



SIF Object Usage Guide



Table of Contents

Introduction	2
How to use this document	2
Object Descriptions	3
Student.....	3
Definition	3
Related Objects.....	3
Object Management	3
Person	3
Definition	3
Related Objects.....	3
Object Management	3
Student School Association	4
Definition	4
Related Objects.....	4
Object Management	4
Use Cases	5
Register Student in District	5
Description.....	5
Scenario: Add Student to System - Single Event.....	6
Scenario: Add Student to System - Multiple Events	9
Enroll Student in School	13
Scenario: Associate Student with School – Home School.....	13
Scenario: Associate Student with School – Concurrent School	16
Change Grade Level	20
Scenario: Mid-Year Grade Level Change – Student is Currently Enrolled.....	20
Scenario: Grade Level Change – Correction.....	24
Attachments.....	27
SIF Example XML.....	27

Introduction

This document supplements the normative specification artifacts with object usage information. Documentation such as element definitions, xml types, cardinality of elements, readable names, and other information are embedded into the XSD documents of the specifications.

This document includes two major parts:

1. Object Descriptions - This section lists every SIF object, defines the object, describes how the object relates to key other SIF objects, and describes some issues around managing the object in a SIF Context.
2. Use Cases – This section contains major uses of SIF and describes best practice with respect to the SIF objects and object interactions.

How to use this document

The object description section is suited for technical as well as non-technical readers. Non-technical readers may get a sense of how the object fits into the rest of the data system and how a system might be mapped from a client system to the SIF interoperability environment.

The Use Case section is a little more technical as it describes what a programmer would need to know to construct successful transactions with other system. However, non-technical users might also benefit from seeing the number of decisions that need to be made, beyond having common data structure, in order for systems to interoperate.

Object Descriptions

Student

Definition

Represents a student. This object contains characteristics of the student but not related information such as program participation, grades, test scores, etc.

Related Objects

Student has a list of reflds that point to one or more address objects. This allows multiple addresses for a student to be tracked. It also removes the redundant storage of addresses for students who have the same address (e.g. siblings).

Object Management

- Student instances are created when the student is first known to the system.
- Students may not ever be deleted from a system.
- Student change events that include an Alert Message or a Medical Alert Message are to be handled by a listener for those events in real time. This allows the propagation of messages of varying levels of urgency to other systems and people.

Person

Definition

Represents a person. A person can be (take on the role of) student, contact person, staff person, etc. An instance of a person can be a student, staff person, etc. all at the same time.

Related Objects

Person has a list of reflds that point to one or more address objects. This allows multiple addresses for a person to be tracked. It also removes the redundant storage of addresses for persons who have the same address (e.g. siblings).

A person is associated with other instances of role objects (e.g. student, contact person, etc.) using the Person Longitudinal object. This allows a person to be tracked as they move from one role to another, such as student to parent, or allows data from multiple roles, such as parent and staff person to be unified.

Object Management

- Person instances are created when the person is first known to the system.
- Person instance may not ever be deleted from a system.

Student School Association

Definition

Represents the association of a student with a school. Creating this association is usually called enrolling a student in a school. This is different from registering a student in the district or enrolling a student in a section of a course.

Related Objects

This object specifically associates a student with a school. The instances of student and school specified in the instance of Student School Association must exist when the instance is created.

Object Management

- Student RefId and School RefId are mandatory.
- School Year is mandatory. There must be at least one instance of Student School Association per student per school per school year.
- Entry Type is mandatory.
- Enrollment Status: An indication as to whether a student's name was, is, or will be officially registered on the roll of a school or schools. Following are the CEDS/NCES approved values:
 - Concurrently enrolled
 - Currently enrolled
 - Previously enrolled
 - Transferring (will enroll)
- Clarification of Exit Status, Exit Type, and Record Closure Reason
 - Exit Status – A determination of whether the exit is temporary or permanent. See CEDS element 000108. CEDS values:
 - Permanent
 - Temporary
 - Exit Type – The reason the student exited such as enrolled in another school in district, transferred to home schooling, withdrew due to illness, graduated, etc. See CEDS element 000110.
 - Record Closure Reason – The reason why the enrollment record was closed. This is information about the record and not necessarily about the student. Values are:
 - SchoolExit
 - TimeDependentDataChange
 - EndOfYear - The EndOfYear option must be used to convey status change due to rollover activity.

Use Cases

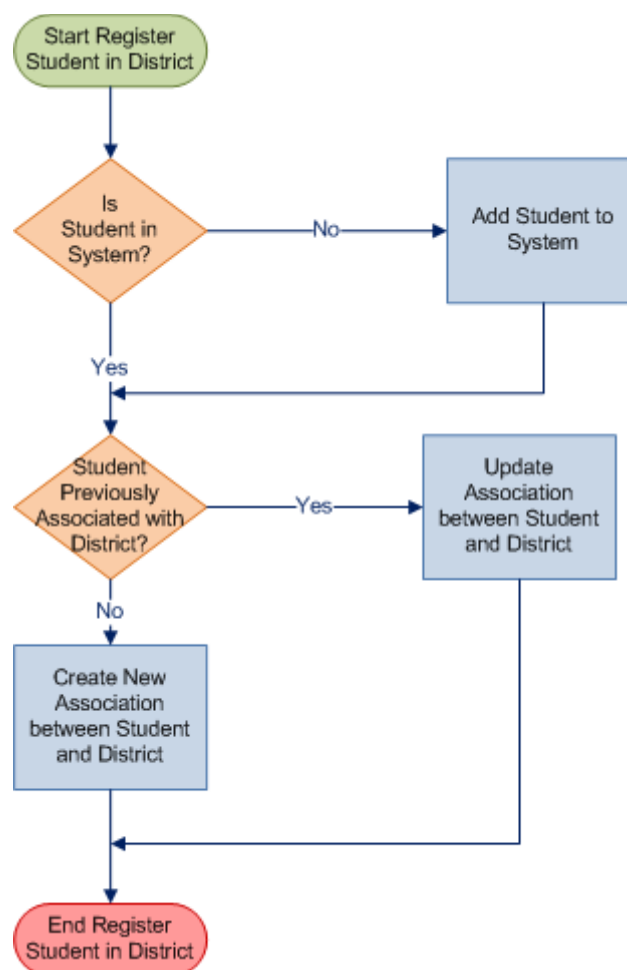
Register Student in District

Description

Registering a student in a district requires the creation of a new student entity if one does not already exist, altering the data for the student to current values if the student entity does exist and then creating an association of the student entity to the district entity. The Student object represents the student entity. The district entity is represented by the LEA object and the association of student and district is represented by the Student LEA Association object. For each object, only the elements required to address the Use Case and to fulfill the Task are included.

Use Case: Register Student in District - Overview

Figure 1



Scenario: Add Student to System - Single Event

Description

If a student does not exist in the system, the user will use the application functionality to add the student. For this scenario, the user knows everything needed to register the student with the mandatory and required elements needed for the profile. The user creates the new student record in one step. The student entity is represented by the Student object.

Assumptions

- All information needed to add a student is known before the student is entered.
- The application is capable of publishing all needed information in a single add event.

Pre-Conditions

- The student does not exist in the provider application that manages students.

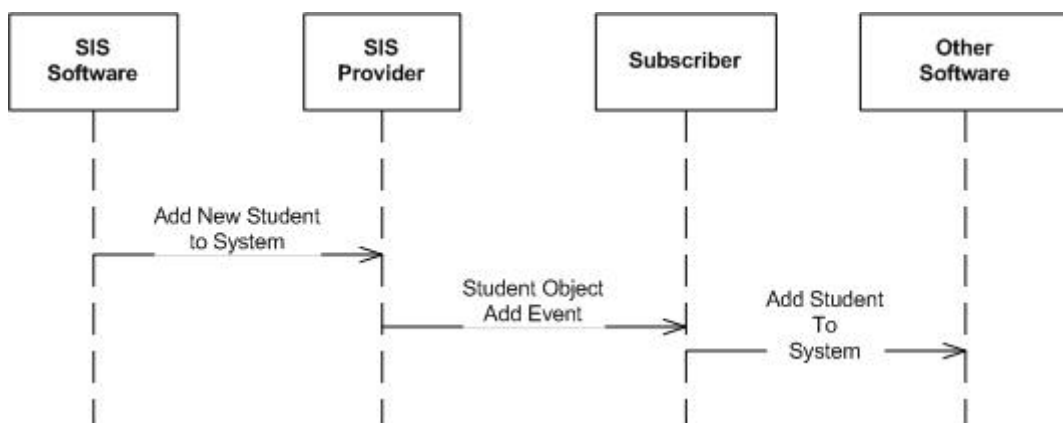
Post-Conditions

- The student exists in the provider application.
- The provider has received the SIF Event message containing all of the mandatory and required elements as defined in Data below.
- The provider will also send the event to the subscribing application subscribers when they are available but this is not required to complete the scenario.

Sequence Diagram

Scenario: Add Student to System - Single Event

Figure 2



Data Structure and Requirements

A Student Add event **must** be published containing at least the Mandatory and Required elements as defined below.

Table 1

Elements/Attributes	M/O	Requirements
@refId	M	Immutable
addressRefIdList/addressRefId	R	At least one occurrence of addressRefId
name/@nameRole	M	Value="04"
name/nameOfRecord/lastName	M	
name/nameOfRecord/firstName	M	
localId	M	
demographics/sexus	M	
demographics/birthDate	M	
...	O	Other elements should be included if available in the provider's application
demographics/ethnicityList/ethnicity	R	At least one occurrence of ethnicity element

XML Instance Document

XML Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<student refId="D3E34B359D75101A8C3D00AA001A1652">
  <addressRefIdList>
    <addressRefId addressRole="0123">
      A3E34B359D75101A8C3D00AA001A1652
    </addressRefId>
  </addressRefIdList>
  <name nameRole="04">
    <nameOfRecord>
      <lastName>Student</lastName>
      <firstName>Joe</firstName>
    </nameOfRecord>
  </name>
  <localId>P00001</localId>
```



```
<demographics>
  <ethnicityList>
    <ethnicity>
      <code>1002</code>
    </ethnicity>
  </ethnicityList>
  <sexus>M</sexus>
  <birthDate>1990-09-26</birthDate>
</demographics>
</student>
```

Scenario: Add Student to System - Multiple Events

Description

If a student does not exist in the system, the user will use the application functionality to add the student. For this scenario, the user does not know everything at the beginning that is needed by the system to register the student. The user enters the student with the mandatory data and then later adds the additional required information. Since this operation may be accomplished with two or more steps in the application, the detail of the SIF events published will likely vary from application to application. The student entity is represented by the Student object.

Assumptions

- Not all information to register a student is known when the student is added or the application publishes the data in multiple steps as the default method whether the data is entered all at once or not.

Pre-Conditions

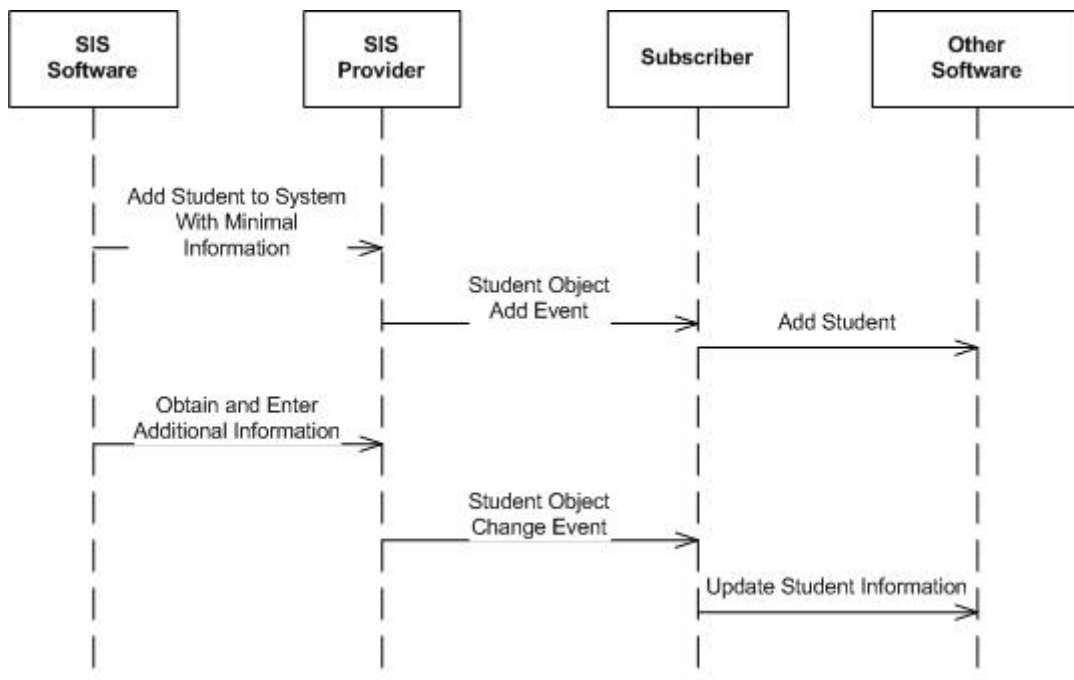
- The student does not exist in the provider application that manages students.

Post-Conditions

- The student exists in the provider application.
- The provider has received the SIF Event messages containing all mandatory and required elements as defined in the Data Structure and Requirement section below.
- The provider will also send the events to the subscribing application subscribers when they are available but this is not required to complete the scenario.

Sequence Diagram

Figure 3



Data Structure and Requirements

A Student Add event **must** be published containing at least the Mandatory elements below.

Table 2

Elements/Attributes	M/O	Requirements
refId	M	Immutable
name/@nameRole	M	Value="04"
name/nameOfRecord/lastName	M	
name/nameOfRecord/firstName	M	
localId	M	
demographics/sexus	M	
demographics/birthDate	M	
...	O	Other elements should be included if available in the provider's application

XML Instance Document

XML Example 2

```
<?xml version="1.0" encoding="UTF-8"?>
<student refId="D3E34B359D75101A8C3D00AA001A1652">
  <name nameRole="04">
    <nameOfRecord>
      <lastName>Student</lastName>
      <firstName>Joe</firstName>
    </nameOfRecord>
  </name>
  <localId>P00001</localId>
  <demographics>
    <sexus>M</sexus>
    <birthDate>1990-09-26</birthDate>
  </demographics>
</student>
```

Additional Student Change events **may** be published containing various elements as they are entered in the application. The elements below are **required** to be published in one or more of these change events or the initial add event.

Table 3

Elements/Attributes	M/O	Requirements
refId	M	Immutable
addressRefIdList/addressRefId	R	At least one occurrence of addressRefId
demographics/ethnicityList/ethnicity	R	At least one occurrence of ethnicity element
demographics/sexus	M	
demographics/birthDate	M	
...	O	Other address types and other elements may be included if available in the provider's application

XML Instance Document

XML Example 3

```
<?xml version="1.0" encoding="UTF-8"?>
<student refId="D3E34B359D75101A8C3D00AA001A1652">
  <addressRefIdList>
    <addressRefId addressRole="0123">
      A3E34B359D75101A8C3D00AA001A1652
    </addressRefId>
  </addressRefIdList>
  <demographics>
    <ethnicityList>
      <ethnicity>
        <code>1002</code>
      </ethnicity>
    </ethnicityList>
    <sexus>M</sexus>
    <birthDate>1990-09-26</birthDate>
  </demographics>
</student>
```

```
</demographics>
```

```
</student>
```

XML Example 4

```
<?xml version="1.0" encoding="UTF-8"?>
<student refId="D3E34B359D75101A8C3D00AA001A1652">
  <demographics>
    <ethnicityList>
      <ethnicity>
        <code>1002</code>
      </ethnicity>
      <ethnicity>
        <code>1004</code>
      </ethnicity>
    </ethnicityList>
  </demographics>
</student>
```

Enroll Student in School

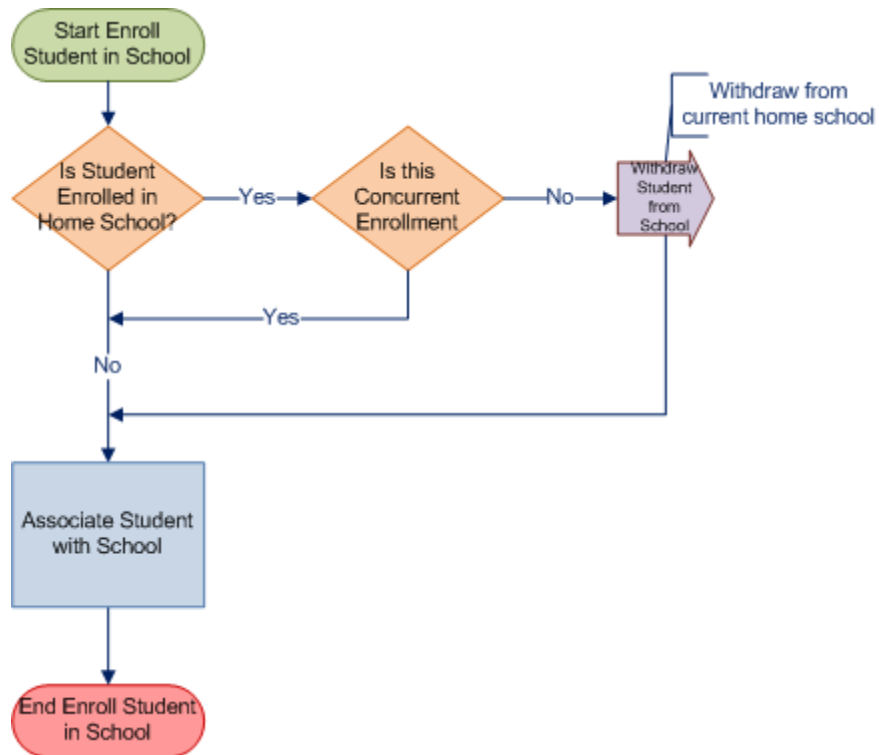
Description

This user story describes only the student enrollment at a school. If the student does not exist in the system or if student information needs to be modified prior to enrollment, other user stories describe those activities. If the student is being enrolled at their home school and they are already enrolled in a home school, they must first be withdrawn from that home school. If the student is being enrolled at a concurrent school (i.e. they are concurrently enrolled at more than one school), they must be enrolled in their home school first.

Enrolling a student in a school requires an existing student entity and an existing school entity in order to create the association between them along with the information for that association. The student entity is represented by the Student object, the school entity by the School object and the association of student and school by the Student School Enrollment object.

User Story: Enroll Student in School - Overview

Figure 4



Scenario: Associate Student with School – Home School

Description

This scenario creates the association between a student and a school, which represents an enrollment of a student at their home school. This scenario **must** be completed before the student can be enrolled in a concurrent school, assigned classes in the school, or other school related scenarios.

Assumptions

Pre-Conditions

- The student already exists in the system.
- The school already exists in the system.
- The student does not have a home school enrollment.

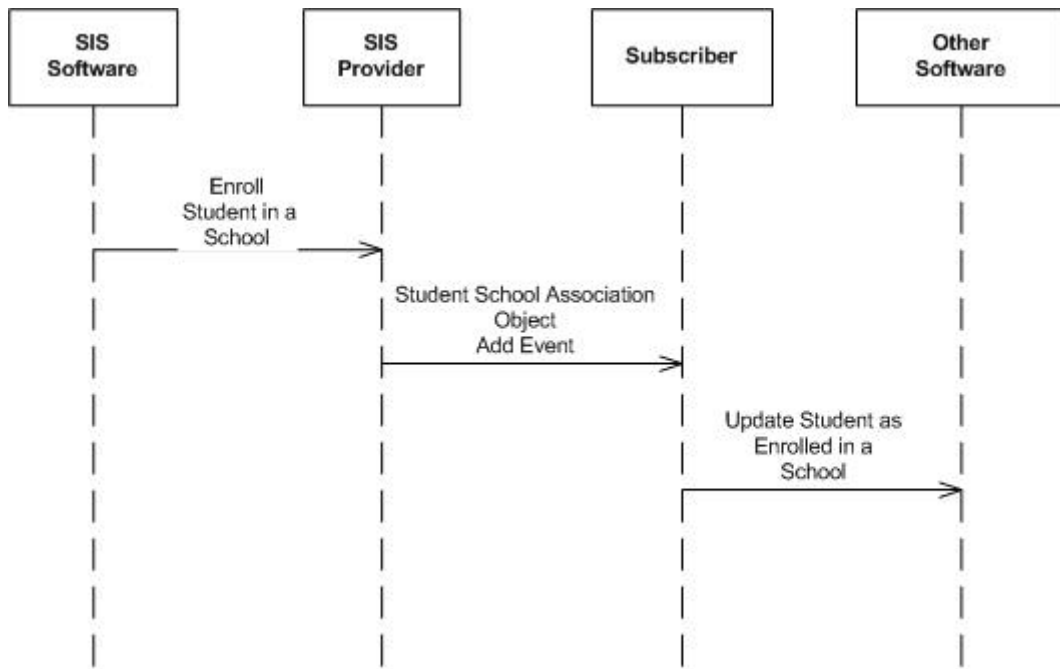
Post-Conditions

- The student is associated with their home school (i.e. home school enrollment) in the provider application.
- The provider has received the SIF Event message containing all of the mandatory and required elements as defined in Data below.
- The provider will also send the event to the subscribing application subscribers when they are available but this is not required to complete the scenario.

Sequence Diagram

Scenario: Associate Student with School – Home School

Figure 5



Data Structure and Requirements

A studentSchoolAssociation Add event **must** be published containing at least the Mandatory elements below.

Table 4

Elements/Attributes	M/O	Requirements
refId	M	New and Immutable
studentRefId	M	Immutable

schoolRefId	M	Immutable except for a data correction
membershipType	M	Value="Home". There must be one and only one "Home" at any point in time, for all timeframes.
schoolYear	M	Immutable
entryDate	M	The possibilities are 1) date the student registers at the school (recommended), 2) first date of actual membership for the school year (this would not allow "no-shows"), 3) first date of actual instruction for the school year (this would not allow "no-shows").
entryType	M	
yearGroup	M	Immutable except for a data correction. This field is typically called grade level.
...	O	Other elements may be included if available in the provider's application

XML Instance Document

XML Example 4

```
<?xml version="1.0" encoding="UTF-8"?>
<StudentSchoolAssociation refId="A8C3D3E34B359D75101D00AA001A1652">
  <studentRefId>D3E34B359D75101A8C3D00AA001A1652</studentRefId>
  <schoolRefId>B3E34B359D75101A8C3D00AA001A1651</schoolRefId>
  <membershipType>Home</membershipType>
  <schoolYear>2011</schoolYear>
  <entryDate>2011-01-29</entryDate>
  <entryType>
    <Code>1838</Code>
  </entryType>
  <yearGroup>
    <Code>10</Code>
  </yearGroup>
  <homeroomRefId>C7510D3E34B3591A8C3D00AA001A165</homeroomRefId>
</StudentSchoolAssociation>
```


Scenario: Associate Student with School – Concurrent School

Description

This scenario creates the association between a student and a school, which represents an enrollment of a student at a concurrent school. This scenario **must** be completed before the student can be assigned classes in the school, or other school related scenarios.

Assumptions

Pre-Conditions

- The student already exists in the system.
- The school already exists in the system.
- The student already has a home school enrollment.

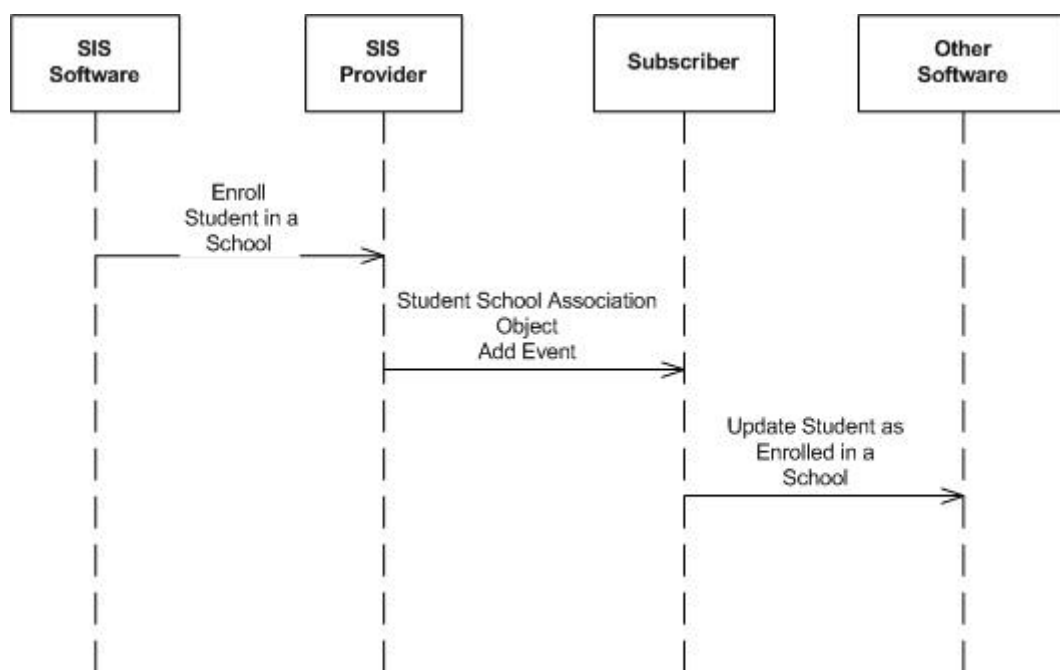
Post-Conditions

- The student is associated with their concurrent school (i.e. concurrent school enrollment) in the provider application.
- The provider has received the SIF Event message containing all of the mandatory and required elements as defined in Data below.
- The provider will also send the event to the subscribing application subscribers when they are available but this is not required to complete the scenario.

Sequence Diagram

Scenario: Associate Student with School – Concurrent School

Figure 6



Data Structure and Requirements

A Student School Association Add event **must** be published containing at least the Mandatory elements below.

Table 5

Elements/Attributes	M/O	Requirements
refId	M	New and Immutable
studentRefId	M	Immutable
schoolRefId	M	Immutable except for a data correction
membershipType	M	Value="Concurrent"
schoolYear	M	Immutable
entryDate	M	A date between the beginning and end of the school year.
entryType	M	
yearGroup	M	Immutable except for a data correction
...	O	Other elements may be included if available in the provider's application

XML Instance Document

XML Example 6

```
<?xml version="1.0" encoding="UTF-8"?>
<studentSchoolAssociation refId="A8C3D3E34B359D75101D00AA001A1652">
  <studentRefId>D3E34B359D75101A8C3D00AA001A1652</studentRefId>
  <schoolRefId>D3E34B359D75101A8C3D00AA001A1651</schoolRefId>
  <membershipType>Concurrent</membershipType>
  <schoolYear>2011</schoolYear>
  <entryDate>2011-01-29</entryDate>
  <entryType>
    <code>1838</code>
  </entryType>
  <yearGroup>
```

```
<code>10</code>
```

```
</yearGroup>
```

```
<homeRoomRefId>D7510D3E34B3591A8C3D00AA001A1651</homeRoomRefId>
```

```
</studentSchoolAssociation>
```

Retrieve Information from SIF Providers

Scenario: *Retrieve all the students enrolled in a given school*

If the Service Path binding “schools/{}/students” exists for Student Data Entity objects in the Run Time Data Model, then a Consumer issuing a Query request with a URL ending in:

../schools/1234/students

will receive in the response, the subset of all student objects for only those students currently enrolled in the school represented by a school object with a RefId of 1234.

Change Grade Level

Description

For each day of student enrollment, students are enrolled into a single, distinct grade level. The typical student is in a grade level for an entire school year. After the end of the year, the student is promoted to the next grade level for the next school year. This year-end scenario is discussed in the year-end rollover process user story.

Another scenario, the one addressed in this document, is where an individual student grade level must be changed during the school year. More specifically, the scenario addressed in this document describes the best practice for changing the GradeLevel element in the Student School Association object.

The student grade level that exists in the Student School Association object is the most accurate and timely representation of grade level. It can provide the current grade level status of a student as well as a detailed history, by date, of the a student's grade level. Student grade level appears other places in the Specification, e.g., GradeLevel in Student, but is more ambiguous with respect to time or specific only for a single point in time, such as GradeLevel in a snapshot object.

Rules describing pre- and post-dated changes are to student grade level in the Student School Association object are not addressed in this document.

Scenario: Mid-Year Grade Level Change – Student is Currently Enrolled

Description

A currently enrolled student is promoted or demoted mid-year. In this case, the user wants both grade levels to be maintained with their appropriate date ranges. For this scenario, only the grade level and the exit/entry dates in Student School Association are affected.

The current Student School Association instance for the student must be published (using a change event) with an ExitDate representing the last date the student was in the prior grade level. A new Student School Association instance, with a new Student School AssociationRefId, must be published with an EntryDate of the first day the student is in the new grade level. There must not be any date overlap between the prior grade date range and the new grade date range.

The provider must republish (using a change event) Student to reflect the change to the MostRecent/GradeLevel.

Assumption

- The user wants both grade levels to be maintained with their appropriate date ranges.

Pre-Conditions

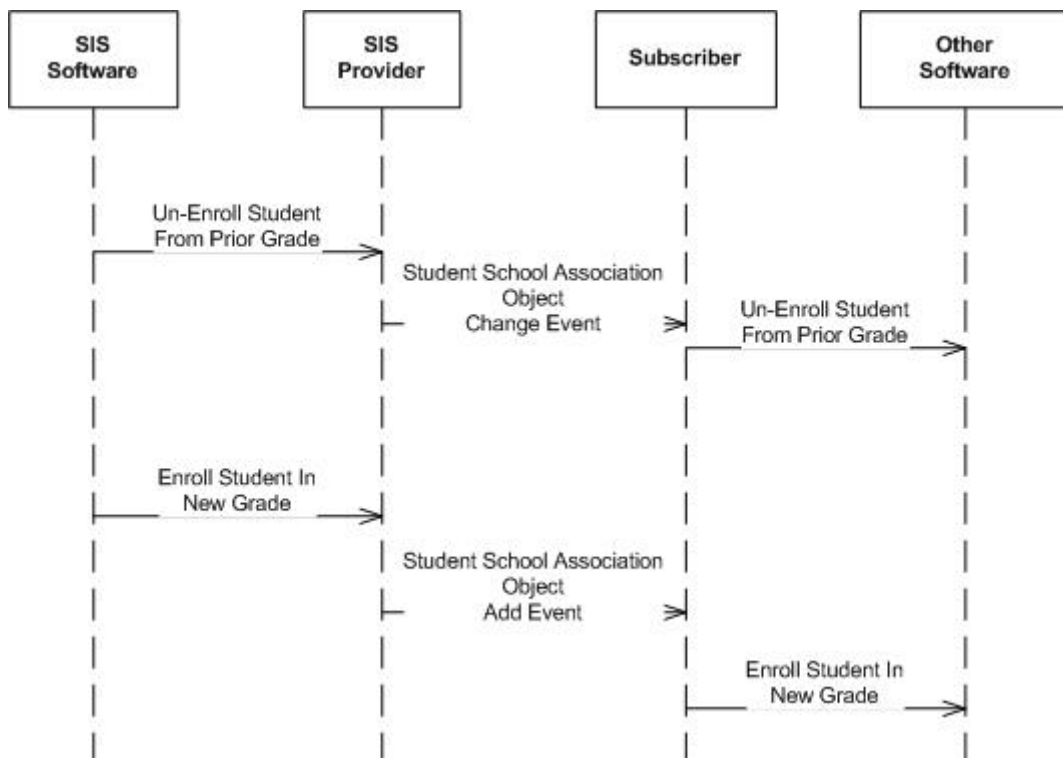
- The student is currently enrolled in the current year.

Post-Conditions

- In the Student Information System (SIS), the student is un-enrolled from the prior grade level, and re-enrolled in the new grade level. Records for both grade levels exist, with non-overlapping dates. The provider has received all of the event messages and data elements described below.

Sequence Diagram

Scenario: Mid-Year Grade Level Change – Student is Currently Enrolled



Data

The following three steps must be performed.

1. Change Student School Association: Un-enroll prior grade.

A Student School Association Change event **must** be published containing at least the Mandatory and Required elements as defined below.

Elements/Attributes	M/O	Requirements
refld	M	Immutable

studentRefId	M	Immutable
schoolRefId	M	Immutable
membershipType	M	Immutable
schoolYear	M	Immutable
exitDate	M	This is the last day of the existing grade level
exitStatus/Code	O	Code = 1906 - Temporary exit/withdrawal
exitType/Code	O	Code = 9999 - Other
recordClosureReason	M	Value = TimeDependentDataChange
promotionStatus	O	Value = Promoted or Demoted

XML Instance Document

```

<studentSchoolAssociation refId="A8C3D3E34B359D75101D00AA001A1652">
  <studentRefId>D3E34B359D75101A8C3D00AA001A1652</studentRefId>
  <schoolRefId>D3E34B359D75101A8C3D00AA001A1651</schoolRefId>
  <membershipType>Home</membershipType>
  <schoolYear>2011</schoolYear>
  <exitDate>2011-01-28</exitDate>
  <exitStatus>
    <code>1906</code>
  </exitStatus>
  <exitType>
    <code>9999</code>
  </exitType>
  <recordClosureReason>TimeDependentDataChange</recordClosureReason>
  <promotionStatus>Promoted</promotionStatus>
</studentSchoolAssociation>

```

2. Add Student School Association: Enroll new grade

A Student School Enrollment Add event **must** be published containing at least the Mandatory and Required elements as defined below.

Elements/Attributes	M/O	Requirements
refId	M	Immutable
studentRefId	M	Immutable
schoolRefId	M	Immutable
membershipType	M	Immutable
schoolYear	M	Immutable
entryDate	M	Date must fall after ExitDate in "Change Student School Association: Un-enroll prior grade" above
entryType/Code	O	Code= 1835 - Re-entry from the same school with no interruption of schooling
yearGroup/Code	M	New grade level for student

XML Instance Document

```

<studentSchoolAssociation refId="A8C3D3E34B359D75101D00AA001A1652">
  <studentRefId>D3E34B359D75101A8C3D00AA001A1652</studentRefId>
  <schoolRefId>D3E34B359D75101A8C3D00AA001A1651</schoolRefId>
  <membershipType>Home</membershipType>
  <schoolYear>2011</schoolYear>
  <entryDate>2011-01-29</entryDate>
  <entryType>
    <code>1835</code>
  </entryType>
  <yearGroup>
    <code>11</code>
  </yearGroup>
</studentSchoolAssociation>

```


3. Change Student (Optional): Update with new Projected Graduation Year, if applicable.

A Student change event **must** be published containing at least the Mandatory and Required elements as defined below.

Elements/Attributes	M/O	Requirements
refId	M	Immutable
projectedGraduationYear	M	

XML Instance Document

```
<student refId="D3E34B359D75101A8C3D00AA001A1652">
  <projectedGraduationYear>2049</projectedGraduationYear>
</student>
```

Scenario: Grade Level Change – Correction

Description

A grade has been incorrectly entered and the SIS user corrects the mistake.

Alternately, a currently enrolled student's grade level is changed mid-year and the user does not need to maintain the original grade level history. This scenario is not a widely-accepted practice, but may occur in some implementations. This is not an accepted SIF Functional Profile and this practice is discouraged.

Providers SHOULD NOT send Student School Association Delete and Add events to indicate a grade level correction. The recommended method is to publish a Student School Association Change event with the same RefId and the corrected grade level.

The SIS MUST also publish a Student Change event to reflect the MostRecent/GradeLevel change.

Assumption

The change is the result of a mistake and not an actual change in grade level for the student.

Pre-Conditions

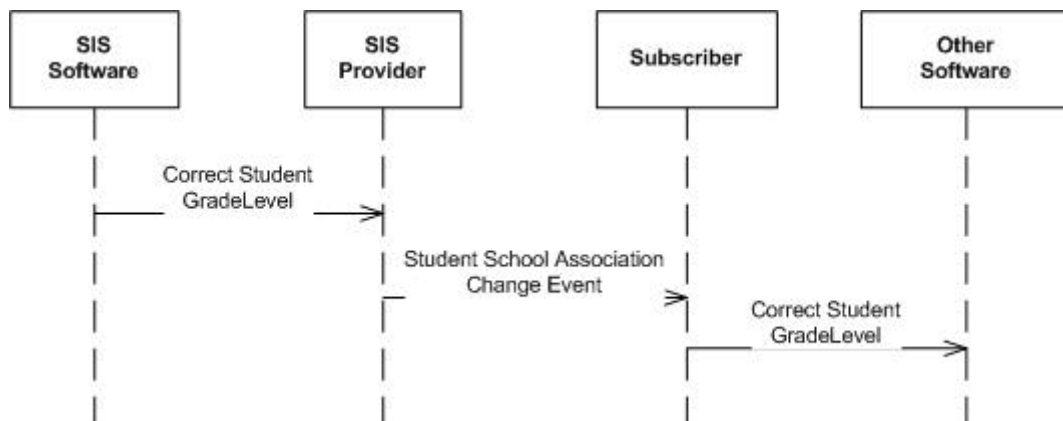
- The student is or has been enrolled in the current year.
- There is a prior Student School Association record for this student for the current year.
- There is a Student record for this student.

Post-Conditions

- In the SIF, the student's grade in the Student School Association record has been changed. The provider has received all of the events and elements as defined in Data below.

Sequence Diagram

Scenario: Grade Level Change – Correction



Data

The following two steps must be performed.

- Change Student School Association: Enroll new grade
A Student School Association change event **must** be published containing at least the Mandatory and Required elements as defined below.

Elements/Attributes	M/O	Requirements
refId	M	Immutable
studentRefId	M	Immutable
schoolRefId	M	Immutable
membershipType	M	Immutable
schoolYear	M	Immutable
yearGroup/Code	M	New grade level for student

XML Instance Document

```

<studentSchoolAssociation refId="A8C3D3E34B359D75101D00AA001A1652"
  studentRefId="D3E34B359D75101A8C3D00AA001A1652"
  schoolRefId="D3E34B359D75101A8C3D00AA001A1651"
  membershipType="Home"
  schoolYear="2011">
  <yearGroup>
    <Code>11</Code>
  </yearGroup>
</studentSchoolAssociation>

```

2. Change the Composite object StudentCore: Update with new grade level
A StudentCore change event **must** be published containing at least the Mandatory and Required elements as defined below.

Elements/Attributes	M/O	Requirements
refId	M	Immutable
gradeLevel/code	M	New grade level for student

XML Instance Document

```

<studentCore refId="D3E34B359D75101A8C3D00AA001A1652">
  <gradeLevel>
    <code>11</code>
  </gradeLevel>
</student>

```

Attachments

SIF Example XML