

S-PLUS JavaServer Pages

Tag Reference

S-PLUS JavaServer Pages (S+JSP) technology provides a way to embed S-PLUS code in a Web page. When a user requests the page, the Web server sends the S-PLUS code to S-PLUS Server to be run, and embeds the resulting output, results and graphs in the copy of the Web page it sends to the user.

S+JSP consists of a number of custom elements (also called tags) to use in a Web page. These elements set off the S-PLUS code and indicate what output is expected from S-PLUS and how it should be included in the page. For example, an `<splus:graphlet>` element marks S-PLUS code that generates a Graphlet, which the element adds to the Web page.

S+JSP represents a new paradigm for developing Web applications that use S-PLUS to analyze data and produce graphs on demand. This new paradigm is an alternative to the one used by S+Server for Windows and a number of Web applications for S-PLUS Server for UNIX. The hope is that S+JSP will prove easier to learn and that S+JSP code will prove quicker to write and simpler to debug. Preliminary results from people who have tried S+JSP are quite encouraging.

This document describes each S+JSP custom element in turn. After a brief description of the element, it gives an example using the element. The attributes for each element are summarized in a table. The table gives the name and a description of each attribute, and tells the attribute's type, whether the attribute is required, and whether it evaluates embedded elements in its value at run time.

Page Directives

A page that uses S+JSP needs two lines at the top of the page:

```
<%@ page language="java" contentType="text/html" %>
<%@ taglib uri="/sjsp" prefix="splus" %>
```

The first of these lines declares that the page uses JavaServer Pages technology. The second line declares that the page uses S+JSP technology. (Depending on your setup, the second line might vary a bit.)

A page can have other page directives as well. For example, the line

```
<%@ page errorPage="/error.jsp" %>
```

forwards the request to the page error.jsp if an error occurs. For more information on page directives, see any good book on JavaServer Pages.

<plus:connect>

The <plus:connect> element governs connections with S-PLUS. The start tag establishes an S-PLUS connection and the end tag releases the connection. All other plus tags must appear in the body of an <plus:connect> element; otherwise they will not be able to access an S-PLUS connection and an error will result.

The <plus:connect> tag implements a fault tolerance mechanism by default. Should the connection to S-PLUS fail due to a problem with the network or with the remote machine running S-PLUS, the tag will establish a new S-PLUS connection and will rerun everything in its body from the beginning. You can turn off the fault tolerance mechanism by setting the optional `faulttolerant` attribute to `false` or `no`.

Example:

```
<plus:connect>
:
:
<%--
  All other S-PLUS elements go here
  (together with HTML to format them)
--%>
:
:
</plus:connect>
```

Attribute:

The <plus:connect> element admits a single attribute, shown in the table below.

Name	Info	Description
faulttolerant	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	Whether to re-evaluate the body with a new connection if the previous attempt fails due to an RMI error. If this attribute is missing, the fault tolerance mechanism is employed. This method sets the flag to true unless the attribute has the value <code>false</code> or <code>no</code> .

`<splus:graphlet>`

The `<splus:graphlet>` element produces a Graphlet file and displays it on the Web page. The body of this element must be S-PLUS code that creates a graph. This code must not open or close graphics devices, as the element will take care of this automatically. If the code creates more than one graph, all the graphs will appear as pages in the Graphlet.

The `<splus:graphlet>` adds an HTML `<applet>` element to the Web page to display the Graphlet. To add additional `<param>` tags to the body of this element, for example to turn off the Graphlet's page tabs, use the `<splus:param>` element.

Example:

This example adds a two-page Graphlet to the Web page. The graphs are cluster trees based on data giving the percentage of people in the 50 states of the USA that voted Republican in the presidential elections between 1856 and 1956. The first page uses the agnes algorithm to cluster the states; the second page uses the diana algorithm.

```
<splus:graphlet width="480" height="440">
  plot(agnes(votes.repub), which = 2)
  java.set.page.title("agnes")
  plot(diana(votes.repub), which = 2)
  java.set.page.title("diana")
</splus:graphlet>
```

Attributes:

The `height` and `width` attributes are required. All other attributes are optional. The `colorscheme` attribute is passed to S-PLUS. All others are copied to the HTML `<applet>` element that displays the image. For more information about these attributes, read about the HTML `<applet>` element in any good book on HTML.

Name	Info	Description
height	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> No	The height of the Graphlet in pixels.
width	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> No	The width of the Graphlet in pixels.
colorscheme	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The name in S-PLUS for a color scheme for the Graphlet. If this attribute is missing, S-PLUS uses <code>java.colorscheme.default</code> .
name	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	A name to identify this Graphlet on the Web page. (More precisely, the name identifies the applet that is displaying this Graphlet.) JavaScript code can refer to the Graphlet by this name in order to get the list of the tags of the selected regions.
align	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The placement of the Graphlet with respect to surrounding text on the page. This attribute is deprecated in HTML.
alt	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The alternative text. A browser might display this text while the Graphlet's applet is loading, and in place of the applet if it does not load it for some reason.
hspace	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The horizontal space separating the Graphlet from other items on the page.
vspace	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The vertical space separating the Graphlet from other items on the page.

Static Graphlets

The `staticFile` attribute tells the `<splus:graphlet>` element to display a previously generated Graphlet from the specified file without calling S-PLUS. In this case, the element simply adds the appropriate HTML `<applet>` tag to the output page. The `<splus:graphlet>` element itself still needs to be included within an `<splus:connect>` element, even though it normally does not try to call S-PLUS.

However the `<splus:graphlet>` element does check to see whether the specified Graphlet file actually exists, and if not it attempts to call S-PLUS to generate it. This behavior is useful in some cases to initially generate the Graphlet and install it in the right place in the Web application.

Example:

This example adds to the Web page an already existing Graphlet in the file `StaticGraph.spj`. The element has no body containing S-PLUS code as it will never call S-PLUS to generate the Graphlet.

```
<splus:graphlet width="400" height="300" staticFile="StaticGraph" />
```

Attributes:

The table below summarizes the attributes useful for static Graphlets.

Name	Info	Description
staticFile	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The name of a previous generated Graphlet file to add to the output page. If the file exists, the tag does not call S-PLUS; it simply outputs HTML to display the Graphlet. If the file doesn't exist, the tag attempts to call S-PLUS to generate it.
codebase	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The directory where the Graphlet file is stored. A copy of spjgraph.jar must be in this directory. If this attribute is missing, the element will use the standard code base directory for this application. This attribute is useful primarily for static Graphlets.

<splplus:img>

The <splplus:img> element creates a graph in S-PLUS and places it on the Web page.

The body of an <splplus:img> element must be a set of S-PLUS commands that create a graph. The commands must not open or close graphics devices, as the element will take care of this automatically. If the commands create more than one graph, only the last one will appear on the Web page.

Consider using Graphlets (generated with the <splplus:graphlet> tag) rather than images generated with this tag. Graphlet files are considerably smaller than JPEG files, unless the graph is extremely complicated. Graphlets also allow the user to pan and zoom, and they can have active regions with labels and links to other pages.

Required: An X11 Server

To generate a JPEG, whether for an <splplus:img> element or otherwise, S-PLUS needs access to an X11 server. If your system isn't configured to provide an X11 server, you will get an error when you try to use this tag. The X11 server needs to be running on the computer set as the X Display when S-PLUS Server started the S-PLUS session. By default this computer will be the one running the Web server, but you can change that using the <display> element in the SplplusConnections.xml file, or by providing an X Display value when establishing a connection with the AddConnection.jsp page. The X11 server needs to act as a host for the computer running S-PLUS; use the `xhost` command to make this true. (The command `xhost +` opens up the X11 server to all clients.)

You can use a virtual frame buffer as an X11 server. To do this, use the command

```
/usr/X11R6/bin/Xvfb :1 -screen 0 800x600x24 &
```

on the computer running S-PLUS, and use `:1.0` as the X Display value. X11R6 is part of the standard RedHat distribution. Source code is available from <http://www.x.org>, and precompiled binaries for many platforms can be found online.

Example:

This example shows a hexbin plot for earthquake locations in the San Francisco Bay area for 1962 to 1981.

```
<splplus:img width="400" height="300"
  colorscheme="java.colorscheme.trellis.black.on.white">
  plot(
    hexbin(quakes.bay$longitude, quakes.bay$latitude),
    col.regions = 80:15,
    at = c(0, 10, 20, 30, 40, 50, 150))
</splplus:img>
```

Attributes:

The `<splus:img>` element takes three types of attributes. Attributes specifying the size of the image are sent to S-PLUS and are also included in the `` element places on the Web page. Attributes governing how a graph is to be generated are sent to S-PLUS, and attributes governing how it is to be displayed are added to the `` element.

The `width` and `height` attributes are sent to S-PLUS to govern the size of the generated image. They are also included in the HTML `` element to ensure that the image is rendered at the same size. Their use is recommended; if no width and height are specified, S-PLUS renders a rather large image. The table below gives their properties.

Name	Info	Description
height	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The height of the image in pixels.
width	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The width of the image in pixels.

The optional attributes in the next table are sent to S-PLUS to govern the type of image it generates. They are included in the call to `java.graph` that opens the graphics device. For more information about these attributes, see the S-PLUS Help file for `java.graph`.

Name	Info	Description
format	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The image file format. Typically the value will be JPEG or PNG (an alternative to GIF). S-PLUS can also generate BMP and PMN files. Internet Explorer can display BMP files, but Netscape cannot. Neither browser can display PMN files. If this attribute is missing, S-PLUS generates a JPEG file.
colorscheme	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The name in S-PLUS for a color scheme for the graph. If this attribute is missing, S-PLUS uses <code>java.colorscheme.default</code> .
pngcolor	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The color map for PNG output. This attribute is ignored if the <code>format</code> attribute is not set to PNG. If this attribute is missing, S-PLUS generates a color graph.
quality	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The JPEG quality. Valid values range from 0.0 (lowest) to 1.0 (highest). If this attribute is missing, S-PLUS uses a quality value of 1.0. This attribute is ignored if the <code>format</code> attribute changes to file type to anything other than JPEG.

The optional attributes in the following table determine how the image is displayed on the Web page. They are copied to the HTML `` element that displays the image. For more information about these attributes, read about the HTML `` element in any good book on HTML.

Name	Info	Description
align	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The placement of the image with respect to surrounding text on the page. This attribute is deprecated in HTML.
alt	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The alternative text. A browser might display this text while it downloads the image, or in place of the image if it does not download it for some reason.
border	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The border size in pixels.
hspace	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The horizontal space separating the image from other items on the page.
ismap	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The ismap attribute.
usemap	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The usemap attribute.
vspace	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The vertical space separating the image from other items on the page.

Static Images

The `staticFile` attribute tells the `<splus:img>` element to display a previously generated image from the specified file without calling S-PLUS. In this case, the element simply adds the appropriate HTML `` tag to the output page. The `<splus:img>` element itself still needs to be included within an `<splus:connect>` element, even though it normally does not try to call S-PLUS.

However the `<splus:img>` element does check to see whether the specified image file actually exists, and if not it attempts to call S-PLUS to generate it. This behavior is useful in some cases to initially generate the image and install it in the right place in the Web application.

Example:

This example adds to the Web page an already existing image in the file `StaticGraph.png`.

```
<splus:img width="400" height="300" staticFile="StaticGraph" format="PNG" >
plot(1:10, 1:10)
</splus:img>
```

Attributes:

The table below summarizes the attributes useful for static images.

Name	Info	Description
staticFile	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The name of a previous generated image file to add to the output page. If the file exists, the tag does not call S-PLUS; it simply outputs HTML to display the image. If the file doesn't exist, the tag attempts to call S-PLUS to generate it.

<plus:output>

The <plus:output> element evaluates an S-PLUS expression and adds the S-PLUS output to the Web page. The output is the text that a command line session of S-PLUS would print after executing the command.

The <plus:output> element provides two ways to specify the S-PLUS expression. The first way is to use the **expr** attribute in an element with no body. The second way is to use a begin and an end tag, with the S-PLUS expression as the body, as shown in the example below. The second option is handy when you use an <plus:sarg> element for an argument in the expression, as at least some containers will not interpret this tag when used in an **expr** attribute.

Example:

The example prints the output from a two-sided *t*-test. The data for this example give the weight gains (in grams) for rats fed high- and low-protein diets respectively. The <plus:output> element is embedded in a <pre> element so the text won't be reformatted.

```
<pre>
<plus:output>
  t.test(
    c(134, 146, 104, 119, 124, 161, 107, 83, 113, 129, 97, 123),
    c(70, 118, 101, 85, 107, 132, 94))
</plus:output>
</pre>
```

The example for the <plus:result> tag shows another way to display the same information.

Attribute:

The <plus:output> element takes a single attribute, described in the table below.

Name	Info	Description
expr	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The S-PLUS expression to evaluate. If this attribute is missing, the element uses its body as the S-PLUS expression to evaluate.

<spplus:param>

The <spplus:param> element, when used within the body of an <spplus:graphlet> element, sets a parameter governing how the applet displays the Graphlet. The <spplus:param> element works by inserting a corresponding <param> element in the body of the <applet> element that the <spplus:graphlet> element adds to the Web page.

The chapter about Graphlets in the *S-PLUS Server Programmer's Guide* describes the valid parameters. As of this writing, the valid parameter names are as follows:

```
spjgraph.mouse.position
spjgraph.mouse.position.checkbox
spjgraph.active.regions
spjgraph.active.regions.checkbox
spjgraph.rect.button
spjgraph.resize.buttons
spjgraph.options.buttons
spjgraph.help.button
spjgraph.tabs
spjgraph.select.button
```

Example:

This example is a variation of the example for the <spplus:graphlet> element. It displays a Graphlet with only a single page. The body of the <spplus:graphlet> element includes an <spplus:param> element that turns off the page tab that would otherwise appear at the bottom of the Graphlet.

```
<spplus:graphlet width="480" height="440">
  <spplus:param name="spjgraph.tabs" value="off" />
  plot(agnes(votes.repub), which = 2)
</spplus:graphlet>
```

Attributes:

The <spplus:param> element requires two attributes, described in the table below.

Name	Info	Description
name	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> No	The parameter name.
value	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> No	The parameter value.

<plus:result>

The <plus:result> element evaluates an S-PLUS expression and attempts to add the results from S-PLUS to the Web page. A result from S-PLUS can be any S-PLUS object. This element will add it to the page if it is a numeric, logical, or character string value, or a vector of character strings. Otherwise this element will add nothing to the page.

The <plus:result> element provides two ways to specify the S-PLUS expression. The first way is to use the **expr** attribute in an element with no body, as shown in the first example below. The second way is to use a begin and an end tag, with the S-PLUS expression as the body, as shown in the second example below. The second option is handy when you use an <plus:sarg> element for an argument in the expression, as at least some containers will not interpret this tag when used in an **expr** attribute.

Examples:

The code below shows two examples. The first example obtains a single real number from S-PLUS, namely the value of pi. The second example shows how to generate an HTML table in S-PLUS and include it in the output. This example displays the results of a two-sided *t*-test. The data for this example give the weight gains (in grams) for rats fed high- and low-protein diets respectively.

The value of pi is <plus:result expr="pi" />.

```
<p>
<plus:result>
  html.table(unlist(t.test(
    c(134, 146, 104, 119, 124, 161, 107, 83, 113, 129, 97, 123),
    c(70, 118, 101, 85, 107, 132, 94))))
</plus:result>
```

The example for the <plus:output> tag shows another way to display the same *t*-test information.

Attribute:

The <plus:result> element takes a single attribute, described in the table below.

Name	Info	Description
expr	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The S-PLUS expression to evaluate. If this attribute is missing, the element uses its body as the S-PLUS expression to evaluate.

`<splus:sarg>`

The `<splus:sarg>` element finds a parameter passed in by the calling page and formats it as an argument for an S-PLUS function call. If the request parameter doesn't exist, then this element adds nothing to the Web page.

The `prepend` and `append` attributes allow you to add text at the beginning and end of the request parameter value before sending it to S-PLUS as an argument. For example, if the request parameter value is a list of tags of selected Graphlet regions such as "21,22,23,24", you can prepend "c(" and append ")" to get the vector "c(21,22,23,24)" to pass to S-PLUS.

A request parameter can have multiple values. (For example, if several checkboxes or other controls on the calling page have the same name, all of their values are passed in a single request parameter.) The `<splus:sarg>` element formats multiple values as a single S-PLUS vector. For example if the values are 1, 2, and 3, the `<splus:sarg>` element sends to S-PLUS the single value `c(1,2,3)`. Any text in the `prepend` and `append` attributes is prepended and appended to each value.

Example:

This example is similar to the t-test example for the `<splus:output>` tag, except in this case we allow the calling page to pass in a request parameter named `alternative` specifying the alternative hypothesis. The value of this variable should be `greater`, `less` or `two.sided`, or the initial letter of one of these. If the calling page passes in such a parameter, the `<splus:sarg>` element passes it along in the call to the S-PLUS `t.test` function. Otherwise, the `<splus:sarg>` element adds nothing to the page, and S-PLUS will use the default value (`two.sided`).

```
<pre>
<splus:output>
  t.test(
    c(134, 146, 104, 119, 124, 161, 107, 83, 113, 129, 97, 123),
    c(70, 118, 101, 85, 107, 132, 94),
    <splus:sarg name="alternative" />)
  </splus:output>
</pre>
```

Attributes:

The table on the next page summarizes the attributes for this element.

Name	Info	Description
name	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> No	The name of the request parameter that gives the value for this argument.
sname	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	The name to give the argument when passing it to S-PLUS. If this attribute is missing, the name attribute is used for this purpose as well.
prepend	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	Text to prepend to the request parameter value before sending it to S-PLUS as an argument. If the request parameter has multiple values, this text is prepended to each value.
append	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	Text to append to the request parameter value before sending it to S-PLUS as an argument. If the request parameter has multiple values, this text is appended to each value.

`<spplus:script>`

The `<spplus:script>` element evaluates one or more S-PLUS statements without adding anything to the Web page returned to the user. This element can be used to set up data within S-PLUS for later elements to access.

Example:

This example creates a linear model relating radiation to ozone. The `<spplus:script>` element calls S-PLUS to create the model and to store it as a variable named `lmAns`. Subsequent elements (`<spplus:output>` and `<spplus:graphlet>`) display a text summary and a graph of the results.

```
<spplus:script>
  lmAns <- lm(formula = ozone ~ radiation, data = air)
</spplus:script>

<pre>
<spplus:output expr="print(lmAns)" />
</pre>

<spplus:graphlet width="480" height="360">
  plot(lmAns, which.plots = 3)
</spplus:graphlet>
```

Attributes:

The `<spplus:script>` element takes no attributes.

`<splus:useResult>`

The `<splus:useResult>` element evaluates an S-PLUS expression and assigns the result to a Java variable. Subsequent Java code on the JSP page might use this variable to output the result in some specially formatted way, or the code might use the variable in an `if` or `while` statement to control flow of control based on the S-PLUS result.

The Java type of variable to which the S-PLUS result is assigned must either match that of the result or be a superclass. Currently S-PLUS Server can return the result only as one of these array types: `boolean[]`, `long[]`, `float[]`, `double[]`, or `java.lang.String[]`. The sole superclass of these types is `java.lang.Object`. Thus the variable must be declared as one of these types.

The `<splus:useResult>` element provides two ways to specify the S-PLUS expression. The first way is to use the `expr` attribute in an element with no body, as shown in the example below. The second way is to use a `begin` and an `end` tag, with the S-PLUS expression as the body. The second option is handy when you use an `<splus:sarg>` element for an argument in the expression, as at least some containers will not interpret this tag when used in an `expr` attribute.

The `<splus:useResult>` element is similar to the `<splus:result>` element, which outputs the S-PLUS result directly to the page sent to the user rather than assigning it to a Java variable.

Example:

This example declares a Java variable named `strColumns` and assigns to it the column names of the data frame `fuel.frame`. The code then creates a drop down box with the names, so that the user can select a column.

```
<splus:useResult expr="names(fuel.frame)"
  id="strColumns" className="java.lang.String[]" />

<form>
<select name=column>
<% for (int i = 0; i < strColumns.length; i++) { %>
  <option value="<%= strColumns[i] %>">
    <%= strColumns[i] %>
  </option>
<% } %>
</select>
</form>
```

Attributes:

The `<splus:useResult>` attributes are described in the following table.

Name	Info	Description
expr	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The S-PLUS expression to evaluate. If this attribute is missing, the element uses its body as the S-PLUS expression to evaluate.
id	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> No	The name of the Java variable to which the S-PLUS result will be assigned.
className	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> No	The Java type of the variable to which the S-PLUS result will be assigned. This type can either match the Java type of the result that S-PLUS returns, or it can be the superclass <code>java.lang.Object</code> . A type mismatch will throw an exception at run time.
declare	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	Whether the tag needs to declare the variable before assigning to it the S-PLUS results. The tag declares the variable unless this attribute is present and has the value <code>false</code> or <code>no</code> .

<splus:batch>

The <splus:batch> element runs S-PLUS code as a batch job. This element is intended to be used for code that takes a while to run, longer than a user will typically want to wait for a Web page to return.

The <splus:batch> element spawns a separate thread that connects to an S-PLUS session, runs the S-PLUS code from the element's body, and sends email to notify the user when the job is finished. If the S-PLUS code in the body of the <splus:batch> element returns text (character data), this text will be included in the email message sent to the user when the job completes.

An application that supports batch jobs will have a separate pool of S-PLUS connections to run these jobs. These connections will be configured specifically for batch jobs and might run on a different machine. The application will typically include JSP pages that list running batch jobs and provide a way to cancel them.

Unlike most other elements, the <splus:batch> element does not need to be embedded within an <splus:connect> element, as it does not use S-PLUS connections from the regular pool.

Attributes:

The <splus:batch> attributes are described in the following table.

Name	Info	Description
description	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	A text description of the batch job, for use in JSP pages that report running jobs and in email sent to the user when the job finishes.
user	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The name of the user who submitted the job, for use in JSP pages that report running jobs.
email	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The email address, typically of the person who submitted the job, to which email should be sent when the job completes. If this attribute is missing, no email will be sent.
workingdir	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	If the working directory is specified, then the Batch working directory manager is used to manage the working directories. The working directory is kept around after the job is completed so that the user can return and continue analysis or view more complicated results that cannot be sent through e-mail.
bodyofemail	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	The string specified in this attribute will be included in the body of the email sent to the user specified in the e-mail tag. This allows one to add some brief descriptive text to the e-mail sent.
userjobid	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> Yes	An external job id that can be used to find a specific job. The batch tag does create an internal job id, but the userjobid can be created externally and used to find a specific job. The userjobid could be created using the current time or session id or some other unique identifier.

<splus:file>

The <splus:file> element moves files back and forth between the Web server computer and the computer running an S-PLUS session.

The <splus:file> tag copies files from the Web server to the computer running the S-PLUS session using the same file movement operations that the S-PLUS Server client uses. With the <splus:file> tag, a web application can upload a file to the web server and then move this file to the S-PLUS Server so that it can be processed by the S-PLUS session. For example, one could copy a data file to the server and process it there using this tag. The web application developer will have to determine the best way to upload the file to the web application.

Attributes:

The <splus:file> attributes are described in the following table.

Name	Info	Description
command	<i>Java Type:</i> String <i>Required:</i> No <i>Accepts Tags:</i> No	Specifies the action to be performed on the file - put to put the file on the S-PLUS computer or get to get it from the S-PLUS computer.
remotedir	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> Yes	The directory of the file on the S-PLUS Server.
remotename	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> Yes	The name of the file on the S-PLUS Server.
localdir	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> Yes	The directory for the file on the Web server computer.
localname	<i>Java Type:</i> String <i>Required:</i> Yes <i>Accepts Tags:</i> Yes	The name of the file on the Web server computer.
deleteRemote	<i>Java Type:</i> boolean <i>Required:</i> No <i>Accepts Tags:</i> No	Sets whether to delete the remote file on a get command. This value is ignored if the command is put .