
Stage 3 proposal: Feature #252 add @outputclass to DITAVAL

For flagging purposes, DITAVAL today allows you to associate a variety of styles with a specific property or revision. Today, the @outputclass attribute is frequently used to associate a much broader range of processing or even CSS styling with an element or group of elements; allowing DITAVAL to associate that same token as a flag will give DITAVAL based flagging the same full range of possibilities.

Champion

Robert D. Anderson

Tracking information

Event	Date	Links
Stage 1 proposal accepted	June 11 2019	Minutes
Stage 2 proposal submitted	Sept 4 2019	PDF , DITA Source
Stage 2 proposal discussed	Sept 10 2019	Minutes
Stage 2 proposal approved	Sept 17 2019	Minutes
Stage 3 proposal submitted to reviewers	October 1, October 8	Eliot Kimber, Scott Hudson, Eric Sirois, Carsten Brennecke
Stage 3 proposal (this document) submitted to TC	October 10, 2019	

Approved technical requirements

Add @outputclass to the <prop> and <revprop> elements in the DITAVAL format. The attribute is CDATA, allowing one or more tokens (exactly the same syntax as @outputclass in DITA).

Dependencies or interrelated proposals

Feature #17 made @outputclass universal. These features do not depend on each other, just noting that the already-completed proposal #17 ensures that @outputclass is already available as a universal attribute, so this new feature is not providing a *new* way to add output classes to elements that would otherwise not allow them.

Modified grammar files

Add @outputclass declaration to ditaval.rng:

```
<define name="attlist.prop" combine="interleave">
  <optional>
    <attribute name="att"/>
  </optional>
  <optional>
    <attribute name="val"/>
  </optional>
  <attribute name="action">
    <choice>
      <value>flag</value>
      <value>include</value>
    </choice>
  </attribute>
</define>
```

```

    <value>exclude</value>
    <value>passthrough</value>
  </choice>
</attribute>
<optional>
  <attribute name="outputclass"/>
</optional>
<optional>
  <attribute name="color"/>
</optional>
<optional>
  <attribute name="backcolor"/>
</optional>
<optional>
  <attribute name="style">
    <data type="NMOKENS" dita:since="1.3"/>
  </attribute>
</optional>
</define>

```

```

<define name="attlist.revprop" combine="interleave">
  <optional>
    <attribute name="val"/>
  </optional>
  <attribute name="action">
    <choice>
      <value>include</value>
      <value>passthrough</value>
      <value>flag</value>
    </choice>
  </attribute>
  <optional>
    <attribute name="outputclass"/>
  </optional>
  <optional>
    <attribute name="changebar"/>
  </optional>
  <optional>
    <attribute name="color"/>
  </optional>
  <optional>
    <attribute name="backcolor"/>
  </optional>
  <optional>
    <attribute name="style">
      <a:documentation>
        The style attribute should be a color value (either a name, or a SRGB value).
        See below for the supported color names (taken from the XHTML DTD).

        There are 16 widely known color names with their sRGB values:

        black = #000000    green  = #008000
        silver = #C0C0C0   lime   = #00FF00
        gray  = #808080    olive  = #808000
        white = #FFFFFF    yellow = #FFFF00
        maroon = #800000   navy   = #000080
        red   = #FF0000    blue   = #0000FF
        purple = #800080   teal   = #008080
        fuchsia = #FF00FF  aqua   = #00FFFF
      </a:documentation>
      <data type="NMOKENS" dita:since="1.3"/>
    </attribute>
  </optional>
</define>

```

Add @outputclass to ditaval.dtd:

```

<!ATTLIST prop
  att      CDATA      #IMPLIED
  val      CDATA      #IMPLIED
  action   (flag|include|exclude|passthrough) #REQUIRED
  outputclass CDATA    #IMPLIED

```

```
color      CDATA      #IMPLIED
backcolor  CDATA      #IMPLIED
style      NMTOKENS   #IMPLIED
>
```

```
<!ATTLIST revprop
  val      CDATA      #IMPLIED
  action   (include|passthrough|flag) #REQUIRED
  outputclass CDATA      #IMPLIED
  changebar CDATA      #IMPLIED
  color     CDATA      #IMPLIED
  backcolor CDATA      #IMPLIED
  style     NMTOKENS   #IMPLIED
>
```

Modified terminology

N/A

Modified specification documentation

Add a definition of @outputclass to the following topics: <prop>, <revprop>

@outputclass

If flag has been set, treat the flagged element as if the full @outputclass value in the DITAVAL was specified on that element's @outputclass attribute. If two or more DITAVAL properties apply @outputclass flags to the same element, treat the flagged element as if each value was specified on that element's @outputclass attribute; in that case, the order of those DITAVAL-based tokens is undefined. If the flagged element already specifies @outputclass, treat the flagged element as if all DITAVAL-based @outputclass values come first in the attribute.

Migration plans for backwards incompatibilities

N/A