Properties for indirection:

The following properties allow for data to be accessed in a non-sequential or non-contiguous format by using indirection values to locate the data contents. When parsing, the indirection value is used to access the contents in the data stream to populate the infoset. When unparsing, an indirection value is created to store the contents in the data stream when creating the data stream from the infoset. The indirection value does not exist in the infoset, but only in the data stream.

The LeadingAlignment and TrailingAlignment regions apply to the indirection value in the data stream. The initiator and terminator as well as all regions between the LeftFraming and RightFraming regions are applied to the data content accessed through the indirection value.

|  |  |
| --- | --- |
| Property Name | Description |
| indirectKind | EnumSpecifies the type of indirection value used to access the the contents in the data stream.Valid values ‘pointer’, ‘offset’, ‘none’.‘pointer’ means a numeric value gives an implementation-dependent starting position of the content.‘offset’ means a numeric value that when added to indirectBase gives the starting position of the content.‘none’ means no indirection is used and is the default value.Annotation: dfdl:choice, dfdl:element, dfdl:group, dfdl:simpleType, dfdl:sequence |
| indirectType | QNameName of a simple type derived from xs:integer or any subtype of it.This type, with its DFDL annotations specifies the representation of the indirection value.It is a schema definition error if the xs:simpleType specifies any of: * dfdl:lengthKind 'delimited', 'endOfParent', or 'pattern'
* dfdl:lengthKind 'explicit' where length is an expression
* dfdl:outputValueCalc
* dfdl:initiator or dfdl:terminator other than empty string
* dfdl:alignment other than '1'
* dfdl:leadingSkip or dfdl:trailingSkip other than '0'.

Annotation: dfdl:choice, dfdl:element, dfdl:group, dfdl:simpleType, dfdl:sequence |
| indirectBase | StringProvides the relative path to a prior element upon which the offset is based.Required only when dfdl:indirectKind is ‘offset’.Annotation: dfdl:choice, dfdl:element, dfdl:group, dfdl:simpleType, dfdl:sequence |
| indirectEmptyValue | IntegerSpecifies the indirection value that indicates when the data content is empty. If this property is specified, all underlying elements must contain a default value specification or it is a schema definition error.Annotation: dfdl:choice, dfdl:element, dfdl:group, dfdl:simpleType, dfdl:sequence |

The following example illustrates how to describe a pointer to a null-terminated string (common in languages like C).

<xs:element name="myString" type="xs:string"
 dfdl:lengthKind="delimited" dfdl:encoding="UTF-8"
 dfdl:terminator="%NUL;" dfdl:indirectKind="pointer"
 dfdl:indirectType="ptr32\_t" dfdl:indirectEmptyValue="0"
 default=""/>

<xs:simpleType name="ptr32\_t" dfdl:representation="binary"
 dfdl:lengthKind="explicit" dfdl:length="4"
 dfdl:byteOrder="bigEndian">
 <xs:restriction base="integer"/>
</xs:simpleType>

The data stream may look like the following for a string value of “test”.

Location Hex values
00000000 0012A000

0012A000 7465737400
 t e s t

The following example defines an array of 3 pointers to complex elements defined by ns0:myStruct.

<xs:element name="myArray" type="ns0:myStruct"
 dfdl:lengthKind="implicit" dfdl:indirectKind="pointer"
 dfdl:indirectType="ptr32\_t" dfdl:indirectEmptyValue="0"
 minOccurs="3" maxOccurs="3" dfdl:occursCountKind="fixed"/>

<xs:simpleType name="ptr32\_t" dfdl:representation="binary"
 dfdl:lengthKind="explicit" dfdl:length="4"
 dfdl:byteOrder="bigEndian">
 <xs:restriction base="integer"/>
</xs:simpleType>

The data stream may look like the following with the contents of each occurrence in a different location and the contents of the 2nd occurrence being empty.

Location Hex values
00000000 00147000 00000000 00146000

00146000 ...

00147000 ...

The following example defines an offset to a 6 byte hexBinary value with the offset being calculated from the start of the current element.

<xs:element name="myData" type="xs:hexBinary"
 dfdl:lengthKind="explicit" dfdl:length="6"
 dfdl:indirectKind="offset" dfdl:indirectType="uint"
 dfdl:indirectBase="."/>

<xs:simpleType name="uint" dfdl:representation="binary"
 dfdl:lengthKind="explicit" dfdl:length="4"
 dfdl:byteOrder="bigEndian">
 <xs:restriction base="unsignedInt"/>
</xs:simpleType>

The data stream may look like the following for a hexBinary value of “123456789ABC”.

Location Hex values
00000100 00000108

00000208 123456789ABC