

NEMESIS V3 UUID Guide

Date

October 1, 2019

Authors

Joshua Legler – NEMESIS Consultant
N. Clay Mann – NEMESIS P.I.

Overview

Universally unique identifiers (UUIDs), also known as globally unique identifiers (GUIDs), were introduced in NEMESIS 3.5.0. UUIDs are used to uniquely identify agency demographic data objects (representing agency locations, vehicles, personnel, facilities, etc.) and patient care reports within an agency over time. Each demographic data object and patient care report has a software-generated unique ID that never changes and is never assigned to any other object.

Prior to NEMESIS 3.5, it was difficult for NEMESIS-compliant Receive & Process systems to track agency resource changes over time when processing demographic data submissions. For example, it was difficult for a Receive & Process system to determine whether a vehicle is a new object (a newly purchased vehicle) or an update to an existing object (an existing vehicle for which the VIN was corrected). With UUIDs, objects within an agency can be reliably tracked over time by Receive & Process systems even if all data elements within the objects are modified.

The following sequence illustrates the problem that UUIDs are intended to solve in agency demographic data:

1. A vehicle object is created within an agency in a “Collect Data” system:

Vehicle Object	
🔑 UUID	8c229a9b-00d5-4417-891f-1a2e9a4998c1
Vehicle Number	<i>bLank</i>
VIN	<i>bLank</i>
Call Sign	E1
Type	Rescue
Additional data	...

2. The “Collect Data” system sends demographic data, including the vehicle object, to a “Receive and Process” system. The “Receive and Process” system stores the vehicle object attached to the agency.

- The vehicle object is modified in the “Collect Data” system:

Vehicle Object		
🔑 UUID	8c229a9b-00d5-4417-891f-1a2e9a4998c1	
Vehicle Number	blank	101
VIN	blank	1GTDK14K4JZ512094
Call Sign	E1	Engine 1
Type	Rescue	Fire Apparatus
Additional data	...	

- The “Collect Data” system re-sends demographic data, including the updated vehicle object, to the “Receive and Process” system. Even though various data elements within the vehicle object have changed, the “Receive and Process” system uses the UUID to identify and update the correct vehicle object within the agency.

Prior to NEMSIS 3.5, patient care reports were identified by a combination of state, agency ID, agency number, patient care report number, unit call sign, and unit notified by dispatch date/time. If a change was made to any of these elements in a PCR and the PCR was resubmitted to the national EMS database, a duplicate PCR was created in the national database. Duplicates occurred due to corrections in unit call sign or unit notified by dispatch date/time on individual PCRs as well as by more large-scale updates such as a change in an agency’s ID or a time zone correction on unit notified by dispatch date/time on many PCRs. Patient Care Report Number was intended to be “a unique number for the EMS agency for all of time” (NEMSIS Data Dictionary), but since no algorithm was specified for generating patient care report numbers, there is no guarantee that the same number will not be generated on future PCRs in an agency. With UUIDs, PCRs can be reliably identified even if all data within the PCRs are modified.

UUIDs are automatically generated by software and have no business meaning. UUIDs are generated using a standard algorithm that virtually guarantees that the same UUID will never be generated more than once, thus allowing for a decentralized model that does not require the NEMSIS TAC or any other single entity to generate UUIDs.

Purpose

This guide describes the use of UUIDs in the NEMSIS standard and sets forth requirements for the implementation of UUIDs in NEMSIS-compliant software. The following terms in this document have special meaning when displayed in uppercase:

- MUST:** Specified feature is mandatory and will be verified through NEMSIS TAC compliance testing processes.
- SHOULD:** Specified feature is recommended but will not be verified through NEMSIS TAC compliance testing processes.
- MAY:** Specified feature is allowed but not mandatory.

Scope

UUIDs are used to track data structures, not real-world entities. While UUIDs are universally unique (in other words, the UUID algorithm will never generate the same UUID more than once), in the NEMESIS standard they identify data structures within the scope of an agency using a software product. For example, if any agency acquires a vehicle, the agency's software will assign a UUID to the new data object that represents the information about the vehicle. If the agency sells the vehicle, the agency's software will delete or inactivate the data object representing the information about the vehicle. If a second agency acquires that same vehicle, the second agency's software will likely assign its own UUID to the new data object that represents the information about the vehicle. The NEMESIS standard does not require the second agency's software to preserve the UUID that was generated by the first agency's software. Likewise, if an agency switches software, the new software may assign its own UUID to the new data object that represents the information about the vehicle.

Elements That Have UUIDs

In DEMDataSet, UUID attributes exist on repeating group elements (in other words, elements that have maxOccurs="unbounded" and that have child elements). The following demographic data elements have a UUID attribute:

- dAgency.AgencyServiceGroup
- dAgency.AgencyYearGroup
- dContact.ContactInfoGroup
- dConfiguration.ConfigurationGroup
- dConfiguration.ProcedureGroup
- dConfiguration.MedicationGroup
- dLocation.LocationGroup
- dVehicle.VehicleGroup
- dVehicle.VehicleCertificationLevelsGroup
- dVehicle.YearGroup
- dPersonnel.PersonnelGroup
- dPersonnel.ImmunizationsGroup
- dPersonnel.LicensureGroup
- dPersonnel.CertificationLevelGroup
- dDevice.DeviceGroup
- dFacilityGroup
- dFacility.FacilityGroup

In EMSDataSet, a UUID attribute exists on patient care reports:

- PatientCareReport

The UUID attribute is mandatory on all elements that have it.

UUID Standard

Systems MUST generate UUIDs using the [IETF RFC 4122](#) standard. Valid UUIDs match the following regular expression:

```
[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[1-5][a-fA-F0-9]{3}-[89abAB][a-fA-F0-9]{3}-[a-fA-F0-9]{12}
```

A sample valid UUID is:

```
e48cd734-01cc-4da4-ae6a-915b0b1290f6
```

A sample data element with a UUID attribute added is:

```
<dVehicle.VehicleGroup UUID="e48cd734-01cc-4da4-ae6a-915b0b1290f6">...
```

UUIDs are hexadecimals and are case-insensitive. The following two UUIDs are equal to each other:

- e48cd734-01cc-4da4-ae6a-915b0b1290f6
- E48CD734-01CC-4DA4-AE6A-915B0B1290F6

The hyphens in a UUID are for formatting only but MUST be present in NEMSIS XML documents.

Requirements for “Collect Data” Systems

This section describes requirements for software products to receive NEMSIS compliance certification as “Collect Data” systems in NEMSIS version 3.5 or above.

Generate and Store UUIDs

A “Collect Data” system MUST generate a UUID for all objects that have a UUID attribute. The system MUST generate the UUID for an object at the time the object is created or before the first time the object is exported from the system in NEMSIS V3 format. The system MUST store the UUID. Each time the object is exported, the system MUST use the same UUID. The system MUST not reuse the same UUID for a different object.

Scenario: A user of a “Collect Data” system adds a new vehicle to an agency. The system generates and stores a UUID for that vehicle. When the vehicle is included in a NEMSIS demographic data export, the UUID is sent as an attribute of the dVehicle.VehicleGroup data element. Over time, information about the vehicle changes. Each time the vehicle is included in a NEMSIS demographic data export, the same UUID is sent as an attribute of the dVehicle.VehicleGroup data element, thus uniquely identifying the vehicle even if all other information about the vehicle has changed. Eventually, the agency retires the vehicle and a user removes or deactivates the vehicle in the system. The system no longer includes the vehicle in NEMSIS demographic data exports, and the system never reassigns the vehicle’s UUID to another vehicle or any other object.

Algorithm

A “Collect Data” system MUST generate UUIDs using the [IETF RFC 4122](#) standard. RFC 4122 defines multiple algorithms for UUID generation. A system MUST implement one of the defined algorithms. RFC 4122-compliant implementations exist for Java, .Net, JavaScript, PHP, Python, Ruby, etc.

More Details About Uniqueness

The national Schematron schema for DEMDataSet asserts that each UUID must be unique within an agency (DemographicReport). In practice, this means:

- If multiple agencies share a “Collect Data” system and a person works for multiple agencies, the system MAY assign a single UUID to that person. The same UUID MAY be used for that person in each agency’s demographic data export. However, the same UUID MUST not be used more than once within a single agency.
- If a person is on both the contact list and the personnel list within an agency, the software MUST use a different UUID for dContact.ContactGroup and dPersonnel.PersonnelGroup. If the same UUID is used for both, the demographic data will fail national Schematron validation, because the same UUID is used more than once within a DemographicReport.

Requirements for “Receive & Process” Systems

This section describes requirements for software products for certification as “Receive & Process” systems in NEMESIS version 3.5 or above.

Use as Unique Identifier

When receiving data from another system, a “Receive & Process” system MUST use UUIDs as unique identifiers. The system MAY also use additional data to verify, validate, or enforce security on received data.

Scenario 1: A “Receive & Process” system receives agency demographic data from another system for the first time. The data submission includes data about vehicles. The system stores the data about each vehicle, including the UUID that was attached as an attribute of the dVehicle.VehicleGroup data element for each vehicle. Later, the system receives updated agency demographic data. The system compares its stored vehicle list to the vehicle list in the received data using UUIDs. It takes the following actions:

- A. Updates: For vehicles that have a matching UUID in the stored data and the received data, the system updates the stored data to match the received data.
- B. Insertions: For vehicles in the received data that have a UUID that does not exist on any vehicle in the stored data, the system inserts a new vehicle in the stored data, including its UUID.
- C. Deletions: For vehicles in the stored data that do not exist in the received data, the system deletes or deactivates the vehicle in the stored data.

Scenario 2: A “Receive & Process” system receives agency demographic data from another system. The data submission includes a personnel object with a UUID that is attached to an existing personnel record in another agency in the stored data. This scenario may happen when two or more agencies share a single “Collect Data” system installation and the same person works for more than one of the agencies. The “Receive & Process” system may choose how to respond to and store the data.

Scenario 3: A “Receive & Process” system receives a patient care report (PCR) with a UUID that already exists on a PCR in its stored data. The system **MUST** not insert the PCR as a new record in its stored data if the result would be more than one PCR with the same UUID. The system **SHOULD** process the data submission as an update to the existing PCR if the system’s security constraints and other constraints are met (for example, the client has permission to submit or update data for the agency and the client’s credentials are valid).

Validation

UUIDs are generated using the [IETF RFC 4122](#) standard and are validated using a regular expression pattern in the NEMESIS XSDs.

Store UUIDs

A “Receive & Process” system must store the UUIDs that it receives so that it can compare them to UUIDs that are received later. A system **MAY** also generate its own UUIDs for internal use, but it **MUST** use the UUIDs provided by sending systems for uniqueness comparisons.

Exports

Each time an object is exported, a system **MUST** use the same UUID for that object, and it **MUST** be the UUID that was received from the submitting system.

If a data element is included in the national-only XSD and has a UUID attribute, a “Receive & Process” system **MUST** include the UUID attribute in national-only exports.

Additional Topics

Transparent to End Users

Generally speaking, the implementation of UUIDs should not involve end users. End users should not be required to interact with UUIDs. UUIDs may be made visible to system administrative users for the purpose of managing their systems.

UUIDs vs. CorrelationIDs

The NEMESIS standard uses the CorrelationID attribute on repeating elements to support custom elements. The addition of UUIDs in the NEMESIS standard has no impact on CorrelationIDs.

CorrelationIDs may be generated on the fly when a NEMESIS XML document is generated, because their scope of use is limited to the document instance in which they are contained. UUIDs must be stored, because their scope spans across time and they must be the same every time an object is exported.

Translation Limitations

It is possible to translate data from a pre-3.5 version of NEMSIS to version 3.5 or later using an XSL transformation. However, the translated data will fail NEMSIS 3.5 XML Schema (XSD) validation, because it is not possible to create and persistently store UUIDs during the XSL transformation process. Each time an object is translated, it is possible that a different UUID would be generated, which would not meet the requirements for UUID implementation. For data originally created in a pre-3.5 version, it may be possible for a system to implement a one-time process to add UUIDs to all existing objects. Then the system could translate the data prior to export and ultimately export the data in 3.5 format.

Version 3.5 or later data can be translated to earlier versions using an XSL transformation. The translation process ignores the new UUIDs.

Switching Software

If an agency switches software vendor, UUIDs for objects in that agency may change.

Software vendors SHOULD develop processes for maintaining existing UUIDs when new software implementations are replacing a different NEMSIS 3.5 software system. State and local EMS officials should request that software maintain and utilize an existing DEMDataSet when replacing an existing software system.

If the new software can receive and process NEMSIS XML data, then existing data could be exported from the old system (or from a state or regional system that has previously received the demographic data from the old system) and imported into the new system, preserving existing UUIDs.

If the new software cannot receive and process NEMSIS XML data, the agency will set up its data manually in the new system, and the new system will likely generate new UUIDs for all objects. The first time the agency exports demographic data using the new system, it would appear to the receiving system that the agency had removed all previous objects and created all new objects.

Conclusions

UUIDs facilitate the identification and tracking of demographic data objects and patient care reports in the NEMSIS V3 standard. The requirements in this guide ensure that UUIDs are generated and utilized correctly by NEMSIS-compliant software systems.