

XML Zebra – Path Syntax Summary

Version 1.0-pre-1

<http://www.xmlzebra.com/>

This document summarises the syntax that XML Zebra uses for the paths it generates in its “fingerprint” XML files.



Things that are the same as XPath

- Path elements are separated by “/”
- Element names appear directly, e.g. “paragraph”, “html:p”.
- Attribute names are prefixed by “@”, e.g. “@ref”, “@xml:id”.

Things that are different to XPath

- Paths always start with “/”
- Structure elements appear in paths: “xs:sequence”, “xs:choice”, “xs:all”, “xs:any”
 - If the structure element is a mandatory xs:sequence (one occurrence only), then the structure element does **not** appear in the path. These are suppressed because mandatory sequences occur so often in XML Schemas
- Brackets (“[“ and “]”) are used differently to XPath (as discussed below). Single/double quotes are **not** used around values inside brackets
- Where an “xsi:type” extension is allowed for an element, for each extension type there is a path with “[xsi:type=...]” following the element name, e.g. “[xsi:type=a:AttributedType]”
- The XML Schema base type of an XML element (with simple content) or an XML attribute is indicated by the path element suffix “[base=xs:...]”, e.g. “[base=xs:string]”
- The base type can be followed by the default or fixed value, if any, e.g. “[default=3.141592356589]” or “[fixed=some enchanted evening]”
- Any path element can be suffixed with an occurrence specifier:
 - no occurrence specifier = one occurrence only (mandatory)
 - “?” = zero or one (optional)
 - “*” = zero or more (optional, repeatable)
 - “+” = one or more (mandatory, repeatable)
 - “[m..n]” = at least “m” occurrences, no more than “n” occurrences
- A facet for a type with simple content appears as a path element, with its value appearing as a following “@value” (attribute) path element with an explicit value, e.g. “xs:minLength/@value=3” or “xs:enumeration/@value=enum1”

- Where there are multiple facets, each is in a separate path
- A recursion is indicated by a path element of the form “...[recursion=...]” which indicates the complex type for which the recursion was detected, e.g. “...[recursion=b:CircularB]”

Other things to note

- Paths are **not** created for abstract elements
- Simple types which are lists or unions are not yet handled
- “xs:any” is not yet fully handled

Some example XML Zebra paths

- /doc
- /b:docB/b:a/a:c
- /a:doc/xs:sequence*/a:numeric[base=xs:decimal][default=3.141592356589][0..3]
- /a:doc/xs:sequence*/a:elem1[base=xs:string]/xs:minLength/@value=3
- /a:doc/xs:sequence*/a:attributed[base=xs:string]*/@optionalDefaulted[base=xs:string][default=defaultValue]?
- /a:doc/xs:sequence*/a:attributed[base=xs:string]*/@requiredFixed[base=xs:string][fixed=fixedValue]
- /a:doc/xs:sequence*/a:recursion?/a:recursion*/...[recursion=a:RecursiveType]
- /b:base[xsi:type=b:ExtendedType]/b:base1[xsi:type=a:AttributedType]