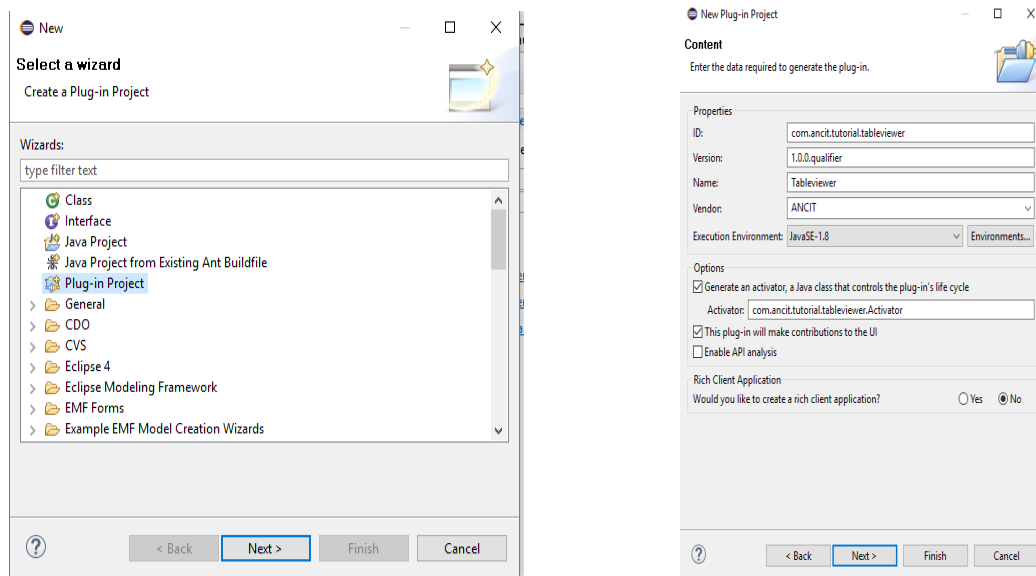


# Table Viewer

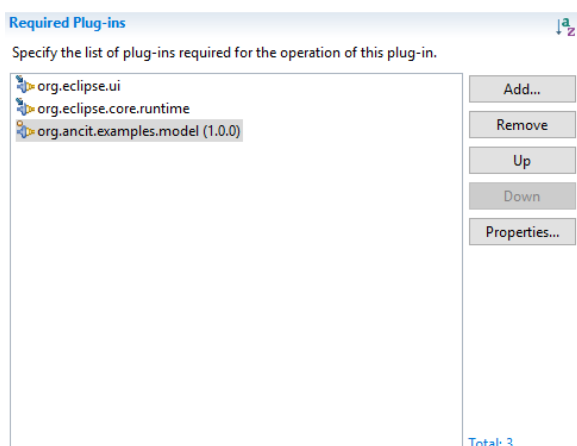
In this tutorial we concentrate on how to create a Table Viewer. Table Viewer is a View in Eclipse which shows items in a table.

## Creating the Table

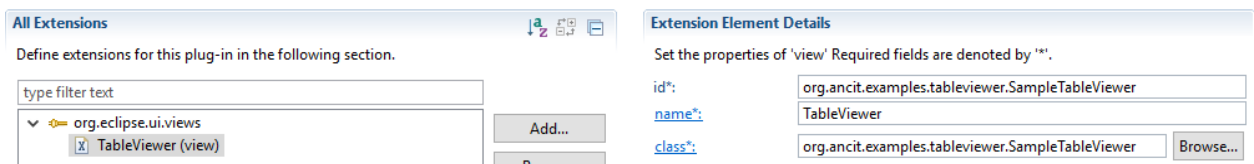
1. Create a new UI plugin project by File->New ->Plugin Project and Configure as in pictures given below. Complete the steps below and click finish.



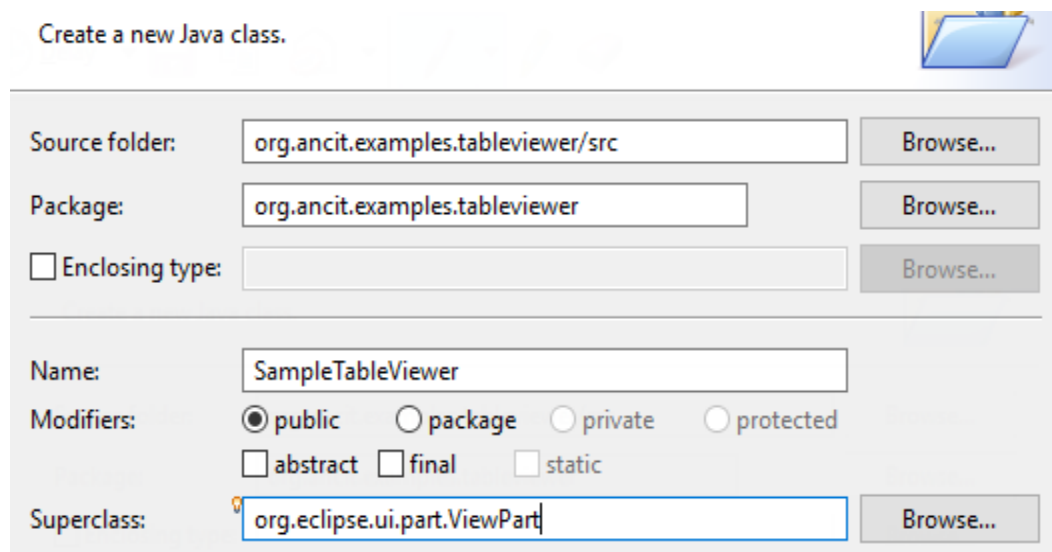
and open MANIFEST.MF, navigate to dependencies tab and add “org.eclispe.ui”, “org.eclipse.core.runtime” if not added and your model.



2. Contribute a new view by adding “org.eclipse.ui.views” extension in the extensions tab and create a new class “SampleTableViewer” extending “ViewPart. Also provide a name for the table viewer and give a unique id for your table viewer(Can be your class name).



Class creation is shown below with superclass.



3. Open the class “SampleTableViewer” and override the method “createPartControl and do the steps given below
  - Instantiate a new Composite and send the parent attribute.
  - Instantiate a new TableColumnLayout and set the created layout to the composite.
  - Instantiate a new TableViewer and send the composite to the constructor.
  - Create a private variable of type “Table” and get the table form the TableViewer using “getTable()” method and assign it to the private variable.

- Create new “TableViewColumn” as many columns needed for your data and send the “TableView” to the constructor. From the “TableViewColumn” get the column using “getColumn()” method and store it in a local variable of type “TableColumn”. Name the Columns using “setText()” method as you wish.

```
@Override
public void createPartControl(Composite parent) {

    Composite composite = new Composite(parent, SWT.NONE);
    TableColumnLayout tcl_composite = new TableColumnLayout();
    composite.setLayout(tcl_composite);

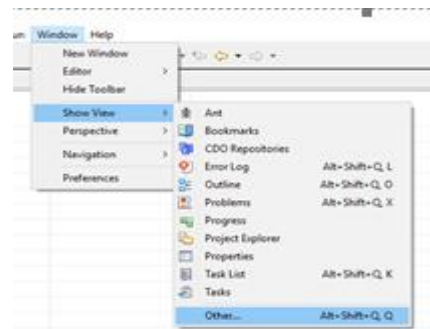
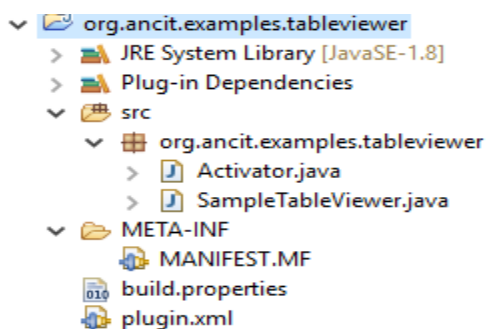
    TableView tableView = new TableView(composite, SWT.BORDER | SWT.FULL_SELECTION);
    table = tableView.getTable();
    table.setHeaderVisible(true);
    table.setLinesVisible(true);

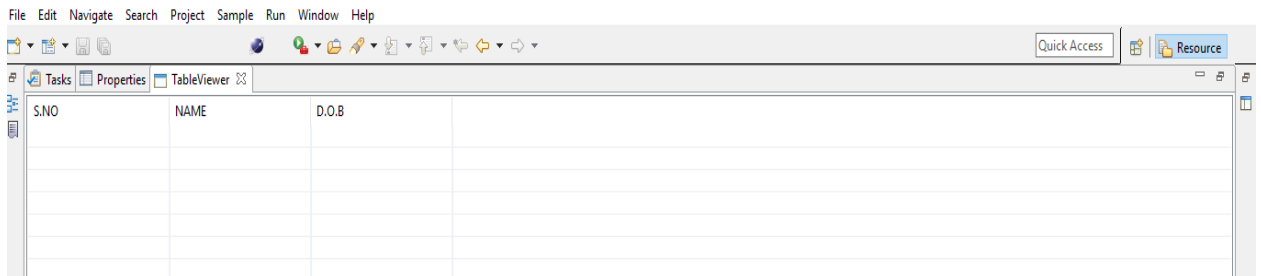
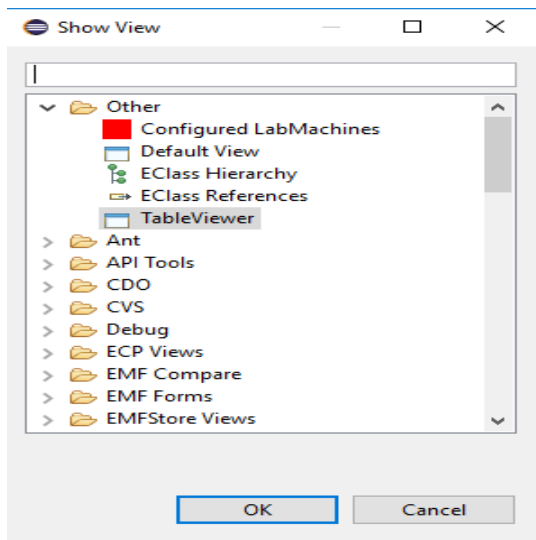
    TableViewColumn tableViewColumnSerialNumber = new TableViewColumn(tableView, SWT.NONE);
    TableColumn tableColumnSerialNumber = tableViewColumnSerialNumber.getColumn();
    tcl_composite.setColumnData(tableColumnSerialNumber, new ColumnPixelData(150, true, true));
    tableColumnSerialNumber.setText("S.NO");

    TableViewColumn tableViewColumnName = new TableViewColumn(tableView, SWT.NONE);
    TableColumn tableColumnName = tableViewColumnName.getColumn();
    tcl_composite.setColumnData(tableColumnName, new ColumnPixelData(150, true, true));
    tableColumnName.setText("NAME");

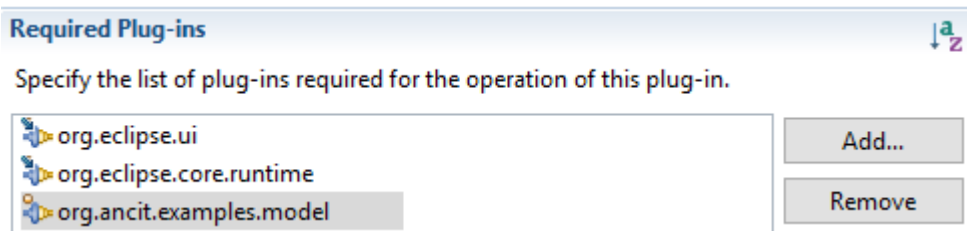
    TableViewColumn tableViewColumnDob = new TableViewColumn(tableView, SWT.NONE);
    TableColumn tableColumnDob = tableViewColumnDob.getColumn();
    tcl_composite.setColumnData(tableColumnDob, new ColumnPixelData(150, true, true));
    tableColumnDob.setText("D.O.B");
```

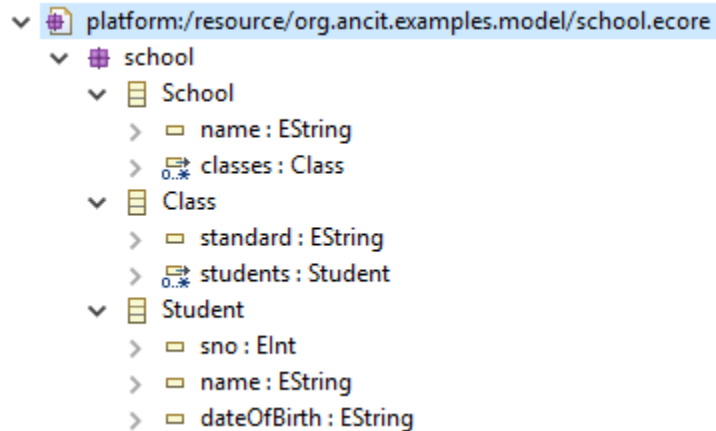
4. Now the project will look like the picture given below. Right Click on you project and run as eclipse application. When opened find “Window” in menu bar navigate to “Show View -> Other” and find “TableView”.





- Now to provide content for the table form a model go to the "MANIFEST.MF" and navigate to dependencies tab and add your model. (In my case its "org.ancit.examples.model") and the model is also shown below.





6. Now again open the class “SampleTableViewer” and set the content provider for the table.

```
tableViewer.setContentProvider(ArrayContentProvider.getInstance());
```

7. Then set the labelProvider and pass a new inner class “TableLabelProvider” that extends form “LabelProvider” and implements “ITableLabelProvider”.

```
tableViewer.setLabelProvider(new TableLabelProvider());
```

Create two methods returning Image and String.

```
public class SampleTableViewer extends ViewPart {
    private class TableLabelProvider extends LabelProvider implements ITableLabelProvider {
        public Image getColumnImage(Object element, int columnIndex) {
            return null;
        }
        public String getColumnText(Object element, int columnIndex) {
            return element.toString();
        }
    }
}
```

8. Now to set input for the table we call the method “setinput()” and pass the List to be shown in the table. First we set the contents to the model and pass the model which contains the data to the tableviewer as shown below.

```
School school = SchoolFactory.eINSTANCE.createSchool();
school.setName("School");
```

```
Class schoolClass = SchoolFactory.eINSTANCE.createClass();
schoolClass.setStandard("1");
school.getClasses().add(schoolClass);
```

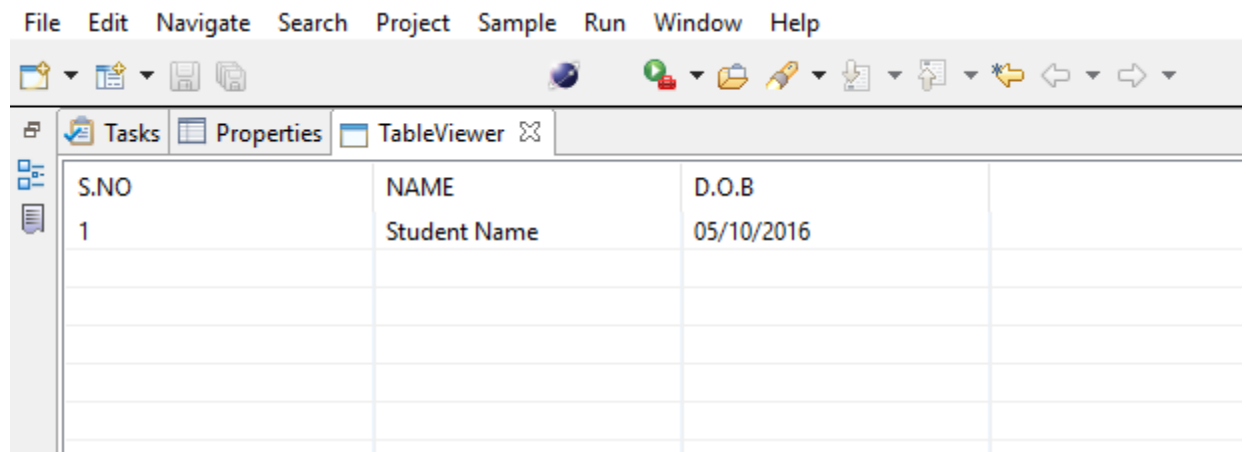
```
Student student1 = SchoolFactory.eINSTANCE.createStudent();
student1.setName("Student Name");
student1.setSno(1);
student1.setDateOfBirth("05/10/2016");
schoolClass.getStudents().add(student1);
```

```
tableViewer.setContentProvider(ArrayContentProvider.getInstance());
tableViewer.setLabelProvider(new TableLabelProvider());
tableViewer.setInput(schoolClass.getStudents());
```

9. Now on the inner class TableLabelProvider , in the method getColumnText check for the element type to be "Student" and type cast it to student and put a switch statement and return the string incoming. Now run the eclipse application and navigate to your view to find the data in the table.

```
public String getColumnText(Object element, int columnIndex) {  
    if (element instanceof Student) {  
        Student student = (Student) element;  
        switch (columnIndex) {  
            case 0:  
                return Integer.toString(student.getSno());  
            case 1:  
                return student.getName();  
            case 2:  
                return student.getDateOfBirth();  
            default:  
                break;  
        }  
    }  
    return null;  
}
```

The table will show the data set to the model.



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Navigate, Search, Project, Sample, Run, Window, and Help. Below the menu is a toolbar with various icons. The main workspace is divided into two panes. The left pane shows a project structure with a folder named 'Tasks' and a file named 'Properties'. The right pane shows a table view titled 'TableViewer'. The table has three columns: 'S.NO', 'NAME', and 'D.O.B'. The first row contains the values '1', 'Student Name', and '05/10/2016'. There are four empty rows below the first row.

S.NO	NAME	D.O.B
1	Student Name	05/10/2016

## 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.