BEA WebLogic Workshop™ Help

Version 8.1 SP4
December 2004

Copyright

Restricted Rights Legend

Trademarks or Service Marks

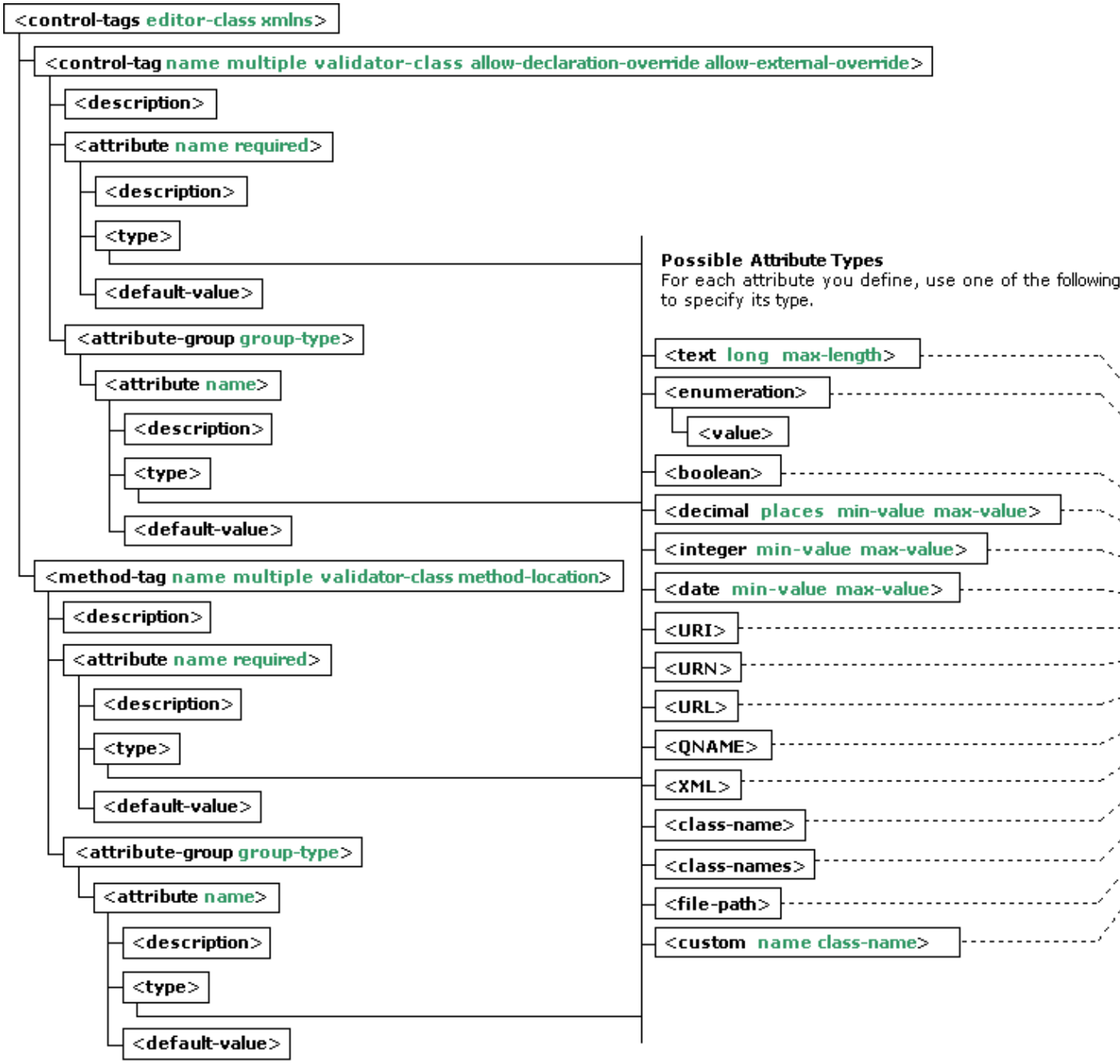
Table of Contents

Control Property Schema Reference

When you define properties that a custom Java control will expose, you specify characteristics for those properties in an XML file. The particular elements and attributes you use are diagrammed in the following figure. You can click each box to see a topic about a particular element and its attributes.

For step−by−step instructions on creating an annotation XML file and connecting it to your control, see How Do I: Define Properties for a Java Control.

Hierarchy



Example

The following example defines a department−description property with two attributes, abbreviation and id. The abbreviation attribute takes a text value that can not be more than four characters long. The id attribute is an integer. Both attributes are defined as not "required," meaning that a developer need not explicitly set them before running an app that uses the control. As a result, default values are provided.

```
<control−tags>
    <control−tag name="department−description">
        <description>Specifies information about the department.</description>
        <attribute name="abbreviation" required="false">
            <description>Specifies the department's four−letter abbreviation.</description>
            <type>
                <text max−length="4"/>
            </type>
            <default−value>ACCT</default−value>
        </attribute>
        <attribute name="id" required="false">
            <description>Specifies the department's ID number.</description>
            <type>
                <integer/>
            </type>
            <default−value>1036</default−value>
        </attribute>
    </control−tag>
</control−tags>
```

A developer could add the control that exposes this property to a component in WebLogic Workshop such as a JWS file. Setting these attributes on its control in a JWS would add an annotation in the JWS source code, immediately preceding the control's variable declaration. If the developer changed the default attribute values to ENGN and 2345, that annotation would look something like this:

```
@jc:department−description abbreviation="ENGN" id="2345"
```

Related Topics

@jcs:control−tags Tag

Tutorial: Java Control

<attribute> Element

Specifies information about an attribute for this property.

Syntax

```
<attribute
    name="attributeName"
    required="true | false"
>
```

Attributes

name	Specifies the name of this attribute. ***Allowable values***: Any string. ***Use***: Required ***Type***: xs:string. ***Default value***: None.
required	Specifies whether this attribute must be set by a developer using the control. You might set the required attribute to true if the attribute to which it applies must have a value, but the value can only be known at design time (when a developer is building an app with the control), or when there is no useful default value. In fact, setting the required attribute to true means that the attribute to which it applies can not have a default value. If you set the required attribute to false, then you must provide a default value because a developer using the control is not required to set a value. ***Allowable values***: true or false. ***Use***: Required. ***Type***: xs:boolean. ***Default value***: None.

Type

This element can contain description, type, and default−value elements.

Hierarchy

Parent: control−tag, method−tag, or attribute−group.

Children: description, type, default−value.

Remarks

None.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<attribute−group> Element

Specifies a group of attributes that must be applied in keeping with specified occurrence rules.

Syntax

```
<attribute−group
    group−type="at−most−one | exactly−one | at−least−one"
>
```

Attributes

group−type	Specifies the occurrence rule for attributes in this group. For attributes in an attribute group, you can specify the following rules that should be used when a developer is using the control and applying attributes from the group: • at−most−one: Only one of the attributes defined in the group may be used, or none may be used. • exactly−one: Only one of the attributes defined in the group may be used, but one must be used. • at−least−one: At least one of the attributes defined in the group must be used, but more may be used. ***Allowable values***: at−most−one, exactly−one, at−least−one ***Use***: Required. ***Type***: AttributeGroupType. ***Default value***: None.

Type

This element contains a list of attribute elements.

Hierarchy

Parent: control−tag or method−tag.

Children: A sequence of attributes.

Remarks

Use this element to define two or more attributes whose presence as annotations on a control or method is mutually dependent. For example, you might have a control that retrieves information from a file, but the file may be available through a file path or over HTTP. You could define an attribute group that includes a fileName attribute whose value must be a file−path and a fileURL attribute whose value must be a URL. Defining this in an attribute group whose group−type is exactly−one ensures that the developer will use only one of the attributes.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<boolean> Element

Specifies that this property attribute takes a boolean value.

Syntax

```
<boolean/>
```

Attributes

None.

Type

This element does not contain a value.

Hierarchy

Parent: attribute.

Children: None.

Remarks

In the WebLogic Workshop Property Editor, a boolean attribute will be displayed with a drop−down listing true and false.

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<class−name> Element

Specifies that this property attribute takes a fully−qualified class name.

Syntax

```
<class−name/>
```

Attributes

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<class−names> Element

Specifies that this property attribute takes a space−separated list of fully−qualified class names.

Syntax

```
<class−names/>
```

Attributes

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<control−tag> Element

Specifies a Java control property. In source code the property will appear as a custom tag; in Design View it will appear in the Property Editor.

Syntax

```
<control−tag
    name="propertyName"
    multiple="true | false"
    validator−class="packageName.ClassName"
    allow−declaration−override="true | false"
/>
```

Attributes

name	Specifies the name of this property. ***Allowable values***: Any string. ***Use***: Required ***Type***: xs:string. ***Default value***: None.
multiple	Specifies whether this property may occur multiple times. ***Allowable values***: true or false. ***Use***: Optional. ***Type***: xs:boolean. ***Default value***: false.
validator−class	Specifies the name of a class to use for validating attribute values. ***Allowable values***: A fully−qualfied class name. ***Use***: Optional. ***Type***: xs:string. ***Default value***: None.
allow−declaration−override	Specifies whether this property may be overridden by a control declaration. If you set allow−declaration−override to true, then a developer using the control can add an annotation to set this tag's value in the control's declaration (in, say, a JWS file). The annotation on the declaration will override a corresponding annotation in the control's JCX or JCS file.

	Allowable values: true or false. ***Use***: Optional. ***Type***: xs:boolean. ***Default value***: true.

Type

This element contain a description, a list of attributes, and/or attribute groups.

Hierarchy

Parent: control−tags.

Children: description, attribute, and attribute−group.

Remarks

This element's attribute and/or attribute−group child elements define the property's attributes.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<control−tags> Element

Top−level tag for an XML file that defines Java control attributes. This tag can contain a control−tag element to define tags that apply to a Java control instance or extension (a JCX file) or a method−tag element to define tags that apply to a control method or callback.

Syntax

```
<control−tags editor−class="nameOfEditorClass" xmlns="http://www.bea.com/2003/03/controls">
```

Attributes

xmlns	Specifies the namespace for elements and attributes in a control−tags XML document. ***Use***: Required. ***Type***: xs:string ***Allowable values***: http://www.bea.com/2003/03/controls ***Default value***: http://www.bea.com/2003/03/controls

editor−class	Specifies the name of the class to use for handling behaviors in the IDE. ***Use***: Optional. ***Type***: xs:string ***Allowable values***: The fully−qualified name of a class that provides editor support. ***Default value***: None.

Hierarchy

Parent: None

Children: control−tag, method−tag.

Remarks

This is the root of a tag XML document.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<custom> Element

Specifies that this property attribute's value is a type defined by the control author.

Syntax

```
<custom
    name="nameForThisValueType"
    class="propertyEditorClass"
/>
```

Attributes

name	Specifies name of the type to be used for this attribute's value. ***Allowable values***: The name of the type. ***Use***: Required. ***Type***: xs:string ***Default value***: None.
class−name	Specifies the path to a class containing an editor for this property's value. ***Allowable values***: The path to a class that implements an editor for this property's value. ***Use***: Required. ***Type***: xs:string ***Default value***: None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

Use the custom element when the attribute's value should be a type you have defined. The class−name attribute's value should be a class that implements support for validating the value of the attribute for which the custom value is defined. For more information on validation classes, see the CustomerData control in the ControlDevKit samples provided with WebLogic Workshop.

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<date> Element

Specifies that this property attribute takes a date value.

Syntax

```
<date
    min-value="minimumValueAllowed"
    max-value="maximumValueAllowed"
/>
```

Attributes

min-value	Specifies the minimum value that this property may have. ***Allowable values***: Any date. ***Use***: Optional. ***Type***: xs:date ***Default value***: None.
max-value	Specifies the maximum value that this property may have. ***Allowable values***: Any date. ***Use***: Optional. ***Type***: xs:date ***Default value***: None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class-name, class-names, date, decimal, enumeration, file-path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<decimal> Element

Specifies that this property attribute takes a decimal value.

Syntax

```
<decimal
    places="numberOfDecimalPlaces"
    min-value="minimumValueAllowed"
    max-value="minimumValueAllowed"
/>
```

Attributes

min-value	Specifies the minimum value that this attribute may have. ***Allowable values***: Any decimal. ***Use***: Optional. ***Type***: xs:decimal. ***Default value***: None.
max-value	Specifies the maximum value that this attribute may have. ***Allowable values***: Any decimal. ***Use***: Optional. ***Type***: xs:decimal. ***Default value***: None.
places	Specifies the number of places to the right of the decimal. ***Allowable values***: Any integer. ***Use***: Optional. ***Type***: xs:int ***Default value***: None.

Type

This element does not contain a value.

Hierarchy

Parent: attribute.

Children: None.

Remarks

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<default−value> Element

Specifies this property attribute's default value.

Syntax

```
<default−value>
    theDefaultValue
<default−value>
```

Attributes

None.

Type

xs:string.

Hierarchy

Parent: attribute.

Children: None.

Remarks

If you have set a default value by adding this element and giving it a value, WebLogic Workshop will insert that value automatically when a developer adds your control to an application. If the developer sets a new value, then later deletes the value, WebLogic Workshop will again use the default.

Note that you can only specify a default if you have set the attribute element's required attribute to false. If the required attribute is set to true, WebLogic Workshop assumes that you want a developer using your control to set the value explicitly, and the IDE will prompt them to do so. For "required" attributes, a default value has no meaning. If you add this default−value element to an attribute that is defined as required, your tags XML file will be invalid.

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<description> Element

Specifies the text that describes a control property or attribute in the Property Editor.

Syntax

```
<description>textDescribingThisPropertyOrAttribute</description>
```

Attributes

None.

Type

xs:string

Hierarchy

Parent: control−tag, method−tag or attribute.

Children: None.

Remarks

None.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<enumeration> Element

Specifies that this property attribute takes an enumerated value. This element also defines the enumerated values that may be used.

Syntax

```
<enumeration>
    <value>value1</value>
    <value>value2</value>
</enumeration>
```

Attributes

None.

Type

None.

Hierarchy

Parent: attribute.

Children: value.

Remarks

Enumerated values will be displayed in a drop−down in the WebLogic Workshop Property Editor.

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<file−path> Element

Specifies that this attribute takes a valid file path.

Syntax

```
<file−path/>
```

Attributes

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<integer> Element

Specifies that this property attribute takes an integer value.

Syntax

```
<integer
    min-value="minimumValueAllowed"
    max-value="maximumValueAllowed"
/>
```

Attributes

min-value	Specifies the minimum value that this attribute may have. ***Allowable values***: Any long. ***Use***: Optional. ***Type***: xs:long ***Default value***: None.
max-value	Specifies the maximum value that this attribute may have. ***Allowable values***: Any long. ***Use***: Optional. ***Type***: xs:long ***Default value***: None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class-name, class-names, date, decimal, enumeration, file-path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<method−tag> Element

Specifies a Java control property that should be applied to a control method defined in a JCX file. In source code the property will appear as a custom tag; in Design View it will appear in the Property Editor.

Syntax

```
<control−tag
    name="propertyName"
    multiple="true | false"
    validator−class="packageName.ClassName"
    method−location="interface−method | callback−method | both"
/>
```

Attributes

name	Specifies the name of this property. ***Allowable values***: Any string. ***Use***: Required. ***Type***: xs:string. ***Default value***: None.
multiple	Specifies whether this property may occur multiple times. ***Allowable values***: true or false. ***Use***: Optional. ***Type***: xs:boolean. ***Default value***: false.
validator−class	Specifies the name of a class to use for validating attribute values. ***Allowable values***: A fully−qualfied class name. ***Use***: Optional. ***Type***: xs:string. ***Default value***: None.
method−location	Defines the types of methods where this tag can exist. The annotation for a property defined as a method tag must occur in a JCX file, and may be used on a method, callback, or both. ***Allowable values***: interface−method, callback−method, both ***Use***: Optional.

	Type: Restricted from xs:string. ***Default value***: interface−method.

Type

This element contain a description, a list of attributes, and/or attribute groups.

Hierarchy

Parent: control−tags.

Children: description, attribute, and attribute−group.

Remarks

This element's attribute and/or attribute−group child elements define attributes for the property this element defines.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<QNAME> Element

Specifies that this property attribute takes an XML qualified name.

Syntax

```
<QNAME/>
```

Attributes

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

A qualified name is a combination of a prefix corresponding to a URI and a local name. An example would be "xs:string", where "xs" corresponds to the URI http://www.w3.org/2001/XMLSchema and "string" is the local name.

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<text> Element

Specifies that this property attribute takes a text value.

Syntax

```
<text
    long="true | false"
    max-length="maximumNumberOfCharactersAllowed"
/>
```

Attributes

long	Specifies whether the IDE should provide an expanded editing area for the attribute's value. Set this attribute to false to provide a typical single-line editing field in the Property Editor. [Ignored in beta.] ***Allowable values***: true or false. ***Use***: Optional. ***Type***: xs:boolean. ***Default value***: false.
max-length	Specifies the maximum number of characters allowed for the value of this attribute. ***Allowable values***: Any integer. ***Use***: Optional. ***Type***: xs:int. ***Default value***: None.

Type

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class-name, class-names, date, decimal, enumeration, file-path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<type> Element

Specifies the type that should be used for the attribute to which this element applies.

Syntax

```
<type>
    typeForThisAttribute
</type>
```

Attributes

None.

Type

This element should contain an element specifying the type that should be used for the attribute to which this element applies. See Children for a list of types that may be used.

Hierarchy

Parent: attribute.

Children: boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom.

Remarks

None.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<URI> Element

Specifies that this property attribute takes a Uniform Resource Identifier.

Syntax

```
<URI/>
```

Attributes

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

URIs are the generic set of all names/addresses that are short strings that refer to resources.

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<URL> Element

Specifies that this property attribute takes a Uniform Resource Locator.

Syntax

```
<URL/>
```

Attributes

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

While no longer used in technical specifications, the term "URL" is associated with popular URI schemes such as http, ftp, mailto, and so on.

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<URN> Element

Specifies that this property attribute takes a Uniform Resource Name.

Syntax

```
<URN/>
```

Attributes

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

A URN identifies a resource in a persistent, location−independent way. URNs are a subset of URIs.

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<value> Element

Specifies one of the enumerated values for this property attribute.

Syntax

```
<value>value1</value>
<value>value2</value>
```

Attributes

None.

Type

xs:string.

Hierarchy

Parent: enumeration.

Children: None.

Remarks

Use this element as a child of the enumeration element to specify one of the enumerated values.

Related Topics

Control Property Schema Reference

Tutorial: Java Control

<XML> Element

Specifies that this property attribute takes an XML value.

Syntax

```
<XML/>
```

Attributes

None.

Hierarchy

Parent: attribute.

Children: None.

Remarks

This element is one of several you can use to specify the type of a property attribute's value. The complete list includes boolean, class−name, class−names, date, decimal, enumeration, file−path, integer, QNAME, text, URI, URL, URN, XML, or custom. Note that you may use only one of these types for a given attribute.

Related Topics

Control Property Schema Reference

Tutorial: Java Control