

NEMESIS V3 Demographic (Agency) Data Guide

Date

November 17, 2011 (Final)
February 1, 2012 (Updated)
January 27, 2017 (Rewritten)

Authors

Joshua Legler – NEMESIS Consultant
Karen E. Jacobson – NEMESIS Director
N. Clay Mann – NEMESIS P.I.

Contributors

Aaron Hart – ZOLL
Chad Perry – Golden Hour
Chris Oloa – IAED
Dan Vanorny, Kevin Ketcher – ImageTrend
David Saylor – Beyond Lucid
Doug Avery – Starwest Tech
Ernie Doss – State of Georgia
Halyna Petetrechko, Iryna Domchak, Jennifer
Correa, Robyn Hughes – AMR

Jeff Robertson – EMS PIC
Jeremy Pate – University of Alabama
Joe Williams – American Heart Association
Jon Burkhart – Intermedix
Juan Esparza – State of Florida
Matthew Curran, Sushil Kamble – Forte
Holdings
Steve Kimball – EMS Data Systems
Taylor Smits – Forward Health Group

Overview

In NEMESIS V3, “Collect and Send” systems at the local agency level send agency demographic data to “Receive and Process” systems at the state level. In turn, the state-level systems send some agency demographic data (“national elements”) to the national EMS database.

This document describes requirements regarding what demographic data to send and the frequency with which to send it, both from an agency to a state, and from a state to the national EMS database.

This document also provides guidance to developers and administrators of “Receive and Process” systems regarding the processing of agency demographic data updates received from “Collect and Send” systems. The following topics are addressed:

- Determining which data elements to process when receiving updates
- Identifying new, updated, and deleted object instances and handling them appropriately
- Using Schematron rules to require unique identifiers for object instances

When to Send Demographic Data

Initially

“Collect and Send” systems SHOULD send agency demographic data to state systems before sending any PCR data for an agency.

“Receive and Process” systems sending data to the national EMS database MUST send agency demographic data at least 24 hours prior to sending any PCR data for an agency.

As Updated

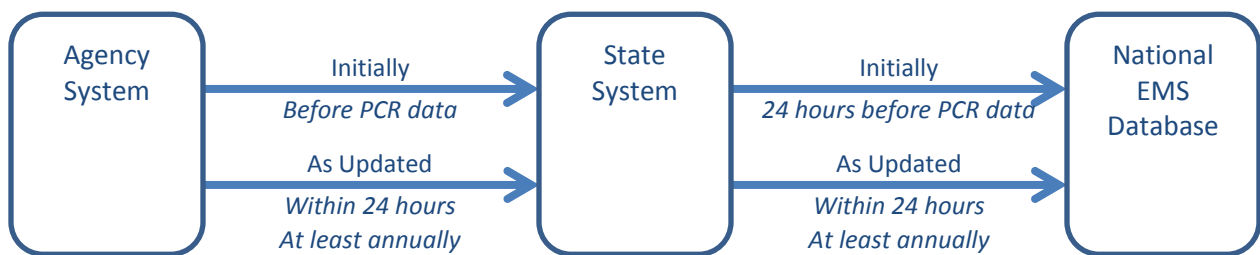
When the value of any national element changes in an agency’s demographic data in a “Collect and Send” system, the system SHOULD send updated agency demographic data to the state system within 24 hours. A “Collect and Send” system SHOULD also be capable of sending updated agency demographic data when triggered by changes in the value of any state-required element.

When the value of any national element changes in an agency’s demographic data in a “Receive and Process” system sending data to the national EMS database, the system MUST send updated agency demographic data to the national EMS database within 24 hours.

A system is not required to send demographic data more frequently than daily, but it may do so. If a system is not capable of monitoring national demographic data elements for changes, it may simply send updated demographic data on a scheduled daily basis, regardless of whether data have changed.

Annually

Agency annual statistics (dAgency.AgencyYearGroup) should be updated at least annually, so the minimum allowable frequency of demographic data updates is annually.



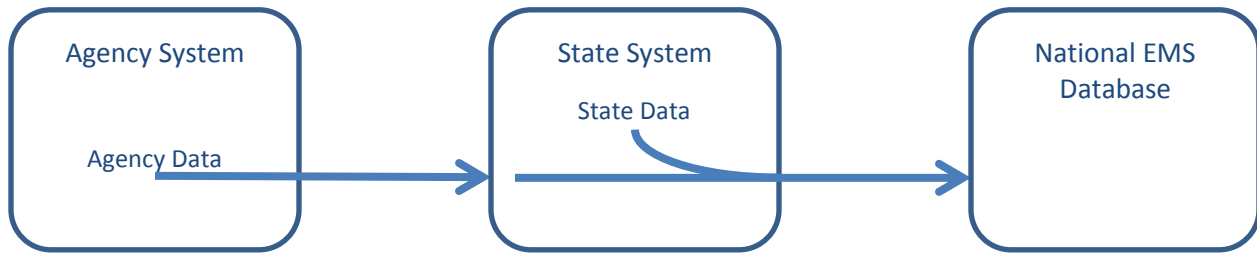
What Demographic Data to Send

“Collect and Send” systems MUST be capable of sending the agency demographic data elements requested or required by a state. “Collect and Send” systems SHOULD give administrators the ability to configure which data elements to send.

“Receive and Process” systems sending data to the national EMS database MUST send only the “national elements” as defined in the NEMSIS Data Dictionary.

For the data elements in the following list, state systems MAY receive and process values submitted from agencies. However, state systems sending data to the national EMS database MUST send values that correctly represent the submitting state and are consistent across agencies. In other words, for the following data elements, the state system should “inject” and send correct information as determined by the state, rather than simply forwarding information as received from agencies.

- dConfiguration.01 State Associated with the Certification/Licensure Levels
- dConfiguration.02 State Certification/Licensure Levels
- dConfiguration.03 Procedures Permitted by the State
- dConfiguration.04 Medications Permitted by the State
- dConfiguration.05 Protocols Permitted by the State



Processing Demographic Data

The remainder of this guide provides recommendations for “Receive and Process” systems to effectively implement the processing of demographic data received from other systems.

Authoritative Source / System of Record

In compliance testing, “Receive and Process” systems must demonstrate the ability to receive and process data for all NEMSIS data elements and send national elements to the national EMS database. However, states may have various business requirements regarding the processing of incoming demographic data.

The information about objects, attributes, and relationships in agency demographic data usually comes from a mixture of sources. For example, the state may be the “authoritative source,” or “system of record,” regarding the ID and name of an agency, while the agency may be the source of information regarding the agency’s statistical year information. As a result, states may require their systems to selectively process information received from agency systems. For example, when processing demographic data received from an agency system, the state system may be configured to ignore the data value contained in dAgency.03 EMS Agency Name, because the state is the licenser of the agency and has regulatory authority over agency name changes.

Each state should identify the authoritative source of each type of object, attribute, and relationship in demographic data and work with the developer of its state data system to implement a system configuration that meets the state’s business needs.

See Appendix A: Authoritative Source, p. 10, for recommendations regarding the authoritative source of specific objects, attributes, and relationships in agency demographic data.

Identifying New, Updated, and Deleted Information

The NEMSIS 3.4 standard (and below) does not explicitly define unique identifiers, or keys, for objects in demographic data. As a result, it can be difficult for a “Receive and Process” system to handle changes in demographic data over time.

“Receive and Process” systems may consider the following options for handling updated demographic data. The illustrations below will use medical devices (dDevice.DeviceGroup) as an example but are applicable to other objects in demographic data.

Delete + Insert (or Deactivate + Insert)

The simplest approach to handling updated demographic data is to simply discard or de-activate all previously-received data (by deleting it or marking it inactive in the database) and insert the newly received data.

When submission #1 is received, the receiving system inserts all data into the database. When submission #2 is received, the receiving system first deletes or deactivates existing records and then inserts new records for the newly received data.

Submission #1

dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
10101	Defib1	AED	Acme	LS100	1/1/2016
10102	Defib2	AED	Acme	LS100	6/1/2016
10103	Defib3	AED	Acme	LS100	9/1/2016

“Receive and Process” System After Submission #1

Key	dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
1	<u>10101</u>	<u>Defib1</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>1/1/2016</u>
2	<u>10102</u>	<u>Defib2</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>6/1/2016</u>
3	<u>10103</u>	<u>Defib3</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>9/1/2016</u>

Submission #2

dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
10102	Defib2	AED	Acme Inc.	LS100	6/1/2016
10103	Defib3	AED	Acme	LS100	9/1/2016
10104	Defib4	AED	Acme	LS100	12/1/2016

“Receive and Process” System After Submission #2

Key	dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
1	10101	Defib1	AED	Acme	LS100	1/1/2016
2	10102	Defib2	AED	Acme	LS100	6/1/2016
3	10103	Defib3	AED	Acme	LS100	9/1/2016
4	<u>10102</u>	<u>Defib2</u>	<u>AED</u>	<u>Acme Inc.</u>	<u>LS100</u>	<u>6/1/2016</u>
5	<u>10103</u>	<u>Defib3</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>9/1/2016</u>
6	<u>10104</u>	<u>Defib4</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>12/1/2016</u>

Pros

- Simple
- Minimizes data storage requirements if records are deleted rather than deactivated
- Guaranteed to represent current status
- Does not require unique keys for records in incoming data

Cons

- Loses history
- Loses referential integrity if records are deleted rather than deactivated

Delete + Insert + Update

This approach attempts to identify objects by their keys. See Appendix B: Object Keys (Unique Identifiers), p. 12, for a list of recommended key elements for each object in demographic data. In this example, the receiving system has been configured to use dDevice.01 Device Serial Number as the key for incoming data.

Submission #1 is simply inserted into the database as new records. For submission #2, the receiving system uses the key (dDevice.01) to uniquely identify each object. The system finds device 10102 in the database but finds that some previously-received data does not match the new incoming data, so the system performs an update. The system finds device 10103 in the database and recognizes that none of its data have changed. The system does not find device 10104 in the database, so it inserts a new record. The system deletes or de-activates any records in the database for which there is no object in the incoming data; as a result, the system deletes or de-activates the record for device 10101, because it does not exist in submission #2.

Submission #1

dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
10101	Defib1	AED	Acme	LS100	1/1/2016
10102	Defib2	AED	Acme	LS100	6/1/2016
10103	Defib3	AED	Acme	LS100	9/1/2016

“Receive and Process” System After Submission #1

Key	dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
<u>1</u>	<u>10101</u>	<u>Defib1</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>1/1/2016</u>
<u>2</u>	<u>10102</u>	<u>Defib2</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>6/1/2016</u>
<u>3</u>	<u>10103</u>	<u>Defib3</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>9/1/2016</u>

Submission #2

dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
10102	Defib2	AED	Acme Inc.	LS100	6/1/2016
10103	Defib3	AED	Acme	LS100	9/1/2016
10104	Defib4	AED	Acme	LS100	12/1/2016

“Receive and Process” System After Submission #2

Key	dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
1	10101	Defib1	AED	Acme	LS100	Jan 1 2016
2	10102	Defib2	AED	Acme Inc.	LS100	Jun 1 2016
3	10103	Defib3	AED	Acme	LS100	Sep 1 2016
<u>4</u>	<u>10104</u>	<u>Defib4</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>Dec 1 2016</u>

Pros

- Minimizes database storage requirements
- Prevents duplicates

Cons

- Loses history
- Requires unique keys for records in incoming data
- Receiving system must compare data within each element to determine whether an update is needed

Versioning

This approach uses data warehousing “slowly changing dimension” concepts to maintain a historically consistent snapshot of demographic data. In other words, a system implementing this approach can know what an agency’s demographic data looked like at any given point in the past.

This approach, like the Delete + Insert + Update approach, uses keys to identify objects in the incoming data. See Appendix B: Object Keys (Unique Identifiers), p. 12, for a list of recommended key elements for each object in demographic data.

The database has some additional columns: “From,” to track the date on which a record became active, and “To,” the date on which a record became inactive. (For convenience, most designs also include a third column containing a flag that indicates whether or not the record is currently active.)

In this approach, every change to an object triggers the insertion of a new record to represent the new version of that object.

Submission #1 is inserted into the database. For submission #2, the receiving system uses the key (dDevice.01) to uniquely identify each object. The system finds device 10102 in the database but finds that some previously-received data does not match the new incoming data, so the system updates the existing record by setting the “To” date to the date on which submission #2 was received; then the system inserts a new record to represent the new version of device 10102, setting the “From” date to the date on which submission #2 was received. The system finds device 10103 in the database and recognizes that none of its data have changed. The system does not find device 10104 in the database, so it inserts a new record, setting the “From” date to the date on which submission #2 was received. Finally, the system updates any records in the database for which there is no object in the incoming data, by setting the “To” date to the date on which submission #2 was received.

Submission #1

dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
10101	Defib1	AED	Acme	LS100	1/1/2016
10102	Defib2	AED	Acme	LS100	6/1/2016
10103	Defib3	AED	Acme	LS100	9/1/2016

“Receive and Process” System After Submission #1

Key	dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date	From	To
1	<u>10101</u>	<u>Defib1</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>1/1/2016</u>		
2	<u>10102</u>	<u>Defib2</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>6/1/2016</u>		
3	<u>10103</u>	<u>Defib3</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>9/1/2016</u>		

Submission #2

dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date
10102	Defib2	AED	Acme Inc.	LS100	6/1/2016
10103	Defib3	AED	Acme	LS100	9/1/2016
10104	Defib4	AED	Acme	LS100	12/1/2016

“Receive and Process” System After Submission #2

Key	dDevice.01 Serial #	dDevice.02 Name or ID	dDevice.03 Type	dDevice.04 Manuf.	dDevice.05 Model #	dDevice.06 Pur. Date	From	To
1	10101	Defib1	AED	Acme	LS100	1/1/2016		9/27/2016
2	10102	Defib2	AED	Acme	LS100	6/1/2016		9/27/2016
3	10103	Defib3	AED	Acme	LS100	9/1/2016		
4	<u>10102</u>	<u>Defib2</u>	<u>AED</u>	<u>Acme Inc.</u>	<u>LS100</u>	<u>6/1/2016</u>	<u>9/27/2016</u>	
5	<u>10104</u>	<u>Defib4</u>	<u>AED</u>	<u>Acme</u>	<u>LS100</u>	<u>12/1/2016</u>	<u>9/27/2016</u>	

Pros

- Preserves history fully

Cons

- Requires more database storage
- Requires unique keys for records in incoming data
- Preserves historically incorrect data (unless “punch-through” updates are employed)
- If records are supplemented with state-provided data, the system must “pull forward” that information when inserting each new record.
- Receiving system must compare data within each element to determine whether an insert is needed for a new version of an object

Enforcing Unique Identifiers

The Delete + Insert + Update method and the Versioning method of handling updated demographic data both require the use of keys to uniquely identify objects in demographic data. NEMESIS 3.4 (and below) does not explicitly define keys for objects in demographic data. However, a state system may identify existing NEMESIS elements (or combinations of elements) to serve as keys in each group within demographic data. In order for elements to serve as keys, they must be unique within their context.

For example, a state may adopt dDevice.01 Device Serial Number as the key for objects in dDevice.DeviceGroup (medical devices). In order for the key to serve its purpose, the state should enforce rules that require it to be unique within an agency. With such rules in place, a state system would reject an agency demographic data submission containing more than one instance of dDevice.DeviceGroup with the same value for dDevice.01.

As needed, states should use a state Schematron schema to publish rules regarding the uniqueness of elements used as keys. Even if a state does not require, but accepts, data for a particular type of object in demographics, Schematron rules may be used effectively. For example, a state may not require medical device data, but it may accept medical device data when an agency submits it. Via Schematron, a state can implement rules that do not require medical device data to be present; but when it is present, dDevice.01 must be unique within the agency.

See Appendix B: Object Keys (Unique Identifiers), p. 12, for guidance regarding data elements and attributes that can be adopted as the unique identifiers, or keys, for group objects in demographic data.

See Appendix C: Sample Schematron Code for Enforcing Unique Identifiers, p. 15, for sample Schematron code that asserts the uniqueness of identifiers in demographic data objects.

Other Considerations

Initially, it may seem simple to recommend any of the above approaches for a “Receive and Process” system to handle updated demographic data. However, since there are various authoritative sources for

the information contained in an agency's demographic data, many systems are integrated with other data sources. When processing NEMSIS demographic data updates, a system may need to not only store the records it receives but also ensure that the records are connected meaningfully to other data.

For example, a state system may provide user accounts to personnel. A person directly accessing the state system may expect to be able to access records or run analytical reports for PCRs on which the person was a crew member. In order to support such needs, a state system must not only properly process incoming demographic data to record the fact that a person works for an agency, but also identify the relationship between the demographic data and the crew member data on a PCR, as well as the relationship between the demographic data and a user account in the state system. And, the system must maintain information about those relationships as data change over time with new updates received through either NEMSIS demographic data submissions or other means.

Conclusions

The handling of updated demographic by "Receive and Process" systems is complex, due to the lack of unique identifiers in demographic data objects and due to the number of sources from which demographic data is obtained. By adopting techniques described in this guide, developers of "Receive and Process" systems can ensure that their systems support state-level business requirements and maintain data integrity. They can also ensure that their systems send accurate demographic data to the national EMS database.

Appendix A: Authoritative Source

Wherever the agency is the authoritative source of data for an object, attribute, or relationship in demographic data, and the state requires the data, the state system should accept the data from the agency's system in NEMSIS V3 DEMDataSet format via NEMSIS V3 web services.

For each object, attribute, or relationship not noted in this appendix, the state should determine whether the state or the agency is the authoritative source of the data. The determination will vary by state, depending on regulatory structure. If the state determines that it is the authoritative source, the state system should ignore incoming data for that object, attribute, or relationship. If the state determines that the agency is the authoritative source, and the state requires the data, the state system should process incoming data for that object, attribute, or relationship.

DEMDataSet

dState

Receiving: Ignore.

Sending: Send state-provided data.

dCustomConfiguration

dCustomConfiguration.CustomGroup

Receiving: If @CustomElementID matches the @CustomElementID of a custom element defined by the state, ignore. Otherwise, process content as an agency-specific custom element configuration.

Sending: If @CustomElementID matches the @CustomElementID of a custom element defined by the state, send state-defined data for the custom element configuration. Otherwise, send data as received.

DemographicReport

dAgency

The combination of dAgency.01 EMS Agency Unique State ID, dAgency.02 EMS Agency Number, and dAgency.04 EMS Agency State serves as the unique key for an agency.

dContact

dConfiguration

dConfiguration.ConfigurationGroup

Receiving: If dConfiguration.01 State Associated with Certification/Licensure Levels is the receiving state, ignore. Otherwise, process content as an out-of-state configuration, or rely on StateDataSet data obtained from other states via NEMSIS.

Sending: Send one instance of dConfiguration.ConfigurationGroup with data correctly representing the state (see What Demographic Data to Send, p. 2). Optionally, send out-of-state configurations as received.

dLocation

dVehicle

dPersonnel

Receiving: If the state regulates EMS personnel certification/licensure, the state should determine which elements are state-regulated, and the state system should ignore submitted data for those elements. The agency is usually the authoritative source of information regarding the relationship between an agency and personnel (i.e., the fact that a person works for the agency) and data about the person's employment within the agency. The state system should process data about a person's agency employment as agency-specific information related to that person.

Sending: Send state-defined data for data elements that are state-regulated. Send other data as received.

dDevice

dFacility

dFacilityGroup

dFacility.FacilityGroup

Receiving: If dFacility.01 Type of Facility, dFacility.03 Facility Location Code, and dFacility.09 Facility State match a facility defined by the state, process the relationship (the fact that the agency serves the facility) and ignore all other data. Otherwise, process content as an agency-specific facility.

Sending: If the agency serves a facility defined by the state, send state-defined data for the facility. Otherwise, send data as received.

dCustomResults

Receiving: If dCustomResults.02 Custom Element ID Referenced matches the @CustomElementID of a custom element defined by the state, process content as state-level custom element results. Otherwise, process content as agency-specific custom element results.

Sending: Send data as received.

Appendix B: Object Keys (Unique Identifiers)

NEMSIS 3.4 (and below) does not explicitly define unique identifiers, or keys, for objects in demographic data. The following information provides guidance regarding data elements and attributes that can be adopted as the unique identifiers, or keys, for group objects in demographic data.

Where multiple elements or attributes are listed, they form a composite key, or combination key. It may be acceptable for certain elements within certain keys to be missing or blank. The absence of a value for an element should be treated as a zero-length string rather than a database null for comparison purposes (or use a null-safe operator).

DEMDataset

dState

dCustomConfiguration

dCustomConfiguration.CustomGroup

🔑 @CustomElementID

DemographicReport

- 🔑 dAgency/dAgency.01 EMS Agency Unique State ID
- 🔑 dAgency/dAgency.02 EMS Agency Number
- 🔑 dAgency/dAgency.04 EMS Agency State

dAgency

dAgency.ServiceGroup

🔑 dAgency.05 EMS Service Area States

dAgency.AgencyYearGroup

🔑 dAgency.15 Statistical Calendar Year

dContact

dContact.ContactInfoGroup

- 🔑 dContact.01 Agency Contact Type
- 🔑 dContact.02 Agency Contact Last Name
- 🔑 dContact.03 Agency Contact First Name
- 🔑 dContact.04 Agency Contact Middle Name/Initial

Even with this combination, it may not be possible to distinguish between different people with the same name holding the same position within an agency, such as Sr. and Jr.

dConfiguration

dConfiguration.ConfigurationGroup

🔑 dConfiguration.01 State Associated with the Certification/Licensure Levels

dConfiguration.ProcedureGroup

- 🔑 dConfiguration.06 EMS Certification Levels Permitted to Perform Each Procedure

dConfiguration.MedicationGroup

- 🔑 dConfiguration.08 EMS Certification Levels Permitted to Administer Each Medication

dLocation

dLocation.LocationGroup

- 🔑 dLocation.02 EMS Location Name
- 🔑 dLocation.03 EMS Location Number

It may be sufficient to use only one of these elements.

dVehicle

dVehicle.VehicleGroup

- 🔑 dVehicle.01 Unit/Vehicle Number
- 🔑 dVehicle.02 Vehicle Identification Number

Both are included because some vehicles may not have a VIN, and some vehicles may not have a state- or agency-assigned vehicle number. It may be sufficient to use only dVehicle.01 Unit/Vehicle Number.

dVehicle.VehicleCertificationLevelsGroup

- 🔑 dVehicle.05 Crew State Certification/Licensure Levels

dVehicle.YearGroup

- 🔑 dVehicle.11 Year Miles/Kilometers Hours Accrued

dPersonnel

dPersonnel.PersonnelGroup

- 🔑 dPersonnel.LicensureGroup/dPersonnel.22 EMS Personnel's State of Licensure
- 🔑 dPersonnel.LicensureGroup/dPersonnel.23 EMS Personnel's State's Licensure ID Number

The occurrence of dPersonnel.LicensureGroup is 0:M. The state should use Schematron rules to require an instance of dPersonnel.LicensureGroup with dPersonnel.23 set to the receiving state.

For personnel not licensed/certified by the state, the state should instruct agencies to create their own value for dPersonnel.22. The state should provide guidance on creating the value so that it will not be easily confused with real state-issued IDs. It is necessary for every personnel to have a value for dPersonnel.22 because that value is used in eCrew.01 in patient care reports and needs to correlate to a personnel record in demographics.

If the state may re-use a particular licensure ID for different people over time (not recommended), then the state system should also include dPersonnel.27 EMS Personnel's Current State's Licensure Expiration Date as part of the key.

dPersonnel.ImmunizationsGroup

- 🔑 dPersonnel.18 EMS Personnel's Immunization Status
- 🔑 dPersonnel.19 EMS Personnel's Immunization Year

dPersonnel.LicensureGroup

- 🔑 dPersonnel.22 EMS Personnel's State of Licensure
- 🔑 dPersonnel.23 EMS Personnel's State's Licensure ID Number
- 🔑 dPersonnel.24 EMS Personnel's State EMS Certification Licensure Level
- 🔑 dPersonnel.27 EMS Personnel's Current State's Licensure Expiration Date

dPersonnel.CertificationLevelGroup

- 🔑 dPersonnel.28 EMS Personnel's Practice Level

dDevice

dDevice.DeviceGroup

- 🔑 dDevice.01 Medical Device Serial Number

dFacility

dFacilityGroup

dFacility.FacilityGroup

- 🔑 ../dFacility.01 Type of Facility
- 🔑 dFacility.03 Facility Location Code
- 🔑 dFacility.07 Facility Street Address
- 🔑 dFacility.08 Facility City
- 🔑 dFacility.09 Facility State

This combination allows for situations where facilities in different states have the same state-issued facility code, where two facility locations/campuses share a state-issued facility code, or where facilities of different types share the same state-issued facility code. It may be possible for a state to use only dFacility.02 and dFacility.09 to uniquely identify a facility.

dCustomResults

dCustomResults.ResultsGroup

- 🔑 dCustomResults.02 Custom Element ID Referenced
- 🔑 dCustomResults.03 CorrelationID of DemographicReport Element or Group

Appendix C: Sample Schematron Code for Enforcing Unique Identifiers

The following sample Schematron code asserts two conditions related to medical device data in a NEMESIS DEMDataSet document:

1. It does not require medical device information to be present, but if it is present, it asserts that dDevice.01 must be non-empty.
2. It asserts that dDevice.01 for a given medical device must not be the same value as dDevice.01 for any previous medical device within the agency.

The rule context (`nem:dDevice.DeviceGroup[* != '']`) causes the rule to fire on each instance of `dDevice.DeviceGroup` where at least one child element is non-empty. If medical device data is not included, the rule does not fire.

The variable `nemisElements` identifies `dDevice.01` as the element that should be highlighted in the software user interface. If `dDevice.01` is missing, the variable `nemisElementsMissing` identifies `dDevice.01` as the missing element.

The first assert requires `dDevice.01` to be non-empty.

The second assert is configured so that it is not triggered if the first assert fails. The second assert requires the value of `dDevice.01` to not equal the value of `dDevice.01` in any previous instance `dDevice.DeviceGroup` within the agency.

Similar rules can be written for other object types in demographic data. A full sample Schematron schema is available from the NEMESIS TAC.

```
<rule id="uniqueness_dDevice.DeviceGroup" context="nem:dDevice.DeviceGroup[* != '']">

  <let name="nemisElements" value="nem:dDevice.01"/>
  <let name="nemisElementsMissing" value=".[not(nem:dDevice.01)]/'dDevice.01'"/>

  <assert id="uniqueness_dDevice.DeviceGroup_present" role="[ERROR]"
    diagnostics="nemisDiagnostic"
    test="normalize-space(nem:dDevice.01) != ''">
    If Medical Device information is included, Medical Device Name or ID should be
    recorded.
  </assert>

  <assert id="uniqueness_dDevice.DeviceGroup_unique" role="[ERROR]"
    diagnostics="nemisDiagnostic"
    test="normalize-space(nem:dDevice.01) = ''
    or not(normalize-space(nem:dDevice.01) =
    preceding-sibling::nem:dDevice.DeviceGroup/normalize-space(nem:dDevice.01))">
    Each Medical Device should have a unique Medical Device Name or ID.
  </assert>

</rule>
```