


# How do we access static parameters from an Action

Static (or pre-defined) parameters can be set to a Map property or to individual JavaBean properties.

- Define the parameters to be set by adding the name(s) and value(s) to the action mapping element (in the application's `struts.xml`).

```
<action name="myAction" class=" MyAction">
  <param name="myStaticParam1">myStaticValue1</param>
  <param name="myStaticParam2">myStaticValue2</param>
  <param name="myStaticParam3">myStaticValue3</param>
</action>
```

- Ensure that `staticParams` Interceptor is included in the Action's stack.
  -  *The default stack already includes `staticParams`.*
- Edit the Action so that it implements the `Parameterizable` interface.

## Map property

- Ensure the Action defines a `setParams(Map)` method.
- The `staticParams` Interceptor will set the defined values to the Map, using the name as the entry key.

key	value
myStaticParam1	myStaticValue1
myStaticParam2	myStaticValue2
myStaticParam3	myStaticValue3

## JavaBean properties

- Ensure that the Action defines JavaBean properties corresponding to the `param` elements in the action mapping.
- The `staticParams` Interceptor will set the defined values to each JavaBean property that corresponds to a `param` element.

```
public String getMyStaticParam1()
public void setMyStaticParam1(String myStaticParam1)

public String getMyStaticParam2()
public void setMyStaticParam2(String myStaticParam2)

public String getMyStaticParam3()
public void setMyStaticParam3(String myStaticParam3)
```

@see `com.opensymphony.xwork.interceptor.StaticParametersInterceptor`  
@see `com.opensymphony.xwork.config.entities.Parameterizable`