

# ConvertBoolean

## Introduction

It is often necessary to display a Boolean (true/false) on a web page using custom Strings to represent the true and false values. For example, you might want to display "Yes" instead of "true" and "No" instead of "false".

This `<s:convertBoolean>` tag, which uses the [BooleanConverter](#) internally, can be used to accomplish such formatting. Its usage is analogous to the use of `<f:convertDateTime>` to format a `java.util.Date`.

## Example

```
<h:outputText value="#{backingBean.customer.enjoysJazz}">
  <s:convertBoolean trueValue="Yes" falseValue="No" />
</h:outputText>
```

You can also use a managed bean for the Yes and No values

```
<h:outputText value="#{backingBean.customer.enjoysJazz}">
  <s:convertBoolean trueValue="#{Message.Yes}" falseValue="#Message.No" />
</h:outputText>
```

Also, it might be quite useful to use it inside a `selectOneRadio`, example:

```
<h:selectOneRadio value="#{backingBean.customer.enjoysJazz}">
  <f:selectItem itemLabel="#{Message.Yes}" itemValue="1" />
  <f:selectItem itemLabel="#{Message.No}" itemValue="0" />
  <s:convertBoolean trueValue="1" falseValue="0" />
</h:selectOneRadio>
```

and in the managed bean...

```
class Customer
{
    private Boolean enjoysJazz;
    // getters and setters here
}
```

The `"backingBean.customer.enjoysJazz"` value binding expression above represents a `java.lang.Boolean` value.