

UsingPortletUtil

My application needs to know if it is running as a portlet.

[MyFaces](#) has a static utility class that can let your application know if it is running as a portlet or not. This class is `org.apache.myfaces.portlet.PortletUtil`. It has two static methods:

```
public static boolean isRenderResponse(FacesContext facesContext)
public static boolean isPortletRequest(FacesContext facesContext)
```

You will probably never use `isRenderResponse()`. It is used by the [MyFaces](#) implementation.

`isPortletRequest()` simply checks for the existence of a flag that is set by [MyFacesGenericPortlet](#). It does not use any portlet api for this so you don't need the portlet jar in your classpath if you are just running as a servlet.

A common use case for [PortletUtil](#) will be to enable or disable a JSF component based on whether or not you are running in a portlet. To do this, you will wrap the [PortletUtil](#) in a managed bean that is placed in application scope. The following example assumes that you want to display some text only in a portlet environment:

First, create the wrapper bean

```
import javax.faces.context.FacesContext;

import org.apache.myfaces.portlet.PortletUtil;

public class PortletUtilBean {

    public boolean isPortletRequest() {
        FacesContext facesContext = FacesContext.getCurrentInstance();
        return PortletUtil.isPortletRequest(facesContext);
    }
}
```

Then, register it in faces-config.xml

```
<managed-bean>
    <managed-bean-name>PortletUtilBean</managed-bean-name>
    <managed-bean-class>com.foo.mybean.PortletUtilBean</managed-bean-class>
    <managed-bean-scope>application</managed-bean-scope>
</managed-bean>
```

Then, use the managed bean as a binding in the rendered attribute of your JSF component

```
<h:outputText escape="true" rendered="#{PortletUtilBean.portletRequest}" style="color: Red;"
    value="This text will only appear in portlets." />
```