

# Table of Contents

If you need native checkboxes that are automatically bound to your model, instead of fake "[ChoiceCellEditor checkboxes](#)", you can use `UICheckBoxCellEditor`.

However, be aware that such native checkboxes only have two states and two values (selected or unselected, e.g., Y or N). You can't set a custom value for null/empty, but this would be possible with `UICheckCellEditor`.

The advantage of native checkboxes is that they will fit your look and feel, but this would be also possible with custom images and choice cell editors.

### Usage

The syntax is similar to [ChoiceCellEditors](#), e.g.:

```
rowDef.getColumnDefinition("CHOOSE").getDataType().  
setCellEditor(ApplicationUtil.YESNO_CHECKBOX);
```

The `ApplicationUtil` defines default cell editors. The `YESNO_CHECKBOX` is defined as:

```
new UICheckBoxCellEditor("Y", "N");
```

It's also possible to define default checkbox cell editors for an application:

```
UICheckBoxCellEditor.addDefaultCheckBoxCellEditor(ApplicationUtil.YESNO_CHEC  
KBOX);
```

From:

<https://doc.sibvisions.com/> - **Documentation**

Permanent link:

<https://doc.sibvisions.com/jvx/client/gui/checkbox>



Last update: **2020/06/26 11:53**