DFDL Escape Blocks

### Author: Avery Bibeau

# Explanation

Escape blocks are one of two methods for escaping separator characters in DFDL elements. Escaping is a useful capability for using alternative data representation that includes the separator character. Escape blocks utilize starting and ending characters to define sections where separator characters will be escaped.

# Problem

This tutorial explores escape blocks with alternative data representation in a list of names where each name can be represented as ‘First Last’ or ‘Last, First’

The following data set will be parsed: a list of names containing two names where each name is separated by a comma. Notice that the second name uses a comma (the separator character), this character needs to be escaped by an escape block.

names=John Doe,"Doe, Jane"

# Solution

A defineEscapeScheme is required to create properties related to escaping. This definition exists outside of the default properties schema.

1. The dfdl:defineEscapeScheme is created in the annotation, immediately below the schema’s dfdl:format and is given the name "DefaultPropertiesEscapeScheme"

<xsd:annotation>

<xsd:appinfo source="http://www.ogf.org/dfdl/">

<dfdl:format ref="DefaultPropertiesFormat"/>

<dfdl:defineEscapeScheme name="DefaultPropertiesEscapeScheme">

<dfdl:escapeScheme …> </dfdl:escapeScheme>

</dfdl:defineEscapeScheme>

</xsd:appinfo>

</xsd:annotation>

1. The property escapeKind set to "escapeBlock" is necessary for using escape blocks.
2. Next, the characters used to define the start and end of escape blocks can be set with the properties escapeBlockStart and escapeBlockEnd. For this schema, both properties are set to "&quot;".
3. The property generateEscapeBlock will be set to "whenNeeded" to indicate that escaping will only occur when the data contains an in-scope terminating delimiter, the escapeBlockStart character, or an extraEscapeCharacter.
4. Next, the escapeEscapeCharacter is used to define a character for escaping the first character of the escapeBlockEnd, this property is set to "\". While this property is not entirely necessary for this example, it is still a useful capability.

The resulting XML infoset looks like this:

<names>

<name>John Doe</name>

<name>Doe, Jane</name>

</names>

1. To further explore the escapeEscapeCharacter, we can use the following data set

names=John Doe,"Doe, Jane \"Green\""

1. This data set demonstrates escaping the escapeBlockEnd character with the escapeEscapeCharacter. The resulting XML infoset looks like this:

<names>

 <name>John Doe</name>

 <name>Doe, Jane "Green"</name>

</names>

# Schema

Here’s a complete schema using escape blocks:

<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:dfdl="http://www.ogf.org/dfdl/dfdl-1.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"

xmlns:fn="http://www.w3.org/2005/xpath-functions">

<xsd:include schemaLocation="DefaultProperties.xsd"/>

<xsd:annotation>

<xsd:appinfo source="http://www.ogf.org/dfdl/">

<dfdl:format ref="DefaultPropertiesFormat"/>

<dfdl:defineEscapeScheme name="DefaultPropertiesEscapeScheme">

<dfdl:escapeScheme escapeBlockEnd="&quot;" escapeBlockStart="&quot;" escapeKind="escapeBlock"

escapeEscapeCharacter="\" generateEscapeBlock="whenNeeded">

</dfdl:escapeScheme>

</dfdl:defineEscapeScheme>

</xsd:appinfo>

</xsd:annotation>

<xsd:element name="names" dfdl:initiator="names=">

<xsd:complexType>

<xsd:sequence dfdl:separator=",">

<xsd:element name="name" type="xsd:string" dfdl:occursCountKind="implicit" minOccurs="1" maxOccurs="unbounded" dfdl:escapeSchemeRef="DefaultPropertiesEscapeScheme"/>

</xsd:sequence>

</xsd:complexType>

</xsd:element>

</xsd:schema>