶ᠎ᠡ ᠲᠠᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠬᠤ ᠪᠣᠯᠪᠠᠴᠣ ᠵᠢᠩᠭᠢᠨᠢ ᠲᠦᠯᠦᠪᠰᠢᠷᠡᠵᠦ ᠬᠦᠭ᠍ᠵᠢᠭᠰᠡᠨ ᠨᠢ ᠮᠣᠩᠭ᠋ᠣᠯᠴᠣᠳ ᠤᠨ ᠮᠠᠯᠵᠢᠯ ᠠᠵᠤ ᠠᠬᠤᠢ ᠨᠡᠯᠢᠶᠡᠳ " +

"ᠬᠦᠭ᠍ᠵᠢᠯᠲᠡ ᠲᠠᠢ ᠪᠣᠯᠤᠭᠰᠠᠨ ᠦᠶ᠎ᠡ ᠲᠠᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠨ᠎ᠠ ᠭᠡᠰᠡᠨ ᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠠᠭᠤᠯᠭ᠎ᠠ ᠡᠴᠡ ᠬᠠᠢ᠌ᠪᠠ᠃ ᠮᠠᠯᠵᠢᠯ ᠰᠣᠶᠣᠯ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠲᠦᠯᠦᠪᠰᠢᠷᠡᠯ ᠢ ᠪᠣᠢ ᠪᠣᠯᠭᠠᠭᠰᠠᠨ ᠢᠶᠡᠷ ᠪᠠᠷᠠᠬᠤ ᠦᠭᠡᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠵᠠᠨ " +

"ᠦᠢᠯᠡ᠂ ᠰᠡᠳᠭᠢᠮᠵᠢ ᠶᠢ ᠢᠯᠡᠳᠲᠡ ᠨᠥᠯᠥᠭᠡᠯᠡᠵᠦ᠂ ᠰᠠᠭᠤᠷᠢ ᠪᠣᠯᠬᠤᠢ᠌ᠴᠠ ᠦᠢᠯᠡᠳᠦᠯ ᠭᠠᠷᠭᠠᠭᠰᠠᠨ ᠪᠠᠢ᠌ᠨ᠎ᠠ. ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ ᠤᠨ ᠠᠭᠤᠯᠭ᠎ᠠ ᠪᠠᠷ ᠢᠶᠡᠨ ᠶᠠᠮᠠᠷ ᠨᠢᠭᠡᠨ ᠶᠠᠭᠤᠮ᠎ᠠ ᠦᠵᠡᠭᠳᠡᠯ᠂ " +

"ᠬᠦᠮᠦᠨ ᠠᠮᠢᠲᠠᠨ ᠢ ᠳᠦᠷᠰᠦᠯᠡᠳᠡᠭ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠰᠡᠳᠭᠢᠮᠵᠢ ᠶᠢᠨ ᠳᠣᠪᠤᠢ᠌ᠮ᠎ᠠ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠯᠵᠠᠢ᠃ ᠡᠳᠡᠭᠡᠷ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠳᠤ ᠮᠣᠩᠭᠣᠯ ᠬᠦᠮᠦᠨ ᠦ ᠮᠣᠯ ᠤᠨᠤᠭ᠎ᠠ ᠳᠤ ᠪᠠᠨ ᠬᠠᠢ᠌ᠷᠠᠲᠠᠢ ᠪᠠᠢ᠌ᠳᠠᠭ ᠲᠦᠭᠡᠮᠡᠯ ᠪᠣᠳᠤᠯ ᠰᠡᠳᠭᠢᠯᠭᠡ ᠪᠠ ᠡᠬᠡ " +

"ᠪᠠᠢ᠌ᠭᠠᠯᠢ ᠪᠠᠨ ᠳᠡᠭᠡᠳᠦᠯᠡᠵᠦ ᠰᠢᠳᠦᠳᠡᠭ ᠦᠵᠡᠯ ᠰᠡᠳᠭᠢᠴᠡ ᠴᠣᠬᠤᠯᠵᠠᠭᠰᠠᠨ ᠪᠠᠢ᠌ᠭ᠎ᠠ ᠶᠢ ᠣᠯᠵᠤ ᠬᠠᠷᠠᠨ᠎ᠠ᠃ ᠳᠠᠷᠤᠢ ᠮᠠᠯᠵᠢᠯ ᠰᠣᠶᠣᠯ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠪᠠᠢ᠌ᠭᠠᠯᠢ ᠶᠢᠨ ᠦᠪᠡᠷᠮᠢᠴᠡ ᠳᠦᠷᠢ ᠲᠦᠷᠬᠦ ᠪᠠ ᠮᠣᠩᠭᠣᠯ ᠬᠦᠮᠦᠨ ᠦ ᠰᠠᠨᠠᠵᠤ ᠰᠡᠳᠭᠢᠬᠦ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠠᠢ᠌ᠳᠠᠯ ᠢ " +

"ᠨᠥᠬᠦᠴᠡᠯᠳᠦᠭᠦᠯᠦᠭᠰᠡᠨ ᠪᠠᠢ᠌ᠭ᠎ᠠ ᠶᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠨᠡᠯᠢᠶᠡᠳ ᠬᠡᠮᠵᠢᠶᠡᠨ᠎ᠡ ᠬᠠᠷᠠᠭᠤᠯᠵᠤ ᠳᠡᠢ᠌ᠯᠬᠦ ᠶᠢ ᠪᠣᠳᠠᠲᠠᠢ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠪᠠᠷ ᠲᠠᠢ᠌ᠯᠪᠤᠷᠢᠯᠠᠪᠠ᠃" ;

MTextArea txtArea2 = new MTextArea( context, 15, 40 );

txtArea2.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

txtArea2.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

// txtArea2.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

txtArea2.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

txtArea2.setLineWrap(true);

txtArea2.setWrapStyleWord(true);

JScrollPane scrollingArea2 = new JScrollPane(txtArea2);

// Dimension preferredSize = new Dimension(800, 400);

// scrollingArea2.setPreferredSize(preferredSize);

gbc.gridx = line;

gbc.gridy = 1;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(scrollingArea2, gbc);

p.add(scrollingArea2);

line++;

MLabel lblTxt9 = new MLabel( "Mongolian" );

lblTxt9.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

lblTxt9.setRotateHint(MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

lblTxt9.setFont(new Font("MS明朝", Font.PLAIN, 16));

gbc.gridx = line;

gbc.gridy = 0;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(lblTxt9, gbc);

p.add(lblTxt9);

context = "ᠲᠤᠰ ᠦᠭᠦᠯᠡᠯ ᠦᠨ ᠤᠳᠤᠷᠢᠳᠭᠠᠯ ᠤᠨ ᠬᠡᠰᠡᠭ ᠲᠥ ᠰᠡᠳᠦᠪ ᠰᠤᠩᠭ᠋ᠤᠭᠰᠠᠨ ᠰᠢᠯᠲᠠᠭᠠᠨ᠂ " +

"ᠠᠴᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠯ᠂ ᠵᠣᠷᠢᠯᠭ᠎ᠠ᠂ ᠠᠷᠭ᠎ᠠ ᠪᠠ ᠰᠣᠳᠤᠯᠤᠯ ᠤᠨ ᠲᠣᠢ᠌ᠮᠣ ᠶᠢ ᠲᠠᠨᠢᠯᠴᠠᠭᠤᠯᠪᠠ᠃ " +

"ᠨᠢᠭᠡᠳᠦᠭᠡᠷ ᠪᠦᠯᠦᠭ ᠲᠥ᠂ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠡᠭᠦᠰᠦᠯ ᠦᠨ ᠱᠠᠲᠤ ᠦᠶ᠎ᠡ ᠨᠡᠯᠢᠶᠡᠳ ᠡᠷᠲᠡ ᠦᠶ᠎ᠡ ᠲᠠᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠬᠤ ᠪᠣᠯᠪᠠᠴᠣ ᠵᠢᠩᠭᠢᠨᠢ ᠲᠦᠯᠦᠪᠰᠢᠷᠡᠵᠦ ᠬᠦᠭ᠍ᠵᠢᠭᠰᠡᠨ ᠨᠢ ᠮᠣᠩᠭ᠋ᠣᠯᠴᠣᠳ ᠤᠨ ᠮᠠᠯᠵᠢᠯ ᠠᠵᠤ ᠠᠬᠤᠢ ᠨᠡᠯᠢᠶᠡᠳ " +

"ᠬᠦᠭ᠍ᠵᠢᠯᠲᠡ ᠲᠠᠢ ᠪᠣᠯᠤᠭᠰᠠᠨ ᠦᠶ᠎ᠡ ᠲᠠᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠨ᠎ᠠ ᠭᠡᠰᠡᠨ ᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠠᠭᠤᠯᠭ᠎ᠠ ᠡᠴᠡ ᠬᠠᠢ᠌ᠪᠠ᠃ ᠮᠠᠯᠵᠢᠯ ᠰᠣᠶᠣᠯ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠲᠦᠯᠦᠪᠰᠢᠷᠡᠯ ᠢ ᠪᠣᠢ ᠪᠣᠯᠭᠠᠭᠰᠠᠨ ᠢᠶᠡᠷ ᠪᠠᠷᠠᠬᠤ ᠦᠭᠡᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠵᠠᠨ " +

"ᠦᠢᠯᠡ᠂ ᠰᠡᠳᠭᠢᠮᠵᠢ ᠶᠢ ᠢᠯᠡᠳᠲᠡ ᠨᠥᠯᠥᠭᠡᠯᠡᠵᠦ᠂ ᠰᠠᠭᠤᠷᠢ ᠪᠣᠯᠬᠤᠢ᠌ᠴᠠ ᠦᠢᠯᠡᠳᠦᠯ ᠭᠠᠷᠭᠠᠭᠰᠠᠨ ᠪᠠᠢ᠌ᠨ᠎ᠠ. ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ ᠤᠨ ᠠᠭᠤᠯᠭ᠎ᠠ ᠪᠠᠷ ᠢᠶᠡᠨ ᠶᠠᠮᠠᠷ ᠨᠢᠭᠡᠨ ᠶᠠᠭᠤᠮ᠎ᠠ ᠦᠵᠡᠭᠳᠡᠯ᠂ " +

"ᠬᠦᠮᠦᠨ ᠠᠮᠢᠲᠠᠨ ᠢ ᠳᠦᠷᠰᠦᠯᠡᠳᠡᠭ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠰᠡᠳᠭᠢᠮᠵᠢ ᠶᠢᠨ ᠳᠣᠪᠤᠢ᠌ᠮ᠎ᠠ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠯᠵᠠᠢ᠃ ᠡᠳᠡᠭᠡᠷ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠳᠤ ᠮᠣᠩᠭᠣᠯ ᠬᠦᠮᠦᠨ ᠦ ᠮᠣᠯ ᠤᠨᠤᠭ᠎ᠠ ᠳᠤ ᠪᠠᠨ ᠬᠠᠢ᠌ᠷᠠᠲᠠᠢ ᠪᠠᠢ᠌ᠳᠠᠭ ᠲᠦᠭᠡᠮᠡᠯ ᠪᠣᠳᠤᠯ ᠰᠡᠳᠭᠢᠯᠭᠡ ᠪᠠ ᠡᠬᠡ " +

"ᠪᠠᠢ᠌ᠭᠠᠯᠢ ᠪᠠᠨ ᠳᠡᠭᠡᠳᠦᠯᠡᠵᠦ ᠰᠢᠳᠦᠳᠡᠭ ᠦᠵᠡᠯ ᠰᠡᠳᠭᠢᠴᠡ ᠴᠣᠬᠤᠯᠵᠠᠭᠰᠠᠨ ᠪᠠᠢ᠌ᠭ᠎ᠠ ᠶᠢ ᠣᠯᠵᠤ ᠬᠠᠷᠠᠨ᠎ᠠ᠃ ᠳᠠᠷᠤᠢ ᠮᠠᠯᠵᠢᠯ ᠰᠣᠶᠣᠯ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠪᠠᠢ᠌ᠭᠠᠯᠢ ᠶᠢᠨ ᠦᠪᠡᠷᠮᠢᠴᠡ ᠳᠦᠷᠢ ᠲᠦᠷᠬᠦ ᠪᠠ ᠮᠣᠩᠭᠣᠯ ᠬᠦᠮᠦᠨ ᠦ ᠰᠠᠨᠠᠵᠤ ᠰᠡᠳᠭᠢᠬᠦ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠠᠢ᠌ᠳᠠᠯ ᠢ " +

"ᠨᠥᠬᠦᠴᠡᠯᠳᠦᠭᠦᠯᠦᠭᠰᠡᠨ ᠪᠠᠢ᠌ᠭ᠎ᠠ ᠶᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠨᠡᠯᠢᠶᠡᠳ ᠬᠡᠮᠵᠢᠶᠡᠨ᠎ᠡ ᠬᠠᠷᠠᠭᠤᠯᠵᠤ ᠳᠡᠢ᠌ᠯᠬᠦ ᠶᠢ ᠪᠣᠳᠠᠲᠠᠢ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠪᠠᠷ ᠲᠠᠢ᠌ᠯᠪᠤᠷᠢᠯᠠᠪᠠ᠃" ;

MTextArea txtArea3 = new MTextArea( context, 15, 40 );

txtArea3.setRotateDirection(MSwingRotateUtilities.ROTATE\_VERTICAL);

txtArea3.setRotateHint(MSwingRotateUtilities.ROTATE\_RIGHTTOLEFT);

// txtArea3.setFont(new Font("Mongolian White Pua", Font.PLAIN, 24));

txtArea3.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

txtArea3.setLineWrap(true);

txtArea3.setWrapStyleWord(true);

JScrollPane scrollingArea3 = new JScrollPane(txtArea3);

// Dimension preferredSize = new Dimension(800, 400);

// scrollingArea2.setPreferredSize(preferredSize);

gbc.gridx = line;

gbc.gridy = 1;

gbc.gridwidth = 1;

gbc.gridheight = 1;

layout.setConstraints(scrollingArea3, gbc);

p.add(scrollingArea3);

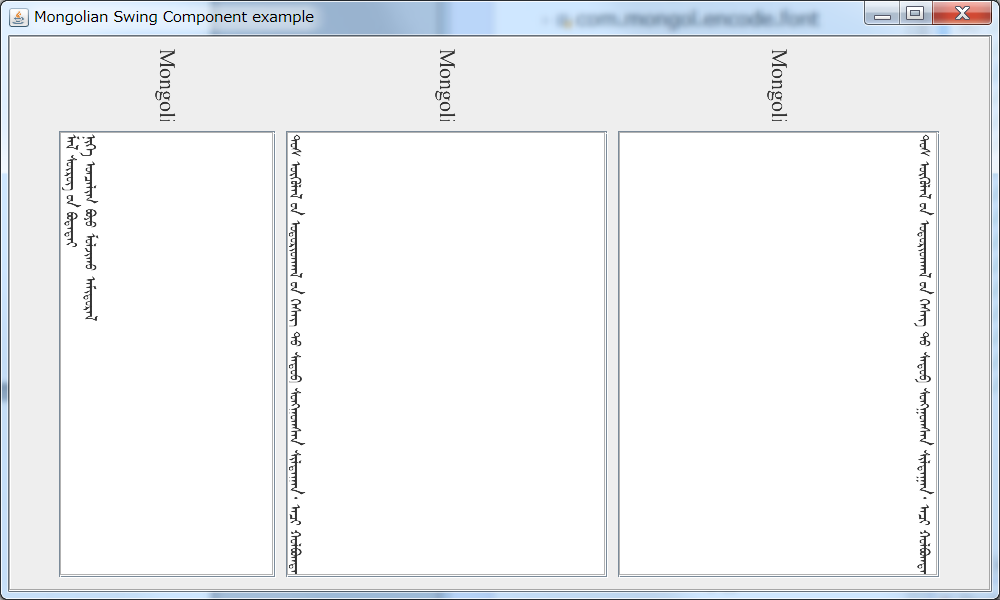
getContentPane().add(p, BorderLayout.CENTER);

p.setSize(980, 680);

}

}

此程序的运行结果如下图所示。



* 1. MTextPane 的使用方法

MTextPane控件是继承Java Swing JTextPane的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ \nᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

MTextPane txtPane2 = **new** MTextPane( MSwingRotateUtilities.***ROTATE\_VERTICAL***, MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

txtPane2.putClientProperty(MEditorPane.***HONOR\_DISPLAY\_PROPERTIES***, Boolean.***TRUE***);

txtPane2.setFont( **new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

txtPane2.setText( context );

txtPane2.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

如上所示:

1. 第一步要构建文本编辑模板MTextPane Object，同时指定文本编辑框的文本显示方向和文本编辑模板的文本的进行方向

MTextPane txtPane2 = **new** TextPane( MSwingRotateUtilities.***ROTATE\_VERTICAL***, MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

1. 指定文文本编辑模板的客户端属性

txtPane2.putClientProperty(MEditorPane.***HONOR\_DISPLAY\_PROPERTIES***, Boolean.***TRUE***);

1. 指定文本编辑模板的字体字号

txtPane2.setFont( **new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

1. 指定文本编辑模板的边框

txtPane2.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

1. 指定文本编辑框的文本内容

txtPane2.setText( context );

下面的程序是，MTextArea使用的完整示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Font;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTextPane;

import javax.swing.border.EtchedBorder;

import javax.swing.text.BadLocationException;

import javax.swing.text.SimpleAttributeSet;

import javax.swing.text.StyleConstants;

import javax.swing.text.StyledDocument;

import com.mongol.swing.MEditorPane;

import com.mongol.swing.MSwingRotateUtilities;

import com.mongol.swing.MTextPane;

public class MTextPaneDemo extends JFrame {

/\*\* serialVersionUID \*/

private static final long serialVersionUID = 1L;

/\*\*

\* @param args

\*/

public static void main(String[] args) {

MTextPaneDemo frame = new MTextPaneDemo();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// frame.setBounds(10, 10, 300, 200);

frame.setTitle("Mongolian Swing Component example");

frame.setSize(1000, 600);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

public MTextPaneDemo() {

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

BorderLayout layout = new BorderLayout();

JPanel p = new JPanel();

p.setLayout(layout);

p.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

context = "ᠲᠤᠰ ᠦᠭᠦᠯᠡᠯ ᠦᠨ ᠤᠳᠤᠷᠢᠳᠭᠠᠯ ᠤᠨ ᠬᠡᠰᠡᠭ ᠲᠥ ᠰᠡᠳᠦᠪ ᠰᠤᠩᠭ᠋ᠤᠭᠰᠠᠨ ᠰᠢᠯᠲᠠᠭᠠᠨ᠂ 日本語" +

"ᠠᠴᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠯ᠂ ᠵᠣᠷᠢᠯᠭ᠎ᠠ᠂ ᠠᠷᠭ᠎ᠠ ᠪᠠ ᠰᠣᠳᠤᠯᠤᠯ ᠤᠨ ᠲᠣᠢ᠌ᠮᠣ ᠶᠢ ᠲᠠᠨᠢᠯᠴᠠᠭᠤᠯᠪᠠ᠃ " +

"ᠨᠢᠭᠡᠳᠦᠭᠡᠷ ᠪᠦᠯᠦᠭ ᠲᠥ᠂ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠡᠭᠦᠰᠦᠯ ᠦᠨ ᠱᠠᠲᠤ ᠦᠶ᠎ᠡ ᠨᠡᠯᠢᠶᠡᠳ ᠡᠷᠲᠡ ᠦᠶ᠎ᠡ ᠲᠠᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠬᠤ ᠪᠣᠯᠪᠠᠴᠣ ᠵᠢᠩᠭᠢᠨᠢ ᠲᠦᠯᠦᠪᠰᠢᠷᠡᠵᠦ ᠬᠦᠭ᠍ᠵᠢᠭᠰᠡᠨ ᠨᠢ ᠮᠣᠩᠭ᠋ᠣᠯᠴᠣᠳ ᠤᠨ ᠮᠠᠯᠵᠢᠯ ᠠᠵᠤ ᠠᠬᠤᠢ ᠨᠡᠯᠢᠶᠡᠳ " +

"ᠬᠦᠭ᠍ᠵᠢᠯᠲᠡ ᠲᠠᠢ ᠪᠣᠯᠤᠭᠰᠠᠨ ᠦᠶ᠎ᠡ ᠲᠠᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠨ᠎ᠠ ᠭᠡᠰᠡᠨ ᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠠᠭᠤᠯᠭ᠎ᠠ ᠡᠴᠡ ᠬᠠᠢ᠌ᠪᠠ᠃ ᠮᠠᠯᠵᠢᠯ ᠰᠣᠶᠣᠯ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠲᠦᠯᠦᠪᠰᠢᠷᠡᠯ ᠢ ᠪᠣᠢ ᠪᠣᠯᠭᠠᠭᠰᠠᠨ ᠢᠶᠡᠷ ᠪᠠᠷᠠᠬᠤ ᠦᠭᠡᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠵᠠᠨ " +

"ᠦᠢᠯᠡ᠂ ᠰᠡᠳᠭᠢᠮᠵᠢ ᠶᠢ ᠢᠯᠡᠳᠲᠡ ᠨᠥᠯᠥᠭᠡᠯᠡᠵᠦ᠂ ᠰᠠᠭᠤᠷᠢ ᠪᠣᠯᠬᠤᠢ᠌ᠴᠠ ᠦᠢᠯᠡᠳᠦᠯ ᠭᠠᠷᠭᠠᠭᠰᠠᠨ ᠪᠠᠢ᠌ᠨ᠎ᠠ. ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ ᠤᠨ ᠠᠭᠤᠯᠭ᠎ᠠ ᠪᠠᠷ ᠢᠶᠡᠨ ᠶᠠᠮᠠᠷ ᠨᠢᠭᠡᠨ ᠶᠠᠭᠤᠮ᠎ᠠ ᠦᠵᠡᠭᠳᠡᠯ᠂ " +

"ᠬᠦᠮᠦᠨ ᠠᠮᠢᠲᠠᠨ ᠢ ᠳᠦᠷᠰᠦᠯᠡᠳᠡᠭ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠰᠡᠳᠭᠢᠮᠵᠢ ᠶᠢᠨ ᠳᠣᠪᠤᠢ᠌ᠮ᠎ᠠ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠯᠵᠠᠢ᠃ ᠡᠳᠡᠭᠡᠷ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠳᠤ ᠮᠣᠩᠭᠣᠯ ᠬᠦᠮᠦᠨ ᠦ ᠮᠣᠯ ᠤᠨᠤᠭ᠎ᠠ ᠳᠤ ᠪᠠᠨ ᠬᠠᠢ᠌ᠷᠠᠲᠠᠢ ᠪᠠᠢ᠌ᠳᠠᠭ ᠲᠦᠭᠡᠮᠡᠯ ᠪᠣᠳᠤᠯ ᠰᠡᠳᠭᠢᠯᠭᠡ ᠪᠠ ᠡᠬᠡ " +

"ᠪᠠᠢ᠌ᠭᠠᠯᠢ ᠪᠠᠨ ᠳᠡᠭᠡᠳᠦᠯᠡᠵᠦ ᠰᠢᠳᠦᠳᠡᠭ ᠦᠵᠡᠯ ᠰᠡᠳᠭᠢᠴᠡ ᠴᠣᠬᠤᠯᠵᠠᠭᠰᠠᠨ ᠪᠠᠢ᠌ᠭ᠎ᠠ ᠶᠢ ᠣᠯᠵᠤ ᠬᠠᠷᠠᠨ᠎ᠠ᠃ ᠳᠠᠷᠤᠢ ᠮᠠᠯᠵᠢᠯ ᠰᠣᠶᠣᠯ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠪᠠᠢ᠌ᠭᠠᠯᠢ ᠶᠢᠨ ᠦᠪᠡᠷᠮᠢᠴᠡ ᠳᠦᠷᠢ ᠲᠦᠷᠬᠦ ᠪᠠ ᠮᠣᠩᠭᠣᠯ ᠬᠦᠮᠦᠨ ᠦ ᠰᠠᠨᠠᠵᠤ ᠰᠡᠳᠭᠢᠬᠦ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠠᠢ᠌ᠳᠠᠯ ᠢ " +

"ᠨᠥᠬᠦᠴᠡᠯᠳᠦᠭᠦᠯᠦᠭᠰᠡᠨ ᠪᠠᠢ᠌ᠭ᠎ᠠ ᠶᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠨᠡᠯᠢᠶᠡᠳ ᠬᠡᠮᠵᠢᠶᠡᠨ᠎ᠡ ᠬᠠᠷᠠᠭᠤᠯᠵᠤ ᠳᠡᠢ᠌ᠯᠬᠦ ᠶᠢ ᠪᠣᠳᠠᠲᠠᠢ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠪᠠᠷ ᠲᠠᠢ᠌ᠯᠪᠤᠷᠢᠯᠠᠪᠠ᠃" ;

// MTextPane txtPane2 = new MTextPane( );

MTextPane txtPane2 = new MTextPane( MSwingRotateUtilities.ROTATE\_VERTICAL, MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

txtPane2.putClientProperty(MEditorPane.HONOR\_DISPLAY\_PROPERTIES, Boolean.TRUE);

txtPane2.setFont( new Font("Mongolian White Pua", Font.PLAIN, 24));

txtPane2.setText( context );

txtPane2.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

JScrollPane scrollingArea2 = new JScrollPane(txtPane2);

p.add(scrollingArea2, BorderLayout.CENTER);

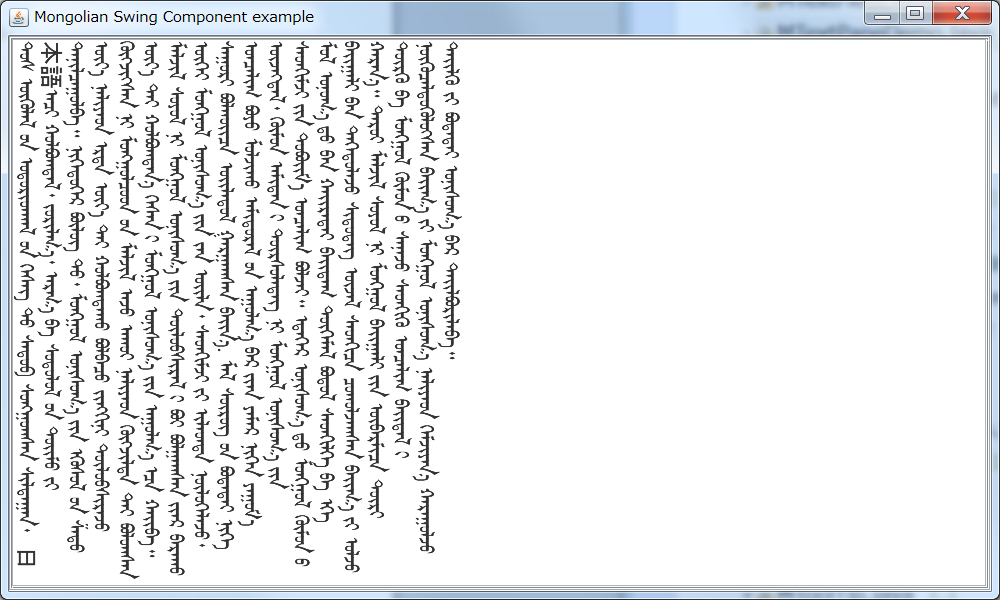
getContentPane().add(p, BorderLayout.CENTER);

p.setSize(980, 680);

}

}

此程序的运行结果如下图所示。



* 1. MEditorPane 的使用方法

MEditorPane控件是继承Java Swing JEditorPane的支持国际标准编码蒙古文的标签控件，能够支持蒙古文的竖写格式。

其使用方法为

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ \nᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

MEditorPane txtPane2 = **new** MEditorPane ( MSwingRotateUtilities.***ROTATE\_VERTICAL***, MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

txtPane2.putClientProperty(MEditorPane.***HONOR\_DISPLAY\_PROPERTIES***, Boolean.***TRUE***);

txtPane2.setFont( **new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

txtPane2.setText( context );

txtPane2.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

如上所示:

1. 第一步要构建文本编辑模板MEditorPane Object，同时指定文本编辑框的文本显示方向和文本编辑模板的文本的进行方向

MEditorPane txtPane2 = **new** MEditorPane ( MSwingRotateUtilities.***ROTATE\_VERTICAL***, MSwingRotateUtilities.***ROTATE\_LEFTTORIGHT***);

1. 指定文文本编辑模板的客户端属性

txtPane2.putClientProperty(MEditorPane.***HONOR\_DISPLAY\_PROPERTIES***, Boolean.***TRUE***);

1. 指定文本编辑模板的字体字号

txtPane2.setFont( **new** Font("Mongolian White Pua", Font.***PLAIN***, 24));

1. 指定文本编辑模板的边框

txtPane2.setBorder(**new** EtchedBorder(EtchedBorder.***LOWERED***));

1. 指定文本编辑框的文本内容

txtPane2.setText( context );

下面的程序是，MEditorPane使用的完整示例程序。

package com.mongol.demo;

import java.awt.BorderLayout;

import java.awt.Font;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.border.EtchedBorder;

import javax.swing.text.BadLocationException;

import javax.swing.text.SimpleAttributeSet;

import javax.swing.text.StyleConstants;

import javax.swing.text.StyledDocument;

import com.mongol.swing.MEditorPane;

import com.mongol.swing.MSwingRotateUtilities;

public class MEditorPaneDemo extends JFrame {

/\*\* serialVersionUID \*/

private static final long serialVersionUID = 1L;

/\*\*

\* @param args

\*/

public static void main(String[] args) {

MEditorPaneDemo frame = new MEditorPaneDemo();

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

// frame.setBounds(10, 10, 300, 200);

frame.setTitle("Mongolian Swing Component example");

frame.setSize(1000, 600);

frame.setLocationRelativeTo(null);

frame.setVisible(true);

}

public MEditorPaneDemo() {

String context = "ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ  " ;

BorderLayout layout = new BorderLayout();

JPanel p = new JPanel();

p.setLayout(layout);

p.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

context = "ᠲᠤᠰ ᠦᠭᠦᠯᠡᠯ ᠦᠨ ᠤᠳᠤᠷᠢᠳᠭᠠᠯ ᠤᠨ ᠬᠡᠰᠡᠭ ᠲᠥ ᠰᠡᠳᠦᠪ ᠰᠤᠩᠭ᠋ᠤᠭᠰᠠᠨ ᠰᠢᠯᠲᠠᠭᠠᠨ᠂ " +

"ᠠᠴᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠯ᠂ ᠵᠣᠷᠢᠯᠭ᠎ᠠ᠂ ᠠᠷᠭ᠎ᠠ ᠪᠠ ᠰᠣᠳᠤᠯᠤᠯ ᠤᠨ ᠲᠣᠢ᠌ᠮᠣ ᠶᠢ ᠲᠠᠨᠢᠯᠴᠠᠭᠤᠯᠪᠠ᠃ " +

"ᠨᠢᠭᠡᠳᠦᠭᠡᠷ ᠪᠦᠯᠦᠭ ᠲᠥ᠂ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠡᠭᠦᠰᠦᠯ ᠦᠨ ᠱᠠᠲᠤ ᠦᠶ᠎ᠡ ᠨᠡᠯᠢᠶᠡᠳ ᠡᠷᠲᠡ ᠦᠶ᠎ᠡ ᠲᠠᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠬᠤ ᠪᠣᠯᠪᠠᠴᠣ ᠵᠢᠩᠭᠢᠨᠢ ᠲᠦᠯᠦᠪᠰᠢᠷᠡᠵᠦ ᠬᠦᠭ᠍ᠵᠢᠭᠰᠡᠨ ᠨᠢ ᠮᠣᠩᠭ᠋ᠣᠯᠴᠣᠳ ᠤᠨ ᠮᠠᠯᠵᠢᠯ ᠠᠵᠤ ᠠᠬᠤᠢ ᠨᠡᠯᠢᠶᠡᠳ " +

"ᠬᠦᠭ᠍ᠵᠢᠯᠲᠡ ᠲᠠᠢ ᠪᠣᠯᠤᠭᠰᠠᠨ ᠦᠶ᠎ᠡ ᠲᠠᠢ ᠬᠣᠯᠪᠣᠭᠳᠠᠨ᠎ᠠ ᠭᠡᠰᠡᠨ ᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠠᠭᠤᠯᠭ᠎ᠠ ᠡᠴᠡ ᠬᠠᠢ᠌ᠪᠠ᠃ ᠮᠠᠯᠵᠢᠯ ᠰᠣᠶᠣᠯ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠲᠦᠯᠦᠪᠰᠢᠷᠡᠯ ᠢ ᠪᠣᠢ ᠪᠣᠯᠭᠠᠭᠰᠠᠨ ᠢᠶᠡᠷ ᠪᠠᠷᠠᠬᠤ ᠦᠭᠡᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠵᠠᠨ " +

"ᠦᠢᠯᠡ᠂ ᠰᠡᠳᠭᠢᠮᠵᠢ ᠶᠢ ᠢᠯᠡᠳᠲᠡ ᠨᠥᠯᠥᠭᠡᠯᠡᠵᠦ᠂ ᠰᠠᠭᠤᠷᠢ ᠪᠣᠯᠬᠤᠢ᠌ᠴᠠ ᠦᠢᠯᠡᠳᠦᠯ ᠭᠠᠷᠭᠠᠭᠰᠠᠨ ᠪᠠᠢ᠌ᠨ᠎ᠠ. ᠮᠠᠯ ᠰᠦᠷᠦᠭ ᠦᠨ ᠪᠣᠳᠠᠲᠠᠢ ᠨᠢᠭᠡ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠶᠣ ᠮᠣᠯᠵᠢᠬᠤ ᠠᠮᠢᠳᠤᠷᠠᠯ ᠤᠨ ᠠᠭᠤᠯᠭ᠎ᠠ ᠪᠠᠷ ᠢᠶᠡᠨ ᠶᠠᠮᠠᠷ ᠨᠢᠭᠡᠨ ᠶᠠᠭᠤᠮ᠎ᠠ ᠦᠵᠡᠭᠳᠡᠯ᠂ " +

"ᠬᠦᠮᠦᠨ ᠠᠮᠢᠲᠠᠨ ᠢ ᠳᠦᠷᠰᠦᠯᠡᠳᠡᠭ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠶᠢᠨ ᠰᠡᠳᠭᠢᠮᠵᠢ ᠶᠢᠨ ᠳᠣᠪᠤᠢ᠌ᠮ᠎ᠠ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠣᠯᠵᠠᠢ᠃ ᠡᠳᠡᠭᠡᠷ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠳᠤ ᠮᠣᠩᠭᠣᠯ ᠬᠦᠮᠦᠨ ᠦ ᠮᠣᠯ ᠤᠨᠤᠭ᠎ᠠ ᠳᠤ ᠪᠠᠨ ᠬᠠᠢ᠌ᠷᠠᠲᠠᠢ ᠪᠠᠢ᠌ᠳᠠᠭ ᠲᠦᠭᠡᠮᠡᠯ ᠪᠣᠳᠤᠯ ᠰᠡᠳᠭᠢᠯᠭᠡ ᠪᠠ ᠡᠬᠡ " +

"ᠪᠠᠢ᠌ᠭᠠᠯᠢ ᠪᠠᠨ ᠳᠡᠭᠡᠳᠦᠯᠡᠵᠦ ᠰᠢᠳᠦᠳᠡᠭ ᠦᠵᠡᠯ ᠰᠡᠳᠭᠢᠴᠡ ᠴᠣᠬᠤᠯᠵᠠᠭᠰᠠᠨ ᠪᠠᠢ᠌ᠭ᠎ᠠ ᠶᠢ ᠣᠯᠵᠤ ᠬᠠᠷᠠᠨ᠎ᠠ᠃ ᠳᠠᠷᠤᠢ ᠮᠠᠯᠵᠢᠯ ᠰᠣᠶᠣᠯ ᠨᠢ ᠮᠣᠩᠭᠣᠯ ᠪᠠᠢ᠌ᠭᠠᠯᠢ ᠶᠢᠨ ᠦᠪᠡᠷᠮᠢᠴᠡ ᠳᠦᠷᠢ ᠲᠦᠷᠬᠦ ᠪᠠ ᠮᠣᠩᠭᠣᠯ ᠬᠦᠮᠦᠨ ᠦ ᠰᠠᠨᠠᠵᠤ ᠰᠡᠳᠭᠢᠬᠦ ᠣᠨᠴᠠᠯᠢᠭ ᠪᠠᠢ᠌ᠳᠠᠯ ᠢ " +

"ᠨᠥᠬᠦᠴᠡᠯᠳᠦᠭᠦᠯᠦᠭᠰᠡᠨ ᠪᠠᠢ᠌ᠭ᠎ᠠ ᠶᠢ ᠮᠣᠩᠭᠣᠯ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠨᠡᠯᠢᠶᠡᠳ ᠬᠡᠮᠵᠢᠶᠡᠨ᠎ᠡ ᠬᠠᠷᠠᠭᠤᠯᠵᠤ ᠳᠡᠢ᠌ᠯᠬᠦ ᠶᠢ ᠪᠣᠳᠠᠲᠠᠢ ᠣᠨᠢᠰᠤᠭ᠎ᠠ ᠪᠠᠷ ᠲᠠᠢ᠌ᠯᠪᠤᠷᠢᠯᠠᠪᠠ᠃" ;

// MEditorPane txtPane2 = new MEditorPane( );

MEditorPane txtPane2 = new MEditorPane( MSwingRotateUtilities.ROTATE\_VERTICAL, MSwingRotateUtilities.ROTATE\_LEFTTORIGHT);

txtPane2.putClientProperty(MEditorPane.HONOR\_DISPLAY\_PROPERTIES, Boolean.TRUE);

// txtPane2.setFont( new Font("Mongolian White Pua", Font.PLAIN, 24));

txtPane2.setText( context );

txtPane2.setBorder(new EtchedBorder(EtchedBorder.LOWERED));

JScrollPane scrollingArea2 = new JScrollPane(txtPane2);

p.add(scrollingArea2, BorderLayout.CENTER);

getContentPane().add(p, BorderLayout.CENTER);

p.setSize(980, 680);

}

}

此程序的运行结果如下图所示。

