**NetBeans Platform Crash Course**

By NetBeans User Group Serbia [www.netbeans-serbia.org](http://www.netbeans-serbia.org/)

In this tutorial we will show some additional concepts, which are addition to previous tutorial.

**Overview:**

1. Adding new property window
2. Connecting windows in our application using Lookup API

Requirements:

Java 1.6 or later

Netbeans 7.0 Beta 2 or later

Prerequisites :

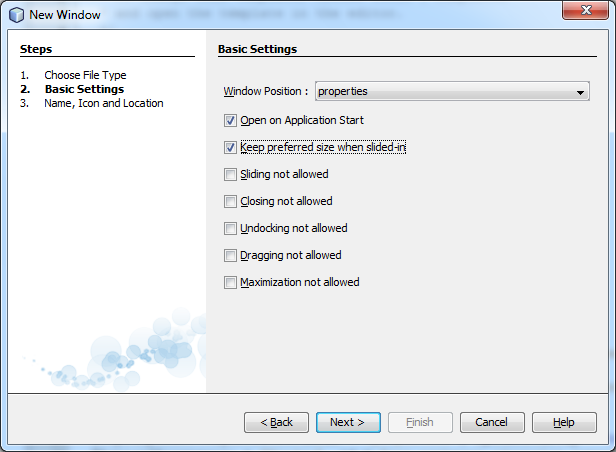
Basic Java and SWING programing.

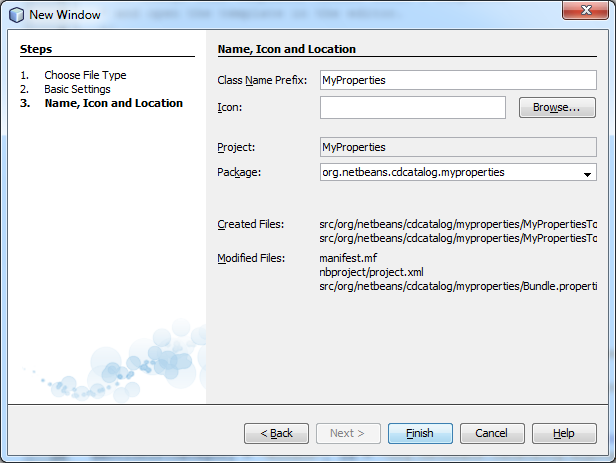
To follow this tutorial, it is necessary to download and open project considering a second day of NetBeans Platform 2011 training (<http://www.netbeans-serbia.org/participants-of-netbeans-platform-training-2011>.). In following sections you will find code modifications, which you will have to implement in project from second day of NetBeans tutorial.

Adding new Property window

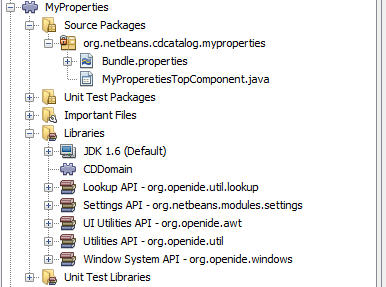
When you start second day project, there will be option for opening a Property window, which displays information about CD you selected (providing it was added by wizard). If you modify data in this window the CD object will update its values accordingly. Our task is to create our own property window, so we could display data in our own format.

In project suite add new module called ***MyProperties***  (package name: ***org.netbeans.cdcatalog.myproperties***). After new module is created, right click on module project and follow next steps: ***New> Window*** and enter following data:





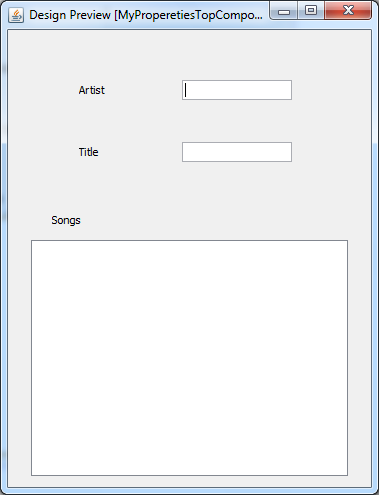
On image below is displayed architecture of ***MyProperty*** module (it is also shown which modules dependencies should be included).



(Hint # 1 -> in case you need to make modules package public for other modules in application these are necessary steps : right click on module project ***-> Properties-> API Versioning*** , then in ***Public Packages*** section chose which packages should be set as public, so the other modules in application can make dependencies to this package.

Hint # 2 -> in case you need to make dependency to other module these are the steps: right click on module project > ***Properties->Libraries->Add Dependency->*** from ***Module*** section chose module you need, and if it’s not available just check ***Show Non-API modules.*** )

After new window steps are completed, in front of you should be displayed a blank JPannel form. Add on it three label fields, two text fields and one text area field:



Next step is to rename our form components (for my text field components I used „txt“+labelText - >txtArtist, txtSongs...).

The following section of code represents part of MyProperetiesTopComponent class, with essential modifications (marked with red color):

|  |
| --- |
| package org.netbeans.cdcatalog.myproperties;  import java.util.Collection;  import org.openide.util.LookupEvent;  import org.openide.util.NbBundle;  import org.openide.windows.TopComponent;  import org.openide.windows.WindowManager;  import org.netbeans.api.settings.ConvertAsProperties;  import org.netbeans.cdcatalog.domain.CD;  import org.openide.awt.ActionID;  import org.openide.awt.ActionReference;  import org.openide.util.Lookup.Result;  import org.openide.util.LookupListener;  import org.openide.util.Utilities;  @ConvertAsProperties(dtd = "-//org.netbeans.cdcatalog.myproperties//MyPropereties//EN",  autostore = false)  @TopComponent.Description(preferredID = "MyProperetiesTopComponent",  //iconBase="SET/PATH/TO/ICON/HERE",  persistenceType = TopComponent.PERSISTENCE\_ALWAYS)  @TopComponent.Registration(mode = "properties", openAtStartup = true)  @ActionID(category = "Window", id = "org.netbeans.cdcatalog.myproperties.MyProperetiesTopComponent")  @ActionReference(path = "Menu/Window" /\*, position = 333 \*/)  @TopComponent.OpenActionRegistration(displayName = "#CTL\_MyProperetiesAction",  preferredID = "MyProperetiesTopComponent")  public final class MyProperetiesTopComponent extends TopComponent implements LookupListener {  …  Result<CD> result;  @Override  public void componentOpened() {  result = Utilities.actionsGlobalContext().lookupResult(CD.class);  result.addLookupListener(this);  }  @Override  public void resultChanged(LookupEvent le) {  Collection <? extends CD> allInstances =result.allInstances();  for (CD cd : allInstances) {  if (cd!=null) {  txtArtist.setText(cd.getArtist().trim());  txtTitle.setText(cd.getTitle().trim());  txtSongs.setText(cd.getSongs().trim());  }  }  }  @Override  public void componentClosed() {  result.removeLookupListener(this);  }  … |

By implementing ***LookupListener*** interface we’ve enabled ***MyProperty*** to monitor state changes on other windows in application. Next step is to register other windows by implementing ***LookupAPI***. More information about ***Lookup*** or ***Lookup API*** is provided here:

<http://bits.netbeans.org/dev/javadoc/org-openide-util-lookup/org/openide/util/lookup/package-summary.html> , and:

<http://bits.netbeans.org/dev/javadoc/org-openide-util-lookup/org/openide/util/class-use/LookupListener.html>

Following code should be added in ***CDCatalogTopComponent*** class( ***CDCatalog***  module):

|  |
| --- |
| package org.netbeans.cdcatalog;  import java.util.ArrayList;  import java.util.logging.Logger;  import org.openide.util.LookupEvent;  import org.openide.util.NbBundle;  import org.openide.windows.TopComponent;  import org.openide.windows.WindowManager;  import org.netbeans.api.settings.ConvertAsProperties;  import org.netbeans.cdcatalog.domain.CD;  import org.openide.awt.ActionID;  import org.openide.awt.ActionReference;  import org.openide.util.Lookup.Result;  import org.openide.util.LookupListener;  import org.openide.util.Utilities;  @ConvertAsProperties(dtd = "-//org.netbeans.cdcatalog//CDCatalog//EN",  autostore = false)  @TopComponent.Description(preferredID = "CDCatalogTopComponent",  //iconBase="SET/PATH/TO/ICON/HERE",  persistenceType = TopComponent.PERSISTENCE\_ALWAYS)  @TopComponent.Registration(mode = "editor", openAtStartup = true)  @ActionID(category = "Window", id = "org.netbeans.cdcatalog.CDCatalogTopComponent")  @ActionReference(path = "Menu/Window" /\*, position = 333 \*/)  @TopComponent.OpenActionRegistration(displayName = "#CTL\_CDCatalogAction",  preferredID = "CDCatalogTopComponent")  public final class CDCatalogTopComponent extends TopComponent implements LookupListener{  …  Result<CD> result;  @Override  public void resultChanged(LookupEvent le) {  CD cd = null;  if (result.allInstances().iterator().hasNext()) {  cd = result.allInstances().iterator().next();  }  if (cd != null) {  txtTitle.setText(cd.getTitle());  txtArtist.setText(cd.getArtist());  }  }    @Override  public void componentOpened() {  result = Utilities.actionsGlobalContext().lookupResult(CD.class);  result.addLookupListener(this);  }  @Override  public void componentClosed() {  result.removeLookupListener(this);  }  … |

Next thing you need to change (or comment the old one and type new one instead ) is constructor in ***CDNode*** class ( ***CDViewer*** module):

|  |
| --- |
| public CDNode(CD bean) throws IntrospectionException {  super(bean, Children.LEAF, Lookups.singleton(bean));  setDisplayName(bean.getTitle());  setShortDescription(bean.getArtist());  } |

Start your application , add new CD using wizards , select a newly added CD from ***CDViewer window*** , and in both ***CDCatalog*** and ***MyProperties*** should fields considering CD data be filled:

