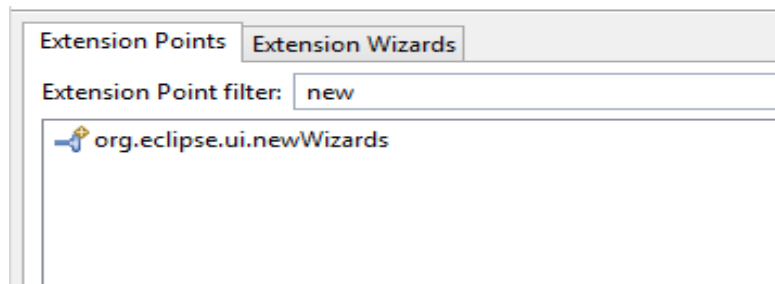


## Introduction to Eclipse Wizards

Assume that you already know how to create a plugin project. In this tutorial we will learn about creating New Wizards. Wizards are a collection of 1 to Many WizardPages. You will find Wizards in Eclipse under File > New, File > Import, File > Export

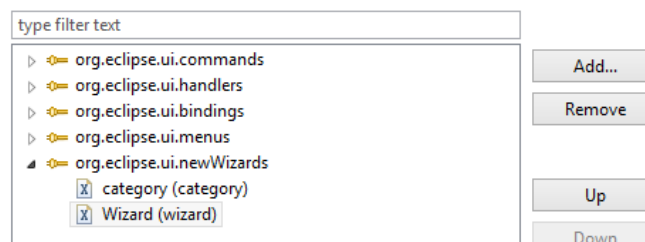
In plugin.xml, select the extensions tab and click on ADD and look for org.eclipse.ui.newWizards



On Right Click on the extension point, you will find Category and Wizard as an option.

Create a Category and provide its Name.

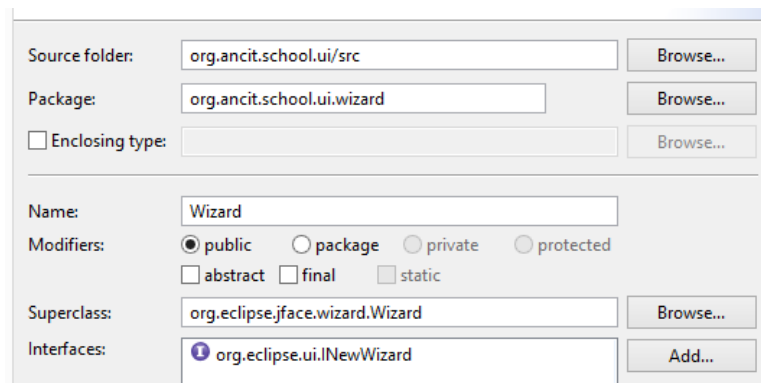
Create a Wizard and then provide a name to the Wizard



Click on the Class Hyperlink to create the Wizard. Provide a ClassName

The class should extend Wizard and implements INewWizard

Then Click finish.



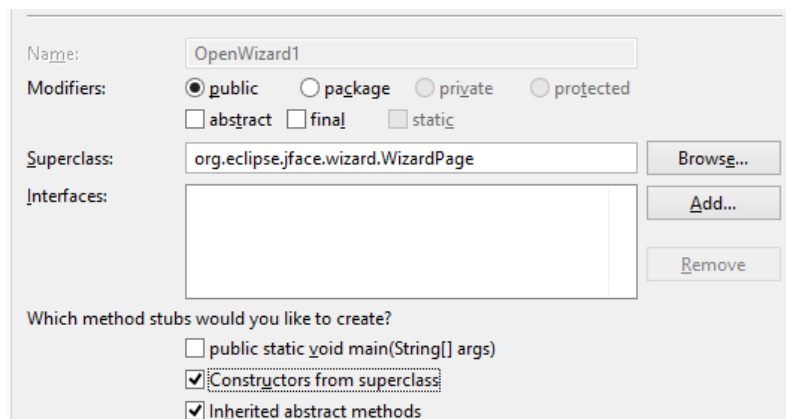
Wizard class would be created. In that class we need to Override the addPages()

To add a Page you would require a WizardPage.

You can use the following code to create instance for new page and add that page there.

```
@Override
public void addPages() {
    openWizard = new OpenWizard("page1");
    addPage(openWizard);
}
```

Create a new class which extends WizardPage.

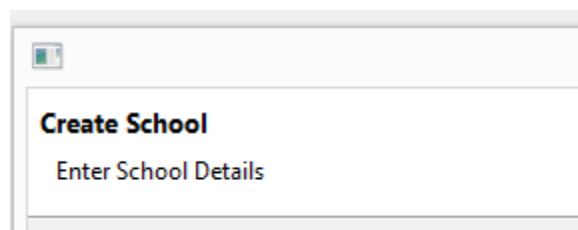


The screenshot shows the 'New Class' dialog in Eclipse. The 'Name' field contains 'OpenWizard1'. Under 'Modifiers', the 'public' radio button is selected. The 'Superclass' field is set to 'org.eclipse.jface.wizard.WizardPage'. In the 'Which method stubs would you like to create?' section, the checkboxes for 'Constructors from superclass' and 'Inherited abstract methods' are checked.

In the constructor of the WizardPage, set title and description for the wizard.

You can use the following code to give Title and description for the wizard.

```
public OpenWizard(String pageName) {
    super(pageName);
    setTitle("Create School");
    setDescription("Enter School Details");
}
```



In the WizardPage, you will find another method called as `createControl()`. This method is responsible for the UI for the WizardPage.

You can use the following code to give Label and Text field for the wizard.

```
@Override
public void createControl(Composite parent) {
    setPageComplete(false);
    composite = new Composite(parent, SWT.NONE);
    composite.setLayoutData(new GridData(SWT.FILL, SWT.FILL, true, true));
    composite.setLayout(new GridLayout(2, false));

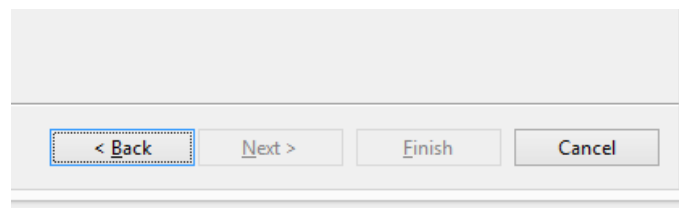
    lblname = new Label(composite, SWT.NONE);
    lblname.setLayoutData(new GridData(SWT.BEGINNING, SWT.CENTER, false, false));
    lblname.setText("Name");
    txtname = new Text(composite, SWT.SINGLE | SWT.LEAD | SWT.BORDER);
    txtname.setLayoutData(new GridData(SWT.FILL, SWT.CENTER, true, false));
    txtname.addModifyListener(this);

    lblid = new Label(composite, SWT.NONE);
    lblid.setLayoutData(new GridData(SWT.BEGINNING, SWT.CENTER, false, false));
    lblid.setText("Id");
    txtid = new Text(composite, SWT.SINGLE | SWT.LEAD | SWT.BORDER);
    txtid.setLayoutData(new GridData(SWT.FILL, SWT.CENTER, true, false));
    txtid.addModifyListener(this);

    setControl(composite);
}
```

To disable the finish button by default using this code inside `createControl()`

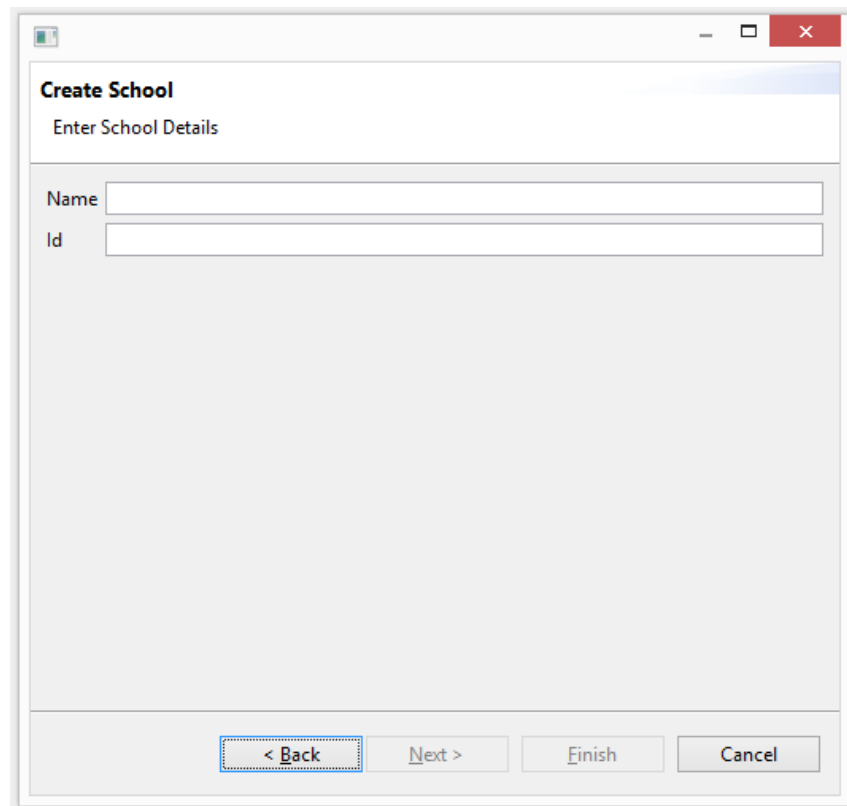
```
setPageComplete(false);
```



Now you may run the Project as an Eclipse Application, by selecting the project, right click on it and run as Eclipse Application.

In the new Instance of Eclipse, select File > New > Others and then you should be able to find your Wizard in this space. Select your Wizard and Click NEXT to see the below output

**output:**



**Create School**  
Enter School Details

Name

Id

< Back   Next >   Finish   Cancel

## **ABOUT ANCIT:**

ANCIT Consulting is an Eclipse Consulting Firm located in the "Silicon Valley of India", Bangalore. Offers professional Eclipse Support and Training for various Eclipse based Frameworks including RCP, EMF, GEF, GMF. Contact us on [annamalai@ancitconsulting.com](mailto:annamalai@ancitconsulting.com) to learn more about our services.