

Value Help Pop-up Dialogs

In case you want to provide help in form of a pop-up dialog - based on a certain field - Application Designer offers a technique for implementing a value selection help:

- The FIELD control offers the property `popupmethod`. With this property, a method of the adapter class is called whenever the user requests a value help inside the field - by pressing F4 or F7 in the field or by double-clicking on the field.
- A pop-up dialog opens displaying possible data selections.

The value help pop-up dialogs are just normal pop-up dialogs which just have a dedicated purpose.

This chapter covers the following topics:

- Standard Method `openIdValueHelp`
 - Standard Method `openIdValueCombo`
-

Standard Method `openIdValueHelp`

Inherited from the `Adapter` class, there is a very simple way to provide a value help pop-up dialog. The method `openIdValueHelp` is implemented in a generic way and can be used as follows:

- In your adapter, implement a method `findValidValuesForXxx()`. Replace "Xxx" with the name of the property field.
- This method must return an array of `com.softwareag.cis.server.util.ValidValueLine` objects. This array contains pairs of IDs and values which are valid data options for the Xxx property.
- When requesting a value help for the corresponding field, a pop-up dialog displays the `ValidValueLine` objects which are passed back from your method. If the user selects an item in the pop-up dialog, the value is placed in the `setXXX` method of the property.

The following Java source shows an example:

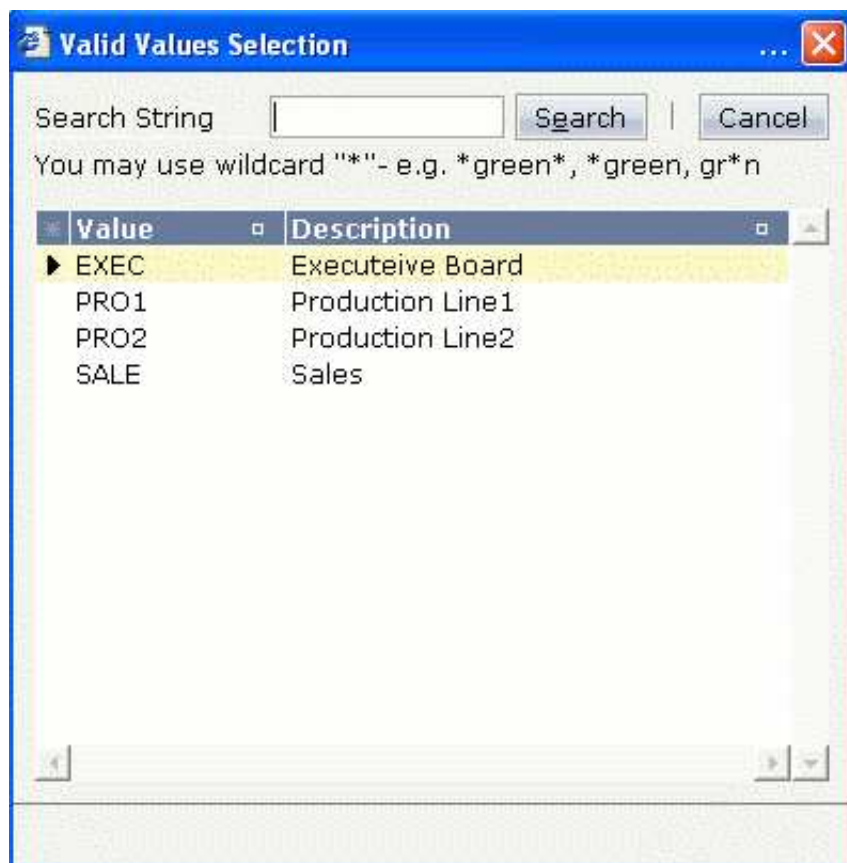
```
// property >department<
String m_department;
public String getDepartment() { return m_department; }
public void setDepartment(String value) { m_department = value; }

public ValidValueLine[] findValidValuesForDepartment()
{
    Vector v = new Vector();
    v.addElement(new ValidValueLine("EXEC", "Executive Board"));
    v.addElement(new ValidValueLine("PRO1", "Production Line1"));
    v.addElement(new ValidValueLine("PRO2", "Production Line2"));
    v.addElement(new ValidValueLine("SALE", "Sales"));
    ValidValueLine[] result = new ValidValueLine[v.size()];
    v.copyInto(result);
    return result;
}
```

The XML layout looks as follows:

```
<rowarea name="Field with Value Help">
  <itr>
    <label name="Department" width="100">
      </label>
    <field valueprop="department" width="150" popupmethod="openIdValueHelp">
      </field>
    </itr>
  </rowarea>
```

The result is a field which automatically opens a pop-up dialog when the user presses F4 or F7, or double-clicks on the field.



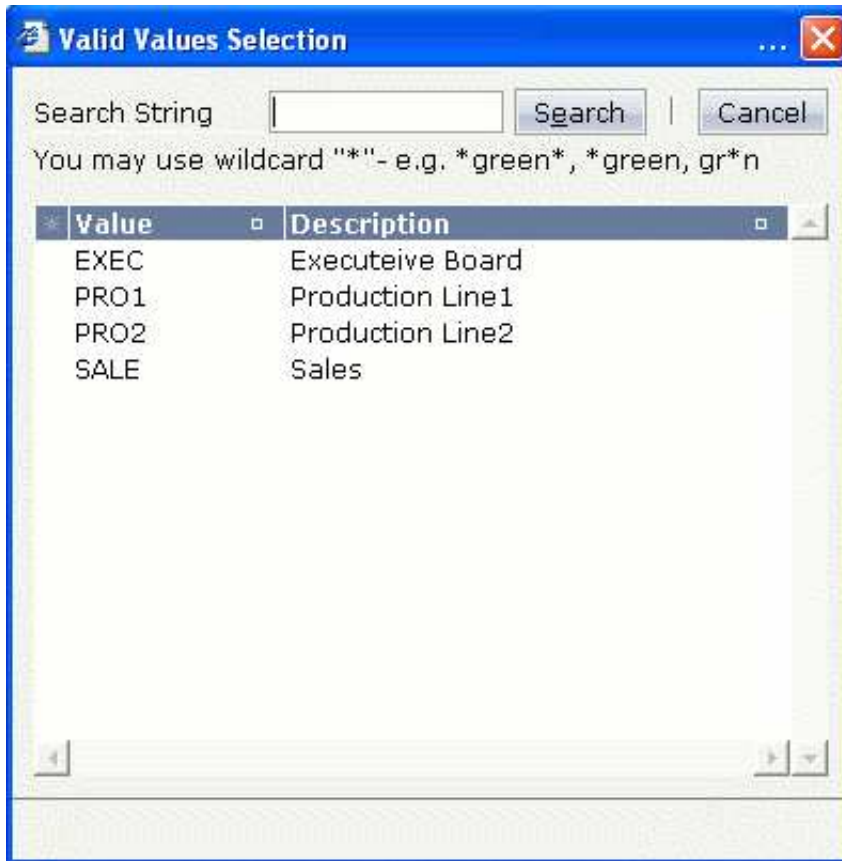
An additional feature available: instead of displaying pairs of ID and name, the dialog can display a list of IDs only. There is a constructor of the `ValidValueLine` class with only passing an ID to it:

```

public ValidValueLine[] findValidValuesForDepartment_02()
{
    Vector v = new Vector();
    v.addElement(new ValidValueLine("EXEC"));
    v.addElement(new ValidValueLine("PRO1"));
    v.addElement(new ValidValueLine("PRO2"));
    v.addElement(new ValidValueLine("SALE"));
    ValidValueLine[] result = new ValidValueLine[v.size()];
    v.copyInto(result);
    return result;
}

```

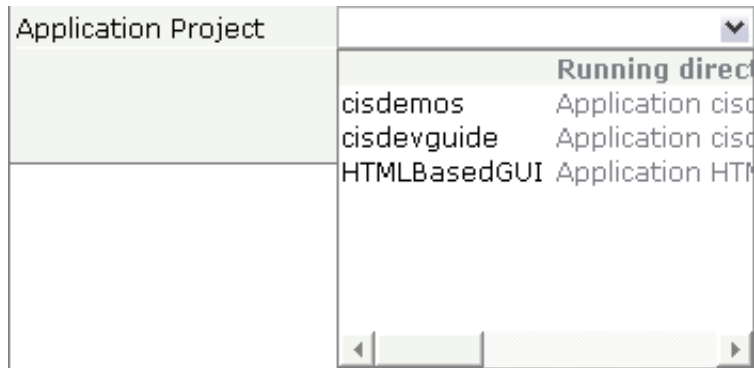
Now the pop-up dialog contains only a column containing the IDs:



Standard Method openIdValueCombo

See the description of the FIELD control for information on how to implement a valid value help with the openIdValueCombo method. This method does not open a pop-up but it open a combo-like selection.

The interface on the server side is exactly the same as for openIdValueHelp - just the rendering result is different:



Further information is provided in the description of the FIELD control.