

NIEM 301 Exercises

****Note:** All stories in the exercises are fictitious, and no bias was used in the selection of actors.**

Exercise 1: Using Inheritance

Situation: U.S. Customs and Border Patrol (CBP) wants to share passport information about passengers arriving at United States ports of entry. CBP would like to use the NIEM core `nc:PassportNumberIdentification` element in this exchange and extend the element to contain the new `nc:IdentificationID` data element. The data element can have an alphanumeric value that identifies the passport. The goal of this exercise is to declare a specialized (inherited) element in an XML schema and instance. When you have completed the XML Schema, create an XML instance that will validate against this schema.

The following information will be used to describe passports for this exchange:

- Passport Number (can be mapped to `nc:PassportNumberIdentification`)
- `nc:IdentificationID` is of type `niem-xsd:string`

Exercise 2: Identify Substitution Groups

Situation: CBP needs to share with Transportation Security Administration (TSA) information about the dates passengers enter the United States. The date information that will be shared may be in different formats; for example, date and time or year and month. The goal of this exercise is to determine the different ways that “Date” can be represented in NIEM.

NOTE: Within this exercise, identify the different substitutions for date representation. A schema, types, and instance do not need to be defined.

Exercise 3a: Using Substitution Groups—Explicit Substitution

Situation: CBP wants to share information about the gender of the passengers arriving in the United States. Along with the identification number that identifies each passport, CBP also wants to share information specific to the person’s sex; however, a male can be identified as either “M” or “male” and female as either “F” or “female.” The goal of this exercise is to create a substitution group for this situation that allows the sex designation to be either a code value or a text value.

The type that will contain this substitution group is `nc:PersonType`. When you have completed the XML Schema, create an XML instance that will validate against this schema.

Exercise 3b: Using Substitution Groups—Implied Substitution

Situation: A situation exists in which an airline wants to share information about the contact numbers of passengers arriving in the United States.

An implied substitution group can be used to address this need so that a mobile, fax, or pager definition of a contact number can also be used within the exchange. The goal of this exercise is to create a substitution group that will enable `nc:ContactTelephoneNumber` to be used in the exchange or, alternatively, `nc:ContactMobileTelephoneNumber`, `nc:ContactPagerNumber`, or `nc:ContactFaxNumber` can take the place of, or “substitute” for, `nc:ContactTelephoneNumber`. When you have completed the XML Schema, create an XML instance that will validate against this schema.

NOTE: `nc:ContactTelephoneNumber` is of type `nc:TelephoneNumberType`.

Exercise 4: Identifying Code Lists

NIEM is designed to be compatible with external data standards. Multiple organizations and domains maintain code lists, including ISO, FIPS, NIST, among others. The NIEM type `fips_10-4:InternationalStateCodeSimpleType` contains data types for states implemented as a code list defined as “Countries, dependencies, areas of special sovereignty, and their principal administrative divisions from the Federal Information Processing Standards (FIPS) 10-4.” The United States is included in this code list. Find other code lists that include the United States.

NOTE: Within this exercise, identify the code lists. A schema, types, and instance do not need to be defined.

Exercise 5: Using Roles

Situation: TSA is validating aliens arriving in the United States and would like to share information with the U.S. Bureau of Citizenship and Immigration Services (USCIS). The goal of this exercise is to create an XML Schema that provides for this situation using type augmentation. When you have completed the XML Schema, create an XML instance that will validate against this schema.

The following information is collected to identify a person arriving in the United States:

- Full Name (maps to `nc:PersonName/nc:PersonFullName`)

NOTE: In this exercise, you do not need to write the XML Schema code that defines the person element; you will reuse elements from `nc:Person`. You can indicate that `nc:Person` is defined elsewhere.

The following information is collected to identify the information specific to the person's role of alien:

- Category of person—Alien (can be defined as of-type `im:AlienType`, maps to `im:Alien`)
- Description of person category (maps to `im:AlienIDDetails`)
- Alien number (maps to `im:AlienNumber`)
 - Identification ID
(`im:AlienIDDetails/im:AlienNumber/nc:IdentificationID`)
 - Effective Date
(`im:AlienIDDetails/im:AlienNumber/nc:IdentificationEffectiveDate`)

NOTE: `nc:RoleOfPersonReference` is also needed within the role type to indicate the person who is playing that role.

Exercise 6: Using Association Types

Situation: U.S. Citizenship and Immigration Services (USCIS) needs to share citizenship applicants' contact information with other federal agencies. Within this exchange, USCIS wants to share information that describes the contacts on record for a citizenship applicant. The goal of this exercise is to create an XML Schema which utilizes associations to address this situation. When you have completed the XML Schema, create an XML instance that will validate against this schema.

The following information is collected to identify a person:

- Person ID (will take the form of an attribute that identifies a person: `s:id`)
- Given Name (maps to `nc:PersonName/nc:PersonGivenName`)
- Last Name (maps to `nc:PersonName/nc:PersonSurName`)

The following data elements will be used for contact information:

- ContactInformationID (will take the form of an attribute that identifies contact information: `s:id`)

- E-mail Address (can be mapped to `nc:ContactEmailID`)
- Telephone Number (can be mapped to `nc:FullTelephoneNumber` within `nc:ContactTelephoneNumber`)

NOTE: Within this exercise, the association can be represented by `nc:PersonContactInformationAssociation` and the association type `nc:PersonContactInformationAssociationType`. All elements and types from NIEM core do not need to be defined in this schema, since they will be defined within the NIEM core subset schema.

Exercise 7: Using Metadata

Situation: TSA has decided that it will share passport information about aliens entering the United States with CBP. Within this exchange, TSA wants to share information that describes the passport expiry date and the date the information is reported. Since the reference information is supplementary and describes the data being sent for a passport, TSA wants to use metadata to represent this reference information. The goal of this exercise is to create an XML Schema which utilizes metadata to address this situation. When you have completed the XML Schema, create an XML instance that will validate against this schema.

NOTE: Within this exercise, create a new metadata element called `local-ns:PassportMetadata` of a new type called `local-ns:PassportMetadataType`. All elements and types mapped from NIEM core do not need to be defined in this schema, since they will be defined within the NIEM core subset schema.

The following information is collected to identify a passport:

- Passport (will take the form of an attribute that identifies a person: `s:id`)
- Passport ID (maps to `nc:PassportNumberIdentification/nc:IdentificationID`)

The following data elements will be used for passport reference information:

- Passport Reference (will be used to associate a person with his or her employment)
- Expiration Date (can be mapped to `nc:ExpirationDate/nc:Date`)
- Comments (can be mapped to `nc:CommentText`)
- Reported Date (can be mapped to `nc:ReportedDate`)

NOTE: Each passport metadata instance will be related to an instance of a `nc:Passport` element through the use of the `s:metadata` attribute.

Exercise 8: Using Type Augmentation

Situation: The Federal Aviation Administration (FAA) is tracking additional education details for select pilots and the aircrafts they operate. FAA would like to share targeted aircraft information collected from this effort with TSA. The goal of this exercise is to create an XML Schema that provides for this situation using type augmentation. When you have completed the XML Schema, create an XML instance that will validate against this schema.

Specifically, FAA would like to share the following information:

- Person Education Level (can be mapped to `intel:PersonAugmentation/intel:PersonEducationDetails/nc:PersonEducationLevelText`)
- Person Degree (can be mapped to `intel:PersonAugmentation/intel:PersonEducationDetails/nc:PersonEducationDegreeCode`)
- Person System ID (can be mapped to `intel:PersonAugmentation/intel:PersonSystemIdentifier/IdentificationID`)
- Aircraft Tail ID (can be mapped to `intel:PersonAugmentation/nc:Aircraft/nc:AircraftTailID`)
- Aircraft Make Code (can be mapped to `intel:PersonAugmentation/nc:Aircraft/nc:AircraftMakeCode`)
- Aircraft Model Code (can be mapped to `intel:PersonAugmentation/nc:Aircraft/nc:AircraftModelCode`)
- Aircraft Style Code (can be mapped to `nc:Aircraft/nc:AircraftStyleCode`)

Exercise 9: Creating Subsets and Wantlists

Situation: Another TSA information exchange scenario will be used within this exercise. TSA wants to track personal biographic information including a person's name. Though there are usually several subset schemas created for an information exchange, the focus of this exercise is to build the subset schema for the NIEM-Core elements involved in this exchange.

After you have completed the subset schema in the situation above, create a wantlist for the scenario. A wantlist is a list of the elements that will be included in the subset schemas for the exchange. The goal of this exercise is to create the wantlist for the TSA information exchange scenario described in the previous exercises.

The following data types and data elements need to be included within the subset schema:

- `nc:PersonName (type="nc:PersonNameType")`
 - `nc:PersonGivenName (type="nc:PersonNameTextType")`
 - `nc:PersonMiddleName (type="nc:PersonNameTextType")`
 - `nc:PersonSurName (type="nc:PersonNameTextType")`

NOTE: The subset schema for this example would include many more elements and types than the types listed above, but for simplicity the exercise will include only the types listed above. As an additional note, subset schemas are typically not hand-coded but instead are generated using one of the various NIEM tools. The primary goal of this exercise is for participants to become familiar with the code in a subset schema and to be able to troubleshoot problems when necessary.

Exercise 10: Coding a Constraint Schema

Situation: TSA would like to build a constraint schema for the information exchange described in the last exercise to restrict the values that can be passed in the exchange. The goal of this exercise is to build a constraint schema that provides the appropriate restrictions desired by TSA for the information involved in the exchange. The following constraints are desired in this exchange:

- Restrict the `nc:PersonGivenName`, field to a maximum of nine occurrences within the `nc:PersonName` element.

NOTE: In practice, constraints are applied through a copy of the subset schema(s), updated to reflect the desired constraints. Use the code snippet below from the NIEM Core subset schema as a reference when creating the constraint schema.

Subset Schema Code Snippet

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

<xsd:element name="PersonGivenName"
```

```

type="nc:PersonNameTextType" nillable="true"/>
<xsd:element name="PersonMiddleName"
type="nc:PersonNameTextType" nillable="true"/>
<xsd:element name="PersonSurName" type="nc:PersonNameTextType"
nillable="true"/>
<xsd:element name="IdentificationID" type="niem-xsd:string"
nillable="true"/>

```

Exercise 11: Creating Extension Schemas

Situation: The goal of this exercise is to create an extension schema for those data elements that could not be mapped to existing data elements within NIEM. Using our TSA person example from the previous exercise, the TSA has determined that among the passport holders are several people of Iberian descent. TSA would like to capture this information using an element to hold the additional surnames of Iberian passengers. When you have completed the XML Schema, create an XML instance that will validate against this schema.

The following data types and data elements need to be included within the extension schema:

- new element `local-ns:IberianPersonName` of type `"local-ns:IberianPersonNameType"` that extends `nc:PersonNameType` and contains:
 - new element `local-ns:PersonSecondarySurName` of type `"nc:PersonNameTextType"`
 - `nc:PersonGivenName`
 - `nc:PersonSurName`

Exercise 12: Creating Exchange Schemas

Situation: Building on the previous three exercises, TSA wants to build an exchange to share information about airline passengers with other federal agencies. The exchange will contain a message with information about the passengers entering the United States. The goal of this exercise is to create an XML Schema that will serve as the exchange schema in this scenario.

The following data elements will be contained under the root element called `"Message"`:

- `local-ns:Passenger`

NOTE: Though you would normally define namespaces as well as import the extension schema namespace within the exchange schema, the focus of this exercise is to define the root element for the exchange as well as the structure of the root element.

Exercise 13: Creating Instances

Situation: Based on the scenario described in the Exchange Schema exercise (Exercise 12), an XML instance for the scenario can be created. The goal of this exercise is to create a valid XML instance for this scenario.

The following data elements should be included:

- `local-ns:Passenger` (`type=nc:PersonType`)
- `local-ns:Message` (`type=local-ns:MessageType`)
- `nc:IberianPersonName` (`type=nc:PersonNameType`)
 - `nc:PersonGivenName` (`type=nc:PersonNameTextType`)
 - `nc:PersonSurName` (`type=nc:PersonNameTextType`)