

Approaches to using NEMESIS V3 Custom Elements

Date

August 17, 2011

July 31, 2013 (added section “Restrictions”, page 11)

March 13, 2014 (“CorrelationID now reads CustomElementID” as do the XML test packages)

May 5, 2016 (updated language regarding use of Not Values)

Authors

Keith R. Davis - NEMESIS Data Architect

Jorge Rojas Jr. - NEMESIS Data Manager / Analyst

Su Shaoyu - NEMESIS Lead Developer

Abstract

This document describes two approaches to extending existing elements found in the NEMESIS V3 XML Schemas (XSDs). These implementations will help agencies and states capture data such as the classic car vs. moose scenario.

Background

With previous schemas, all of the custom element information was contained in eCustom and dCustom. This brute force approach included the custom element definitions in every record resulting in duplication of data. Depending on the number of custom elements, the submission file size could increase dramatically. Starting with the newly released schema, definitions for eCustom elements have been moved into the header configuration section and the results for eCustom elements have been left in the PatientCareReport. The resulting XML file will be smaller through the removal of redundant custom element definitions.

XML Schema (eCustom_v3.xsd)

The eCustom_v3.xsd schema shows, which provides the necessary linkage between the header information and the patient care report. The eCustom_v3.xsd XML Schema can be found in the XSD zip file located at <http://nemsis.org/v3/downloads/emsDataset.html>

Solution #1: Directly Adopt eCustom_v3.xsd

All linking information is included in NEMESIS xml file.

Example EMSDataSet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<EMSDataSet xmlns="http://www.nemesis.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <DemographicGroup><dAgency.01>dAgency.01</dAgency.01></DemographicGroup>
    <ConfigurationGroup>
      <eCustomConfiguration>
        <eCustomConfiguration.CustomGroup CustomElementID="1ABCD"> ←
          <!-- Original eVitals.26 (PatientsHighestLevelResponsiveness) has 4 options:
                3326001 Alert; 3326003 Verbal; 3326005 Painful; 3326007 Unresponsive
                Now we want to add 2 custom levels: 3326088 High_Alert; 3326099 Low_Alert -->
          <eCustomConfiguration.01
                nemsisElement="eVitals.26">custom eVitals.26</eCustomConfiguration.01>
          <eCustomConfiguration.02>associated with eVitals.26</eCustomConfiguration.02>
          <eCustomConfiguration.03>9902009</eCustomConfiguration.03><!--Data Type:9902009 Text/String-->
          <eCustomConfiguration.04>9923001</eCustomConfiguration.04><!--Multiplicity: 9923001 No-->
          <eCustomConfiguration.05>9903001</eCustomConfiguration.05><!--Usage: 9903001 Mandatory-->
          <!--new custom level, High_Alert-->
          <eCustomConfiguration.06 nemsisCode="3326001" customValueDescription="High Alert">3326088</eCustomConfiguration.06>
          <!--new custom level, Low_Alert-->
          <eCustomConfiguration.06 nemsisCode="3326001" customValueDescription="Low Alert">3326099</eCustomConfiguration.06>
          </eCustomConfiguration.CustomGroup>
        </eCustomConfiguration>
      </ConfigurationGroup>
    <PatientCareReport>
      <eRecord><eRecord.01>eRecord.010</eRecord.01></eRecord>
      <eVitals>
        <eVitals.VitalGroup CorrelationID="reference_id_101"> ←
          <eVitals.01>2005-02-03T14:11:11.0+07:00</eVitals.01>
          <eVitals.26>3326001</eVitals.26>
        </eVitals.VitalGroup>
        <eVitals.VitalGroup CorrelationID="reference_id_202"> ←
          <eVitals.01>2006-05-04T18:13:51.0+07:00</eVitals.01>
          <eVitals.26>3326005</eVitals.26>
        </eVitals.VitalGroup>
        <!--This is constructed to demo a possible bad implementation. Generic XML Validator won't catch this error.We need Schematron
                Business Rule or customized XML parsing module to check data integrity. -->
        <eVitals.VitalGroup CorrelationID="non_exist_reference_id">
          <eVitals.01>2006-05-04T18:13:51.0+07:00</eVitals.01>
          <eVitals.26>3326005</eVitals.26>
        </eVitals.VitalGroup>
      </eVitals>
      <eCustomResults>
        <eCustomResults.ResultsGroup>
          <eCustomResults.01>3326088</eCustomResults.01>
          <eCustomResults.02>1ABCD</eCustomResults.02> ●
          <eCustomResults.03>reference_id_101</eCustomResults.03> ●
        </eCustomResults.ResultsGroup>
        <eCustomResults.ResultsGroup>
          <eCustomResults.01>3326099</eCustomResults.01>
          <eCustomResults.02>1ABCD</eCustomResults.02> ●
          <eCustomResults.03> reference_id_202</eCustomResults.03> ●
        </eCustomResults.ResultsGroup>
      </eCustomResults>
    </PatientCareReport>
  </EMSDataSet>

```

<!--This is constructed to demo a possible bad implementation. Generic XML Validator won't catch this error. We need Schematron Business Rule or customized XML parsing module to check data integrity. -->

```
<eCustomResults.ResultsGroup>
  <eCustomResults.01>bad value</eCustomResults.01>
  <eCustomResults.02>bad_link_to_custom_element_definition</eCustomResults.02>
  <eCustomResults.03>never_referred_link_to_real_NEMESIS_element</eCustomResults.03>
</eCustomResults.ResultsGroup>
</eCustomResults>
</PatientCareReport>
</Header>
</EMSDataSet>
```

The above example shows the current schema's fundamental design that separates custom element definitions and results. The complete custom element for eVitals.26 can be constructed by combining eVitals.26, eCustomConfiguration and eCustomResults. There are two links here:

1. From eCustomConfiguration.CustomGroup's CustomElementID to eCustomResults.02. (light blue fonts)
2. From eVitals.VitalGroup's CustomElementID to eCustomResults.03 (purple and dark blue fonts)

This approach doesn't need to modify eCustom_v3.xsd file.

Solution #2: Alternative Schema

This solution replaces an existing element with a new version of the element and includes a new simpleType that defines the possible values for the new version of the element. Linkage is still provided as presented in solution #1.

Modified eCustom_v3.xsd

In the example that follows, we will replace eCustomResult.01 with a modified version and add a new simpleType to provide the element's possible values:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include schemaLocation="commonTypes_v3_su.xsd"/>
  <xs:complexType name="eCustomConfiguration" id="eCustomHeaderInformation">
    <xs:sequence>
      <xs:element name="eCustomConfiguration.CustomGroup" id="eCustomGroup" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>Group Tag to hold custom information</xs:documentation>
        </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="eCustomConfiguration.01" type="CustomTitle" id="eCustomDataElementTitle" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <nemesisTacDoc>
                  <number>eCustomConfiguration.01</number>
                  <name>Custom Data Element Title</name>
                  <national>No</national>
                  <state>No</state>
                </nemesisTacDoc>
                <definition>This is the title of the custom data element created to collect
```

information that is not defined formally in NEMESIS Version 3.</definition>

```
</v2Number/>
<usage>Optional</usage>
<performanceMeasure/>
<comment>This is grouped with all data elements in this section and can have
multiple instances.</comment>
<v3Changes>Added to allow customized data elements to be inserted and
collected from within the NEMESIS Version 3 standard.</v3Changes>
</nemisTacDoc>
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="eCustomConfiguration.02" type="CustomDataType" id="eCustomDefinition" minOccurs="0">
<xs:annotation>
<xs:documentation>
<nemisTacDoc>
<number>eCustomConfiguration.02</number>
<name>Custom Definition</name>
<national>No</national>
<state>No</state>
<definition/>
<v2Number/>
<usage>Optional</usage>
<performanceMeasure/>
<comment/>
<v3Changes/>
</nemisTacDoc>
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="eCustomConfiguration.03" type="CustomDataType" id="eCustomDataType" minOccurs="0">
<xs:annotation>
<xs:documentation>
<nemisTacDoc>
<number>eCustomConfiguration.03</number>
<name>Custom Data Type</name>
<national/>
<state/>
<definition/>
<v2Number/>
<usage>Optional</usage>
<performanceMeasure/>
<comment/>
<v3Changes/>
</nemisTacDoc>
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="eCustomConfiguration.04" type="YesNoValues" id="eCustomDataElementMultiplicity"
minOccurs="0">
<xs:annotation>
<xs:documentation>
<nemisTacDoc>
<number>eCustomConfiguration.04</number>
<name>Custom Data Element Multiplicity</name>
<national>No</national>
<state>No</state>
<definition>Indication if the data element will accept multiple values.</definition>
```

```

</v2Number/>
<usage>Optional</usage>
</performanceMeasure/>
</comment/>
<v3Changes>Added to allow customized data elements to be inserted and
    collected from within the NEMESIS Version 3 standard.</v3Changes>
</nemisTacDoc>
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="eCustomConfiguration.05" type="ElementUsage" id="eCustomDataElementUsage" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      <nemisTacDoc>
        <number>eCustomConfiguration.05</number>
        <name>Custom Data Element Usage</name>
        <national>No</national>
        <state>No</state>
        <definition>The Usage (Mandatory, Required, or Optional) for the Custom Data
            Element.</definition>
        </v2Number/>
        <usage>Optional</usage>
        </performanceMeasure/>
        <comment>Mandatory = Must be completed and will not accept null
            values<br/>&gt;Required = Must be completed but will accept null
            values<br/>&gt;Optional = Not required but if collected, it cannot be a
            null value.</comment>
        <v3Changes>Added to allow customized data elements to be inserted and
            collected from within the NEMESIS Version 3 standard.</v3Changes>
        </nemisTacDoc>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <!-- eCustomConfiguration.06 is not necessary: the list of potential values is defined in eCustomResults.01's data type -->
  <xs:element name="eCustomConfiguration.06" id="eCustomDataElementPotentialValues"
    minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
      <xs:documentation>
        <nemisTacDoc>
          <number>eCustomConfiguration.06</number>
          <name>Custom Data Element Potential Values</name>
          <national>No</national>
          <state>No</state>
          <definition>The values which are associated with the Custom Data Element.
              Values would be the choices provided to the user when they document the
              Custom Data Element</definition>
          </v2Number/>
          <usage>Optional</usage>
          </performanceMeasure/>
          </comment/>
          <v3Changes>Added to allow customized data elements to be inserted and
              collected from within the NEMESIS Version 3 standard.</v3Changes>
          </nemisTacDoc>
        </xs:documentation>
      </xs:annotation>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base="CustomValue">

```

```

        <xs:attribute name="CustomElementID" type="CustomElementID" use="optional"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="CustomElementID" type="CustomElementID" use="optional"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="eCustomResults" id="eCustomResults">
<xs:sequence>
<xs:annotation>
<xs:documentation>Group Tag to hold custom result information</xs:documentation>
</xs:annotation>
<xs:element name="eCustomResults.ResultsGroup" id="eResultsGroup" minOccurs="0" maxOccurs="unbounded">
<xs:annotation>
<xs:documentation>Group Tag to hold custom information</xs:documentation>
</xs:annotation>
<xs:complexType>
<xs:sequence>
<!--
<xs:element name="eCustomResults.01" type="CustomResults" id="eCustomDataElementResult"
minOccurs="0">
<xs:annotation>
<xs:documentation>
<nemisTacDoc>
<number>eCustomResults.01</number>
<name>Custom Data Element Result</name>
<national>No</national>
<state>No</state>
<definition>The actual value or values chosen (if values listed in custom.05) or result
(free text, Date/Time, or number) documented for the Custom Data
Element</definition>
<v2Number/>
<usage>Optional</usage>
<performanceMeasure/>
<comment/>
<v3Changes>Added to allow customized data elements to be inserted and collected from
within the NEMESIS Version 3 standard.</v3Changes>
</nemisTacDoc>
</xs:documentation>
</xs:annotation>
</xs:element>
-->
<xs:element name="eCustomResults.01" type="My__PatientsHighestLevelResponsiveness"
id="eCustomDataElementResult" minOccurs="0">
<xs:annotation>
<xs:documentation>
<nemisTacDoc>
<number>eCustomResults.01</number>
<name>Custom Data Element Result</name>
<national>No</national>
<state>No</state>
<definition>
The actual value or values chosen (if values listed in custom.05) or result (free text, Date/Time, or number)
documented for the Custom Data Element

```

```

</definition>
</v2Number/>
<usage>Optional</usage>
<performanceMeasure/>
<comment/>
<v3Changes>
  Added to allow customized data elements to be inserted and collected from within the NEMESIS Version 3 standard.
</v3Changes>
</nemisTacDoc>
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="eCustomResults.02" type="CorrelationID" id="eCustomDataElementReference" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      <nemisTacDoc>
        <number>eCustomResults.02</number>
        <name>Custom Data Element Reference</name>
        <national>No</national>
        <state>No</state>
        <definition>Refers to the data element being extended</definition>
        </v2Number/>
        <usage>Optional</usage>
        <performanceMeasure/>
        <comment/>
        <v3Changes/>
      </nemisTacDoc>
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="eCustomResults.03" type="CorrelationID" id="eCustomElementID" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      <nemisTacDoc>
        <number>eCustomResults.03</number>
        <name>Custom Correlation ID</name>
        <national>No</national>
        <state>No</state>
        <definition>Correlation id used to map back to set of other elements in other sections.</definition>
        </v2Number/>
        <usage>Optional</usage>
        <performanceMeasure/>
        <comment/>
        <v3Changes/>
      </nemisTacDoc>
    </xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
<xs:attribute name="CorrelationID" type="CorrelationID" use="optional"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

<xs:simpleType name="My__PatientsHighestLevelResponsiveness">
  <xs:annotation>
    <xs:documentation>The patient's highest level of responsiveness.</xs:documentation>
  </xs:annotation>

```

```
</xs:annotation>
<xs:restriction base="xs:string">

  <!-- two new custom levels -->
  <xs:enumeration value="3326088">
    <xs:annotation>
      <xs:documentation>High_Alert</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="3326099">
    <xs:annotation>
      <xs:documentation>Low_Alert</xs:documentation>
    </xs:annotation>
  </xs:enumeration>

  <!-- original responsive levels: it is also OK to remove any choice! -->
  <xs:enumeration value="3326001">
    <xs:annotation>
      <xs:documentation>Alert</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="3326003">
    <xs:annotation>
      <xs:documentation>Painful</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="3326005">
    <xs:annotation>
      <xs:documentation>Unresponsive</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
  <xs:enumeration value="3326007">
    <xs:annotation>
      <xs:documentation>Verbal</xs:documentation>
    </xs:annotation>
  </xs:enumeration>
</xs:restriction>
</xs:simpleType>
</xs:schema>
```


Example EMSDataSet.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<EMSDataSet xmlns="http://www.nemsis.org" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <DemographicGroup>
      <dAgency.01>dAgency.01</dAgency.01>
    </DemographicGroup>
    <ConfigurationGroup>
      <!-- no need to have custom element definition here. -->
    </ConfigurationGroup>
    <PatientCareReport>
      <eRecord>
        <eRecord.01>eRecord.010</eRecord.01>
      </eRecord>
      <eVitals>
        <eVitals.VitalGroup CorrelationID="reference_id_101">
          <eVitals.01>2005-02-03T14:11:11.0+07:00</eVitals.01>
          <eVitals.26>3326001</eVitals.26>
        </eVitals.VitalGroup>
        <eVitals.VitalGroup CorrelationID="reference_id_202">
          <eVitals.01>2006-05-04T18:13:51.0+07:00</eVitals.01>
          <eVitals.26>3326005</eVitals.26>
        </eVitals.VitalGroup>
      <!--This is constructed to demo a possible bad implementation. Generic XML Validator won't catch this error.
      We need Schematron Business Rule or customized XML parsing module to check data integrity. -->
        <eVitals.VitalGroup CorrelationID="non_exist_reference_id">
          <eVitals.01>2006-05-04T18:13:51.0+07:00</eVitals.01>
          <eVitals.26>3326005</eVitals.26>
        </eVitals.VitalGroup>
      </eVitals>
      <eCustomResults>
        <eCustomResults.ResultsGroup>
          <eCustomResults.01>3326088</eCustomResults.01>
          <eCustomResults.02>1ABCD</eCustomResults.02><!-- Not used -->
          <eCustomResults.03>reference_id_101</eCustomResults.03>
        </eCustomResults.ResultsGroup>
        <eCustomResults.ResultsGroup>
          <eCustomResults.01>3326099</eCustomResults.01>
          <eCustomResults.02>1ABCD</eCustomResults.02><!-- Not used -->
          <eCustomResults.03>reference_id_202</eCustomResults.03>
        </eCustomResults.ResultsGroup>
      <!--This is constructed to demo a possible bad implementation. Generic XML Validator won't catch this error.
      We need Schematron Business Rule or customized XML parsing module to check data integrity. -->
        <eCustomResults.ResultsGroup>
          <eCustomResults.01>3326099</eCustomResults.01>
          <eCustomResults.02>bad_link_to_custom_element_definition</eCustomResults.02>
          <eCustomResults.03>never_referred_link_to_real_NEMSYS_element</eCustomResults.03>
        </eCustomResults.ResultsGroup>
      </eCustomResults>
    </PatientCareReport>
  </Header>
</EMSDataSet>
```

The above example uses only one link, from eVitals.VitalGroup’s CustomElementID to eCustomResults.03.

Restrictions

	NEMESIS element values or codes <i>(options listed in order of priority)</i>	Element Example
Mandatory	Required to Map to NEMESIS Code	eResponse.05 Type of Service Requested
Required	1. Map to NEMESIS code 2. Use NOT Value*	eResponse.08 Type of Dispatch Delay
Recommended	1. Map to NEMESIS code 2. Use NOT Value 3. Not include in the xml	eSituation.03 Complaint Type
Optional	1. Map to NEMESIS code 2. Not include in the xml	eSituation.15 Patient’s Occupational Industry

* When “extending” NEMESIS elements with custom codes, NOT values are only to be used when no available NEMESIS code (found in the Element code list) is even vaguely related to the custom value under consideration. Thus, two uses of Not values in custom elements when extending NEMESIS elements would be: 1) A custom value under consideration cannot be even remotely associated with an available NEMESIS code for the element to be extended; 2) A NOT value appropriately represents the custom value under consideration (for example – extending eDisposition.16 to include a “No Transport” code would be appropriately mapped back to the NOT value “Not Applicable”).

Conclusion

These solutions may not be the only way to solve the linkage problem. Solution #1 does not break the current XML schema implementation; however, this solution cannot be validated using an XSD. Solution #2 does break the current implementation and should only be considered for the most complex situations. For custom values, solution #2 can be validated using an XSD in contrast solution #1. As shown in both examples, it is necessary to carefully construct the link information. Otherwise, invalid linking could occur.