

Validate Field Based On Another Field

In this article, we will show how to make a field mandatory based on the value of **another** field.

The screenshot shows a web application interface with a blue header bar containing a menu icon and the text 'VALIDATION BY FIELD VALUE'. Below the header, a breadcrumb trail reads 'Home > Manage Request'. The main content area is titled 'Request' and contains three fields: 'Name', 'Type', and 'Attachment'. The 'Type' field is a dropdown menu with two options: 'Without Evidence' and 'With Evidence'. The 'With Evidence' option is selected and highlighted in green. The 'Attachment' field is a file upload area with a dashed border and the text 'Drop files here or click to upload.' Below this text, two files are listed: 'logs.zip' and 'wrapperPlugin.png', each with a 'remove' link. At the bottom of the form, there are two orange buttons: 'Save' and 'Back'.

Figure 1: When dropdown "Type" is set to "With Evidence", "Attachment" becomes mandatory field.

There are 2 methods, first is by making use of [Section Visibility in Section properties](#), and secondly, by using beanshell script in one of the fields.

Method 1: Using Multiple Sections and Section Visibility

Place the file upload in its own section, configure the section visibility to only show such that when the dropdown value is "With Evidence".

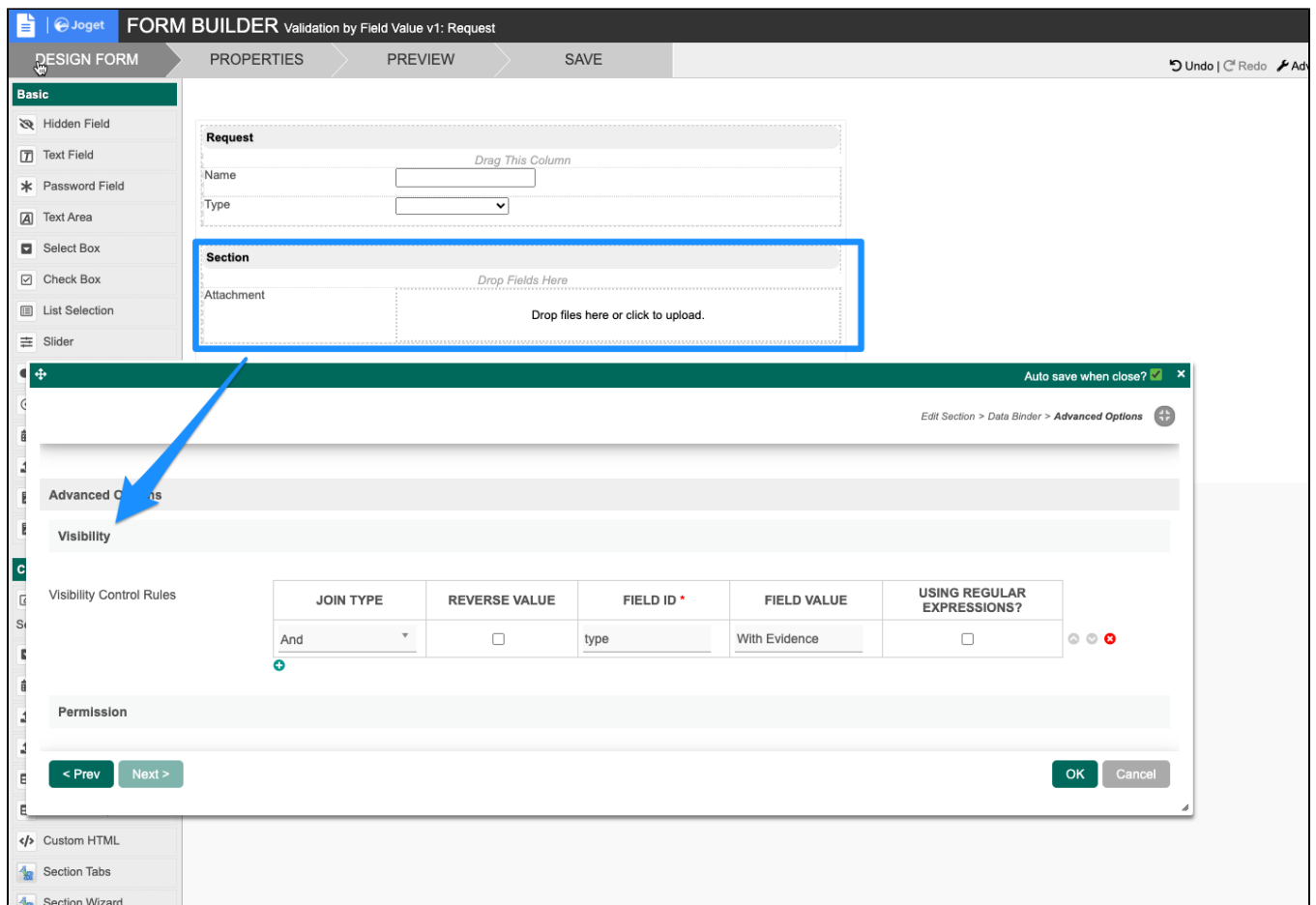


Figure 2: File Upload in its own section

Secondly, set the file upload's validator to mandatory.

Method 2: Using Bean Shell Validation Script

In this method, there's no need to have separate sections as validation will take place in the script itself, instead of relying on section visibility's logic.

We will place the script as "Bean Shell Validator" in one of the field. The following sample script is to be put in File Upload's Bean Shell Validator.

In this script, we will obtain the value selected from the dropdown, then based on its value, we will choose to/not to enforce the need for file upload.

Bean Shell Validator in File Upload

```
import java.util.Arrays;
import org.joget.apps.app.service.AppUtil;
import org.joget.apps.form.model.Element;
import org.joget.apps.form.model.Form;
import org.joget.apps.form.model.FormData;
import org.joget.apps.form.service.FormUtil;

public boolean validate(Element element, FormData formData, String[] fileUploadValues) {
    boolean result = true;

    //get dropdown select box value
    String field1Id = "type";
    Form form = FormUtil.findRootForm(element);
    Element field1 = FormUtil.findElement(field1Id, form, formData);

    if (field1 != null) {
        //get value of field 1
        String[] field1Values = FormUtil.getElementPropertyValues(field1, formData);

        if(field1Values[0].equalsIgnoreCase("With Evidence") && fileUploadValues[0].isEmpty() ){
            String id = FormUtil.getElementParameterName(element);
            formData.addFormError(id, "Attachment is required when type is 'with evidence'");
            result = false;
        }
    } else {
        //ignore if the dropdown does not exist
    }

    return result;
}

//call validate method with injected variable
return validate(element, formData, values);
```

You may check out the demo app below for this method.



APP_validationB...11105071511.jwa