

## <binding> Element Definition

The **binding** element is the root of a binding definition. This defines characteristics for the entire binding. It supports several unique attributes as well as two common attribute groups:

### Attributes

add-constructors	When this optional attribute is present with value "true", the binding compiler will add a default constructor to a bound class when necessary to make it usable in the binding. Default constructors are needed when there's no unmarshaller or factory method for an object reference. Constructors can only be added to classes which are available as class files for modification during the binding compile.
direction	Binding direction, which must be "input" (unmarshalling only), "output" (marshalling only), or "both" (both marshalling and unmarshalling, the default if this attribute is not given).
force-classes	When this optional attribute is present with value "true", it forces generation of marshaller/unmarshaller classes for top-level abstract mappings which are not extended by other mappings. Normally these classes would not be generated, since such mappings are never used directly within the binding. The classes can be used at runtime by custom code, though, as the equivalent of type mappings. The default value is "false".
forwards	Controls whether forward references to ids are supported when unmarshalling, at the cost of some (minor) additional overhead and code size. If supplied the value must be "true" (forwards supported) or "false" (ids must be defined before they are referenced). The default is "true".
name	The name for this binding. The supplied name must

consist only of characters which are valid as part of a Java identifier (so no spaces, periods, etc.). The default if this attribute is not used is to take the name of the file containing the definition document, with any file extension suffix stripped off and invalid characters replaced by underscores (so a binding definition in a file named "binding-2.xml" has the default name "binding\_2").

## package

Java package used for created binding factory class. By default this is the same package as the class associated with the first **mapping** child element. If present, the value must be a Java package name (as in "org.jibx.runtime").

## track-source

When this optional attribute is present with value "true", the binding compiler adds code to each bound object class to implement the `org.jibx.runtime.ITrackSource` interface and store source position information when instance objects are unmarshalled. This interface lets you retrieve information about the source document and specific line and column location of the document component associated with that object. The default value is "false".

## trim-whitespace

When this optional attribute is present with value "true", whitespace is trimmed from all simple text values before the value is deserialized. The effect is the same as if all the built-in [conversions](#) were defined using `whitespace="trim"` (see the [string](#) attribute group). This makes the conversions compatible with schema types.

## style

A **value-style** attribute present on the **binding** element sets a default for all contained elements. See the [style attribute group](#) description for usage details.