Draft NIST Special Publication 1500-101

Election Event Logging Common Data Format Specification

Version 1.0

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.1500-101

National Institute of
Standards and Technology
U.S. Department of Commerce

NIST Special Publication 1500-101

Election Event Logging Common Data Format Specification

Version 1.0

This publication is available free of charge from:
http://dx.doi.org/10.6028/NIST.SP.1500-101

May 2016

U. S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Willie May, Under Secretary of Commerce for Standards and Technology and Director

National Institute of Standards and Technology (NIST) Special Publication 1500-101
37 pages (May 2016)

NIST Special Publication series 1500 is intended to capture external perspectives related to NIST standards, measurement, and testing-related efforts. These external perspectives can come from industry, academia, government, and others. These reports are intended to document external perspectives and do not necessarily represent official NIST positions.

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedback to NIST. All NIST publications are available at http://www.nist.gov/publication-portal.cfm.

National Institute of Standards and Technology
Attn: Software and Systems Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8970) Gaithersburg, MD 20899-8930
Email: voting@nist.gov

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. This document reports on ITL's research, guidance, and outreach efforts in Information Technology and its collaborative activities with industry, government, and academic organizations.

Abstract

This publication describes an election event logging common data format specification for devices used in U.S. elections. The data that would be logged generally contains information about the conduct of the election, such as when the polls open, when a voter starts a voting session or casts a ballot, when administrators logon to the devices, etc. The publication contains a UML model of the relevant election data and an XML format derived from the UML model. It also contains background information regarding requirements for election event logging in the Election Assistance Commission's Voluntary Voting System Guidelines Version 1.1. It is part of a series of planned common data format specifications for voting equipment.

Keywords

Common data format; disposition; elections; event; logging; severity; timestamp; voting; VVSG.

Acknowledgements

The authors wish to thank their colleagues of the National Institute of Standards and Technology VVSG-Interoperability Public Working Group, who reviewed drafts of this document and contributed to its technical content. The authors gratefully acknowledge and appreciate the following contributors for their keen and insightful assistance with developing this specification:

In addition to the above acknowledgments, the authors also gratefully acknowledge and appreciate:

Executive Summary

This publication is a specification for a common data format (CDF) for the election-related logging information produced by election devices, including voting devices in polling places or other voting equipment used to manage elections. This publication contains a definition for an XML (eXtensible Markup Language) schema [1] that specifies the common data format and how it is used.

Election logs generally contain information relevent to the conduct of the election for which the election device is being used. This information includes important events such as when voting operations are enabled on the device, or when a voter initiates a voting session, or when the device records that the voter has cast her ballot. Log can include errors such as the inability of a device to record a vote due to an internal error or that the polls have been opened or closed prematurely multiple times during the election day. Election analysts can use this information to determine not only whether the device itself was performing correctly but also whether the device was used correctly in the election, that is, used accordingly to election procedures. Additionally, analysts can derive various statistics from the log files, such as how often voters arrived and initiated voting sessions or the amount of time on average it took to cast a ballot.

Currently, election devices do not create election logs in an interoperable common data format, but rather the log files are in proprietary formats and as a result are more difficult for election analysts to read and analyze. If the election logging documentation is not at hand, the logs can be indecipherable. Thus, a common format for the election log information will make it easier for election officials and analysts to understand the log files and, potentially, make more use of the log files for purposes of election auditing, research, and testing.

This publication contains discussion of the requirements in the Election Assistance Commission's (EAC) Voluntary Voting System Guidelines (VVSG) Version 1.1 [2] that specify the required and optional election event information to be logged. The publication also includes a second schema whose purpose is for manufacturers to associate their specific event code documentation with the log files.

This specification is geared towards the following audiences:

- Election officials;
- Voting equipment manufacturers;
- Election-affiliated organizations;
- Election analysts and the general public.

The XML schema associated with this specification is generated from a UML (Unified Modeling Language) [3] model that defines the types, structure, and interrelationships of the data used in election event logs. The advantages to using a UML model include that the model can be more easily understood and subsequently modified, if required, and that formats such as XML or other formats can be generated or derived from the UML model.

Table of Contents

List of Appendices

List of Figures

1 Introduction

This publication is a specification for an XML-based (eXtensible Markup Language) [1] common data format (CDF) for election event logs that are produced by election devices used in U.S. states and territories. The logs contain information generated by voting-related applications such as for election management systems (EMS), electronic pollbook applications, or vote-capture applications that are operating on the election devices. The sorts of information logged includes information required in the U.S. Election Assistance Commission (EAC) Voluntary Voting System Guidelines (VVSG) Version 1.1 [2], however manufacturers may include additional information in the logs.

This specification includes a data model in UML (Unified Modeling Language) [3] that specifies and defines the data fields that are logged. The XML format is generated from the UML model.

1.1 Purpose

The purpose of this specification is to provide a concise, interoperable XML format for manufacturers to integrate into their voting equipment and for election offices, researchers, testing laboratories and other groups to use in their own software. The advantages of using this specification include:

- Election logs are in the same, defined format regardless of device manufacturer;
- Manufacturers can use the same, defined format for defining event codes and other information that may be specific to their own equipment;
- Analyzing election logs produced by different types of equipment and different manufacturers is made significantly easier.

1.2 Audience

The intended audience of this specification includes election officials, manufacturers and developers, as well as others in the election community including the general public.

1.3 Motivation and methodology

This specification was motivated primarily to assist testing labs, election officials, and election analysts when analyzing manufacturer log files and understanding their content. Currently, manufacturers produce log files in proprietary formats, and are inconsistent in the fields used for logging events and the meaning of those fields. This makes analysis of log files more difficult, especially when multiple devices produced by multiple manufacturers are involved in the analysis. This specification provides a format that includes required fields to describe an event as well as optional fields to contain additional details about the event. This specification also includes a companion format for including documentation about a device's event log.

Note that this specification addresses U.S. election devices and is not necessarily intended for use “as is” in in other countries.

1.4 Document Structure

Section 2 starts with background and overview material on logging requirements in the EAC VVSG Version 1.1 and how they are implemented in the UML model and XML schemas. Section 3 contains documentation for the XML schemas.

The appendices include references, definitions, acronyms, instructions for downloading the files associated with this specification, and the XML schemas.

2 Background and Overview

This section provides background information about election equipment logging and requirements in the EAC VVSG Version 1.1 that pertain to logging and are thus addressed by this specification. This section also shows how the requirements are implemented in the UML model.

Election applications such as EMS generally operate on devices including personal computers that themselves have an operating system and a logging capability. The EAC VVSGs require the election applications to generate logs of events that are deemed in the VVSGs as significant, such as when the application allows a login by an administrator, or when the polls are opened and voting is enabled on the application, or when the application records a cast ballot (on an electronic vote capture device, for example). These events are generally written to a separate log file generally called the *election event log*.

2.1 VVSG 1.1 logging requirements implemented

In section 2.1.5.1 of the EAC's VVSG 1.1, Operational Requirements, Requirement D itemizes the data that voting equipment shall at a minimum log. Requirement D is as follows[1]:

> *The voting system equipment shall log at a minimum the following data characteristics for each type of event:*
>
> 1. *system ID;*
> 2. *unique event ID and/or type;*
> 3. *timestamp;*
> 4. *success or failure of event, if applicable;*
> 5. *User ID trigger* [sic] *the event, if applicable;*
> 6. *Resources requested, if applicable.*
>   - i. *Timekeeping mechanisms shall generate time and date values.*
>   - ii. *The precision of the timekeeping mechanism shall be able to distinguish and properly order all audit records.*
>   - iii. *Timestamps shall include the date and time, including hours, minutes and seconds.*
>   - iv. *Timestamps shall comply with ISO 8601 and provide all four digits of the year and include the applicable time zone.*
>   - v. *Voting system equipment shall only allow administrators to set or adjust the clock.*
>   - vi. *Voting system equipment shall limit clock drift to a minimum of one minute within a 15 hour period after the clock is set.*

[1] Clauses v and vi deal with access control to the time adjustment mechanism and capabilities to limit clock drift, and thus are not addressed by the data format in this specification.

The UML model and XML schema in the specification implement the requirements within Requirement D and add several additional optional fields for documentation purposes. Those systems that satisfy Requirement D can export directly into the format described by this specification or can include a translation capability to convert from the manufacturer format into this specification's format.

2.2 Use cases for this specification

There are three general use cases that this specification is intended to satisfy:

1. An election device creates an election event log and writes events to that log. The log may be on removable media or other memory.
2. An election device creates an election event log on removable media and writes events to that log; during election day the media may be removed and reinserted into a different election device, which continues to write events to the same log created by the first device.
3. Multiple election devices are connected to a logging device, which creates a single election event log and writes events from the multiple devices to that log.

The first use case results in the election device creating an event log and writing the device identification and other related information into the election event log, subsequently followed by the events recorded by the device. There is thus a one-to-one correspondence between the device and the election event log.

The second use case comes into play generally if a device being used in, say, a polling place, malfunctions and has to be replaced. The election event log is being recorded on the device’s removable media. When the removable media is inserted into the replacement device, the replacement device will continue to write to the same election event log file.

In the second use case, there is a many-to-one correspondence between the devices and the election event log. Thus, the election event log will need to be arranged such that the election events are properly associated with the corresponding generating devices.

The third use case is much like the second in that there will be multiple devices, however they will be connected to a logging device such as a server that creates an election event log and associates events with each of the connected devices. There is a one-to-one correspondence between the logging device and the election event log, and there is a many-to-one correspondence between the connected devices and the election event log. As with the second use case, the election event log will need to be arranged such that election events are properly associated with the corresponding generating devices.

2.3 UML Model

Figure 1 shows the UML model, which consists of 3 classes, one for describing information about the log file such as when generated, a second class for describing information about the device model, manufacturer, and other related information, and a third class to contain the logged details for individual events. The third class is associated with the second class such that the election events are properly associated with the generating device. All 3 classes and their attributes correspond very closely to major XML elements and their attributes. Note: UML attributes that begin with a capital letter get generated as XML elements, whereas those UML attributes that begin with a lower-case letter get generated as XML attributes.

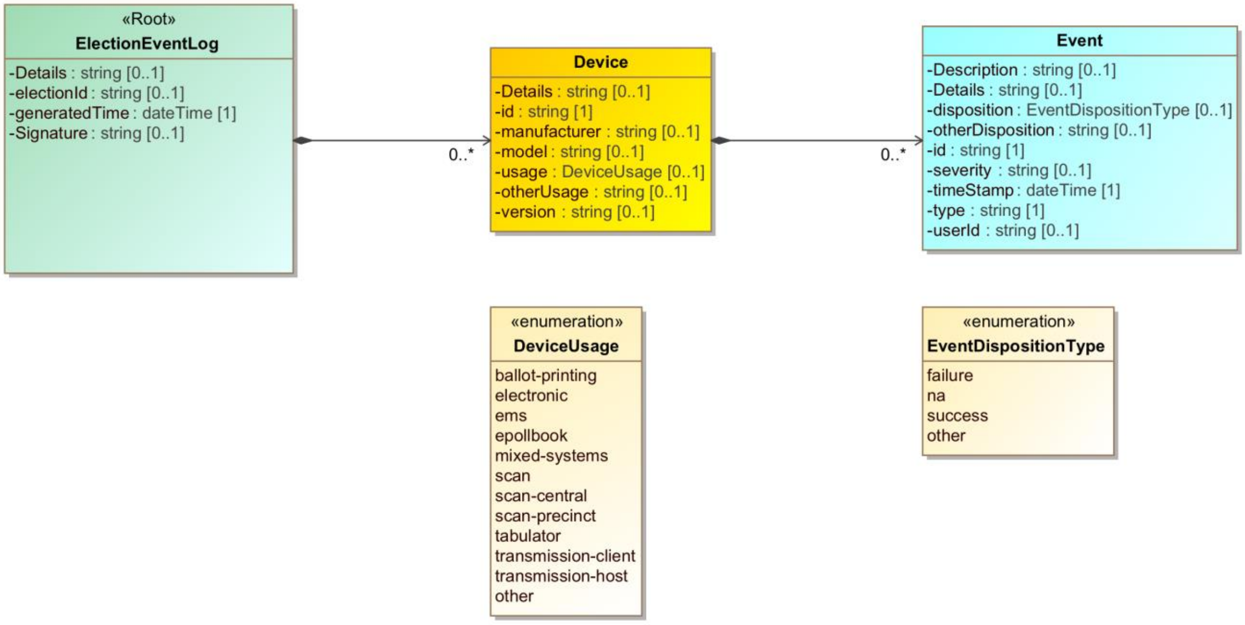

Figure 1 - Election Event Logging UML Class Diagram

2.3.1 The ElectionEventLog Class

The ElectionEventLog class is the root class and contains information about the election event log file itself (as opposed to information about devices and election events). The sole required attribute is generatedTime, for the generation date/time of the election event log file, which satisfies clause 3 and clauses iii and iv of Requirement D. The optional attributes are:

- Details - for any details about the log file
- electionId - for identifying the election that the log file is specific to
- Signature - for associating a digital signature with the election event log file [4]

The ElectionEventLog class is the parent of one or more Device classes, described in the next section.

2.3.2 The Device Class

The Device class contains information about the device(s) generating the election events. There can be multiple instances of the Device class depending on whether multiple devices are generating the election events. There is one required attribute, id, which is for identifying the device, e.g., by using an identifier such as a serial number (Requirement D refers to this attribute in clause 1 as *system ID*). The optional attributes are:

- Details - for any details about the device
- manufacturer - for identifying the name of the manufacturer
- model - for identifying the model of the device
- usage - for associating a pre-defined usage with the device, using the DeviceUsage enumeration
- otherUsage - used if the desired device usage is not found in the DeviceUsage enumeration
- version - for identifying the version of the device

Each instance of the Device class is a parent of one or more ElectionEvent classes, which contain information about specific election events.

2.3.3 The ElectionEvent Class

The ElectionEvent class holds information about an election event, and there can be multiple ElectionEvent classes per Device class. The required attributes are:

- id - the identification number or string used by the manufacturer to identify the event, so as to satisfy clause 2 of Requirement D
- type - the type of event, so as to satisfy clause 2 of Requirement D
- timeStamp - when the event occurred, so as to identify clause 3 of Requirement D

The optional attributes are:

- Description - a description of the event corresponding to the event ID
- Details - any additional information the manufacturer may include
- disposition - an indication of the event status, if applicable, such as success, failure, etc., using the DispositionType enumeration, so as to satisfy clause 4 of Requirement D
- otherDisposition - used if the desired disposition type is not found in the DispositionType enumeration
- severity - an indication of severity, e.g., whether an event is critical, informational, etc.
- userId - an identification of the user associated with or triggering the event, so as to satisfy clause 5 of Requirement D

2.3.4 Examples of class associations to support use cases

Figures 2, 3, and 4 illustrate how the classes are associated so as to support the use cases. Figure 2 shows the associations for the first use case in which there is a single log file per device and multiple, unbounded election events associated with the device.

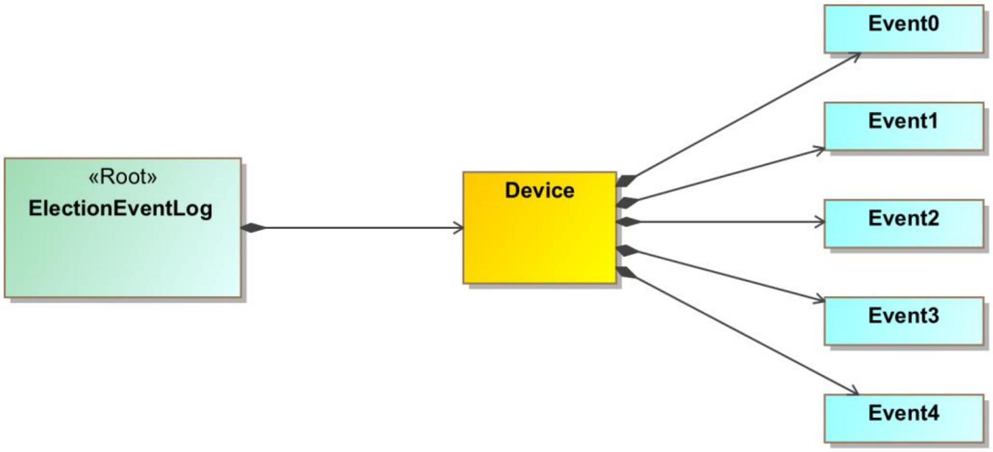

Figure 2 - First use case for a single log file per device

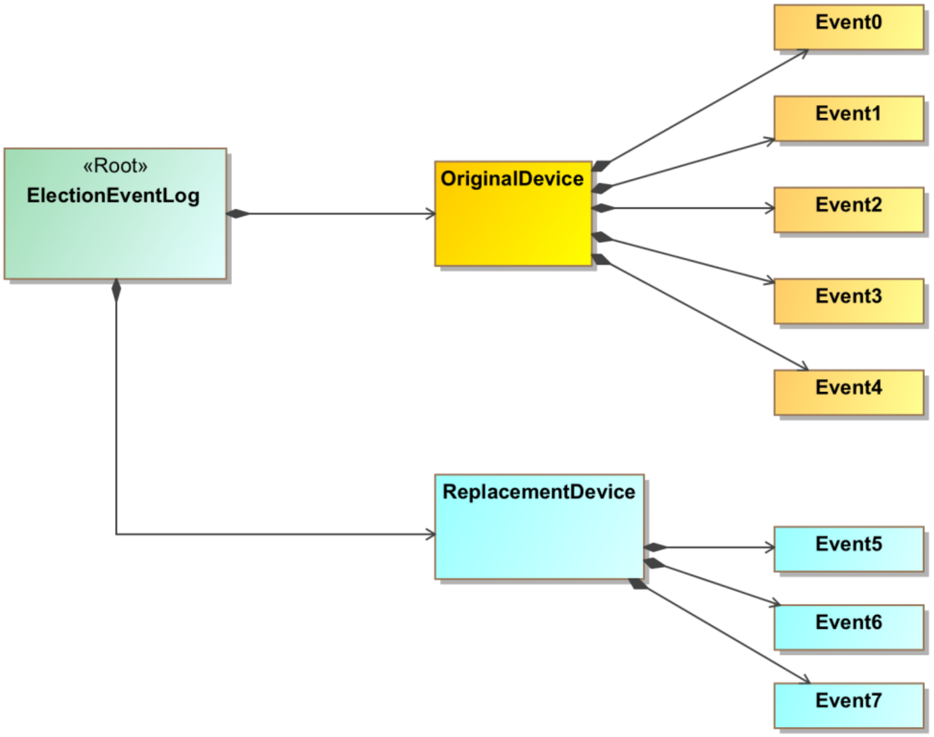

Figure 3 - Second use case for successive devices writing to same log file

Figure 3 shows the associations for the second use case in which there may be multiple devices used successively to write to the same log file, which could occur if a device malfunctions, and the removable media containing the log file is re-inserted into a replacement device.

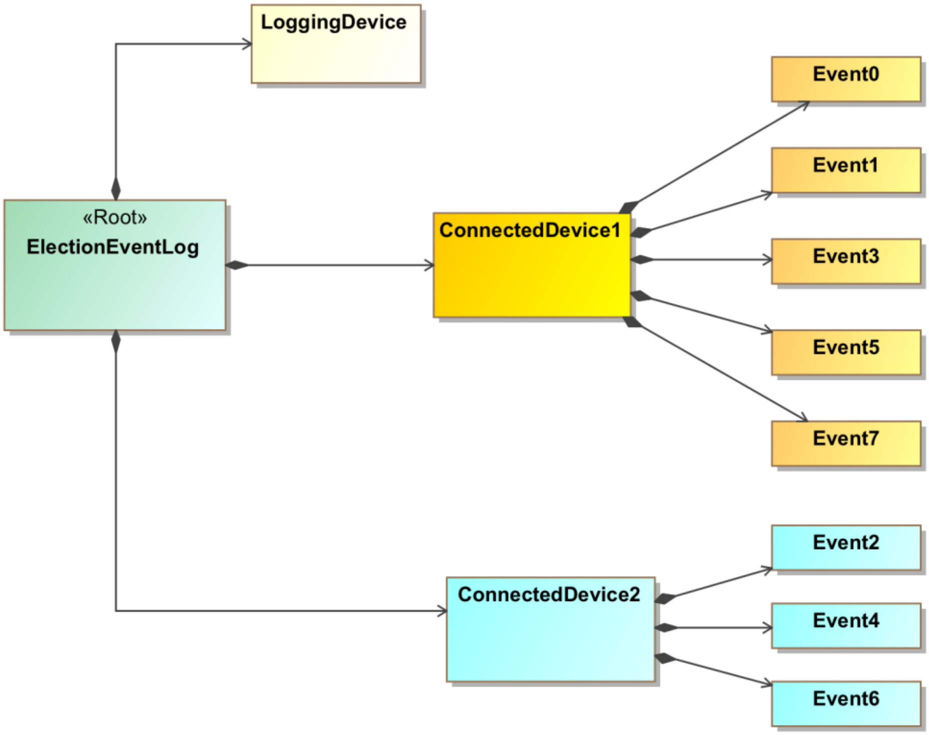

Figure 4 - Third use case for a logging device connected to event generating devices

Figure 4 shows the associations for the third use case in which a logging device is connected to multiple devices. The connected devices would send election events to the logging device, which writes events to the log file and maintains the correspondence between each device and its events. Figure 4 does not show the logging device itself generating any events, however this could happen in practice.

2.4 Documentation schema

To analyze an election log, one must understand the meaning of the various event IDs used in the log as well as other fields. Currently, each manufacturer can use their own specific event IDs, thus there is no consistency of the ID values or their meaning across manufacturers.

One approach to making event IDs easier to understand when analyzing multiple manufacturer devices is to make them consistent in meaning and use. One could provide an event ID lexicon that would describe all known events and assign definitions/meanings to them, and each manufacturer could use this lexicon in all its election logs. Ultimately, however, this approach was not taken in this specification, as defining such a lexicon across all devices as well as future devices will be time-consuming and may still require periodic updates as new devices come on the market.

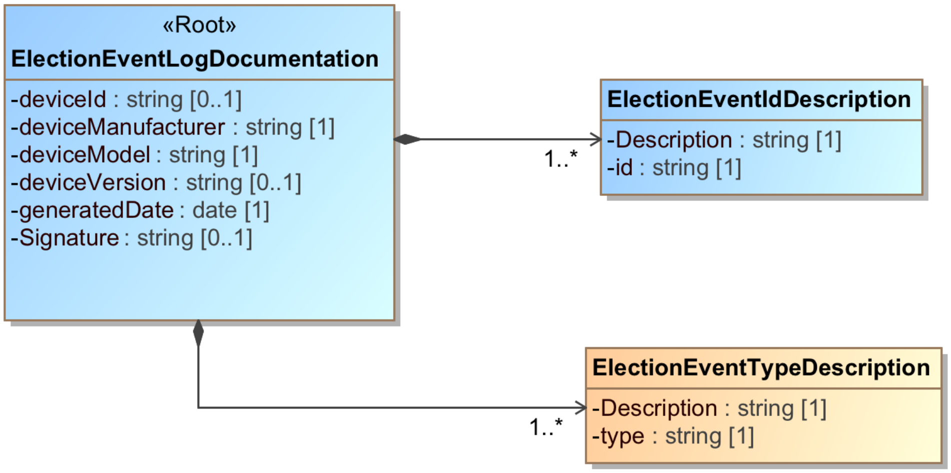
«Root»
ElectionEventLogDocumentation
-deviceId : string [0..1]
-deviceManufacturer : string [1]
-deviceModel : string [1]
-deviceVersion : string [0..1]
-generatedDate : date [1]
-Signature : string [0..1]
ElectionEventIdDescription
-Description : string [1]
-id : string [1]
1..*
ElectionEventTypeDescription
-Description : string [1]
-type : string [1]
1..*

Figure 5 - Election Event Logging Documentation UML Class Diagram

The approach taken by this specification is to provide a second Election Event Logging Documentation XML schema for containing the event ID documentation, along with documentation for each event type. This format includes fields for device and manufacturer documents, additional details, and then a series of fields for containing each event ID and its corresponding definition, and each event type and its corresponding definition. Manufacturers could include a documentation file in this format for each of their devices. While a standard, consistent lexicon may be simpler, the approach taken here provides more flexibility for manufacturers and permits development of new event IDs as necessary, e.g., for new devices.

3 XML Schema Documentation

This section contains documentation and discussion of the features included in the Election Event Logging and Election Event Logging Documentation XML schemas.

3.1 Schema Stylistic Conventions

The XML schemas were written observing the following stylistic conventions:

- Element and attribute names observe variations of "CamelCase" conventions[2], that is:
  - Element and enumeration and primitive names begin with a capital letter and names that consist of multiple words are concatenated and each word begins with a capital letter, thus "CamelCase". For example, *<ElectionEventLog>*.
  - Enumeration value names are in non-capital letters, and names that consist of multiple words are separated by hyphens. For example, *scan-central*.
- Element and enumeration value ordering is generally alphabetical, with the following exceptions:
  - Element (or attribute) names such as *<Type>* are followed by *<OtherType>*.
  - If there is an enumeration value of *other*, it comes last in the list of values.

In the sections below, an element or an enumeration name is denoted using italics and angle brackets, e.g., *<ElectionEvent>*. Enumeration values are in italics, e.g., *other*. An element is sometimes referred to as a "sub-element" when it is included in another element, e.g., *<ElectionEvent>* is a sub-element of *<ElectionEventLog>*. "Includes" is used to denote that an element contains another element as a sub-element, e.g., *<ElectionEventLog>* includes *<ElectionEvent>*.

3.2 Elements and Complex Types - Election Event Logging Schema

The following sections deal with major elements and complex types in the Election Event Logging schema. The XML format includes an *<ElectionEventLog>* root element, followed by potentially multiple <*Device*> elements, and then multiple occurrences of *<ElectionEvent>* elements, the elements corresponding to the generating devices. A brief example showing only the attributes and sub-elements that are required is as follows:

```
<ElectionEventLog generatedTime="2013-13-01T17:45:52.629000">
  <Device deviceId=”XDL31021A”>
    <ElectionEvent id="1004022" timeStamp="2013-12-13T16:51:43" type="P_LOG"/>
    <ElectionEvent id="1004022" timeStamp="2013-12-13T16:51:55" type="P_LOG"/>
    <ElectionEvent id="1004022" timeStamp="2013-12-13T16:52:00" type="P_LOG"/>
  </Device>
</ElectionEventLog>
```

[2] See https://en.wikipedia.org/wiki/CamelCase.

3.2.1 The <ElectionEventLog> Element

<ElectionEventLog> is the root element. It includes one or more *<Device>* elements for identifying the device(s) generating the election events, the date and time when the election event log was created, and an identification of the election. *<Details>* is used as needed for additional description/details.

The optional *<Signature>* element is used for an XML digital signature [4]. *<Signature>* must be the last element of *<ElectionEventLog>*.

Table 3.1 – Attributes for <ElectionEventLog>

Attribute	Required	Type	Attribute Description
<electionId>	no	xsd:string	Identifies the election associated with the log.
<generatedTime>	yes	xsd:dateTime	Identifies the date and time the log was generated.

Table 3.2 – Elements for <ElectionEventLog>

Element	Multiplicity	Type	Element Description
<Details>	0 or 1	xsd:string	Used to associate any details with the event log.
<Device>	0 or more	ElectionEvent	Used to describe the device(s) generating the election events.
<Signature>	0 or 1	Signature	Reference to the *<Signature>* element of the W3C digital signature schema imported into this schema.

Schema Definition:

```
<xsd:complexType name="ElectionEventLog">
  <xsd:sequence>
    <xsd:element name="Details" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Device" type="Device" minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element ref="ds:Signature" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="electionId" type="xsd:string"/>
  <xsd:attribute name="generatedTime" type="xsd:dateTime" use="required"/>
</xsd:complexType>"/>
```

3.2.2 The <Device> Element

The Device element holds information about the device generating election event logs. *<id>* is the only required attribute, all other attributes and elements are optional. If the device usage is not found in the *DeviceUsage* enumeration, <*usage*> should be *other* and <*otherUsage*> should contain the appropriate usage.

Table 3.3 – Attributes for <Device>

Attribute	Required	Type	Attribute Description
<id>	yes	xsd:string	A serial number or otherwise identifier associated with the device.
<manufacturer>	no	xsd:string	Manufacturer of the device.
<model>	no	xsd:string	Model of the device.
<usage>	no	DeviceUsage	Enumerated usage of the device, e.g., DRE, opscan-precinct, etc.
<otherUsage>	no	xsd:string	Used when *<deviceUsage>* is *other*.
<version>	no	xsd:string	Version identification of the device.

Table 3.4 – Elements for <Device>

Element	Multiplicity	Type	Element Description
<Details>	0 or 1	xsd:string	Used to associate any details with the event log.
<ElectionEvent>	1 or more	ElectionEvent	Used to describe a logged event.

Schema Definition:

```
<xsd:complexType name="Device">
  <xsd:sequence>
    <xsd:element name="Details" type="xsd:string" minOccurs="0"/>
    <xsd:element name="ElectionEvent" type="ElectionEvent" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
  <xsd:attribute name="manufacturer" type="xsd:string"/>
  <xsd:attribute name="model" type="xsd:string"/>
  <xsd:attribute name="usage" type="DeviceUsage"/>
  <xsd:attribute name="otherUsage" type="xsd:string"/>
  <xsd:attribute name="version" type="xsd:string"/>
</xsd:complexType>
```

3.2.3 The <ElectionEvent> Element

<ElectionEvent> holds information about a specific event. <*severity*> is an optional attribute for describing a severity indication for the event. If the event disposition is not found in the

DispositionType enumeration, <*disposition*> should be *other* and <*otherDisposition*> should be used to contain the other disposition.

Table 3.5 – Attributes for <ElectionEvent>

Attribute	Required	Type	Attribute Description
<disposition>	no	DispositionType	The disposition, e.g., success or failure, of the event.
<otherDisposition>	no	xsd:string	Used when <*dispositionType*> is *other*.
<id>	yes	xsd:string	An identifier associated with the event.
<severity>	no	xsd:string	Used for an indication of the severity of the event, as determined by the device vendor.
<timeStamp>	yes	xsd:dateTime	Identifies the date and time the event was generated.
<type>	yes	xsd:string	Used for the type of event, as determined by the device vendor.
<userId>	no	xsd:string	An identifier associated with a user, as relevant.

Table 3.6 – Elements for <ElectionEvent>

Element	Multiplicity	Type	Element Description
<Description>	0 or 1	xsd:string	Used for a brief description of the event.
<Details>	0 or 1	xsd:string	Used for additional information about the event, e.g., vendor reserved information.

Schema Definition:

```
<xsd:complexType name="ElectionEvent">
  <xsd:sequence>
    <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Details" type="xsd:string" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="disposition" type="DispositionType"/>
  <xsd:attribute name="otherDisposition" type="xsd:string"/>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
  <xsd:attribute name="severity" type="xsd:string"/>
  <xsd:attribute name="timeStamp" type="xsd:dateTime" use="required"/>
  <xsd:attribute name="type" type="xsd:string" use="required"/>
  <xsd:attribute name="userId" type="xsd:string"/>
</xsd:complexType>
```

3.3 Elements and Complex Types - Election Event Logging Documentation Schema

The following sections deal with major elements and complex types in the Election Event Logging Documentation schema. The purpose of the schema is to provide a format for vendor documentation of the election event identifiers used in the log files. Rather than requiring vendors to standardize on a specific set of election event identifiers and their meaning, the approach represented here is for vendors to use their own specific identifiers but to provide descriptions for those identifiers. The XML format represented by the schema is very simple, with a root *<ElectionEventLogDocumentation>* element followed by multiple occurrences of *<ElectionEventIdDescription>* elements and *<ElectionEventTypeDescription>* elements, each element providing a description for an election event identifier or election event type. A brief example is as follows:

```
<ElectionEventLogDocumentation deviceManufacturer="Blackburd" deviceModel="SR-71"
generatedDate="2010-10-01T16:50:46">
  <ElectionEventIdDescription id="1004022">
    <Description>Voting session complete</Description>
  </ElectionEventIdDescription>
  <ElectionEventIdDescription id="1004150">
    <Description>Attempting to Close Poll</Description>
  </ElectionEventIdDescription>
  <ElectionEventIdDescription id="6004041">
    <Description>Close process complete</Description>
  </ElectionEventIdDescription>
  <ElectionEventTypeDescription id="INFO">
    <Description>Informative message</Description>
  </ElectionEventTypeDescription>
  <ElectionEventTypeDescription id="CRT">
    <Description>Critical exception</Description>
  </ElectionEventTypeDescription>
</ElectionEventLogDocumentation>
```

3.3.1 The <ElectionEventIdDescription> Complex Type

For associating a brief description with an election event log ID.

Table 3.7 – Attributes for <ElectionEventIdDescription>

Attribute	Required	Type	Attribute Description
<id>	yes	xsd:string	An identifier associated with the event.

Table 3.8 – Elements for <ElectionEventIdDescription>

Element	Multiplicity	Type	Element Description
<Description>	1	xsd:string	Used for a brief description of the event.

Schema Definition:

```
<xsd:complexType name="ElectionEventIdDescription">
  <xsd:sequence>
    <xsd:element name="Description" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string" use="required"/>
</xsd:complexType>
```

3.3.2 The <ElectionEventLogDocumentation> Complex Type

<ElectionEventLogDocumention> is the root element. It includes one or more *<ElectionEventIdDescription>* elements and *<ElectionEventTypeDescription>* elements, as well as other information for identifying the specific device associated with the election event documentation.

The optional *<Signature>* element is used for an XML digital signature. *<Signature>* must be the last element of *<ElectionEventLogDocumention>*.

Table 3.9 – Attributes for <ElectionEventLogDocumentation>

Attribute	Required	Type	Attribute Description
<deviceId>	no	xsd:string	A serial number or otherwise identifier associated with the device.
<deviceManufacturer>	yes	xsd:string	Manufacturer of the device.
<deviceModel>	yes	xsd:string	Model of the device.
<deviceVersion>	no	xsd:string	Version identification of the device.
<generatedDate>	yes	xsd:date	Identifies the date the documentation report was generated.

Table 3.10 – Elements for <ElectionEventLogDocumentation>

Element	Multiplicity	Type	Element Description
<ElectionEventIdDescription>	1 or more	ElectionEventIdDescription	For associating a description with an event ID.
<ElectionEventTypeDescription>	1 or more	ElectionEventTypeDescription	For associating a description with an event type.
<Signature>	0 or 1	Signature	Reference to the *<Signature>* element of the W3C digital signature schema imported into this schema.

Schema Definition:

```
<xsd:complexType name="ElectionEventLogDocumentation">
  <xsd:sequence>
    <xsd:element name="ElectionEventIdDescription" type="ElectionEventIdDescription" maxOccurs="unbounded"/>
    <xsd:element name="ElectionEventTypeDescription" type="ElectionEventTypeDescription"
maxOccurs="unbounded"/>
    <xsd:element ref="ds:Signature" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="generatedDate" type="xsd:date" use="required"/>
  <xsd:attribute name="deviceId" type="xsd:string"/>
  <xsd:attribute name="deviceManufacturer" type="xsd:string" use="required"/>
  <xsd:attribute name="deviceModel" type="xsd:string" use="required"/>
  <xsd:attribute name="deviceVersion" type="xsd:string"/>
</xsd:complexType>
```

3.3.3 The <ElectionEventTypeDescription> Complex Type

For associating a brief description with an election event log type.

Table 3.11 – Attributes for <ElectionEventTypeDescription>

Attribute	Required	Type	Attribute Description
<type>	yes	xsd:string	An identifier associated with the event type.

Table 3.12 – Elements for <ElectionEventIdDescription>

Element	Multiplicity	Type	Element Description
<Description>	1	xsd:string	Used for a brief description of the event type.

Schema Definition:

```
<xsd:complexType name="ElectionEventTypeDescription">
  <xsd:sequence>
    <xsd:element name="Description" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="type" type="xsd:string" use="required"/>
</xsd:complexType>
```

3.4 Enumerations

The following sections deal with simple type enumerations in the Election Event Logging schema.

3.4.1 The DeviceUsage Enumeration

Enumeration for the usage of the device in the *<deviceUsage>* attribute.

Table 3.13 – Values for <DeviceUsage>

Value	Value Description
adjudication	Electronic adjudication function for reviewing absentee/mail-in ballots anomalies (blanks/overvotes/write-ins/unreadable ballots).
ballot-activation	Devices for enabling a vote capture device (VCD) to display a ballot, possibly directly connected to the VCD or through a smart card interface.
ballot-marking	Ballot marking devices (voter facing).
ballot-printing	Marked ballot printing devices (voter facing).
blank-ballot-printing	On-demand blank ballot printers.
dre	Electronic voter stations, standalone or daisy chained to a DRE-controller (voter facing).
dre-controller	Network controller for electronic voting (poll worker facing)
electronic-cast	DREs, or other devices that store cast vote records electronically (voter facing).
electronic-cast-paper	DREs, or devices that store cast vote records electronically and also print a paper record (voter facing).
electronic-poll-book	Electronic poll book devices.
ems	Election management systems, including for pre- and post-election administration and reporting functions.
scan-batch	Scanning devices for batches of ballots, auto-feeding, e.g., Central Count (poll worker facing).
scan-single	Scanning devices for single-sheets, e.g., Precinct Count (voter facing), but could be used for Central Count by an election official.
transmission-sending	Remote transmission clients, e.g., for sending of unofficial results from a remote location to a central location (sending station).
transmission-receiving	Remote transmission hosts, e.g., for the receiving of unofficial results at a central location from a remote location (receiving station).
other	Used when the device type is not listed in this enumeration.

Schema Definition:

```
<xsd:simpleType name="DeviceUsage">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="adjudication"/>
    <xsd:enumeration value="ballot-activation"/>
    <xsd:enumeration value="ballot-marking"/>
    <xsd:enumeration value="ballot-printing"/>
    <xsd:enumeration value="blank-ballot-printing"/>
    <xsd:enumeration value="dre"/>
    <xsd:enumeration value="dre-controller"/>
    <xsd:enumeration value="electronic-cast"/>
    <xsd:enumeration value="electronic-cast-paper"/>
    <xsd:enumeration value="electronic-poll-book"/>
    <xsd:enumeration value="ems"/>
    <xsd:enumeration value="scan-batch"/>
    <xsd:enumeration value="scan-single"/>
    <xsd:enumeration value="transmission-sending"/>
    <xsd:enumeration value="transmission-receiving"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
```

3.4.2 The DispositionType Enumeration

Enumeration for types of dispositions in the *<disposition>* attribute.

Table 3.14 – Values for <DispositionType>

Value	Value Description
failure	For a failure disposition.
na	Used when the disposition is not applicable or there is no disposition.
success	For a successful disposition.
other	Used when the type of disposition is not included in this enumeration.

Schema Definition:

```
<xsd:simpleType name="DispositionType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="failure"/>
    <xsd:enumeration value="na"/>
    <xsd:enumeration value="success"/>
    <xsd:enumeration value="other"/>
  </xsd:restriction>
</xsd:simpleType>
```

Appendix A—Acronyms

Selected acronyms and abbreviations used in this document are defined below.

CDF	Common Data Format
DRE	Direct Record Electronic
EAC	Election Assistance Commission
EMS	Election Management System
ISO	International Standards Organization
JSON	JavaScript Object Notation
NIST	National Institute of Standards and Technology
SP	Special Publication
UML	Unified Modeling Language
VVSG	Voluntary Voting Systems Guidelines
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

Appendix B—Glossary

Selected terms used throughout this document are defined below. In some of the definitions, there is ancillary information that is not part of the definition but helpful in understanding the definition; this ancillary information is preceded with "*Note:*". Synonyms are preceded with "*Syn:*".

Adjudication: As used in elections, may be applied to the process of resolving centrally processed paper ballots that have been flagged for various reasons, including write-ins, overvotes, marginal marks, having no contests marked on the entire ballot, or the ballot being unreadable by a tabulator.

Direct record electronic (DRE): An electronic vote-capture device that provides choices visible to the voter on a front panel of the machine in which voters directly enter choices into electronic storage with the use of a touch-screen, pushbuttons, or similar device. *Note:* An alphabetic keyboard is often provided with the entry device to allow for the possibility of write-in votes. The voter's choices are stored in these machines and added to the choices of all other voters.

Election management system (EMS): Computer systems used to perform such tasks as preparing ballots, setting up tally systems, maintaining voter registration information, generating reports, and to consolidate, report, and display election results. *Note:* This device receives results data from the vote-capture devices or by manual input, accumulates the results, and reports the accumulated results.

Overvote: Occurs when a voter selects more than one candidate in a 1-of-M contest or more than N candidates in an N-of-M contest. The vote for that contest is considered an overvote and not counted towards any candidate in that contest (unless approval voting applies for that contest). *Note:* Usually the rest of a properly marked ballot is counted. Large numbers of overvotes can be indicative of confusing ballot layout or confusing instructions.

Scanner: A device that scans a marked ballot and interprets the voter's marks so as to produce a record of the voter's choices on the ballot. A precinct-count scanner generally is used in a polling place by voters to individually scan ballots; the scanner is generally configured to warn the voter if there are mistakes on the ballot such as overvotes or undervotes. A central-count scanner generally is used to high-speed scan large amounts of ballots in a batch mode, with no opportunity for voter correction of mistakes on the ballot.

Tabulator: A programmed device that counts votes.

Undervote: Occurs when the voter does not select a candidate in a 1-of-M contest or selects fewer than N candidates in an 1-of-M contest. *Note:* can indicate a conscious choice of the voter to not vote in the contest. As with over votes, large numbers of under votes can be indicative of confusing ballot layout or confusing instructions.

Vote-capture device: Device that is used directly by a voter to cast a ballot. *See also:* direct record electronic (DRE).

Appendix C—References

[1] W3C, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, W3C Recommendation, November 26, 2008, http://www.w3.org/TR/xml/ [accessed 2/1/2016].

[2] Election Assistance Commission, *Voluntary Voting System Guidelines Version 1.1 Volume 1*, https://eac926.ae-admin.com/assets/1/Documents/VVSG.1.1.VOL.1.FINAL.pdf [accessed 3/31/2016].

[3] Object Management Group (OMG), *UML Specification version 1.1* (OMG document ad/97-08-11) September 22, 2011, http://omg.org/ [accessed 2/1/2016].

[4] W3C, *XML Signature Syntax and Processing (Second Edition)*, W3C Recommendation, June 10, 2008, http://www.w3.org/TR/xmldsig-core/ [accessed 2/1/2016].

Appendix D—File Download Locations

The files associated with this specification are available for download from a NIST repository. These files are:

- This specification,
- XML schema,
- Example XML files,
- Validation tools, and
- UML model.

Other files or updates to the files may be added. The repository URL is located at:

http://vote.nist.gov

Appendix E—Election Event Logging XML Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- Version 1.0, 2016-04-15, Election Event Logging, National Institute of Standards and Technology -->
<!-- See the NIST software disclaimer http://www.nist.gov/public_affairs/disclaimer.cfm -->
<xsd:schema xmlns="NIST_V1_election_event_logging.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="NIST_V1_election_event_logging.xsd"
elementFormDefault="qualified" version="1.0">
  <!-- ========== Imports ========== -->
  <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="http://www.w3.org/2000/09/xmldsig#"/>
  <!-- ========== Roots ========== -->
  <xsd:element name="ElectionEventLog" type="ElectionEventLog"/>
  <!-- ========== Primitives ========== -->
  <!-- ========== Enumerations ========== -->
  <xsd:simpleType name="DeviceUsage">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="adjudication"/>
      <xsd:enumeration value="ballot-activation"/>
      <xsd:enumeration value="ballot-marking"/>
      <xsd:enumeration value="ballot-printing"/>
      <xsd:enumeration value="blank-ballot-printing"/>
      <xsd:enumeration value="dre"/>
      <xsd:enumeration value="dre-controller"/>
      <xsd:enumeration value="electronic-cast"/>
      <xsd:enumeration value="electronic-cast-paper"/>
      <xsd:enumeration value="electronic-poll-book"/>
      <xsd:enumeration value="ems"/>
      <xsd:enumeration value="scan-batch"/>
      <xsd:enumeration value="scan-single"/>
      <xsd:enumeration value="transmission-sending"/>
      <xsd:enumeration value="transmission-receiving"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="DispositionType">
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="failure"/>
      <xsd:enumeration value="na"/>
      <xsd:enumeration value="success"/>
      <xsd:enumeration value="other"/>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- ========== Elements ========== -->
  <xsd:complexType name="Device">
    <xsd:sequence>
      <xsd:element name="Details" type="xsd:string" minOccurs="0"/>
      <xsd:element name="ElectionEvent" type="ElectionEvent" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="manufacturer" type="xsd:string"/>
    <xsd:attribute name="model" type="xsd:string"/>
    <xsd:attribute name="usage" type="DeviceUsage"/>
    <xsd:attribute name="otherUsage" type="xsd:string"/>
    <xsd:attribute name="version" type="xsd:string"/>
  </xsd:complexType>
  <xsd:complexType name="ElectionEvent">
    <xsd:sequence>
      <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
      <xsd:element name="Details" type="xsd:string" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="disposition" type="DispositionType"/>
    <xsd:attribute name="otherDisposition" type="xsd:string"/>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="severity" type="xsd:string"/>
    <xsd:attribute name="timeStamp" type="xsd:dateTime" use="required"/>
    <xsd:attribute name="type" type="xsd:string" use="required"/>
    <xsd:attribute name="userId" type="xsd:string"/>
  </xsd:complexType>
  <xsd:complexType name="ElectionEventLog">
    <xsd:sequence>
      <xsd:element name="Details" type="xsd:string" minOccurs="0"/>
      <xsd:element name="Device" type="Device" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="ds:Signature" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="electionId" type="xsd:string"/>
    <xsd:attribute name="generatedTime" type="xsd:dateTime" use="required"/>
  </xsd:complexType>
</xsd:schema>
```

Appendix F—Election Event Logging Documentation XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Version 1.0, 2016-04-03,  Election Event Logging Documentation, National Institute of Standards and Technology -->
<!-- See the NIST software disclaimer http://www.nist.gov/public_affairs/disclaimer.cfm -->
<xsd:schema xmlns="NIST_V1_election_event_logging_documentation.xsd" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="NIST_V1_election_event_logging_documentation.xsd"
elementFormDefault="qualified" version="1.0">
  <!-- ========== Imports ========== -->
  <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#" schemaLocation="http://www.w3.org/2000/09/xmldsig#"/>
  <!-- ========== Roots ========== -->
  <xsd:element name="ElectionEventLogDocumentation" type="ElectionEventLogDocumentation"/>
  <!-- ========== Primitives ========== -->
  <!-- ========== Enumerations ========== -->
  <!-- ========== Elements ========== -->
  <xsd:complexType name="ElectionEventIdDescription">
    <xsd:sequence>
      <xsd:element name="Description" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="ElectionEventLogDocumentation">
    <xsd:sequence>
      <xsd:element name="ElectionEventIdDescription" type="ElectionEventIdDescription" maxOccurs="unbounded"/>
      <xsd:element name="ElectionEventTypeDescription" type="ElectionEventTypeDescription" maxOccurs="unbounded"/>
      <xsd:element ref="ds:Signature" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="deviceId" type="xsd:string"/>
    <xsd:attribute name="deviceManufacturer" type="xsd:string" use="required"/>
    <xsd:attribute name="deviceModel" type="xsd:string" use="required"/>
    <xsd:attribute name="deviceVersion" type="xsd:string"/>
    <xsd:attribute name="generatedDate" type="xsd:date" use="required"/>
  </xsd:complexType>
  <xsd:complexType name="ElectionEventTypeDescription">
    <xsd:sequence>
      <xsd:element name="Description" type="xsd:string"/>
    </xsd:sequence>
    <xsd:attribute name="type" type="xsd:string" use="required"/>
  </xsd:complexType>
</xsd:schema>
```