

Pre-populate Form Select List from External Source

To pre-populate a single form select list from an external source, choose the select field and click on 'Edit'.

The screenshot shows a form titled "Parent Information" with a header bar containing buttons for "Add Section", "Add Column", "Edit Section", and "Delete Section". Below the header, there is a "Drag This Column" instruction and an "Edit Column" button. The form contains four fields: "First Name *" (text input), "Last Name *" (text input), "Email address *" (text input), and "Relationship Category *" (dropdown menu). The "Relationship Category" dropdown is currently set to "Mom" and has "Edit" and "Delete" buttons next to it.

Under the 'Choose Options Binder' menu, click on 'Bean Shell Form Binder' in the Options Binder select field.

The screenshot shows the "Property Editor" dialog box with a "Choose Options Binder" section. The breadcrumb trail is: "Edit Select Box > Choose Options Binder > Configure Bean Shell Form Binder > Choose Validator > Configure Validator > Workflow Variables". The "Options Binder" dropdown menu is open, showing three options: "Bean Shell Form Binder" (highlighted in blue), "Bean Shell Form Binder", and "Default Form Options Binder". At the bottom of the dialog, there are buttons for "< Prev", "Next >", "OK", and "Cancel".

Under 'Configure Bean Shell Form Binder', enter your Bean shell script to populate the select fields. For each row that you add to the FormRowSet, be sure to put the 'FormUtil.PROPERTY_VALUE' and 'FormUtil.PROPERTY_LABEL' fields in the FormRow that will be added to the FormRowSet.

Property Editor [close]

Configure Bean Shell Form Binder

Edit Select Box > Choose Options Binder > **Configure Bean Shell Form Binder** > Choose Validator > Configure Validator > Workflow Variables

Script *

```
import org.joget.apps.form.model.*;
import org.joget.apps.form.service.*;

public FormRowSet test() {
    FormRowSet f = new FormRowSet();
    f.setMultiRow(true);

    FormRow r1 = new FormRow();
    r1.put(FormUtil.PROPERTY_VALUE, "test1");
    r1.put(FormUtil.PROPERTY_LABEL, "Mom");
    f.add(r1);

    FormRow r2 = new FormRow();
    r2.put(FormUtil.PROPERTY_VALUE, "test2");
    r2.put(FormUtil.PROPERTY_LABEL, "Dad");
    f.add(r2);

    return f;
}
```

< Prev Next > OK Cancel

```
import org.joget.apps.form.model.*;
import org.joget.apps.form.service.*;

public FormRowSet test() {
    FormRowSet f = new FormRowSet();
    f.setMultiRow(true);

    FormRow r1 = new FormRow();
    r1.put(FormUtil.PROPERTY_VALUE, "test1");
    r1.put(FormUtil.PROPERTY_LABEL, "tester1");
    f.add(r1);

    FormRow r2 = new FormRow();
    r2.put(FormUtil.PROPERTY_VALUE, "test2");
    r2.put(FormUtil.PROPERTY_LABEL, "tester2");
    f.add(r2);

    return f;
}

return test();
```

Of course, you would want to populate the values for PROPERTY_LABEL and PROPERTY_VALUE from your external datasource. For more information on how to connect to a database source, see the example code within 'Pre-populate Form Fields with Data from External Source' or this example on the Joget Blog.

Related Elements

- Bean Shell Form Binder
- Select Box