Creating a NIEM IEPD, Part 2: Map and subset NIEM

Design an XML information exchange between U.S. government entities

Priscilla Walmsley

May 18, 2010
(First published February 09, 2010)

Part 1 of this series described the process of creating a UML model of an XML information exchange to be implemented in the National Information Exchange Model (NIEM). In this article, take the next step—map the model to NIEM to determine what parts of NIEM the exchange can reuse. Also learn how to create a subset of the NIEM model to include in an IEPD.

View more content in this series

18 May 2010: Added links to Part 4 of this series in Introduction, Conclusion, and Resources sections.

09 Mar 2010: Added links to Part 3 of this series in Introduction, Conclusion, and Resources sections.

Other articles in this series

- Part 1: Model your NIEM exchange
- Part 3: Extend NIEM
- Part 4: Assemble the IEPD

Now that you've created a UML model of your exchange (see Part 1), the next step is to map your model to NIEM to determine what parts of NIEM you will reuse in your messages. This mapping is most commonly done in a spreadsheet, known as a *Component Mapping Template* (CMT). The CMT is useful for several reasons:

- It provides a detailed, human-readable definition of your exchange model with places for comments and extra documentation.
- It makes explicit which parts of the model reuse NIEM components and which are custom to the Information Exchange Package Documentation (IEPD).
- It serves as a convenient checklist when you make a subset of the NIEM model.

Creating a component mapping template

Frequently used acronyms

- CMT: Component Mapping Template
- IEPD: Information Exchange Package Documentation
- NIEM: National Information Exchange Model
- SSGT: NIEM Schema Subset Generation Tool
- UML: Unified Modeling Language
- XML: Extensible Markup Language

You typically create CMTs in Microsoft® Office Excel® or other spreadsheet software, such as OpenOffice.org Calc. However, you can create them in any tabular format. There is no one required format for a CMT, but a typical CMT has, at a minimum, the following columns:

- **Source type.** The name of the class in the UML model
- **Source property.** The name of the property in the UML model
- **Data type.** The data type of the property
- **Description.** A short description of the type or property
- **Cardinality.** How many of the properties are allowed to appear
- **Extension indicator.** Whether the model matches a component in the NIEM model
- **XPath.** The path to the element in an XML message

Some NIEM implementers add more columns to the CMT to represent the details of extending NIEM. Part 3 of this article series will look further into extending NIEM.

Recording your model in the CMT

Your first step is to record your UML model in the first five columns of the CMT. Part 1 of this article series introduced a UML diagram of a simple example case that involved reporting on the theft of registered vehicles. Based on that UML class diagram—shown again in Figure 1—Table 1 shows the `TheftLocation` class in CMT format. Descriptions are omitted from the table to save space, but a completed example CMT is available from Downloads. (View a text-only version of Figure 1.)

Figure 1. UML model diagram from Part 1

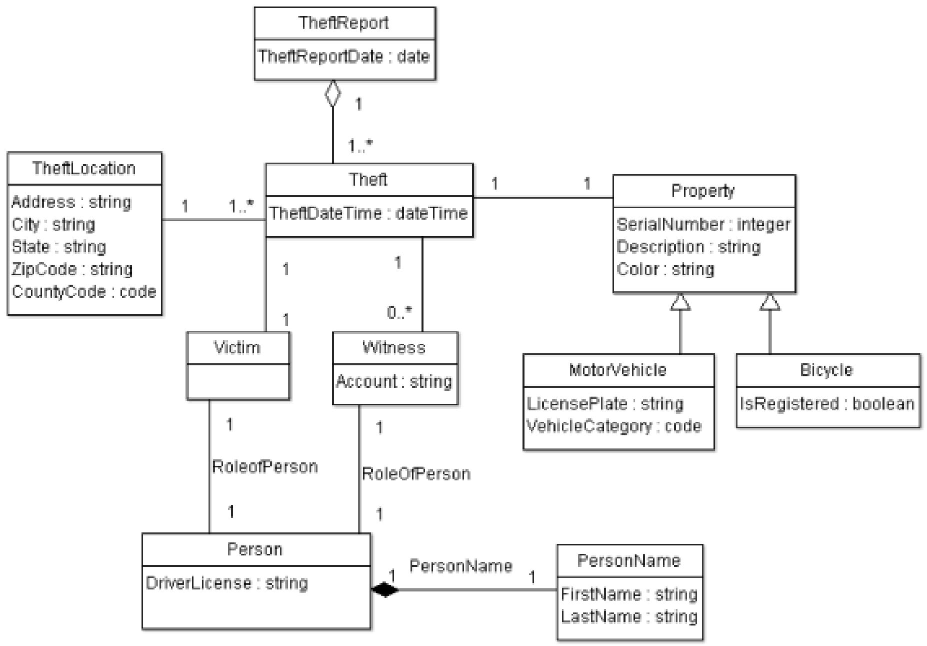


Table 1. Representing a type and properties in the CMT

Source type	Source property	Data type	Description	Cardinality
`TheftLocation`			...	
`TheftLocation`	`Address`	String	...	0..1
`TheftLocation`	`City`	String	...	0..1
`TheftLocation`	`State`	String	...	0..1
`TheftLocation`	`ZipCode`	String	...	0..1
`TheftLocation`	`CountyCode`	CountyCode	...	0..1

In the **Data type** column, XML Schema simple type names are used. In the case of code lists, a code list name is specified, and the valid values are documented in another tab of the spreadsheet. **Cardinality** shows the minimum and maximum number of occurrences, where `*` represents an unbounded number.

Each association should have a row in the CMT, along with rows for references to the types involved in the association. Table 2 shows a CMT representation of the `Theft/TheftLocation` association.

Table 2. Representing an association in the CMT

Source type	Source property	Data type	Description	Cardinality
`Theft / TheftLocation Assn`				
`Theft / TheftLocation Assn`	`Theft`	Reference	...	1..1

`Theft / TheftLocation Assn`	`TheftLocation`	Reference	...	1..1

Role types should be shown with references from the role to the type that is playing the role. The `Witness` role type in Table 3 contains a reference to `Person`, labelled `RoleOfPerson`.

Table 3. Representing a role in the CMT

Source type	Source property	Data type	Description	Cardinality
`Witness`				
`Witness`	`Account`	String	...	0..1
`Witness`	`RoleOfPerson`	Reference	...	1..1

Searching for NIEM equivalents

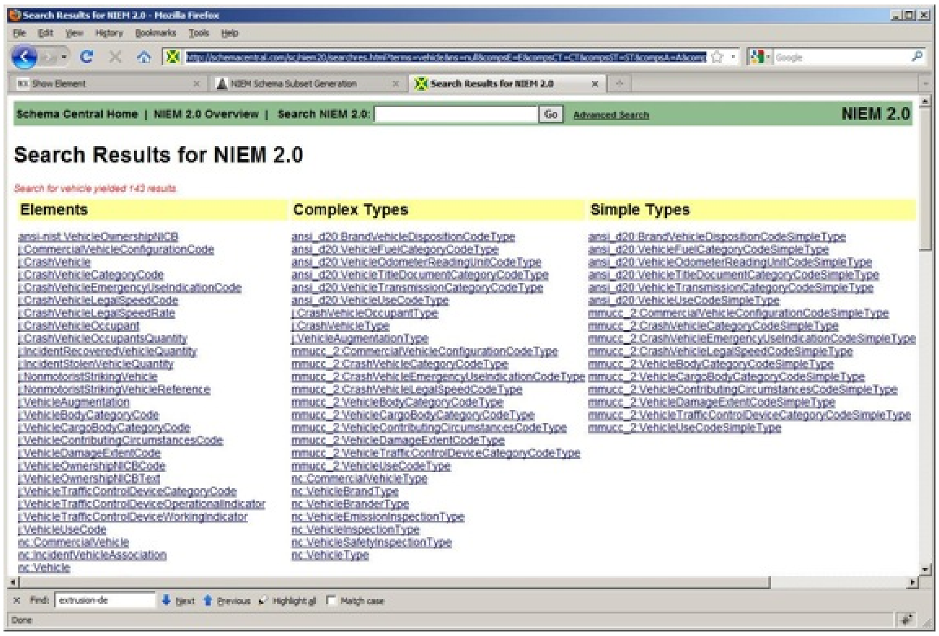
The next task in mapping your exchange is to determine where your model overlaps with NIEM and record those elements in the CMT. You want to reuse NIEM as much as possible to maximize interoperability with other NIEM applications. An IEPD is not NIEM-conformant if it adds new components when semantically equivalent components already exist in the NIEM model. That said, you should not force data into NIEM if it really doesn't fit. Part 3 of this article series will explain how to add new components to the model.

Because the NIEM model is very large, you do not want to scan the schemas by hand looking for matching components. Fortunately, several online tools are available to find components in the NIEM model (see Resources for links to these tools):

- **NIEM Wayfarer.** Using this tool, you can search for NIEM components and traverse through the model with one page per component.
- **Schema Central.** This tool has similar capabilities to NIEM Wayfarer but works with a variety of XML vocabularies, not just NIEM.
- **NIEM Schema Subset Generation Tool (SSGT).** Use this tool to search and navigate the NIEM model in a slightly more graphical fashion. It has the added capability of generating a NIEM subset once you find the components of interest.

Use one of these tools to look for all the components in your CMT that might already exist in NIEM. As an example, when you search for the term *Vehicle* in Schema Central, you see the search results page in Figure 2.

Figure 2. Schema Central search results page



Search Results for NIEM 2.0 - Mozilla Firefox

http://schemacentral.com/sc/niem20/searchres.html?terms=vehicle&ns=null&compsE=E&compsCT=CT&compsST=ST&compsA=A&comp

Schema Central Home | NIEM 2.0 Overview | Search NIEM 2.0: Go Advanced Search NIEM 2.0

Search Results for NIEM 2.0

Search for vehicle yielded 143 results

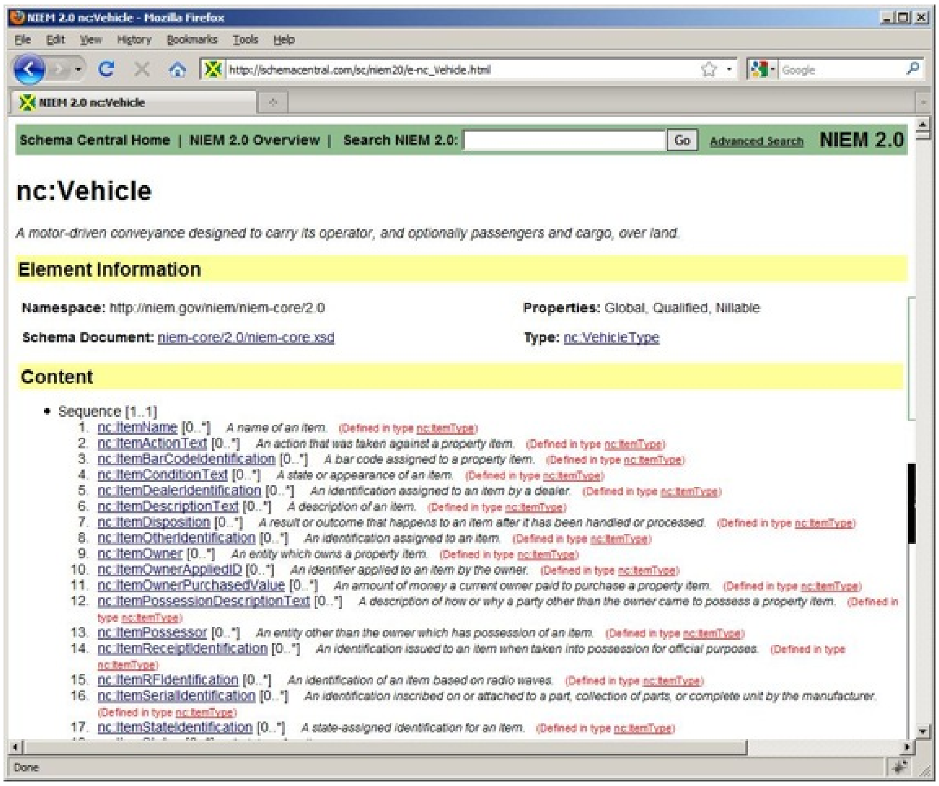
Elements	Complex Types	Simple Types
ansi-nist:VehicleOwnershipNICB	ansi_d20:BrandVehicleDispositionCodeType	ansi_d20:BrandVehicleDispositionCodeSimpleType
j:CommercialVehicleConfigurationCode	ansi_d20:VehicleFuelCategoryCodeType	ansi_d20:VehicleFuelCategoryCodeSimpleType
j:CrashVehicle	ansi_d20:VehicleOdometerReadingUnitCodeType	ansi_d20:VehicleOdometerReadingUnitCodeSimpleType
j:CrashVehicleCategoryCode	ansi_d20:VehicleTitleDocumentCategoryCodeType	ansi_d20:VehicleTitleDocumentCategoryCodeSimpleType
j:CrashVehicleEmergencyUseIndicationCode	ansi_d20:VehicleTransmissionCategoryCodeType	ansi_d20:VehicleTransmissionCategoryCodeSimpleType
j:CrashVehicleLegalSpeedCode	ansi_d20:VehicleUseCodeType	ansi_d20:VehicleUseCodeSimpleType
j:CrashVehicleLegalSpeedRate	j:CrashVehicleOccupantType	mmucc_2:CommercialVehicleConfigurationCodeSimpleType
j:CrashVehicleOccupant	j:CrashVehicleType	mmucc_2:CrashVehicleCategoryCodeSimpleType
j:CrashVehicleOccupantsQuantity	j:VehicleAugmentationType	mmucc_2:CrashVehicleEmergencyUseIndicationCodeSimpleType
j:IncidentRecoveredVehicleQuantity	mmucc_2:CommercialVehicleConfigurationCodeType	mmucc_2:CrashVehicleLegalSpeedCodeSimpleType
j:IncidentStolenVehicleQuantity	mmucc_2:CrashVehicleCategoryCodeType	mmucc_2:VehicleBodyCategoryCodeSimpleType
j:NonmotoristStrikingVehicle	mmucc_2:CrashVehicleEmergencyUseIndicationCodeType	mmucc_2:VehicleCargoBodyCategoryCodeSimpleType
j:NonmotoristStrikingVehicleReference	mmucc_2:CrashVehicleLegalSpeedCodeType	mmucc_2:VehicleContributingCircumstancesCodeSimpleType
j:VehicleAugmentation	mmucc_2:VehicleBodyCategoryCodeType	mmucc_2:VehicleDamageExtentCodeSimpleType
j:VehicleBodyCategoryCode	mmucc_2:VehicleCargoBodyCategoryCodeType	mmucc_2:VehicleTrafficControlDeviceCategoryCodeSimpleType
j:VehicleCargoBodyCategoryCode	mmucc_2:VehicleContributingCircumstancesCodeType	mmucc_2:VehicleUseCodeSimpleType
j:VehicleContributingCircumstancesCode	mmucc_2:VehicleDamageExtentCodeType	
j:VehicleDamageExtentCode	mmucc_2:VehicleTrafficControlDeviceCategoryCodeType	
j:VehicleOwnershipNICBCode	mmucc_2:VehicleUseCodeType	
j:VehicleOwnershipNICBText	nc:CommercialVehicleType	
j:VehicleTrafficControlDeviceCategoryCode	nc:VehicleBrandType	
j:VehicleTrafficControlDeviceOperationalIndicator	nc:VehicleBranderType	
j:VehicleTrafficControlDeviceWorkingIndicator	nc:VehicleEmissionInspectionType	
j:VehicleUseCode	nc:VehicleInspectionType	
nc:CommercialVehicle	nc:VehicleSafetyInspectionType	
nc:IncidentVehicleAssociation	nc:VehicleType	
nc:Vehicle		

Find: extrusion-de Next Previous Highlight all Match case

Done

When you click **nc:Vehicle**, the page in Figure 3 is displayed. It shows some general characteristics of the element, followed by a complete listing of its possible children.

Figure 3. Schema Central element display page



NIEM 2.0 nc:Vehicle - Mozilla Firefox

http://schemacentral.com/sc/niem20/e-nc_Vehicle.html

Schema Central Home | NIEM 2.0 Overview | Search NIEM 2.0: Go Advanced Search NIEM 2.0

nc:Vehicle

A motor-driven conveyance designed to carry its operator, and optionally passengers and cargo, over land.

Element Information

Namespace: http://niem.gov/niem/niem-core/2.0

Schema Document: niem-core/2.0/niem-core.xsd

Properties: Global, Qualified, Nillable

Type: nc:VehicleType

Content

- Sequence [1..1]
  1. nc:ItemName [0..*] *A name of an item.* (Defined in type nc:ItemType)
  2. nc:ItemActionText [0..*] *An action that was taken against a property item.* (Defined in type nc:ItemType)
  3. nc:ItemBarCodeIdentification [0..*] *A bar code assigned to a property item.* (Defined in type nc:ItemType)
  4. nc:ItemConditionText [0..*] *A state or appearance of an item.* (Defined in type nc:ItemType)
  5. nc:ItemDealerIdentification [0..*] *An identification assigned to an item by a dealer.* (Defined in type nc:ItemType)
  6. nc:ItemDescriptionText [0..*] *A description of an item.* (Defined in type nc:ItemType)
  7. nc:ItemDisposition [0..*] *A result or outcome that happens to an item after it has been handled or processed.* (Defined in type nc:ItemType)
  8. nc:ItemOtherIdentification [0..*] *An identification assigned to an item.* (Defined in type nc:ItemType)
  9. nc:ItemOwner [0..*] *An entity which owns a property item.* (Defined in type nc:ItemType)
  10. nc:ItemOwnerAppliedID [0..*] *An identifier applied to an item by the owner.* (Defined in type nc:ItemType)
  11. nc:ItemOwnerPurchasedValue [0..*] *An amount of money a current owner paid to purchase a property item.* (Defined in type nc:ItemType)
  12. nc:ItemPossessionDescriptionText [0..*] *A description of how or why a party other than the owner came to possess a property item.* (Defined in type nc:ItemType)
  13. nc:ItemPossessor [0..*] *An entity other than the owner which has possession of an item.* (Defined in type nc:ItemType)
  14. nc:ItemReceiptIdentification [0..*] *An identification issued to an item when taken into possession for official purposes.* (Defined in type nc:ItemType)
  15. nc:ItemRFIdentification [0..*] *An identification of an item based on radio waves.* (Defined in type nc:ItemType)
  16. nc:ItemSerialIdentification [0..*] *An identification inscribed on or attached to a part, collection of parts, or complete unit by the manufacturer.* (Defined in type nc:ItemType)
  17. nc:ItemStateIdentification [0..*] *A state-assigned identification for an item.* (Defined in type nc:ItemType)

Done

All of the NIEM components have a namespace prefix: *nc* refers to NIEM Core, the namespace in which the most fundamental types reside. There is also a namespace for each of the domains (for example, *j* for *Justice*). Feel free to use NIEM components from any domain as long as they are semantically equivalent to your model. You don't have to be implementing an immigration-related exchange to use an element from the immigration domain.

Guidelines for searching the NIEM model

Regardless of which tool you use, you can make searching the model easier by following these tips:

- It is often easier to start by looking for the highest-level types/classes (in the example case, `Theft`, `Property`, `Location`, and so on) first, and then finding the appropriate properties.
- Don't forget to search for synonyms. If you don't find *License Plate,* look for *Registration.*
- If you can't find your specific component, look for a more general one. Some of the most general types in NIEM are `Person`, `Organization`, `Location`, `Activity`, and `Item`. For example, if you don't find *Theft Location,* you can look for *Location* more generally and use `nc:Location`. If there is not a specific type for *Theft,* consider using the more generic `nc:Activity`.
- Don't just search names. If you expand your scope to search descriptions and enumerations, it might lead you to the appropriate type.

Finding components in the NIEM model might seem daunting at first, but it gets easier as you become more familiar with the general naming and structural patterns of the NIEM model.

Recording NIEM components in the CMT

When you find an equivalent NIEM component, record it in the CMT in the **XPath** column. Generally, simple XPath expressions are used—element and/or attribute names are separated by slashes (`/`). Type names do not need to be included in the XPath. Use namespace prefixes such as *nc:*, because element names are not necessarily unique across namespaces.

Table 4 shows the XPath mappings for `TheftLocation`. **Note:** For formatting purposes, the longer XPath mappings are split to multiple ines in the table. The mappings are normally a single string without blank spaces.

Table 4. TheftLocation XPath mappings

Source type	Source property	...	Ext?	XPath
`TheftLocation`		...		nc:Location
`TheftLocation`	`Address`	...	N	nc:Location/nc:LocationAddress /nc:StructuredAddress /nc:LocationStreet /nc:StreetFullText
`TheftLocation`	`City`	...	N	nc:Location/nc:LocationAddress /nc:StructuredAddress /nc:LocationCityName
`TheftLocation`	`State`	...	N	nc:Location/nc:LocationAddress /nc:StructuredAddress /nc:LocationState USPostalServiceCode
`TheftLocation`	`Zip`	...	N	nc:Location/nc:LocationAddress /nc:StructuredAddress /nc:LocationPostalCode
`TheftLocation`	`CountyCode`	...	Y	

You should include enough steps in the XPath to uniquely identify it. For example, don't just put `nc:StreetFullText` in the row for `Address`. Sometimes, multiple paths can lead to an element in NIEM, and the entire path is needed for precision.

In the example, the `CountyCode` property, which is a state-specific county code, is not found in NIEM, so it will require an extension. Therefore, the **Ext?** column is set to `Y`, and the XPath is left blank for now. Part 3 of this article series will walk through the process of filling in the XPaths for extensions.

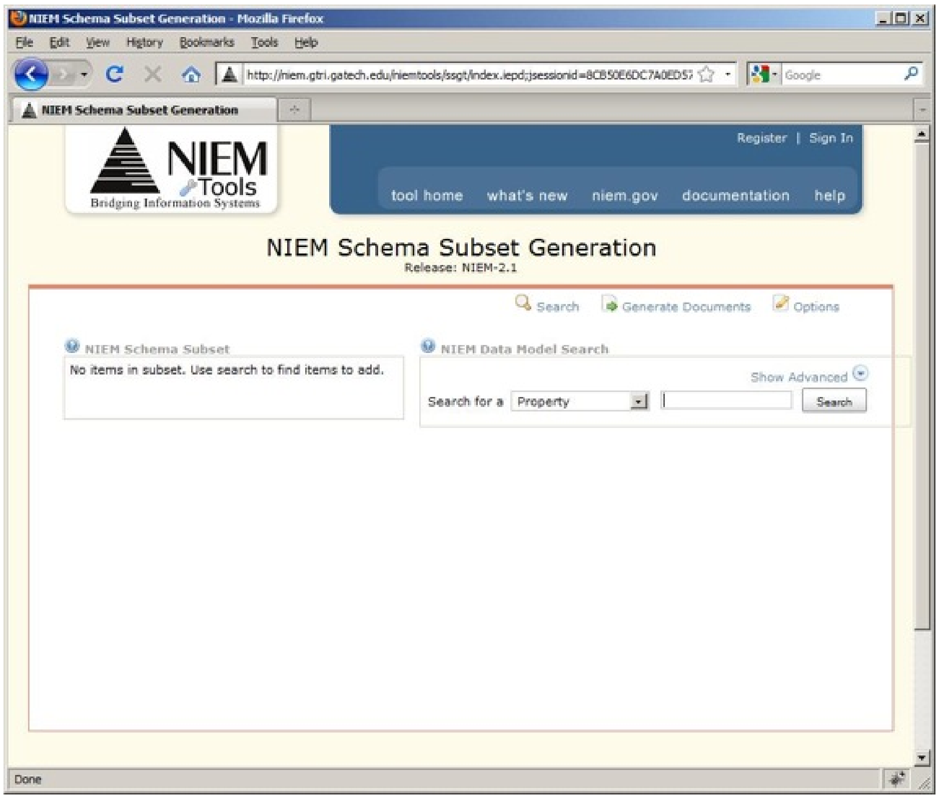
For a complete mapping of the Theft Report example model to NIEM, see the spreadsheet available from Downloads.

Creating a NIEM subset

When you have decided which components of NIEM you want to use in your exchange, you create a subset of the NIEM model that takes the form of a set of XML Schema documents. Because the full NIEM model is so large and loosely constrained, a NIEM subset is necessary to validate your exchange more precisely. The NIEM subset restricts the elements and attributes allowed, the number of times they can occur, and—in some cases—their allowed values. Creating a NIEM subset also speeds up validation of XML messages, because the schemas are much smaller.

You create NIEM subsets using the NIEM SSGT. The initial page of the SSGT in Figure 4 has two panes. The right pane is where you search and navigate the model, and the left pane shows your subset as you add components to it.

Figure 4. SSGT main page

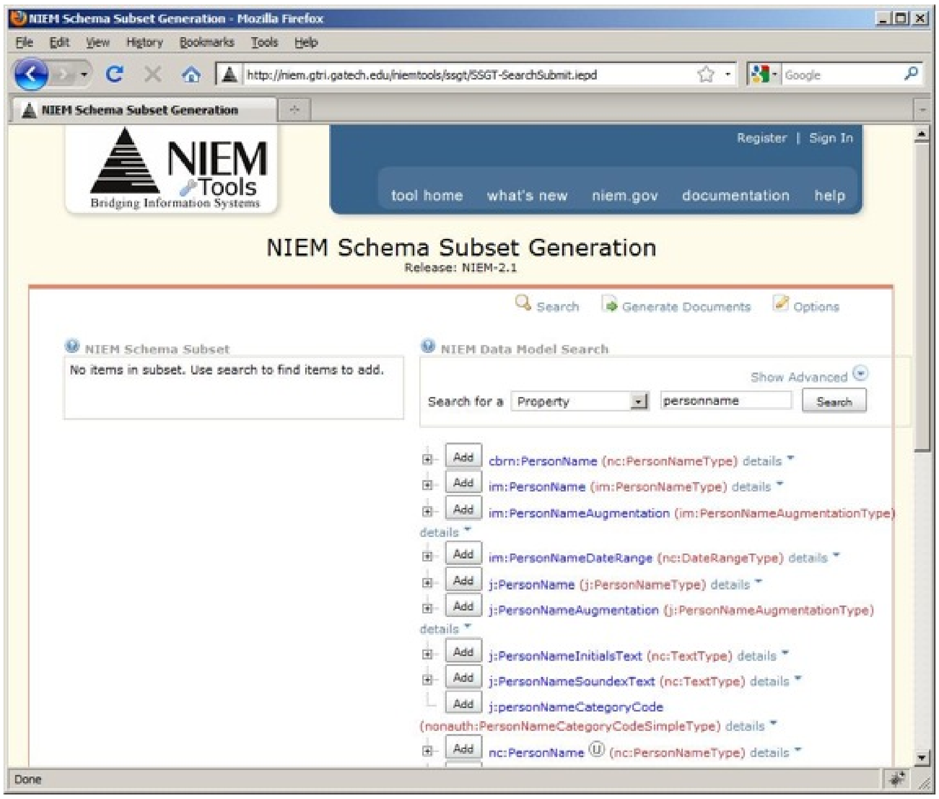

NIEM Schema Subset Generation - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://niem.gtri.gatech.edu/niemtools/ssgt/index.iepd;jsessionid=8CB50E6DC7A0ED57
NIEM Tools
Bridging Information Systems
Register | Sign In
tool home what's new niem.gov documentation help
NIEM Schema Subset Generation
Release: NIEM-2.1
Search Generate Documents Options
NIEM Schema Subset
NIEM Data Model Search
Show Advanced
Search for a Property
Search
Done

No items in subset. Use search to find items to add.

Based on your CMT, you perform searches to find components to add to your subset. Using the SSGT, you can choose to search either properties (element or attribute names), types, associations, or other components. Because you have the names of the element in your CMT, it makes sense to search properties. Sample search results are in Figure 5.

You might wonder why mapping and subsetting are two separate steps when you can perform the tasks in the same tool (the SSGT). It is certainly possible to perform the mapping and subsetting at the same time using the SSGT. However, many NIEM practitioners find it easier to do the mapping with NIEM Wayfarer or Schema Central, which shows the actual (flattened) structure of the types more clearly. The SSGT requires more knowledge of NIEM (and more clicking) to navigate, so going to the SSGT prepared with a CMT that lists exactly what you want from NIEM makes subsetting more efficient.

Figure 5. SSGT search results page


NIEM Schema Subset Generation - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://niem.gtri.gatech.edu/niemtools/ssgt/SSGT-SearchSubmit.iepd
Register | Sign In
tool home what's new niem.gov documentation help
Search Generate Documents Options
Show Advanced
Search
Done

NIEM Schema Subset Generation
Release: NIEM-2.1

NIEM Schema Subset

No items in subset. Use search to find items to add.

NIEM Data Model Search

Search for a Property personname

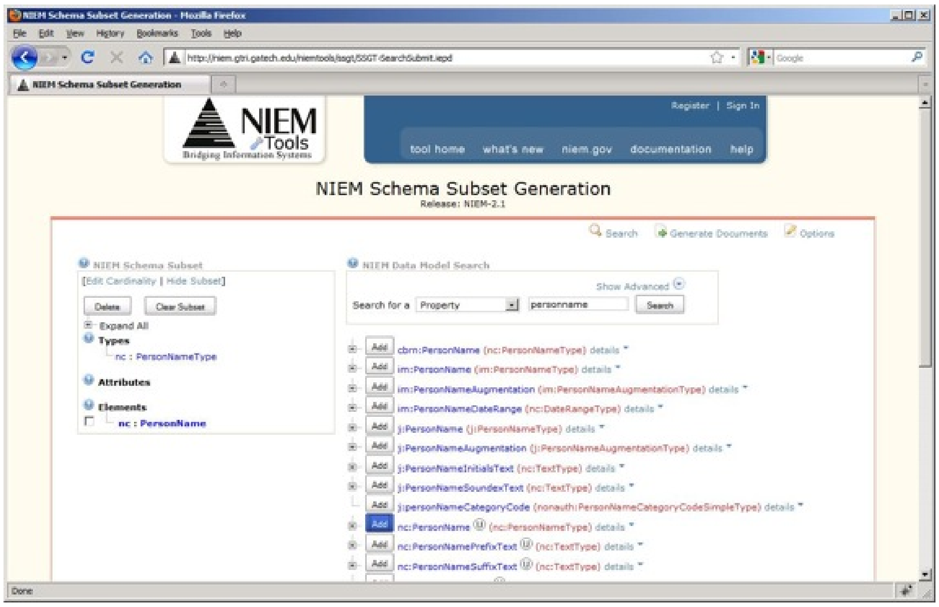
Add cbrn:PersonName (nc:PersonNameType) details
Add im:PersonName (im:PersonNameType) details
Add im:PersonNameAugmentation (im:PersonNameAugmentationType) details
Add im:PersonNameDateRange (nc:DateRangeType) details
Add j:PersonName (j:PersonNameType) details
Add j:PersonNameAugmentation (j:PersonNameAugmentationType) details
Add j:PersonNameInitialsText (nc:TextType) details
Add j:PersonNameSoundexText (nc:TextType) details
Add j:personNameCategoryCode (nonauth:PersonNameCategoryCodeSimpleType) details
Add nc:PersonName Ⓤ (nc:PersonNameType) details

Adding properties to the subset

When the NIEM component of interest is displayed, click **Add** to add it to your subset. It then appears in the left pane under **NIEM Schema Subset**, as in Figure 6.

When you add a property, its type automatically goes with it. For example, if you add `nc:PersonName`, `nc:PersonNameType` is automatically added to the subset, as well. The components you explicitly selected appear in the left pane in bold, with a check box next to them, while the dependent components are not in bold.

Figure 6. SSGT subset



NIEM Schema Subset Generation - Mozilla Firefox

http://niem.gtri.gatech.edu/niemtools/ssgt/SSGT-SearchSubmit.iepd

NIEM Schema Subset Generation
Release: NIEM-2.1

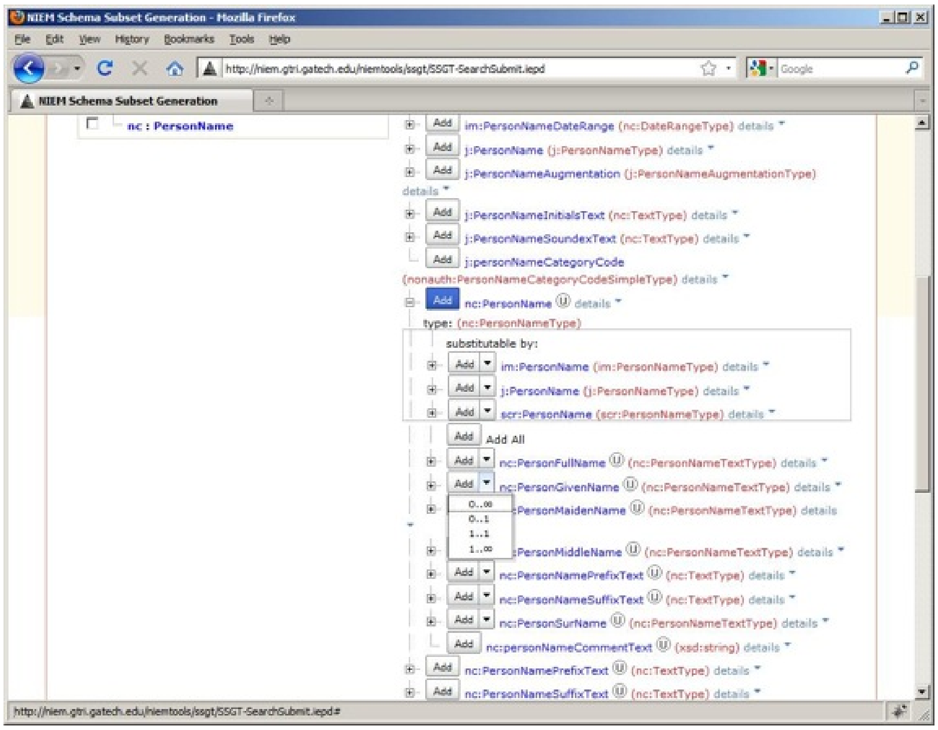
NIEM Schema Subset
[Edit Cardinality | Hide Subset]
Delete | Clear Subset
Expand All
Types
nc : PersonNameType
Attributes
Elements
nc : PersonName

NIEM Data Model Search
Search for a Property: personname

```
Add  cbrn:PersonName (nc:PersonNameType) details
Add  im:PersonName (im:PersonNameType) details
Add  im:PersonNameAugmentation (im:PersonNameAugmentationType) details
Add  im:PersonNameDateRange (nc:DateRangeType) details
Add  j:PersonName (j:PersonNameType) details
Add  j:PersonNameAugmentation (j:PersonNameAugmentationType) details
Add  j:PersonNameInitialsText (nc:TextType) details
Add  j:PersonNameSoundexText (nc:TextType) details
Add  j:personNameCategoryCode (nonauth:PersonNameCategoryCodeSimpleType) details
Add  nc:PersonName (nc:PersonNameType) details
Add  nc:PersonNamePrefixText (nc:TextType) details
Add  nc:PersonNameSuffixText (nc:TextType) details
```

The SSGT does not add the child properties of a type by default. For example, if you add `nc:PersonName`, it does not include the properties `nc:PersonGivenName` and `nc:PersonSurName`. These you must add to the subset separately. When you add them, you must do so in the context of `nc:PersonName` so that the parent-child relationship between, for example, `nc:PersonName` and `nc:PersonGivenName` is maintained. To do this, expand the `nc:PersonName` tree in the SSGT search results and click **Add** next to `nc:PersonGivenName`, as in Figure 7.

Figure 7. Adding a child using the SSGT



NIEM Schema Subset Generation - Mozilla Firefox

http://niem.gtri.gatech.edu/niemtools/ssgt/SSGT-SearchSubmit.iepd

nc : PersonName

```
Add  im:PersonNameDateRange (nc:DateRangeType) details
Add  j:PersonName (j:PersonNameType) details
Add  j:PersonNameAugmentation (j:PersonNameAugmentationType) details
Add  j:PersonNameInitialsText (nc:TextType) details
Add  j:PersonNameSoundexText (nc:TextType) details
Add  j:personNameCategoryCode (nonauth:PersonNameCategoryCodeSimpleType) details
Add  nc:PersonName details
  type: (nc:PersonNameType)
    substitutable by:
      Add  im:PersonName (im:PersonNameType) details
      Add  j:PersonName (j:PersonNameType) details
      Add  scr:PersonName (scr:PersonNameType) details
    Add  Add All
    Add  nc:PersonFullName (nc:PersonNameTextType) details
    Add  nc:PersonGivenName (nc:PersonNameTextType) details
         0..∞
         0..1
         1..1
         1..∞
         PersonMaidenName (nc:PersonNameTextType) details
         PersonMiddleName (nc:PersonNameTextType) details
    Add  nc:PersonNamePrefixText (nc:TextType) details
    Add  nc:PersonNameSuffixText (nc:TextType) details
    Add  nc:PersonSurName (nc:PersonNameTextType) details
    Add  nc:personNameCommentText (xsd:string) details
Add  nc:PersonNamePrefixText (nc:TextType) details
Add  nc:PersonNameSuffixText (nc:TextType) details
```

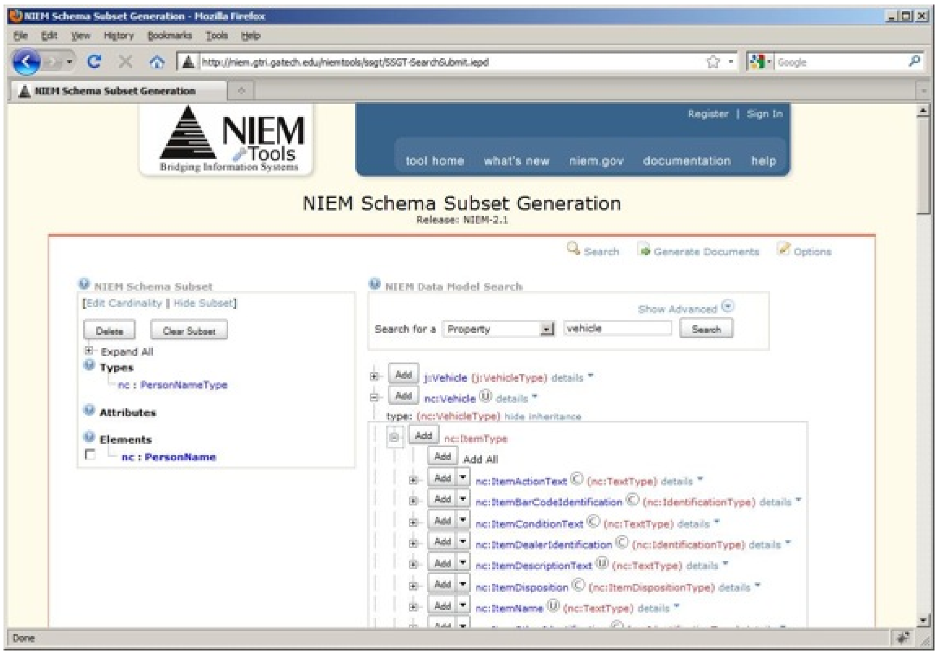
http://niem.gtri.gatech.edu/niemtools/ssgt/SSGT-SearchSubmit.iepd#

If instead you separately searched for `PersonGivenName` and added it from the search results, the element would be added but not as a child of `nc:PersonName`.

Figure 7 also shows that when you add a property of a type, you can specify the cardinality. Clicking the right down arrow on the **Add** button brings up a drop-down menu that shows possible cardinalities. The default is 0 to infinity.

If a property is included by inheritance, it is not displayed in the SSGT hierarchy by default. For example, expanding `nc:Vehicle` in the SSGT search results does not automatically show the `nc:ItemDescriptionText` that is mapped to the `Property Description` property. To see these inherited properties, click **show inheritance** (next to `nc:VehicleType`) and expand the type that contains the property of interest—in this case, `nc:ItemType`, as in Figure 8.

Figure 8. Adding an inherited property using the SSGT

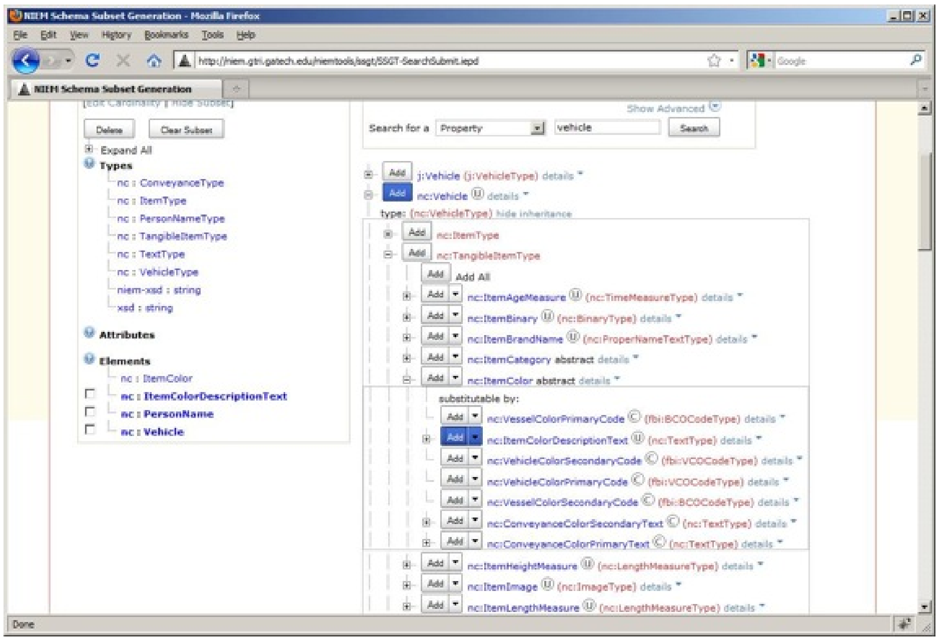

NIEM Schema Subset Generation - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://niem.gtri.gatech.edu/niemtools/ssgt/SSGT-SearchSubmit.iepd
NIEM Tools
Bridging Information Systems
Register | Sign In
tool home what's new niem.gov documentation help
NIEM Schema Subset Generation
Release: NIEM-2.1
Search Generate Documents Options
NIEM Schema Subset
[Edit Cardinality | Hide Subset]
Delete Clear Subset
Expand All
Types
nc : PersonNameType
Attributes
Elements
nc : PersonName
NIEM Data Model Search
Show Advanced
Search for a Property vehicle Search
Add j:Vehicle (j:VehicleType) details
Add nc:Vehicle details
type: (nc:VehicleType) hide inheritance
Add nc:ItemType
Add Add All
Add nc:ItemActionText (nc:TextType) details
Add nc:ItemBarCodeIdentification (nc:IdentificationType) details
Add nc:ItemConditionText (nc:TextType) details
Add nc:ItemDealerIdentification (nc:IdentificationType) details
Add nc:ItemDescriptionText (nc:TextType) details
Add nc:ItemDisposition (nc:ItemDispositionType) details
Add nc:ItemName (nc:TextType) details
Done

Abstract elements and the subset

The NIEM model commonly uses XML Schema abstract elements and substitution groups. For example, there are several ways to represent the color of an item. NIEM has an *abstract element*—`nc:ItemColor`—that cannot appear anywhere in an XML instance. Instead, it must be substituted by one of several elements, such as `nc:VehicleColorPrimaryCode` or `nc:ItemColorDescriptionText`. In XML Schema terminology, `nc:VehicleColorPrimaryCode` and `nc:ItemColorDescriptionText` are said to be members of a *substitution group* whose *head* is `nc:ItemColor`.

The abstract elements add some complexity to the creation of a subset, because you are required to add the substitutable elements in your subset, not just the abstract element. The SSGT marks all abstract elements with the word *abstract* and allows you to expand them to see the substitutable elements, as in Figure 9.

Figure 9. Adding a substitutable element using the SSGT


NIEM Schema Subset Generation - Mozilla Firefox
http://niem.gtri.gatech.edu/niemtools/ssgt/SSGT-SearchSubmit.iepd
Delete
Clear Subset
Expand All
Types
nc : ConveyanceType
nc : ItemType
nc : PersonNameType
nc : TangibleItemType
nc : TextType
nc : VehicleType
niem-xsd : string
xsd : string
Attributes
Elements
nc : ItemColor
nc : ItemColorDescriptionText
nc : PersonName
nc : Vehicle
Search for a Property vehicle Search
Show Advanced
j:Vehicle (j:VehicleType) details
nc:Vehicle details
type: (nc:VehicleType) hide inheritance
nc:ItemType
nc:TangibleItemType
Add All
nc:ItemAgeMeasure (nc:TimeMeasureType) details
nc:ItemBinary (nc:BinaryType) details
nc:ItemBrandName (nc:ProperNameTextType) details
nc:ItemCategory abstract details
nc:ItemColor abstract details
substitutable by:
nc:VesselColorPrimaryCode (fbi:BCOCodeType) details
nc:ItemColorDescriptionText (nc:TextType) details
nc:VehicleColorSecondaryCode (fbi:VCOCodeType) details
nc:VehicleColorPrimaryCode (fbi:VCOCodeType) details
nc:VesselColorSecondaryCode (fbi:BCOCodeType) details
nc:ConveyanceColorSecondaryText (nc:TextType) details
nc:ConveyanceColorPrimaryText (nc:TextType) details
nc:ItemHeightMeasure (nc:LengthMeasureType) details
nc:ItemImage (nc:ImageType) details
nc:ItemLengthMeasure (nc:LengthMeasureType) details
Done

Most date-related types also contain an abstract element `nc:DateRepresentation` that is substitutable by `nc:Date`, `nc:DateTime`, and so on. It is an easy mistake to simply add a date-related property, such as `nc:ActivityDate`, without expanding it to click on `nc:DateRepresentation`, and then `nc:Date` to allow for the appropriate child elements.

Fine-tuning your subset

When you have created your subset, you can modify it using the left pane of the SSGT. You can choose to delete any component by selecting the check box next to it, and then clicking **Delete**. You can also delete allowed code list values by expanding the appropriate simple types in the left pane. By default, all code list values from a simple type are included in the subset.

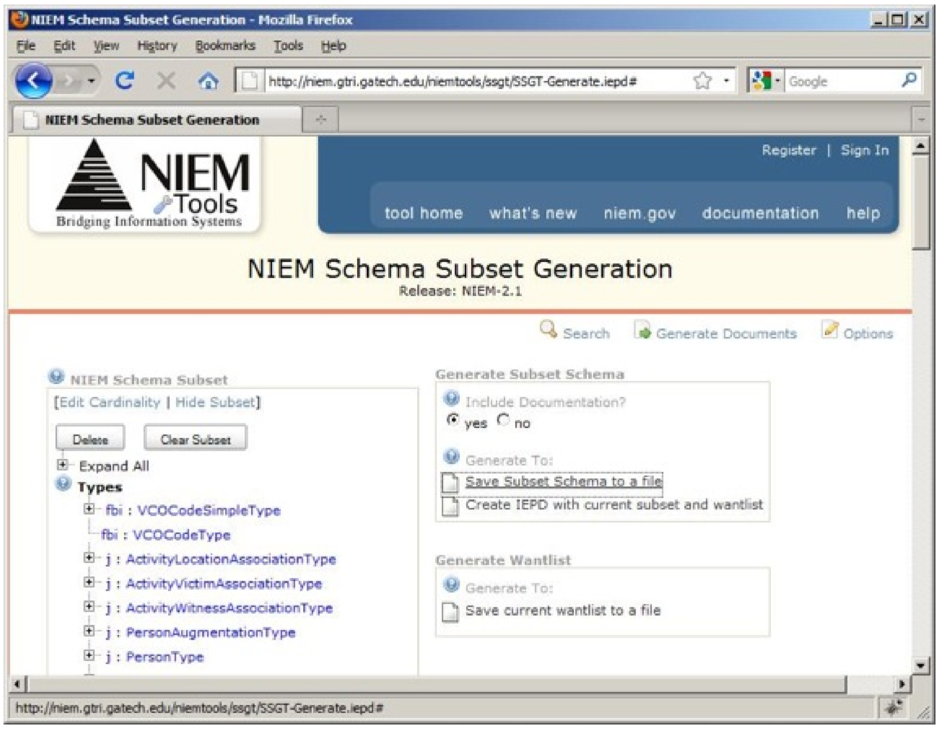
You can also choose to change the cardinalities by clicking **Edit Cardinality** at the top of the left pane. Doing so gives you another opportunity to decide how many of a particular property are allowed in a parent type.

Your NIEM subset does not have to be perfect at this point. NIEM subsetting is often an iterative process. You can save and modify your subset as needed during the final stages of IEPD development.

Generating your NIEM subset

To generate your subset, click **Generate Documents** in the upper right corner of the page. Doing so brings up a window similar to Figure 10 that shows some generation options. Select **Save Subset Schema to a file**, and choose the location in which to save it.

Figure 10. Generating a subset using the SSGT


NIEM Schema Subset Generation - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://niem.gtri.gatech.edu/niemtools/ssgt/SSGT-Generate.iepd#
NIEM Tools
Bridging Information Systems
Register | Sign In
tool home what's new niem.gov documentation help
NIEM Schema Subset Generation
Release: NIEM-2.1
Search Generate Documents Options
NIEM Schema Subset
[Edit Cardinality | Hide Subset]
Delete Clear Subset
Expand All
Types
fbi : VCOCodeSimpleType
fbi : VCOCodeType
j : ActivityLocationAssociationType
j : ActivityVictimAssociationType
j : ActivityWitnessAssociationType
j : PersonAugmentationType
j : PersonType
Generate Subset Schema
Include Documentation?
yes no
Generate To:
Save Subset Schema to a file
Create IEPD with current subset and wantlist
Generate Wantlist
Generate To:
Save current wantlist to a file

Include Documentation? [✓ yes] no

Doing so creates a .zip file called *Subset.zip* with a niem subfolder that contains the NIEM subset. It has a schema document for every namespace from which you chose elements in the SSGT plus a few standard schemas that come with all subsets.

Only the types you chose are included in the schema documents, and those types only contain the chosen properties. For example, although the `nc:PersonNameType` has seven possible children in the entire NIEM model and they all have cardinalities 0..*, your subset schema will contain only what is in Listing 1.

Listing 1. nc:PersonNameType in NIEM subset

```
<xsd:complexType name="PersonNameType">
  <xsd:complexContent>
    <xsd:extension base="s:ComplexObjectType">
      <xsd:sequence>
        <xsd:element ref="nc:PersonGivenName" minOccurs="0" maxOccurs="1"/>
        <xsd:element ref="nc:PersonSurName" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

The subset also contains an XML document called *wantlist.xml*, which lists all of the components you added to your subset along with their cardinalities. The wantlist is useful if you need to make changes later: You can re-upload the wantlist to the SSGT, modify the subset, and regenerate. Listing 2 shows part of the wantlist.

Listing 2. Partial NIEM subset wantlist

```
<w:WantList w:release="2.1" w:product="NIEM" ...>
  <w:Element w:name="j:Person" w:isReference="false"/>
  <w:Element w:name="j:Witness" w:isReference="false"/>
  ...
  <w:Type w:name="j:PersonType" w:isRequested="false">
    <w:ElementInType w:minOccurs="0" w:maxOccurs="1"
      w:name="j:PersonAugmentation" w:isReference="false"/>
  </w:Type>
  <w:Type w:name="j:WitnessType" w:isRequested="false">
    <w:ElementInType w:minOccurs="0" w:maxOccurs="1"
      w:name="j:WitnessAccountDescriptionText" w:isReference="false"/>
    <w:ElementInType w:minOccurs="1" w:maxOccurs="1"
      w:name="nc:RoleOfPerson" w:isReference="true"/>
  </w:Type>
  ...
</w:WantList>
```

Conclusion and next steps

Other articles in this series

- Part 1: Model your NIEM exchange
- Part 3: Extend NIEM
- Part 4: Assemble the IEPD

In this article, I showed you how to map a UML exchange model to NIEM using a CMT. I then described the process of creating a NIEM subset using the NIEM SSGT. In Part 3 of this series, I will address the rows of the CMT that were not filled in yet: the extensions. I will explain the different approaches to extending NIEM and take you through the process of creating Exchange and Extension schemas.

Downloadable resources

Description	Name	Size
Component Mapping Template (CMT)	niem2mapping.zip	62KB
NIEM Subset	niem2subset.zip	11KB