Izpack installer guide

intagleo systems

v 1.0

author: Dev Team

Contents

CHAPTER 1

Simple Installer

Download izpack from http://izpack.org/downloads/

Run IzPack-install.jar. jdk should be installed on your pc.

After installation open folder where it is installed.

Normally its in C:\Program Files\IzPack

Go to bin folder

Make install.xml file. Sample file is given below. You can learn about different tags of this file from manual.pdf given in IzPack\doc\izpack\pdf or visit this link:
http://izpack.org/documentation/installation-files.html

::::::::::::::::::::::::::::::
install.xml
::::::::::::::::::::::::::::::

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>

<!--
   A sample installation file.
   Use it as a base for your own installers :-)

   To compile it :
   - go in the bin directory where you installed IzPack
    compile install.xml -b . -o install.jar -k standard
-->

<installation version="1.0">

   <!--
      The info section.
      The meaning of the tags should be natural ...
   -->
   <info>
      <appname>Test installer</appname>
      <appversion>1.0</appversion>
      <authors>
         <author name="Imran Tariq" email="m.imran.tariq@gmail.com"/>
      </authors>
      <url>http://www.imran.com/</url>
   </info>
```

```
<!--
   The gui preferences indication.
   Sets the installer window to 640x480. It will not be able to change the size.
-->
<guiprefs width="660" height="480" resizable="yes">
        <modifier key="allXGap" value="0"/>
        <modifier key="allYGap" value="0"/>
        <modifier key="useHeadingPanel" value="yes"/>
        <modifier key="useButtonIcons" value="yes"/>
        <modifier key="useHeadingForSummary" value="yes"/>
        <modifier key="headingImageOnLeft" value="yes"/>
        <modifier key="headingLineCount" value="1"/>
        <modifier key="headingFontSize" value="1.5"/>
        <modifier key="headingBackgroundColor" value="0x00ffffff"/>
        <modifier key="headingPanelCounter" value="text"/>
        <modifier key="headingPanelCounterPos" value="inHeading"/>
</guiprefs>
<!--
   The locale section.
   Asks here to include the English and French langpacks.
-->
<locale>
   <langpack iso3="eng" />
</locale>

<!--
   The resources section.
   The ids must be these ones if you want to use the LicencePanel and/or the InfoPanel.
-->
<resources>
   <res id="HTMLLicencePanel.licence" src="license.html"/>
   <res id="InfoPanel.info" src="Readme.txt"/>
   <res id="SummaryPanel.info" src="Summry.txt"/>
</resources>

<!--
   The panels section.
   We indicate here which panels we want to use. The order will be respected.
-->
<panels>
   <panel classname="HelloPanel"/>
   <panel classname="HTMLLicencePanel"/>
  <panel classname="SimpleFinishPanel"/>

</panels>
```

```
   <!--
      The packs section.
      We specify here our packs.
   -->ÿ
   <packs>
   <pack name="Base" required="yes" preselected="yes">
            <description>The base files</description>
            <file src="Readme.txt" targetdir="$INSTALL_PATH"/>
            <file src="Licence.txt" targetdir="$INSTALL_PATH"/>
   </pack>
</packs>


</installation>
```

::::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::::

Make sure that the resources specified in the <resources> tag are available in the bin folder.

to compile it. Open command prompt. Go to bin folder and run this command.

::::::::::::::::::::
::::::::::::::::::::
compile install.xml -b . -o install.jar -k standard
::::::::::::::::::::
::::::::::::::::::::

It will make install.jar in the directory.
Double click it to run your simple installer.

CHAPTER 2

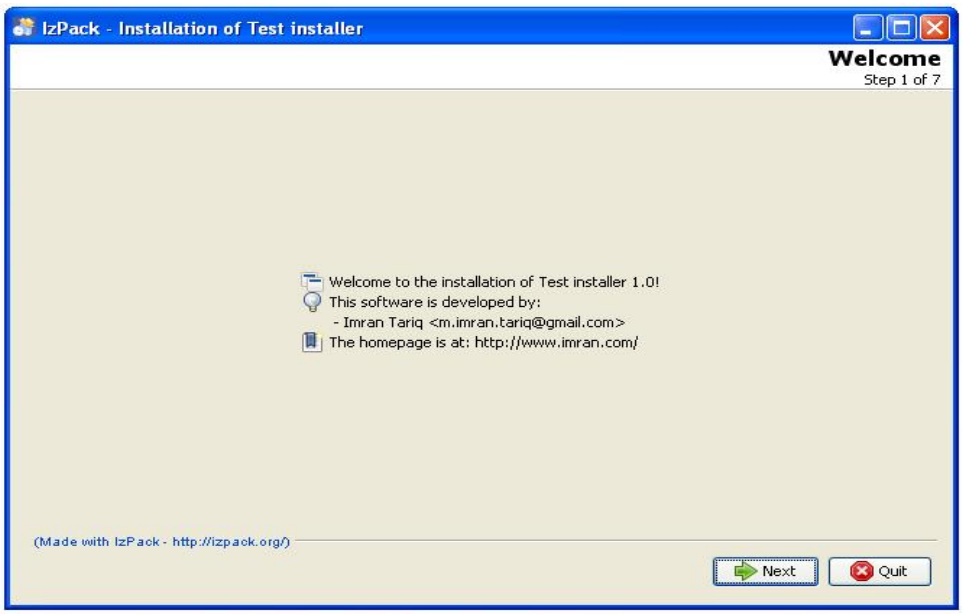
Understanding Install.xml

install.xml is the main file that contains all the information to make an installer.
e.g how many panels to show in installer. what kind of information it contain etc.
Now we'll briefly discuss different tags in this file.

<info>
---

info to be shown under "hellopanel" as show below.

```
<info>
   <appname>Test installer</appname>
   <appversion>1.0</appversion>
   <authors>
      <author name="Imran Tariq" email="m.imran.tariq@gmail.com"/>
   </authors>
   <url>http://www.imran.com/</url>
</info>
```



<guiprefs>

The gui preferences indication.

Set the preferences as you want your installer look and feel like color, fonts etc.

```
<guiprefs width="660" height="480" resizable="yes">
        <modifier key="allXGap" value="0"/>
        <modifier key="allYGap" value="0"/>
        <modifier key="useHeadingPanel" value="yes"/>
        <modifier key="useButtonIcons" value="yes"/>
        <modifier key="useHeadingForSummary" value="yes"/>
        <modifier key="headingImageOnLeft" value="yes"/>
        <modifier key="headingLineCount" value="1"/>
        <modifier key="headingFontSize" value="1.5"/>
        <modifier key="headingBackgroundColor" value="0x00ffffff"/>
        <modifier key="headingPanelCounter" value="text"/>
        <modifier key="headingPanelCounterPos" value="inHeading"/>
</guiprefs>
```

<locale>

Sets the different languages for you installer.

```
    <locale>
        <langpack iso3="pol"/>
        <langpack iso3="eng" />
    </locale>
```

<resources>

Set resources here that are to be used by panels. e.g. liscence.txt file.

```
<resources>
   <res id="HTMLLicencePanel.licence" src="license.html"/>
   <res id="InfoPanel.info" src="Readme.txt"/>
   <res id="SummaryPanel.info" src="Summry.txt"/>
</resources>
```

<panels>

Include panels for installer.
Panels can be built in or you custom panels which can be java classes having swing objects etc.

```
<panels>
   <panel classname="HelloPanel"/>
   <panel classname="HTMLLicencePanel"/>
   <panel classname="TargetPanel"/>
   <panel classname="InstallPanel"/>
   <panel classname="SimpleFinishPanel"/>
</panels>
```

Pictures of Different panels are shown at the end of this document.

<packs>

This tag will add a set of files written in it to the installation path.
e.g if you want your liscence.txt file to be copied to installation path then you will write it as
<file src="Licence.txt" targetdir="$INSTALL_PATH"/>
more info about this tag can be seen here

http://docs.codehaus.org/display/IZPACK/Packs

```
<packs>
  <pack name="Base" required="yes" preselected="yes">
            <description>The base files</description>
            <file src="Readme.txt" targetdir="$INSTALL_PATH"/>
            <file src="Licence.txt" targetdir="$INSTALL_PATH"/>
           <fileset dir="lib" targetdir="$INSTALL_PATH\lib" />
    </pack>
  </packs>
```

<executable>

The <executable> tag is a very useful thing if you need to execute something during the installation process.
It can also be used to set the executable flag on Unix-like systems.

CHAPTER 3

Adding panels to your installer

First step was a simple installer with 3 pnaels. Izpack has built in panels for various tasks.
e.g. HelloPanel is the first panel that shows the information in the <info> tag in install.xml

To make your own panel we'll create JAVA classes. We'll make swing objects to get inputs or show other things.

Following is the sample java class for a panel. Use it to see how constructors are used and what classes are inherited etc.
This java class will get two paths from user. After getting those paths you can add you logic you want.

To configure izpack environment see this link. http://blog.samratdhillon.com/archives/426 having 3 steps
Part 1 Setup on Eclipse
Part 2 Building the Installer
Part 3 Customizing the installer

you just only need to setup on eclipse.

::
mypaneltest.java
::

```
package com.custom;

import java.awt.LayoutManager2;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JTextField;
import java.util.logging.*;
```

```
import com.izforge.izpack.gui.ButtonFactory;
import com.izforge.izpack.gui.IzPanelLayout;
import com.izforge.izpack.installer.InstallData;
import com.izforge.izpack.installer.InstallerFrame;
import com.izforge.izpack.installer.IzPanel;

public class MyPanelTest extends IzPanel implements ActionListener {

/**
 *
ÿ */
private static final long serialVersionUID = 1L;
private static Logger logger = Logger.getLogger("com.custom.MyPanelTest");

private JLabel lblTest;
private JLabel lblTest1;

private JButton browseButton;
private JButton browseButton1;

private JLabel lblEmpty;
private JTextField txtPath;
private JTextField txtPath1;

String srcPath = null;
String destPath = null;

String srcPath1 = null;
String destPath1 = null;

public MyPanelTest(InstallerFrame parent, InstallData idata) {
   this(parent, idata, new IzPanelLayout());
}

public MyPanelTest(InstallerFrame parent, InstallData idata,
     LayoutManager2 lm) {
     super(parent, idata, lm);

    lblTest = new JLabel("Select any folder to copy ... ");
    lblTest1 = new JLabel("Select location to copy to ... ");

    lblEmpty = new JLabel(" ");
    txtPath = new JTextField(49);
     txtPath1 = new JTextField(49);
```

```
browseButton = ButtonFactory.createButton(getInstallerFrame().langpack
.getString("TargetPanel.browse"), getInstallerFrame().icons
.getImageIcon("open"), idata.buttonsHColor);

browseButton1 = ButtonFactory.createButton(getInstallerFrame().langpack
.getString("TargetPanel.browse"), getInstallerFrame().icons
.getImageIcon("open"), idata.buttonsHColor);

add(lblTest, NEXT_LINE);
add(lblEmpty, NEXT_LINE);
add(txtPath, NEXT_LINE);
add(browseButton);
add(lblEmpty, NEXT_LINE);

add(lblTest1, NEXT_LINE);
add(lblEmpty, NEXT_LINE);
add(txtPath1, NEXT_LINE);
add(browseButton1);

browseButton.addActionListener(this);
browseButton1.addActionListener(this);
}

@Override
public void actionPerformed(ActionEvent e) {

  Object source = e.getSource();

  if (source == browseButton) {
     JFileChooser fc = new JFileChooser();
     // fc.setCurrentDirectory(new File(txtPath.getText()));
     fc.setMultiSelectionEnabled(false);
     fc.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
     fc.addChoosableFileFilter(fc.getAcceptAllFileFilter());

     if (fc.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
        String srcPath = fc.getSelectedFile().getAbsolutePath();
        txtPath.setText(srcPath);
     }
  }

  if (source == browseButton1) {
     JFileChooser fc = new JFileChooser();
     // fc.setCurrentDirectory(new File(txtPath.getText()));
     fc.setMultiSelectionEnabled(false);
     fc.setFileSelectionMode(JFileChooser.DIRECTORIES_ONLY);
     fc.addChoosableFileFilter(fc.getAcceptAllFileFilter());
```

```
            if (fc.showOpenDialog(this) == JFileChooser.APPROVE_OPTION) {
                String destPath = fc.getSelectedFile().getAbsolutePath();
                txtPath1.setText(destPath);
            }
        }

    }

    /**
ÿ * Indicates wether the panel has been validated or not.
ÿ *
ÿ * @return Always true.
ÿ */
    public boolean isValidated() {

    File src = new File(srcPath);
    File dest = new File(destPath);

    MyPanelTest cd = new MyPanelTest(parent, idata, null);
    try {
    cd.copyDirectory(src, dest);
    } catch (IOException e1) {
    e1.printStackTrace();
    }
    return true;
    }

}
```

::
::

Now we'll use this panel in installer.

1: Make a jar of its class file. In eclipse you can export mypaneltest.java to make its jar file.
2: Place this jar file in \bin\panels
3: Add this class in <panels> tag in install.xml like this
<panel classname="mypaneltest"/>
add it after any existing panel you want it to work. After adding it above install.xml will become.

```
<panels>
    <panel classname="HelloPanel"/>
    <panel classname="HTMLLicencePanel"/>
    <panel classname="mypaneltest"/>
   <panel classname="SimpleFinishPanel"/>
</panels>
```

4: Now again run compile code from command prompt.
:::::::::::::::::::
compile install.xml -b . -o install.jar -k standard
:::::::::::::::::::

5: run install.jar to check you new panel.

CHAPTER 4

Adding Process panel (batch process)

Now we'll add processes that will do some batch processing.
e.g we may want to install some software through our installer or copy directories etc.
we can do most of the same task through our java classes.

to add process panel change install.xml like this

1: add <res id="ProcessPanel.Spec.xml" src="ProcessPanel.Spec.xml"/> in <resources> tag.

```
<resources>
      <res id="HTMLLicencePanel.licence" src="license.html"/>
      <res id="InfoPanel.info" src="Readme.txt"/>
     <res id="ProcessPanel.Spec.xml" src="ProcessPanel.Spec.xml"/>
      <res id="SummaryPanel.info" src="Summry.txt"/>
</resources>
```

2: add its panel in <panels> tag

```
   <panels>
     <panel classname="HelloPanel"/>
     <panel classname="HTMLLicencePanel"/>
     <panel classname="mypaneltest"/>
     <panel classname="ProcessPanel"/>
    <panel classname="SimpleFinishPanel"/>
   </panels>
```

3: Make ProcessPanel.Spec.xml in bin folder.
sample ProcessPanel.Spec.xml is given below.

::
ProcessPanel.Spec.xml
::

```
   <processing>

     <job name="Starting Server ...">
     <os family="windows" />
     <executefile name="script/start_server.bat" >
     <env>SERVER_PATH=$WAS_PATH</env>
     <env>PROFILE=$profile</env>
     </executefile>
     </job>

     <job name="Copying Reports and Themes ...">
     <os family="windows" />
     <executefile name="script/copy_reports.bat" >
     </executefile>
     </job>

   </processing>
```

In this file we can see <job> tag. there are 2 jobs. one will start server and second will copy some files.
we can set environment variables for our jobs. Each job will execute a batch file. Batch file will do its task.
sample copy_reports.bat is given below.

```
::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::
copy_reports.bat
::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::

d:
mkdir test
cd test
COPY c:\folder\* d:\test

::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::
```

CHAPTER 5

Working on other tasks

Working on unistalling application

To add uninstall option to your installer, add these izpack panels to your install.xml file in <panels> tag.

```
<panel classname="TargetPanel"/>
<panel classname="InstallPanel"/>
```

Now compile your install.xml file as discussed above and see the changes.

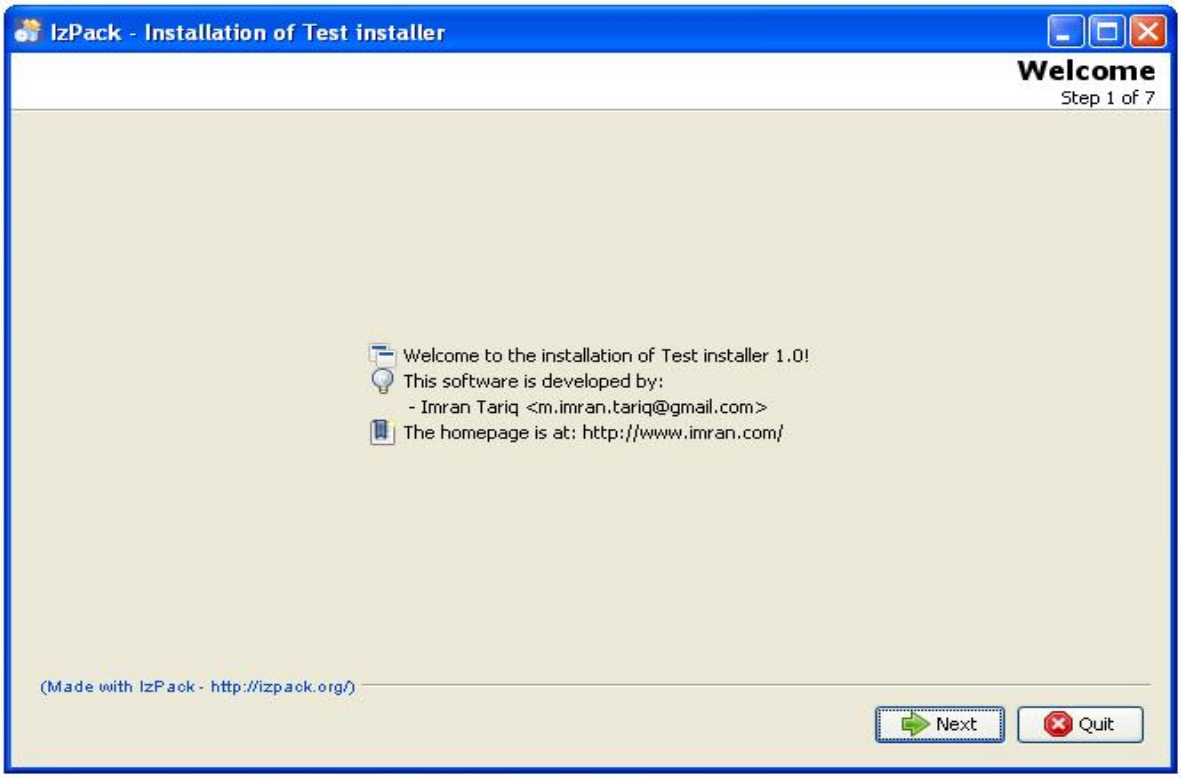
Adding headline to you custom panel

To add heading to your custom panel add following line to you lanuage pack xml file.
Go to IzPack\bin\langpacks\installer
open your language pack xml file e.g eng.xml.
Add this line

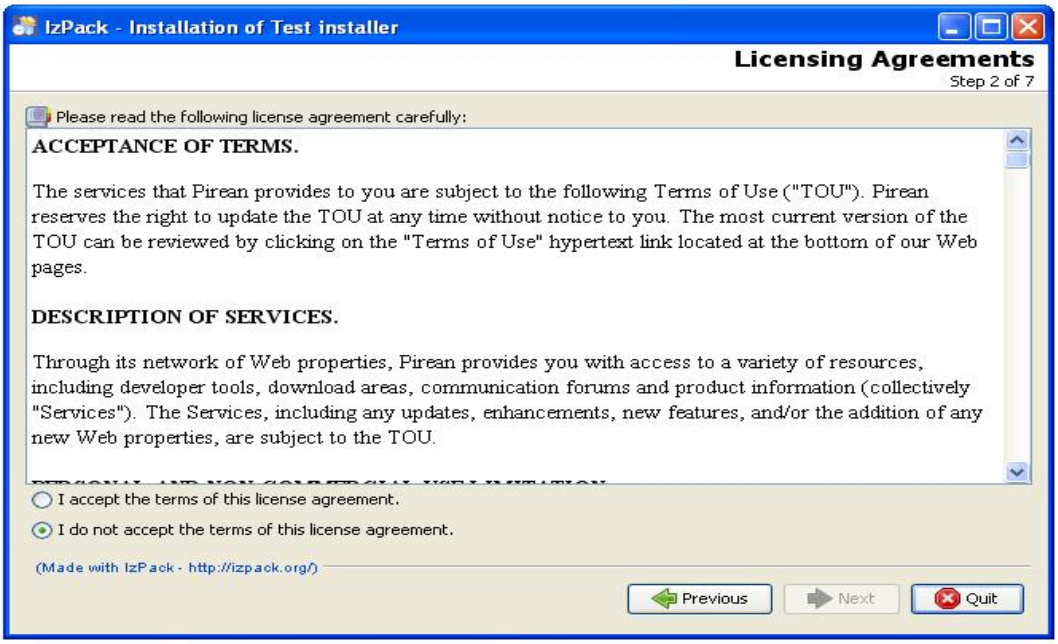
```
<str id="TestPanel.headline" txt="Test Heading"/>
```

Here TestPanel is the name of your panel and text in txt="Test Heading" is the heading to be shown in TestPanel.

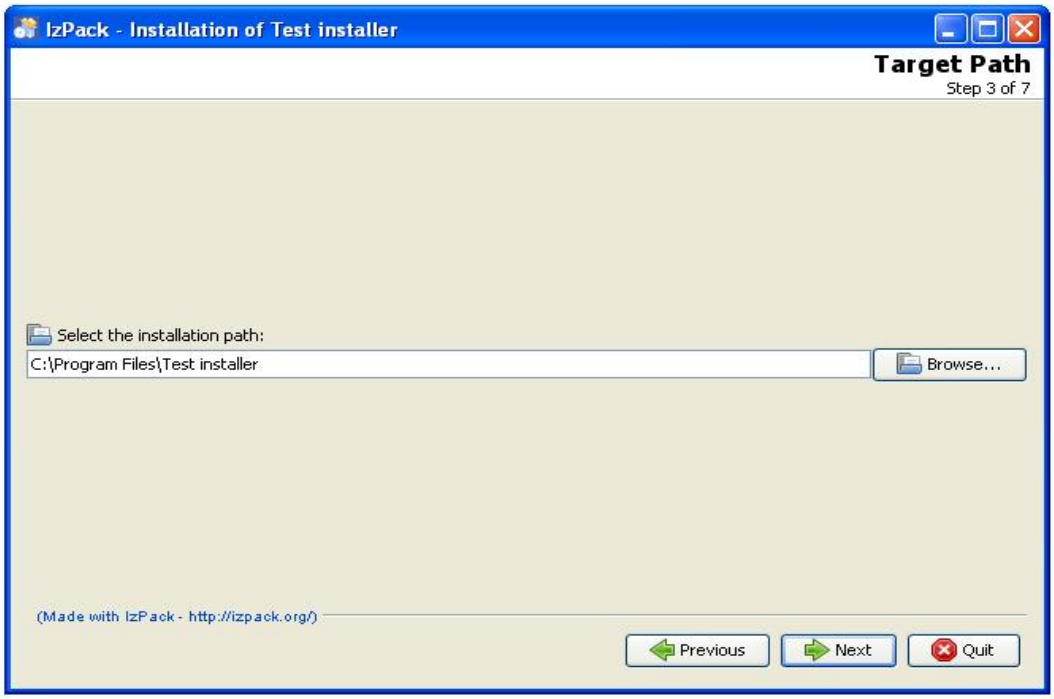
Different Panels screenshots



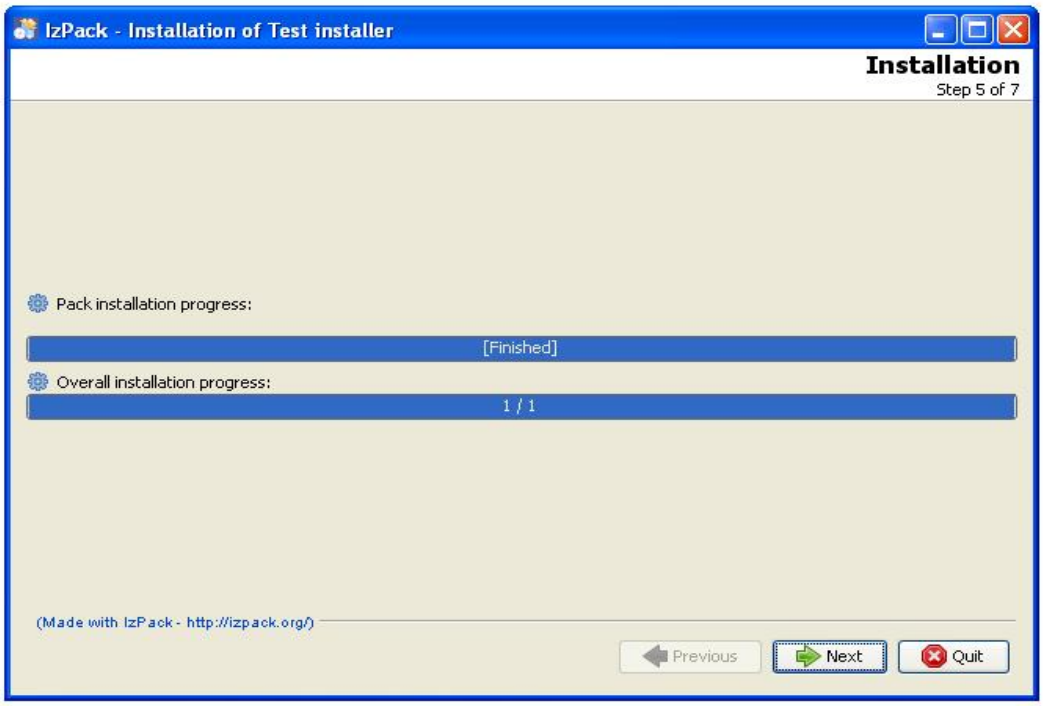
HelloPanel



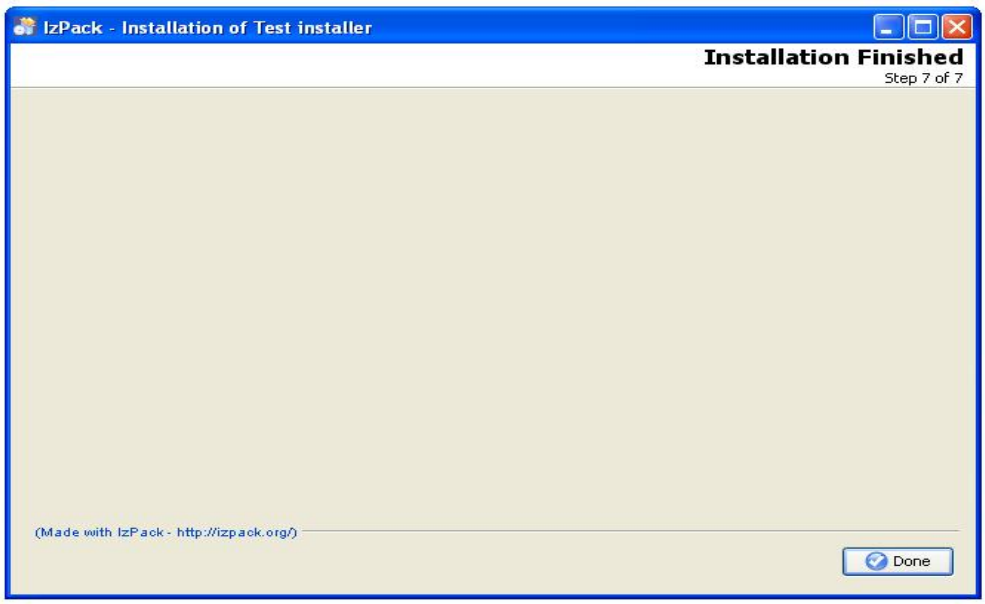
HTMLLicencePanel



TargetPanel



InstallPanel



SimpleFinishPanel

Reference

http://izpack.org/downloads/
http://izpack.org/documentation/installation-files.html
http://blog.samratdhillon.com/archives/426
http://docs.codehaus.org/display/IZPACK/Packs