<value> Element Definition

The **value** element defines a simple value, one that is bound to either an attribute or a simple element (one with no attributes or child elements). This always makes use of a format definition. If no format is explicitly named in the value definition the default format for the property type is used (see Conversions for information on default formats.

The **value** element supports four unique attributes along with several common attribute groups, listed below.

Attributes

constant

This optional attribute is used to define a constant value. If present, none of the attributes from the **string** group and only the **usage** attribute from the **property** group can be present, the **format** attribute is not allowed, and the only **ident** value allowed is "none". If the value is defined as required it must be present and match the constant on input; if optional, it is checked for on input and if found must match the constant. The constant value is always included in the output when marshalling.

format

If present, this gives the name of the format to be used for converting the property value to and from text. The named format must be defined in an enclosing context to the **value** element. As of JiBX 1.1, the value of this attribute is interpreted as namespace qualified.

ident

This optional attribute is used to designate a property value as an identifier type. Possible values are "none" (not an identifier type, the default if this attribute is not used), "def" (value is a unique identifier for the containing object), and "ref" (value is an object with an identifier property and the identifier property of the object is used in the XML representation rather than the actual object). Only one property with **ident="def"** is allowed for a mapping; if one is present the property must be a String, and must be defined directly as a child of the **mapping** element; it is not

allowed as a child of a **structure** element.

Values with **ident** value "ref" cannot be used directly as children of a collection; they *can* be used if there is a wrapper object for the referenced object, as defined by a **structure** element with an object type defined. References contained directly in a collection can be handled by using custom marshaller/unmarshallers - see the JiBX extras description for a base custom marshaller/unmarshaller you can extend for this purpose, along with another custom marshaller/unmarshaller you can extend to include the full representation of an object only at the point of first use (with a reference if the same object is later used again).

style

A **style** attribute present on the **value** element determines the type of XML component used to represent the value. The allowed values are "attribute", "element", "text", and "cdata". The last two choices are subject to some restrictions. They cannot be used directly within a **collection** definition (in other words, for **value** elements that have a **collection** element as their parent), and also cannot be used for direct children of an unordered **mapping** or **structure**. Even within an ordered **mapping** or **structure**, multiple **value** elements using these choices must be separated by a required **value** using the "element" choice.

The default handling for this attribute is also special. If it isn't specified, it defaults to the value set by the innermost containing element with a **value-style** attribute, which will always be either the "attribute" or "element" choice. See the style attribute group description for details of this type of usage.

name

Attributes from the name group supply the element or attribute name for this value. At least the **name** attribute is required if the XML component type used for this value is an attribute or element, and forbidden otherwise. See the name attribute group description for usage details.

property

Attributes from the property group define a property value, including how it is accessed and whether it is optional or

required. A property value is required for **value** elements unless they're contained within a **collection** element. In this case the value is always an item from the collection, so the only attributes from this group allowed are the "usage" and "type" attributes. See the property attribute group description for usage details.

string	Attributes from the string group set the conversion handling of this value. See the string attribute group description for usage details.