

Ed-Fi XSD Guidelines Home	2
Ed-Fi XSD Design & Implementation Guidelines	3
XSD Scope	4
XSD Design Guidelines	5
XSD Implementation Guidelines	9
XML Schema - Standard Data Exchange Schema	10
XML Schema - Extended Reference Types	12
XML Schema - Ed-Fi Descriptors	15
XML Schema - Extensions Framework	17
XML Schema - Custom Data Exchange Schema	18

Ed-Fi XSD Guidelines Home

Ed-Fi XSD Design & Implementation Guidelines

Product: Ed-Fi XSD Design & Implementation Guidelines
Version: 3.0
Generation: Technical Suite 3
Obsoletes: --
Obsoleted By: --
Status: Current

Ed-Fi Alliance
Contact: Eric Jansson
May 31, 2018

This section provides design guidelines for the Ed-Fi XML data standards, which are technically expressed as XML Schema Definition (XSD) files. These guidelines describe the properties to which an XML specification and related implementation must adhere in order to be considered aligned with Ed-Fi technology standards. They are guidelines only. They do not describe a specific implementation or particular XSD specification. Where helpful, a rationale for specific guidelines is also provided.



Version History

Date	Version	Change
January 22, 2019	v3.0.1	Non-substantive updates to v3.0. Updated XML examples to align with Ed-Fi Data Standard v3.1.

Contents

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- › [XSD Implementation Guidelines](#)

XSD Scope

The essential features that characterize an Ed-Fi XSD specification are the data model that serves as its basis.

Data Model

The Ed-Fi Unifying Data Model (UDM)^[1] provides the basis for the data exchanged via Ed-Fi Bulk XML standards. The Ed-Fi UDM is a structured, conceptual model of common K–12 education data. The model includes entities that are easily recognized by educators and administrators: schools, students, teachers, attendance, grades, assessment results, and many others. These entities contain attributes (i.e., properties) that are also easily recognized. For example, assessment results contain data, such as a score and the date the assessment was administered. The UDM also includes associations (i.e., relationships) between entities, such as the association between students and schools.

Ed-Fi-aligned XSD specifications are generally built around entities that define nouns. In the education domain, these nouns include such things as schools, students, and teachers. In the Ed-Fi UDM these nouns have been rigorously defined as "entities," with specific attributes and associations. Compositions of entities, with their attributes and associations, are called "domain aggregates." These are identified from the Ed-Fi UDM according to the principles of Domain-Driven Design (DDD).^[2]

The full Ed-Fi UDM covers a broad set of information, and so it is impractical to send the entire representation of the UDM via a single XML document. Instead, the Ed-Fi Alliance publishes interchange standards as XSD that reference a common XSD Core, but that are composed of different logical sets of information that make sense to transfer in a single exchange. By referencing a standardized XSD Core, similar information will adhere to the same standards even when exchanged via a different specification.

Ed-Fi Domain Scope

The Ed-Fi Unifying Data Model is organized into 16 base domains:

- Alternative/Supplemental Services
- Assessment
- Bell Schedule
- Discipline
- Education Organization
- Enrollment
- Finance
- Graduation
- Intervention
- School Calendar
- Staff
- Student Academic Record
- Student Attendance
- Student Cohort
- Student Identification and Demographics
- Teaching and Learning

These domains are generally well-recognized in the K–12 education data space.

Additional subdomains pre-defined in the Ed-Fi UDM are for specific and relatively common instances of Alternative/Supplemental Services:

- Career and Technical Education
- Migrant Education
- Special Education
- Title I Part A Services

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- › [XSD Implementation Guidelines](#)

¹ For more information about the Ed-Fi Unifying Data Model, see [here](#).

² See, e.g., Evans, Eric, et al. (2006), [Domain-Driven Design Quickly](#), C4Media Inc., for a brief outline of Domain-Driven Design principles.

XSD Design Guidelines

The Ed-Fi Core XML Schema is the embodiment of the [Ed-Fi Unifying Data Model](#) (UDM) in an XML schema format that is designed to meet the requirements for XML data interchange, and has the following attributes:

- A core set of domain, association, and attribute types that directly map to the Ed-Fi UDM
- A method and examples of composing interchange schemas, reusing the types defined in the core schema
- A method and examples for extending the core schema and interchange schema to account for implementation-specific, or even interchange-specific, data (discussed in the [Ed-Fi Extension Framework](#) section of this documentation)

Ed-Fi Core XML Schema Expresses the Ed-Fi UDM

The Ed-Fi Core XML Schema expresses the Ed-Fi UDM's entities, associations, and attributes for education data. The Ed-Fi UDM is organized into 16 domains. These domains are organized into 66 different entities, 29 associations, and supported by 160 Descriptors — all of which have representative XML schema complex types.

Principles

XML schemas enforce rules around the content of XML-instance documents. However, there are many different ways and styles for XML schemas to accomplish the same goal.¹

The Ed-Fi Core XML Schema is designed with the following principles:

- **Consistency.** The XML schema should have a consistent organization and design pattern.
- **Extensibility.** The XML schema should be designed to easily allow for customizations to meet specific interchange requirements.
- **Flexibility.** The XML schema should be flexible to support the interchange of different subsets and collections of education data.
- **Reuse.** The XML schema should be designed to facilitate the reuse of types of elements in constructing the interchange schemas used for data transfer.
- **Composition Using XML Mechanisms.** The creation of interchange schemas from core and extended schemas should be accomplished using native XML mechanisms, not through a cut-and-paste operation.

Schema Design Pattern

The Ed-Fi Core XML Schema uses the Venetian Blind design pattern to facilitate reuse while also hiding namespace complexities. For each interchange schema, the style defines a single global element that nests local elements that use types (simple or complex) that are defined within the global namespace.

For example, consider the data exchange schema for parents and their student relationship depicted below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- (c)2019 Ed-Fi Alliance, LLC. All Rights Reserved. -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://ed-fi.org/
  <xs:include schemaLocation="Ed-Fi-Core.xsd" />
  <xs:annotation>
    <xs:documentation>==== Parent Interchange Model =====</xs:documentatio
  </xs:annotation>
  <xs:element name="InterchangeParent">
    <xs:annotation>
      <xs:documentation>This interchange defines parents and captures the
    </xs:annotation>
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Parent" type="Parent" />
        <xs:element name="StudentParentAssociation" type="StudentParent
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A single element `InterchangeParent` defines the interchange format. The two elements of the interchange, `Parent` and `StudentParentAssociation`, are nested within the single element, referencing types in `Ed-Fi-Core.xsd` (see the `include` statement above).

To provide maximum flexibility for the interchange, we chose to encapsulate the elements of the interchange in an unbounded choice statement rather than in an XML sequence. As a result, the various elements of the interchange are optional and can be provided in any order.

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- › [XSD Implementation Guidelines](#)

Version Identification

Versions of the Ed-Fi Core XML Schema are identified by major and minor version number, such as "3.1". A version may be appended with RFC ("Request for Comment"), RC ("Release Candidate"), or DRAFT. Versions are identified by the following:

- A unique default namespace
- A unique location for the file on the web

Namespace

The Ed-Fi Core XML Schema identifies the target namespace `http://ed-fi.org/nxxx` (with an appending version) as the default namespace. This is done so references to complex types do not have to qualify each reference, hence improving readability.

The implication is that any interchange (or customized core schema) that references the Ed-Fi-Core.xsd would need to use the same namespace, as shown here:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://ed-fi.org/
  <xs:include schemaLocation="Ed-Fi-Core.xsd"/>
  . . .
</xs:schema>
```

Ed-Fi Core XML Schema Organization

The primary Ed-Fi Core XML Schema types listed below are the building blocks of interchange schemas. The domain entities and associations match exactly those in the Ed-Fi UDM.

Table 1. Ed-Fi Core Schema Primary Types

Name	Definition
Domain entities	XML complex types representing the major entities in the education domain, as modeled in the Ed-Fi UDM.
Descriptors	XML complex types representing enumerations that cannot be standardized, and are determined by the content of their application and require loading in an interchange.
Associations	XML complex types representing those associations between the domain and Descriptor entities that require attributes.

Domain entities, Descriptors, and association types are composed from a second tier of XML simple and complex types and are organized as follows:

Table 2. Ed-Fi Supporting Types

Name	Definition
Base types	Template structures for entities, Descriptors, reference types, and Descriptor reference types.
Extended reference types	Supporting associations via XML IDREFs or via a lookup. Extended reference types provide a sort of "off page connector" for interchange schemas, pointing to entities that are already known by the receiving system.
Extended Descriptor reference types	Referencing the Descriptor values for an attribute.
Common types	Complex types composed of cohesive records of attributes, such as Name, Address, or Telephone.
Enumeration type	Standard code lists or controlled vocabulary that are used for attributes or are used to map Descriptor values.
String simple types and numeric simple types	Strings, integers, and doubles with specific constraints, such as length or value range, for the various simple typed attributes. While strings can be restricted in-line, this approach creates an anonymous type that can cause trouble when extending the parent complex type. Therefore, all restricted types are named.

Domain Entities

All domain entities are XML extensions of the type `ComplexObjectType`. This base type defines the common ID attribute that holds the assignment of an XML IDREF to instances of the domain type used in interchange schemas, as shown below.

```
<xs:complexType name="ComplexObjectType" abstract="true">
  <xs:annotation>
    . . .
  <xs:attribute name="id" type="xs:ID">
    . . .
```

```
</xs:complexType>
```

The complex type for a domain entity corresponds to the class defined in the Ed-Fi UDM. The elements of the complex type correspond to the attributes of the UDM class, as shown in the following example for the CourseOffering entity:

```
<xs:complexType name="CourseOffering">
  <xs:complexContent>
    <xs:extension base="ComplexObjectType">
      <xs:sequence>
        <xs:element name="LocalCourseCode" type="LocalCourseCode" />
        . . .
        <xs:element name="LocalCourseTitle" type="CourseTitle" />
        . . .
        <xs:element name="InstructionalTimePlanned" type="Duration" />
        . . .
        <xs:element name="CurriculumUsed" type="CurriculumUsedDefinition" />
        . . .
        <xs:element name="SchoolReference" type="SchoolReference" />
        . . .
        <xs:element name="SessionReference" type="SessionReference" />
        . . .
        <xs:element name="CourseReference" type="CourseReference" />
        . . .
        <xs:element name="CourseLevelCharacteristic" type="CourseLevelCharacteristic" />
        . . .
        <xs:element name="OfferedGradeLevel" type="GradeLevelDescription" />
        . . .
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

XML Attributes

Following the tenet that XML attributes contain data that is only used in the processing of the XML, the ComplexObjectType and ReferenceType base types provide the XML-internal Identifiers, as shown below.

```
<xs:complexType name="ComplexObjectType" abstract="true">
  . . .
  <xs:attribute name="id" type="xs:ID" />
  . . .
</xs:complexType>

<xs:complexType name="ReferenceType">
  . . .
  <xs:attribute name="id" type="xs:ID" />
  . . .
  <xs:attribute name="ref" type="xs:IDREF" />
  . . .
</xs:complexType>
```

Generalizations

Generalizations are specified using the XML schema extension. For example, EducationOrganization is a generalization of School, LocalEducationAgency, EducationServiceCenter, EducationOrganizationNetwork, and StateEducationAgency as shown below. Conversely, it can also be said that School is a specialization of EducationOrganization.

The complex type EducationOrganization is identified as an abstract type denoting that it is only defined as a generalization and may not be used on its own.

```
<xs:complexType name="EducationOrganization" abstract="true">
  . . .
  <xs:complexContent>
    <xs:extension base="ComplexObjectType">
      <xs:sequence>
        <xs:element name="EducationOrganizationIdentificationCode" type="EducationOrganizationIdentificationCode" />
        . . .
        <xs:element name="NameOfInstitution" type="NameOfInstitution" />
        . . .
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

        . . .
        <xs:element name="ShortNameOfInstitution" type="NameOfInsti
        . . .
        <xs:element name="EducationOrganizationCategory" type="Educ
        . . .
        <xs:element name="Address" type="Address" maxOccurs="unbound
        . . .
        <xs:element name="InternationalAddress" type="International
        . . .
        <xs:element name="InstitutionTelephone" type="InstitutionTe
        . . .
        <xs:element name="WebSite" type="URI" minOccurs="0">
        . . .
        <xs:element name="OperationalStatus" type="OperationalStatu
        . . .
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

Specializations are reflected as extensions of the base type, specifying the additional elements particular to School, as shown below. Note that School inherits all of the attributes of EducationOrganization.

```

<xs:complexType name="School">
    . . .
    <xs:complexContent>
        <xs:extension base="EducationOrganization">
            <xs:sequence>
                <xs:element name="SchoolId" type="xs:int">
                . . .
                <xs:element name="GradeLevel" type="GradeLevelDescriptorRefer
                . . .
                <xs:element name="SchoolCategory" type="SchoolCategoryType" m
                . . .
                <xs:element name="SchoolType" type="SchoolType" minOccurs="0"
                . . .
                <xs:element name="CharterStatus" type="CharterStatusType" min
                . . .
                <xs:element name="TitleIPartASchoolDesignation" type="TitleIP
                . . .
                <xs:element name="MagnetSpecialProgramEmphasisSchool" type="M
                . . .
                <xs:element name="AdministrativeFundingControl" type="Adminis
                . . .
                <xs:element name="InternetAccess" type="InternetAccessType" m
                . . .
                <xs:element name="LocalEducationAgencyReference" type="LocalE
                . . .
                <xs:element name="CharterApprovalAgencyType" type="CharterApp
                . . .
                <xs:element name="CharterApprovalSchoolYear" type="SchoolYear
                . . .
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

¹ Adapted from "PESC Guidelines for XML Architecture and Data Modeling", Version 3.0, Postsecondary Electronic Standards Council (PESC), April 29, 2005.

XSD Implementation Guidelines

Ed-Fi bulk data exchange designs typically follow specific patterns and practices.

This section covers the following topics in detail:

- [XML Schema - Standard Data Exchange Schema](#)
- [XML Schema - Extended Reference Types](#)
- [XML Schema - Ed-Fi Descriptors](#)
- [XML Schema - Extensions Framework](#)
- [XML Schema - Custom Data Exchange Schema](#)

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- › [XSD Implementation Guidelines](#)

XML Schema - Standard Data Exchange Schema

Different use cases and different periodicities of data exchange require different interchange schema.

For example, the [Ed-Fi Bulk Data Exchange for XML v3.1](#) contains a set of 21 Standard Interchange Schemas to handle common data exchange use cases, including basic student profile information, attendance records, assessment results, program participation, and many other information-transfer scenarios.

Standard Definitions and the Core XSD

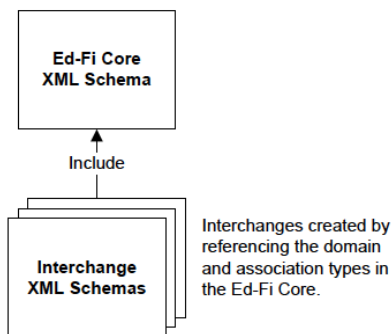
The Ed-Fi data exchange specifications are XSD files, all of which primarily align with the domain entities and other types defined in the [Ed-Fi Unifying Data Model](#). Some specialized exchanges may contain metadata or may convey data not present in the Ed-Fi UDM, but, generally speaking, most substantive information in any given data exchange will be defined by the UDM.

In keeping with that alignment to the Ed-Fi UDM, most Standard data exchange schema reference the Ed-Fi Core XML Schema — which is essentially an expression of the Ed-Fi UDM made concrete as an XSD.

Data Exchange Schema Construction

Through field-testing, the Ed-Fi data exchange schema standards provide a rigorous structure for data, but allow a reasonable degree of flexibility to account for different levels of source data availability. For example, source systems at the start of the year may have a less complete set of data than the same systems at the end of the year. Source assessment systems may have different ways of providing a means of identifying a student. The flexibility in the Ed-Fi Standard Interchange Schemas supports these realities in the field.

The Ed-Fi Standard Interchange Schemas are composed of elements from the domain and association types available in the Ed-Fi Core XML Schema, as depicted below.



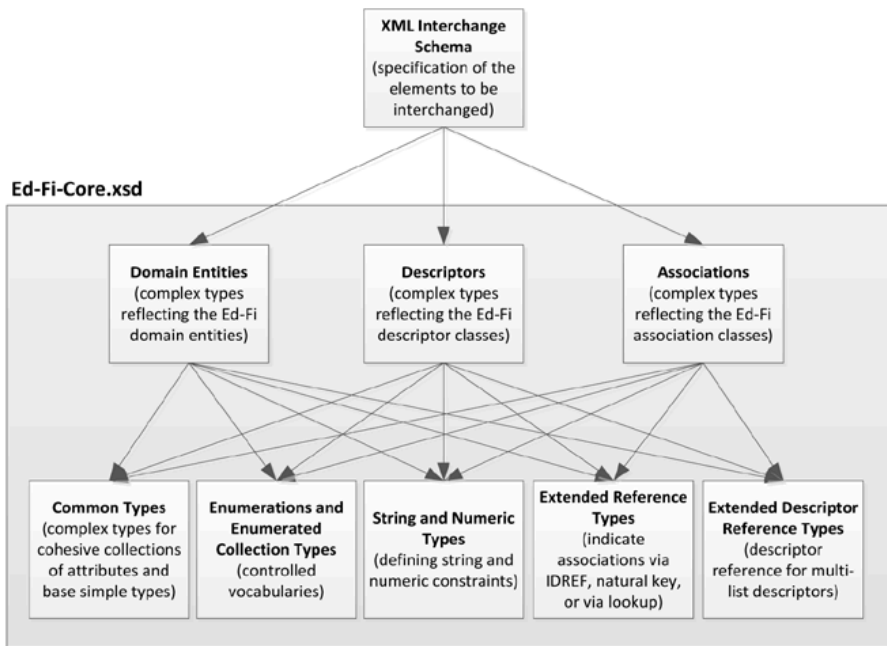
High-level construction of Ed-Fi Interchange Schemas

The Ed-Fi Core XML Schema provides a library of building blocks from which to compose interchange schemas, as depicted in the figure below.

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- ✓ [XSD Implementation Guidelines](#)
 - [XML Schema - Standard Data Exchange Schema](#)
 - [XML Schema - Extended Reference Types](#)
 - [XML Schema - Ed-Fi Descriptors](#)
 - [XML Schema - Extensions Framework](#)
 - [XML Schema - Custom Data Exchange Schema](#)



XML Schema - Extended Reference Types

Extended reference types provide the mechanism for denoting associations in Ed-Fi XML interchange data. The reference may be to an entity in the same interchange file, an entity in a separate interchange file, or an entity presumed to exist in the system receiving the bulk data. For example, assessment results generally need to reference student records (which are often already in the data store) and assessment metadata (which may be in a separate interchange file traveling with the results, or in the case of statewide tests, may already be in the data store).

Extended reference types define associations in one of three ways:

- Using an XML IDREF to refer to an ID defined for an entity within the same interchange file.
- Specifying attributes that represent the natural key for the entity (which may be either in the same interchange file or previously loaded).
- Optionally, specifying attributes that provide enough information to look up the entity from those previously loaded, when appropriate.

Extended reference types are used when an entity or association must reference another, as follows:

- When an entity has an association with another and that association has no attributes, then the entity uses an extended reference type to reference the other entity.
- When an association has attributes, the association uses an extended reference type to reference the associated entity.

Each extended reference type extends the base ReferenceType complex type, as defined below:

```
<xs:complexType name="ReferenceType">
  . . .
  <xs:attribute name="id" type="xs:ID">
  . . .
  <xs:attribute name="ref" type="xs:IDREF">
  . . .
</xs:complexType>
```

For example, the extended reference type for a SessionReferenceType is:

```
<xs:complexType name="SessionReferenceType">
  . . .
  <xs:complexContent>
    <xs:extension base="ReferenceType">
      <xs:sequence>
        <xs:element name="SessionIdentity" type="SessionIdentityType" m
          . . .
        <xs:element name="SessionLookup" type="SessionLookupType" minOc
          . . .
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

The SessionReferenceType allows the reference to be made in one of three ways:

- Using an XML IDREF (if interchanged in the file)
- Specifying the natural key information contained in the type's SessionIdentityType, as follows:

```
<xs:complexType name="SessionIdentityType">
  . . .
  <xs:sequence>
    <xs:element name="SessionName" type="IdentificationCode">
    . . .
    <xs:element name="SchoolYear" type="SchoolYearType">
    . . .
    <xs:element name="SchoolReference" type="SchoolReferenceType">
    . . .
  </xs:sequence>
</xs:complexType>
```

- Specifying attributes in the types SessionLookupType to provide enough information to look up the entity from those previously loaded, as follows:

```
<xs:complexType name="SessionLookupType">
```

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- ✓ [XSD Implementation Guidelines](#)
 - [XML Schema - Standard Data Exchange Schema](#)
 - [XML Schema - Extended Reference Types](#)
 - [XML Schema - Ed-Fi Descriptors](#)
 - [XML Schema - Extensions Framework](#)
 - [XML Schema - Custom Data Exchange Schema](#)

```

. . .
<xs:sequence>
  <xs:element name="SessionName" type="IdentificationCode" minOccurs="0">
    . . .
  <xs:element name="SchoolYear" type="SchoolYearType" minOccurs="0">
    . . .
  <xs:element name="Term" type="TermDescriptorReferenceType" minOccurs="0">
    . . .
  <xs:element name="SchoolReference" type="SchoolReferenceType" minOccurs="0">
    . . .
</xs:sequence>
</xs:complexType>

```

Thus, when CourseOffering refers to its association with a Session, it defines a SessionReference of the type SessionReferenceType, as follows:

```

<xs:complexType name="CourseOffering">
  . . .
  <xs:complexContent>
    <xs:extension base="ComplexObjectType">
      <xs:sequence>
        . . .
        <xs:element name="SessionReference" type="SessionReferenceType" minOccurs="0">
          . . .
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

In associations with attributes, extended reference types are similarly used to relate the associated entities. For example, the StudentParentAssociation has a StudentReference and a ParentReference, as shown below. The StudentReference is of type StudentReferenceType and the ParentReference is of type ParentReferenceType.

```

<xs:complexType name="StudentParentAssociation">
  . . .
  <xs:complexContent>
    <xs:extension base="ComplexObjectType">
      <xs:sequence>
        <xs:element name="StudentReference" type="StudentReferenceType" minOccurs="0">
          . . .
        <xs:element name="ParentReference" type="ParentReferenceType" minOccurs="0"/>
          . . .
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

```

The StudentIdentityType and the StudentLookupType (used within the StudentReferenceType) provide several options for specifying the identity of a student, as is shown below.

```

<xs:complexType name="StudentIdentityType">
  . . .
  <xs:sequence>
    <xs:element name="StudentUniqueId" type="UniqueId" minOccurs="1" maxOccurs="1">
      . . .
    </xs:sequence>
</xs:complexType>

<xs:complexType name="StudentLookupType">
  . . .
  <xs:sequence>
    <xs:element name="StudentUniqueId" type="UniqueId" minOccurs="0" maxOccurs="1">
      . . .
    <xs:element name="StudentIdentificationCode" type="StudentIdentificationCode" minOccurs="0" maxOccurs="1">
      . . .
    <xs:element name="Name" type="Name" minOccurs="0" maxOccurs="1">
      . . .
    <xs:element name="OtherName" type="OtherName" minOccurs="0" maxOccurs="1">
      . . .
    <xs:element name="BirthData" type="BirthData" minOccurs="0" maxOccurs="1">
      . . .
  </xs:sequence>
</xs:complexType>

```

```

    . . .
    <xs:element name="EducationOrganizationReference" type="EducationOrgani
    . . .
  </xs:sequence>
</xs:complexType>

```

First, a student may be specified by a StudentUniqueId provided by the StudentIdentityType. Alternatively, additional attributes provided by the StudentLookup may be specified for lookup providing a combination of StudentUniqueId, StudentIdentificationCode, Name, OtherName, BirthData, and/or EducationOrganizationReference.

Reference Types in v3.1

The following is a complete list of reference types in Ed-Fi Data Standard v3.1:

- AcademicWeekReferenceType
- AccountCodeReferenceType
- AccountReferenceType
- AccountabilityRatingReferenceType
- ActualReferenceType
- AssessmentItemReferenceType
- AssessmentReferenceType
- BellScheduleReferenceType
- BudgetReferenceType
- CalendarDateReferenceType
- CalendarReferenceType
- ClassPeriodReferenceType
- CohortReferenceType
- CommunityOrganizationReferenceType
- CommunityProviderLicenseReferenceType
- CommunityProviderReferenceType
- CompetencyObjectiveReferenceType
- ContractedStaffReferenceType
- CourseOfferingReferenceType
- CourseReferenceType
- CourseTranscriptReferenceType
- CredentialReferenceType
- DisciplineActionReferenceType
- DisciplineIncidentReferenceType
- EducationContentReferenceType
- EducationOrganizationInterventionPrescriptionAssociationReferenceType
- EducationOrganizationNetworkAssociationReferenceType
- EducationOrganizationNetworkReferenceType
- EducationOrganizationPeerAssociationReferenceType
- EducationOrganizationReferenceType
- EducationServiceCenterReferenceType
- FeederSchoolAssociationReferenceType
- GeneralStudentProgramAssociationReferenceType
- GradeReferenceType
- GradebookEntryReferenceType
- GradingPeriodReferenceType
- GraduationPlanReferenceType
- InterventionPrescriptionReferenceType
- InterventionReferenceType
- InterventionStudyReferenceType
- LearningObjectiveReferenceType
- LearningStandardReferenceType
- LocalEducationAgencyReferenceType
- LocationReferenceType
- ObjectiveAssessmentReferenceType
- OpenStaffPositionReferenceType
- ParentReferenceType
- PayrollReferenceType
- PostSecondaryEventRef
- PostSecondaryInstitution
- ProgramReferenceType
- ReferenceType
- ReportCardReferenceType
- RestraintEventReferenceType
- SchoolReferenceType
- SectionAttendanceTaken
- SectionReferenceType
- SessionReferenceType
- StaffAbsenceEventRefer
- StaffCohortAssociationR
- StaffEducationOrganizati
- StaffEducationOrganizati
- StaffEducationOrganizati
- StaffLeaveReferenceTyp
- StaffProgramAssociation
- StaffReferenceType
- StaffSchoolAssociationR
- StaffSectionAssociationF
- StateEducationAgencyR
- StudentAcademicRecord
- StudentAssessmentRefe
- StudentCTEProgramAss
- StudentCohortAssociatio
- StudentCompetencyObje
- StudentDisciplineInciden
- StudentEducationOrgani
- StudentEducationOrgani
- StudentGradebookEntryf
- StudentHomelessProgra
- StudentInterventionAsso
- StudentInterventionAtten
- StudentLanguageInstruc
- StudentLearningObjectiv
- StudentMigrantEducatior
- StudentNeglectedOrDelii
- StudentParentAssociatio
- StudentProgramAssociat
- StudentProgramAttendar
- StudentReferenceType
- StudentSchoolAssociatio
- StudentSchoolAttendanc
- StudentSchoolFoodServi
- StudentSectionAssociati
- StudentSectionAttendan
- StudentSpecialEducatior
- StudentTitlePartAProgra

XML Schema - Ed-Fi Descriptors

Ed-Fi Descriptors provide a malleable alternative to structured value lists that are typically expressed via enumerations in XSD. Descriptors are an expanded feature — first introduced in Ed-Fi Data Standard v1.1 — that are vital to accommodate the ways in which users of the Ed-Fi Data Standard need to refer to enumerated collections of values.

Descriptors are enumeration vocabularies that are not “fixed” within the XML schema, but are defined in XML files and linked to their source. Descriptors provide implementers with the flexibility to define their own enumerations. Please note that some downstream Ed-Fi XML standards may require the use of particular enumeration sets in order to guarantee interoperability, so the use of the flexibility in this pattern needs to be exercised with caution.

Key features of the Descriptor Pattern are:

- Descriptors minimally have a ShortDescription and CodeValue, and may also have a LongDescription. Descriptors allow states and other implementers to continue to use the codes associated with their enumerations.
- To support changing enumerations or code sets, Descriptors have an EffectiveBeginDate and EffectiveEndDate that are typically aligned to school years.
- To provide support for longitudinal analysis, Descriptors may capture a PriorDescriptor, as appropriate, when codes may change for the same concept or category.
- For example, the AttendanceEventCategoryDescriptor allows states and other implementers to define their own attendance codes.
- Descriptors are linked to a “namespace” that defines its scope of use. Ideally, a state will publish an enumeration vocabulary or code list at a specific URL. The Namespace element of the Descriptor will contain this URL.

All Descriptors are an extension of the type DescriptorType, shown below:

```
<xs:complexType name="DescriptorType" abstract="true">
  . . .
  <xs:complexContent>
    <xs:extension base="ComplexObjectType">
      <xs:sequence>
        <xs:element name="CodeValue" type="CodeValue">
          . . .
        <xs:element name="ShortDescription" type="ShortDescription">
          . . .
        <xs:element name="Description" type="Description" minOccurs="0">
          . . .
        <xs:element name="EffectiveBeginDate" type="xs:date" minOccurs="0">
          . . .
        <xs:element name="EffectiveEndDate" type="xs:date" minOccurs="0">
          . . .
        <xs:element name="PriorDescriptor" type="DescriptorReferenceType">
          . . .
        <xs:element name="Namespace" type="URI">
          . . .
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Consider the example of PerformanceLevel for an Assessment. The performance levels are custom to each assessment (e.g., Met Standard, Commended, College Ready) and cannot be standardized. The PerformanceLevelDescriptor is shown below.

```
<xs:complexType name="PerformanceLevelDescriptor">
  . . .
  <xs:complexContent>
    <xs:extension base="DescriptorType" />
  </xs:complexContent>
</xs:complexType>
```

The entity PerformanceLevelDescriptor holds the CodeValue, ShortDescription, and Description for each of the performance level enumerations specific to an assessment, as well as other important attributes defined in the base DescriptorType.

All Descriptor references are an extension of DescriptorReferenceType, shown below.

```
<xs:simpleType name="DescriptorReferenceType">
  . . .
```

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- ✓ [XSD Implementation Guidelines](#)
 - [XML Schema - Standard Data Exchange Schema](#)
 - [XML Schema - Extended Reference Types](#)
 - [XML Schema - Ed-Fi Descriptors](#)
 - [XML Schema - Extensions Framework](#)
 - [XML Schema - Custom Data Exchange Schema](#)

```

<xs:restriction base="xs:string">
  <xs:minLength value="1" />
  <xs:maxLength value="255" />
</xs:restriction>
</xs:simpleType>

```

As an example, the Ed-Fi GraduationPlanType Descriptor definition is constructed in an XML file as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<InterchangeDescriptors xmlns="http://ed-fi.org/0310" xmlns:ann="http://ed-fi.o
  <GraduationPlanTypeDescriptor>
    <CodeValue>Career and Technical Education</CodeValue>
    <ShortDescription>Career and Technical Education</ShortDescription>
    <Description>Career and Technical Education</Description>
    <Namespace>uri://ed-fi.org/GraduationPlanTypeDescriptor</Namespace>
  </GraduationPlanTypeDescriptor>
  <GraduationPlanTypeDescriptor>
    <CodeValue>Distinguished</CodeValue>
    <ShortDescription>Distinguished</ShortDescription>
    <Description>Distinguished</Description>
    <Namespace>uri://ed-fi.org/GraduationPlanTypeDescriptor</Namespace>
  </GraduationPlanTypeDescriptor>
  <GraduationPlanTypeDescriptor>
    <CodeValue>Minimum</CodeValue>
    <ShortDescription>Minimum</ShortDescription>
    <Description>Minimum</Description>
    <Namespace>uri://ed-fi.org/GraduationPlanTypeDescriptor</Namespace>
  </GraduationPlanTypeDescriptor>
  <GraduationPlanTypeDescriptor>
    <CodeValue>Recommended</CodeValue>
    <ShortDescription>Recommended</ShortDescription>
    <Description>Recommended</Description>
    <Namespace>uri://ed-fi.org/GraduationPlanTypeDescriptor</Namespace>
  </GraduationPlanTypeDescriptor>
  <GraduationPlanTypeDescriptor>
    <CodeValue>Standard</CodeValue>
    <ShortDescription>Standard</ShortDescription>
    <Description>Standard</Description>
    <Namespace>uri://ed-fi.org/GraduationPlanTypeDescriptor</Namespace>
  </GraduationPlanTypeDescriptor>
</InterchangeDescriptors>

```

Note that for Descriptors, the namespace is required so the source for the Descriptor definition can be uniquely determined. An optional AsOfDate may be supplied to give temporal context to the Descriptor value reference.

Because each value/code for a Descriptor is defined with a namespace, the controlled vocabulary may be defined combining values from more than one namespace. This allows a state to add codes to a federally defined set or to combine two vocabularies from different contexts. For example, two code lists might be combined for the DisabilityDescriptor, the Individuals with Disabilities Education Act (IDEA) set of disabilities, and the Section 504 set of disabilities, with each set referencing a different namespace.

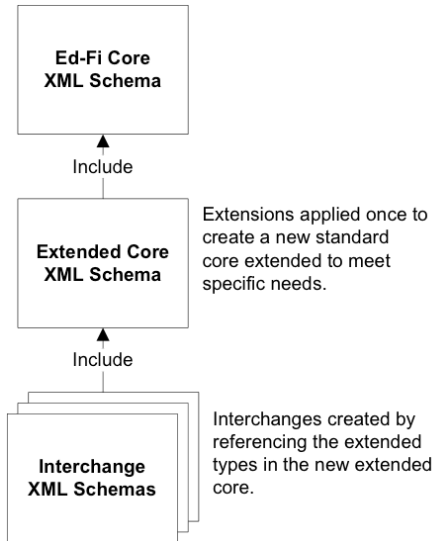
XML Schema - Extensions Framework

Customizations to the Ed-Fi data model – and concrete specifications such as the Ed-Fi data exchange XSD – are explicitly supported for implementation-specific or other purposes using the set of conformant extensions described in this section, known as the Ed-Fi Extension Framework. The [Extension Framework documentation](#) provides detail on extension patterns that are supported throughout Ed-Fi technology.

The Ed-Fi Alliance provides a free tool called the [MetaEd IDE](#) that automates the process of extending Ed-Fi technology, including the Ed-Fi data exchange XSDs. The documentation that follows covers the patterns that are created by the MetaEd IDE.

General Extension Pattern

Extensions are made in an extended core. This allows extensions to be defined once and reused across multiple interchange schemas, as well as supports automated code generation.



Extending the Schema

The [Extension Framework](#) contains the authoritative list of the types of extensions you can do, and the [MetaEd IDE documentation](#) contains everything you need to know about how to extend Ed-Fi XSD definitions. The documentation includes download and installation instructions, plus how-to articles covering the most common extension scenarios.

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- ✓ [XSD Implementation Guidelines](#)
 - [XML Schema - Standard Data Exchange Schema](#)
 - [XML Schema - Extended Reference Types](#)
 - [XML Schema - Ed-Fi Descriptors](#)
 - [XML Schema - Extensions Framework](#)
 - [XML Schema - Custom Data Exchange Schema](#)

XML Schema - Custom Data Exchange Schema

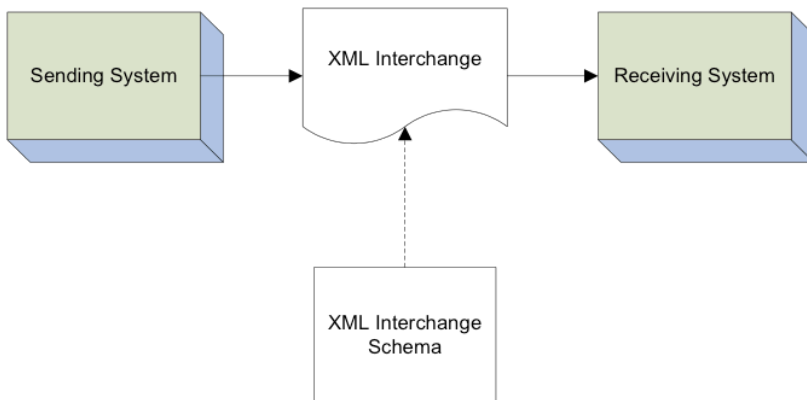
The Ed-Fi Alliance publishes bulk data exchange standards for XML that cover many data exchange scenarios. However, custom data exchange schema can easily be created by anyone familiar with XML Schema.

Ed-Fi data exchange standards leverage the Ed-Fi Core XSD, which contains reusable definitions for hundreds of entities relevant to the K–12 information domain. By using the predefined entities in the Ed-Fi Core XSD, implementers can quickly compose specifications for many different data exchange purposes.

The Ed-Fi Alliance provides a free tool called the [MetaEd IDE](#) that automates the process of extending Ed-Fi technology, including the Ed-Fi data exchange XSDs. The documentation that follows covers the patterns that are created by the MetaEd IDE, and additional considerations when creating a custom exchange.

Custom Exchange Overview

Ed-Fi data exchange schemas define the structure of the XML that transports the data between systems, as shown below. Data exchange schemas can contain as much or as little data as required. In most cases, different interchange schemas will be used to reflect different use cases or to deal with different periodicities of interchange.



The Ed-Fi Core XML Schema provides a library of building blocks from which to compose interchange schemas. However, implementations often have use cases beyond those covered by the Standard Interchange Schemas. This section provides guidance to technical personnel on applying the Standard to create custom interchanges.

Custom Exchange Considerations

In creating interchanges using the Ed-Fi Data Standard, there are several considerations that may require additional analysis, including:

- **Security and FERPA issues.** Based on the users of the receiving system or data store, data may need to be filtered or redacted prior to interchange. For example, a district should only be able to see student data for students enrolled in their district.
- **Periodicity of interchange.** Since most data interchanges are in response to recurring requirements, the periodicity of the interchange must be considered. For example, attendance data may be interchanged daily, where the scores to standardized tests may only be loaded after administration.
- **Incremental changes vs. bulk updates.** Whether complete data sets will be transferred or just the changes that will impact the data being interchanged. Transferring only the changes will greatly reduce the bulk of data to be processed, but will require additional complexity on both sides of the interchange to detect and appropriately handle the changes. Bulk updates are much simpler, but require much larger transfers.
- **Reliability of identity references.** When extended reference types are used to pass identity information for lookup (e.g., students), the reliability of the data and the algorithms used to match instance identities must be considered. For example, is enough data provided to verify identity when matched by a primary ID? If a primary ID is erroneous, is enough data provided to uniquely match on other attributes?
- **Consistency between multiple information sources.** When data is accepted from multiple sources—for example, between different student information systems from different local education agencies—there may be inconsistencies in the data that may require special attention, such as different enumeration lists values or different naming conventions.

Building a Custom Exchange Specification

The [MetaEd IDE documentation](#) contains everything you need to know about building a new XSD specification based on the Ed-Fi Core XSD. The documentation includes download and installation instructions, plus how-to articles.

XSD Guidelines Contents

Find out more about the Ed-Fi XSD Design & Implementation Guidelines:

- [XSD Scope](#)
- [XSD Design Guidelines](#)
- [XSD Implementation Guidelines](#)
 - [XML Schema - Standard Data Exchange Schema](#)
 - [XML Schema - Extended Reference Types](#)
 - [XML Schema - Ed-Fi Descriptors](#)
 - [XML Schema - Extensions Framework](#)
 - [XML Schema - Custom Data Exchange Schema](#)