inBloom Data Ingestion Specification

Release 1.2.70

February 22, 2013

**Copyright © 2012 inBloom, Inc., and its affiliates.**

**inBloom is a trademark of inBloom, Inc.**

This document and the information contained herein is provided on an "AS IS" basis and inBloom, Inc. DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

**Revisions and Status**

This document is adapted from the “SLI Data Ingestion Specification” whitepaper published in May 2012. Ongoing updates to this document will follow for each inBloom release until it is replaced by new instructional and reference documents. See user release notes and ingestion schema change documentation from inBloom for a list of what has changed between versions.

**Purpose and Audience**

This document provides an overview of the Ed-Fi XML schema for describing education data and the data ingestion file formats and procedures for ingesting data in the inBloom Data Store.

The target audience for this document is an inBloom ingestion user, an individual who is preparing for and performing data ingestion for an education organization using inBloom.

**Table of Contents**

Definitions 4

inBloom Overview 6

Architecture Highlights 6

Key Components of the Core Entity Model 7

Ingestion On-boarding and Provisioning 8

Ingestion Process 9

Submitting Ingestion Jobs 9

Authentication 10

Authorization 10

Encryption 10

Ingestion File Formatting 11

Formatting Conventions for All Files 11

Control File Format 11

CSV Support 12

Bulk Upload of Ed-Fi Data 13

Interchange Format 13

InBloom-Ed-Fi Schema Versioning 14

XML Ingestion Sequence 14

Error and Status Logs 16

Job-level Error and Status Reporting 17

Resource- and Record-level Error Reporting 19

General Resource-level Errors 19

XML and Schema Validation Errors and Warnings 20

Record-level Persistence Errors 20

Record-level Persistence Warnings 25

Ingestion Data Dependencies and Constraints 27

Interchange Ordering 27

Interchange-level Constraints 28

EducationOrganization 28

AssessmentMetadata 28

StaffAssociation 29

StudentParent 29

StudentAssessment 30

EducationOrgCalendar 30

StudentEnrollment 31

StudentGrade 31

StudentProgram 31

StudentCohort 31

StudentDiscipline 31

StudentAttendance 32

inBloom-Ed-Fi Entity to inBloom Entity Mapping 33

inBloom-Ed-Fi Entity Type to Interchange Schema Mapping 35

inBloom-Ed-Fi Natural Keys 38

**Definitions**

This section contains definitions of terms that are used in this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| authentication | The process of verifying that a user is who they claim to be. For example, you are authenticating as *jsmith* when you enter the correct username and password for the *jsmith* account. inBloom supports both delegated and federated authentication. |
| authorization | The process of determining which resources a user has permission to access and which actions a user has permission to do. For example, as a teacher, user *jsmith*, probably has permission to view assessment data associated with his current students. |
| directory | A service that manages user identities and those users’ roles and permissions. Commonly-used directories include Microsoft Active Directory, OpenLDAP, and Novell’s eDirectory. |
| effective permissions | The combination of a user’s permissions plus an application’s permissions, which are typically more restrictive than either source of permissions. Effective permissions control access to an education organization’s data when accessed through an application. For example, if a teacher has permission to edit student data, but the application they are using does not, inBloom denies edit access those teachers |
| identity provider (IDP) | A computer system that creates, maintains, and manages identity information and provides user authentication to other service providers within a federation. An IDP may be a directory (as defined above) or it may be a service that acts on behalf of one or more directories. In a Security Assertion Markup Language (SAML) federated environment, a SAML IDP produces SAML assertions, and open standard way of exchanging identity information. |
| education organization | A public or private institution that provides instructional or support services to students or staff. Examples of education organizations include state departments of education, school districts, and schools. |
| permissions | A set of actions that a user is allowed to take. Individual users acquire permissions indirectly through role assignment. An example inBloom permission is “Can see student assessment data for students that the user teaches” or “Can change administrative setting for an account.” |
| personally identifiable information (PII) | Information that can be used to uniquely identify, contact, or locate a single person or can be used with other sources to uniquely identify a single individual. |
| role | A pre-defined relationship between a user and a specific set of permissions. inBloom default roles generally correspond to an person’s job function, such as“teacher” or “principal.” A user may be assigned multiple roles, in which case the roles’ permissions are additive. |
| inBloom, Inc. | inBloom, Inc., is an alliance of states, foundations, educators, content providers, developers and vendors who are passionate about using technology to improve education. |
| inBloom directory | A directory (see above) that is part of inBloom and that is used to authenticate Super Administrator users (see below). |
| Super Administrator | The role that grants complete administrative control over all data within inBloom for a particular education organization. Super Administrator permissions include: adding and removing IT Administrators to the inBloom directory, and setting up IDP services with a district. |
| tenant | A single set of data that roughly corresponds to a logically consistent set of educationally relevant data under a common state education agency (SEA) or local education agency (LEA). These education organization boundaries are referred to as *tenancies*. Tenants never span multiple SEAs. |

# 

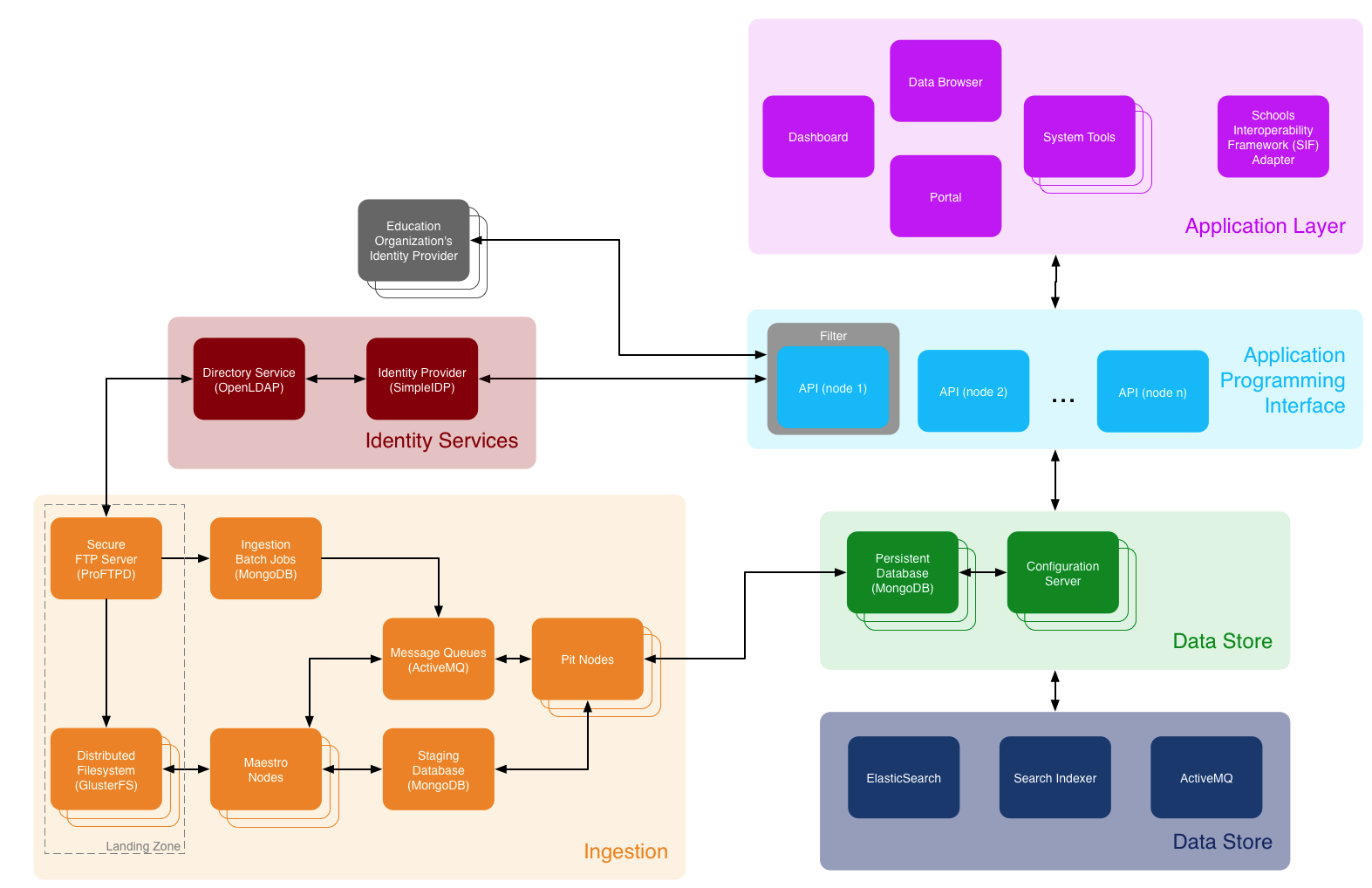
# inBloom Overview

inBloom technology offers the following features:

* A secure, scalable multi-tenant data store that hosts a core set of classroom-level education data elements, which are commonly used in the K-12 education information domain
* A set of lightweight applications that gives educators access to the information and data, helping them analyze student performance and adjust instructional strategies as needed
* An API and SDKs that facilitate software developers in extending inBloom’s education data services to new applications

## Architecture Highlights

inBloom technology is logically divided into a series of subsystems that serve specific purposes in the infrastructure. The diagram below provides a look at the subsystems, their components, and the connections they have to each other.



The three inBloom subsystems that govern data flow in the infrastructure are as follows:

* inBloom Data Store – The secure, scalable multi-tenant data store. Hosts core education data elements as well as custom data for specific organization or application needs.
* inBloom REST API – The application programming interface providing software developers an interface to extend inBloom’s education data services.
* inBloom Data Ingestion – The software and services designed to process bulk/large-scale additions and updates from an education organization for addition to that organization’s tenancy in the inBloom Data Store

This document specifies the detailed processes, schemas, preparation considerations, and error and log reporting for the inBloom Data Store.

## Key Components of the Core Entity Model

The inBloom Core Entity Model (CEM) is based on the Ed-Fi Unifying Data Model. One-to-one mapping is maintained for the majority of the entities, although some Ed-Fi entities have been merged into a single entity or split into multiple entities in the inBloom CEM. This document includes the entity-level mapping between Ed-Fi entities and inBloom CEM entities.

The inBloom Data Ingestion Subsystem uses Ed-Fi XML schemas as the primary format for the source data to be added or updated in the inBloom Data Store. Differences between the inBloom CEM and the Ed-Fi data model require that an Ed-Fi XML schema extension is provided for use with inBloom. This document covers inBloom-specific deviations from published Ed-Fi schemas.

# Ingestion On-boarding and Provisioning

inBloom supports a hierarchical segment of a state school system comprising one or more districts. A tenant consists of the education data within that hierarchy, including a the state and district systems (*EducationOrganizations*), the entities belonging them (*Schools*, *Teachers*, *Students*, *Courses*, *Assessments*, etc.), and their associations. A tenant hierarchy is rooted in an *EducationOrganization* entity, which serves as a logical parent for other *EducationOrganization* entities.

inBloom supports several types of education organizations, including State Education Agency (SEA) and Local Education Agency (LEA) (also known as a school district). The on-boarding process for an LEA starts with the creation of a root entity in the *EducationOrganization* hierarchy (at the SEA level). An inBloom administrator also performs the following tasks:

* Creates a Super Administrator account -- The Super Administrator account is a special inBloom role that authenticates against the inBloom directory rather than through an education organization’s federated IDP. This account is used to manage roles, permissions, and other LEA-specific configuration details within inBloom. The Super Administrator for an LEA can also delegate administrative tasks to either the SEA or another LEA at a higher level of the same education organization hierarchy.
* Creates a landing zone for the LEA -- A *landing zone* is a group of allocated resources used for ingesting data for that LEA. This includes a secure FTP server location for uploading the data files.

If a group of LEAs go through on-boarding at the same time, separate landing zones and Super Administrator accounts are created for each LEA within the tenant.

The following data is required when creating a landing zone during on-boarding:

* Name, institution, and other identifying information for the Super Administrator
* Expected number of records to be loaded for each entity during initial ingestion
* Expected annual growth for key entities, such as *Student* , *StudentAssessment*, and *Session*.

# 

# Ingestion Process

From the technical perspective, inBloom exposes three data ingestion vectors:

* An SFTP interface to upload files to be ingested to a designated landing zone
* A SIF agent interface for event-based data integration
* The inBloom REST API

Ingestion users must prepare data and control files as described in this document and compress them into a single zip file. That file is, in turn, uploaded to the SFTP interface for a landing zone.

This specification refers to the prepared zip file as an *ingestion job*, referring to what inBloom does to process that file. This specification also equates “uploading the zip file to the landing zone” as “submitting an ingestion job.”

## Submitting Ingestion Jobs

The SFTP interface to a landing zone provides a way to submit a job, to monitor its progress throughout the ingestion pipeline, and to be notified of any errors encountered along the way. The following is a summary of the ingestion job submission process:

1. An ingestion user prepares an ingestion job consisting of the following:
   1. A collection of data files in the inBloom-Ed-Fi XML schema.\* Exporting and conversion tools can be used in preparing these XML files.
   2. A job control file containing the ingestion configuration parameters and a list of the data files associated with that ingestion job.\*
2. A Super Administrator for the tenancy makes sure that the user roles in the IDP for the target education organization are mapped to corresponding roles in inBloom, ensuring that access to data is properly controlled by that IDP.\*
3. An ingestion user packages the job as a zip file and performs a secure upload of that file to the SFTP server associated with the target landing zone (which is, in turn, associated with a given tenancy).\*
4. After the upload, the ingestion job is assigned a job number, and a corresponding job progress is added to the landing zone. The job processing begins automatically. During processing, the ingestion user can view the job status by downloading the log files from the landing zone (same SFTP location).\*

\* This specification provides the necessary details an ingestion user needs about the inBloom-Ed-Fi XML schema, the ingestion job control file, role mapping, creating and configuring landing zones, and errors and other information contained in the log files.

## Authentication

The SFTP server authenticates each Super Administrator against the inBloom directory. Thus, a Super Administrator must provide normal authentication in an FTP client used to upload to the SFTP server. This means providing the username and password provided to that Super Administrator at the time of tenant on-boarding.

## Authorization

A Super Administrator is associated with one or more landing zones for a given tenancy. The administrator’s credentials are stored in the inBloom directory. The inBloom CEM maintains an internal mapping between landing zones, tenants, and districts. As a result, a Super Administrator can only upload ingestion jobs to a specific landing zone.

Since tenant and district associations are inherited from the inBloom directory at the time a Super Administrator account is created, they will not change as a result of any subsequent changes to that information in the inBloom directory.  Thus, if important identifying information changes for a Super Administrator, such as a name referenced in the user’s username, administrators (inBloom operators or Super Administrators with user management permissions in the same tenancy) should delete those accounts and create new ones with the desired information.

## Encryption

In order to protect personally identifiable information (PII), the ingested files containing student data are encrypted during transmission and storage, and that data is only delivered to inBloom applications using a secure network protocol.

# Ingestion File Formatting

The files provided as an ingestion job must follow a certain format. The sections that follow cover formatting conventions for all files, the control file format, and the inBloom-Ed-Fi XML schema format.

## Formatting Conventions for All Files

All files should be UTF-8 encoded unless otherwise specified. Machine-readable timestamps use epoch (UNIX) time. Human-readable timestamps use a formatted date/time string. Since an epoch time string is used in the job ID value for ingestion jobs, you can find them in the file names for log and error files produced by each job.

## Control File Format

The control file defines the set of inbound XML data to be ingested as part of the same ingestion job. The format is a text file with a series of lines providing information about the XML files to be ingested and additional parameters to control the ingestion job. inBloom expects this file to have a .ctl file extension (such as *control\_file.ctl*).

For each XML file, there should be a line formatted as a comma-separated list of four values:

<file format>,<interchange>,<file name>,<file checksum>

|  |  |
| --- | --- |
| <file format> | The file format. For inBloom, ***edfi-xml*** is the only supported file format. |
| <interchange> | The applicable Ed-Fi interchange name (covered later in this specification). |
| <file name> | The file name. File names are case insensitive, and should not contain any OS-specific path delimiters (such as “/”, “\”, or “:”).  This field may or may not be enclosed in double quotes. File names containing double quotes or commas should be enclosed in double-quotes. A double-quote appearing inside a field must be escaped by preceding it with another double quote. |
| <file checksum> | The file's MD5 checksum. The MD5 checksum is expressed as 32 hexadecimal digits. |

Additional formatting rules for each row include the following:

* Each of the four fields is case-insensitive.
* Leading or trailing spaces are considered part of the values and are not trimmed when reading the file.
* A comma must not follow the last value in any row.
* Valid line separators include pairs of newlines, carriage returns, or line feeds.

To provide job-level parameters, specify them as additional lines, each preceded with the @ symbol. Job parameters appearing in control files are parsed using the specification for Java properties (but the leading “@” is stripped first). Parameters may not require a value, in which case they are treated like flags. Both parameter names and parameter values are case-insensitive.

The following table describes the parameters currently supported for use in a control file:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Effect** | **Comments** |
| @dry-run | Indicates that the results of ingestion processing should not be written to the core data store. |  |
| @purge | Deletes all previously ingested data from this tenant. All other content of the control file is ignored. |  |
| @purge-keep-edorgs | Performs the same action as @purge above, while retaining the tenant EdOrgs and AppAuthorizations, so the tenant retains access to their registered applications. | Only used in production mode. |

The following shows the contents of an example of a control file for a dry run of a single XML file:

@dry-run

edfi-xml,StudentEnrollment,data.xml,756a5e96e330082424b83902908b070a

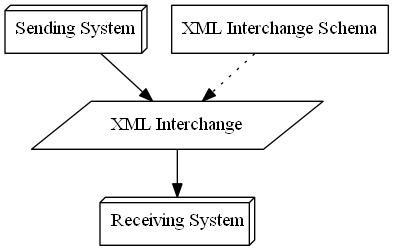
## CSV Support

Though inBloom does not support direct ingestion of comma-separated-value (CSV) files for data, it does provide a CSV-to-Ed-Fi conversion tool. This tools is available as one of the open source offerings at the inBloom Github site (<http://github.com/slcedu/csv2xml>). For more information about how to install and use this tool, see the ingestion sections of the inBloom administrator documentation.

# Bulk Upload of Ed-Fi Data

## Interchange Format

Ed-Fi interchange schemas define XML representations of particular data spaces, or groups of entities and associations, for transport between systems.  This is depicted in the following diagram.



Different interchange schemas may be used to reflect different use cases, such as different groups of source systems. Ed-Fi defines a Core Schema, which provides a library of building blocks referenced from the interchange schemas.

Ed-Fi defines 13 standard interchange schemas that are used by inBloom:

* Interchange-AssessmentMetadata
* Interchange-EducationOrganization
* Interchange-EducationOrgCalendar
* Interchange-MasterSchedule
* Interchange-StaffAssociation
* Interchange-StudentAssessment
* Interchange-StudentAssessment
* Interchange-StudentAttendance
* Interchange-StudentCohort
* Interchange-StudentDiscipline
* Interchange-StudentEnrollment
* Interchange-StudentGrade
* Interchange-StudentParent
* Interchange-StudentProgram

Each XML file that is part of a well-formed ingestion job is validated against an Ed-Fi interchange schema. This is done as part of the ingestion sequence described in the next section.

## InBloom-Ed-Fi Schema Versioning

inBloom maintains and publishes a set of modified Ed-Fi core and interchange schemas for use with inBloom technology. These extensions are based on a public version of Ed-Fi schemas and incorporate the following modifications:

* Exclude unsupported entities in each of the interchanges
* Exclude unsupported attributes of supported entities
* Make optional attributes of supported entities mandatory
* Add attributes to supported entities
* Restructure reference types to use nested references
* Make identity types within reference objects mandatory

inBloom-Ed-Fi schemas are the only XML schemas against which ingested data are validated by inBloom. The inBloom schema names are as follows:

* SLI-Ed-Fi-Core.xsd
* SLI-Ed-Fi-Interchange-<interchange\_name>.xsd

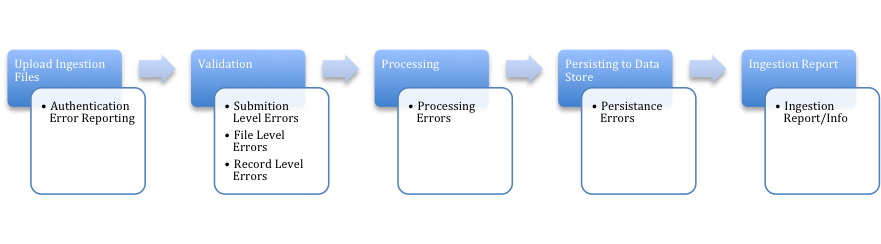
Schema versions are specified as an attribute of the <xs:schema> tag at the top of the .xsd files.

**Caution:** The term “schema” is used to describe both the interchange schema for ingestion and the database schema for the inBloom Data Store. The REST API interacts with data based on the schema in the data store, not the interchange schema. Version numbers of these schemas may not be synchronized. See developer documentation for more information about the API and data store.

## XML Ingestion Sequence

To initiate an ingestion job, a user creates a job control file according to the format described in a previous section. Then, the user combines that file with the XML files as a single zip archive and uses an FTP client to upload that archive to the ingestion landing zone. To perform the upload, the user must authenticate with the SFTP server, which determines whether the use is authorized to ingest data to the tenancy associated with that landing zone.

Once an ingestion job is successfully uploaded, the job is picked up by the ingestion pipeline, which can be visualized as in the following diagram:



During the validation phase, the system checks the consistency of the job control file and validates the XML files against the interchange schemas defined within the inBloom extension of Ed-Fi schemas.

The processing phase is responsible for generating a change set, or “diff”, which can then be persisted to the inBloom Data Store. References to entities in the same job or to previously ingested entities are also resolved in this phase.

The matching algorithm relies on a combination of each entity’s fields to distinguish between insert and update operations in the inBloom Data Store. For entities with unique tenant-wide identifiers (*Student*, *Teacher*, *Staff*, and *Parent*), the *stateUniqueId* is used to uniquely match each ingested entity with the one already in the data store. If a match is found, the operation is an update. Otherwise, it is treated as an insert.

Likewise, newly added association entities are checked against existing association entities referring to the same objects in the data store. If the references match up, the new association is considered an update. Otherwise, it’s an insert.

Since different interchanges allow ingestion of different domain types, they must occur in a certain order, for the matching and reference resolution part of the processing phase to succeed. See the section on the ingestion interchange order and the section that describes additional interchange-level entity dependencies and constraints which must be satisfied for error-free operation. References at the end of this specification include a mapping between domain types (entities) and the interchange schemas used to ingest them.

Once the validation and processing phases are complete, the job passes to the persistence phase, where a generated change set (“diff”) is written to the inBloom Data Store. If this operation is successful, the ingested entities become accessible through the inBloom REST API and an ingestion log file is generated confirming the number of entities persisted to the data store. Likewise, a log file is generated whenever a job is aborted in any of the phases.

Every phase of the ingestion pipeline produces a set of errors and warnings, which are then written to the job log file available from the landing zone. See an upcoming section for the list of errors and warnings you might encounter.

# 

# Error and Status Logs

When inBloom processes an ingestion job, it creates several log files and places those files in the landing zone, accessible from the same SFTP server used to upload the job. These files contain the warnings and errors that occur both at job level (for each control file) and at resource level (for each XML file). The table below summarizes the types of log files, when they are created, and how they are formatted.

|  |  |  |
| --- | --- | --- |
| File Naming Convention | When Created | Content Format |
| job-<jobId>.log | Once for every job | INFO <jobId information> INFO <per resource persisted record count> INFO <configuration parameters> INFO <overall success or failure> INFO <total records processed> |
| job\_warn-<jobId>.log | Job-level (non-resource specific) warnings present | WARN <warning detail> |
| job\_error-<jobId>.log | Job-level (non-resource specific) errors present | ERROR <error detail> |
| warn.<resourceId>-<jobId>.log | Resource-level warnings present | WARN <warning detail> |
| error.<resourceId>-<jobId>.log | Resource-level errors present | ERROR <error detail> |

The job ID (“jobId”) is a unique identifier assigned to each new job by the ingestion pipeline, and it contains the name of the ingested file and an epoch time string. The resourceId refers to the name of a specific file within with a job. Each new log file entry begins with INFO, WARN or ERROR, which improves readability and simplifies multi-line error message parsing.

Deltas are records that are not processed because the data already exists unchanged in the database. The job-<jobId>.log file will log deltas in the following format:

INFO <per resource number of deltas detected>

## Job-level Error and Status Reporting

The job status in job-<jobId>.log includes a series of “INFO” lines, which provide a file-by-file account of how many records were found, ingested, or failed to ingest.

The job errors you might find in job\_error-<jobId>.log are defined in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| Message Code | Message | Scenario | Level |
| BASE\_0002 | No valid files specified in control file. | All the entries specified in the control file are invalid | Error |
| BASE\_0003 | No files specified in control file. | The control file does not have any file entries | Error |
| BASE\_0004 | File ${file}: File name contains path. | Invalid XML file name specified; file name contains a path | Error |
| BASE\_0005 | File ${file}: unknown or empty file ${format | type}: ${value} specified. | Unknown file format or type specified | Error |
| BASE\_0006 | File ${file}: Checksum validation failed. Possible file corruption. | Checksum comparison fails for the specified file | Error |
| BASE\_0007 | File ${file}: No checksum is specified. | Referenced file is missing checksum | Error |
| BASE\_0008 | Could not read .zip archive ${file}. Possible file damage or corruption. Please resubmit. | Either the file was not found or there was an error reading the file | Error |
| BASE\_0009 | No manifest file found in .zip archive ${file}. Please resubmit. | Missing control file in zip archive | Error |
| BASE\_0010 | .zip archive ${file} contains a directory. | Input zip file contains folders | Error |
| BASE\_0011 | Could not read .zip archive ${file}. Unsupported compression method. Please compress with Deflate method. | The input zip file has been compressed with an unsupported method | Error |
| BASE\_0012 | Control file or its parent directory is missing. | Either the control file or its parent directory was not found | Error |
| BASE\_0016 | Invalid control file entry at line number [${number}] Line: ${line content} | The control file entry is invalid possibly due to spaces or incorrect number of entries | Error |
| CORE\_0001 | The tenant is currently being onboarded. Please try ingestion in a few minutes when it has completed. | Indicates that tenant on-boarding is currently in progress and no new ingestion job can be accepted for some time | Error |
| CORE\_0002 | Control file is invalid, the associated files will have to be cleared out manually from the LZ | Warning message generated when the control file is invalid and the xml files can not be cleared out from the landing zone | Warning |
| CORE\_0003 | Failed to parse ctl file. ${message} | There was error processing the control file; the specific failure is provided in the message | Error |
| CORE\_0014 | Failed to process zip file. ${message} | Zip file was not processed successfully as specified by the message | Error |
| CORE\_0021 | Error processing batch job ${batchJobId} due to: ${exception message} | Error processing batch job | Error |
| CORE\_0027 | Error processing ingestion job, data transformation failed :${message} | Failed to process batch job as specified in the message | Error |
| CORE\_0035 | TenantId missing. No purge operation performed. | Purge operation failed due to missing tenant id information | Error |
| CORE\_0036 | Error processing purge job. ${message} | Purge operation failed as specified in the message | Error |

**Example:** Consider an ingestion job submitted as a single zip file named grade12Math.zip with two data files, Grade\_12\_Math\_CCS\_G\_C.xml and Grade\_12\_Math\_CCS\_G\_SRT.xml. The job-level log file for this example is named job-grade12Math.zip-1335552611336-f2ff0a92-c5ba-4aa1-8f28-e52618d35bae.log and contains the following “INFO” lines:

INFO  jobId: grade12Math.zip-1335552611336-f2ff0a92-c5ba-4aa1-8f28-e52618d35bae  
INFO  [file] Grade\_12\_Math\_CCS\_G\_C.xml (neutralrecord/AssessmentMetadata)  
INFO  [file] Grade\_12\_Math\_CCS\_G\_C.xml records considered: 19  
INFO  [file] Grade\_12\_Math\_CCS\_G\_C.xml records ingested successfully: 19  
INFO  [file] Grade\_12\_Math\_CCS\_G\_C.xml records failed: 0  
INFO  [file] Grade\_12\_Math\_CCS\_G\_SRT.xml (neutralrecord/AssessmentMetadata)  
INFO  [file] Grade\_12\_Math\_CCS\_G\_SRT.xml records considered: 0  
INFO  [file] Grade\_12\_Math\_CCS\_G\_SRT.xml records ingested successfully: 0  
INFO  [file] Grade\_12\_Math\_CCS\_G\_SRT.xml records failed: 0  
INFO  [configProperty] dry-run  
INFO  All records processed successfully.  
INFO  Processed 19 records.

If the job produces any warnings or errors, they will be placed in the job-level files job\_warn-grade12Math.zip-1335552611336-f2ff0a92-c5ba-4aa1-8f28-e52618d35bae.log and job\_error-grade12Math.zip-1335552611336-f2ff0a92-c5ba-4aa1-8f28-e52618d35bae.log, respectively.

## Resource- and Record-level Error Reporting

The tables below define the error messages you might find in each error.<resourceId>-<jobId>.log file associated with an ingestion job.

### General Resource-level Errors

|  |  |  |  |
| --- | --- | --- | --- |
| Message Code | Message | Scenario | Level |
| BASE\_0001 | File ${file}: Specified file is missing. | File specified in control file does not exist in the landing zone | Error |
| BASE\_0013 | File ${file}: Not found. | XML file was not found in the landing zone | Error |
| BASE\_0014 | File ${file}: Problem reading file. | There was an error while reading the XML file from the landing zone | Error |
| BASE\_0015 | File ${file}: Empty file. | XML file has no content | Error |
| CORE\_0013 | File type not supported :${file-type} | Ingestion does not support the specified file type | Error |

### XML and Schema Validation Errors and Warnings

inBloom embeds the Xerces XML parser to validate incoming files against the inBloom-Ed-Fi schema. Errors and warnings produced by the parser can be found in the Xerces parser documentation (<http://xerces.apache.org/>). The output of the XML validation phase is written to resource-specific log files.

### Record-level Persistence Errors

These are the errors associated with persisting (permanently saving) the ingested data to the database. The warnings associated with persistence are covered in a later table.

|  |  |  |  |
| --- | --- | --- | --- |
| Message Code | Message | Scenario | Level |
| CORE\_0004 | Failed to process [${0}] due to preceding errors | The current neutral record was not processed due to errors reported earlier. | Error |
| CORE\_0005 | Fatal problem saving records to database: \n \tEntity\t ${entity} \n | Errors occurred while processing the neutral records. | Error |
| CORE\_0006 | There has been a data validation error when saving an entity \n\tError\t ${error}\n\t Entity\t{entity}\n\tInstance\t{Instance}\n\tField\t{Field}\n\tValue\t{value}\n\tExpected\t{expected types}\n | The entity has invalid data or schema. The actual reason for failure is specified by the message. A detailed description of each of these messages is provided in the *Record-level Validation Error* table in this specification. | Error |
| CORE\_0008 | Issue finding key field: ${key-field} for entity of type: ${entity type} | There was an error finding a key filed for the specified entity. | Error |
| CORE\_0009 | Failed to resolve a deterministic id\n\tEntity ${entity-type} : Reference to ${ref-type} is incomplete because the following reference field is not resolved: ${ref-path} | A deterministic ID for the entity as reference path was not found. | Error |
| CORE\_0010 | An entity is missing one or more required natural key fields \n    Entity ${entity-type}\n    Instance   ${record-number}${field-names} | The entity with the given record number is invalid due to missing the specified natural key fields. | Error |
| CORE\_0011 | Cannot find a match for an entity: No key fields specified | No key fields have been specified for this entity | Error |
| CORE\_0012 | Invalid key fields for an entity\n ${message} | Key fields are invalid for the entity. | Error |
| CORE\_0015 | Unrecoverable error encountered while extracting ${collection-name} records from staging database | Mongo (the database software) threw an exception while trying to find all records in the specified collection in the ingestion subsystem’s staging database before persisting to the data store. | Error |
| CORE\_0016 | Could not instantiate smooks, unable to read configuration file. | An error occurred in reading the configuration file. | Error |
| CORE\_0017 | Could not instantiate smooks, problem parsing configuration file. | An error occurred in parsing the configuration file. | Error |
| CORE\_0018 | Index verified: ${Collection} ${keys} , unique ${uniqueness} | The index is verified. | Info |
| CORE\_0019 | Smooks validation failure at element ${element-name} | The element specified is invalid. | Error |
| CORE\_0020 | SmooksException encountered while filtering input: ${file} | Error parsing input XML file. | Error |
| CORE\_0030 | LearningObjective cannot have multiple parents: ${parent} | More than one parent has been specified for this LearningObjective. | Error |
| CORE\_0031 | Could not resolve child learning standard references for learning objective ${attributes} | Learning standard reference is missing. | Error |
| CORE\_0032 | Cannot find AssessmentItem referenced by StudentAssessmentItem.  AssessmentItemIdentificationCode: ${code} | Missing expected *AssessmentItem*. | Error |
| CORE\_0033 | StudentAsessmentItem does not contain an AssessmentItemIdentificationCode referencing an AssessmentItem | Missing *AssessmentItemIdentificationCode* information. | Error |
| CORE\_0034 | Could not resolve LearningObjectiveReference with Objective: ${objective}, AcademicSubject${academicSubject}, ObjectiveGradeLevel${objectiveGradeLevel} | Expected *LearningObjectiveReference* missing. | Error |
| CORE\_0037 | Two or more LearningObjectives have duplicate IdentificationCode, ContentStandardName combination. Objective: ${objective}, AcademicSubject:${academicSubject},ObjectiveGradeLevel:${objectiveGradeLevel} | Multiple LearningObjectives have the same IdentificationCode value. | Error |
| CORE\_0038 | Index missed: ${Collection} ${keys} , unique ${uniqueness} | The index validator indicates an index is missing. | Error |
| CORE\_0039 | Illegal value ${Date} for administration date, must be a string. | An error in administration date. | Error |
| CORE\_0040 | Unable to get key fields for StudentAssessment transform: ${keys} | An error in getting key fields while processing *StudentAssessment*. | Error |
| CORE\_0042 | Exception occurred while retreiving student id: ${ExceptionMessage} | An error exception happens while processing *StudentAssessment*. | Error |
| CORE\_0046 | Exception in matchEntity: ${ExceptionMessage}. | Failed to find a matching entity in the data store. | Error |
| CORE\_0047 | EdFi2SLI Transform has resulted in an empty list of transformed SimpleEntities. | No entity can be transformed. | Error |
| CORE\_0048 | Failed to extract StudentUniqueStateId from attendance entity: ${ExceptionMessage}. | Failed to extract *StudentUniqueStateId*. | Error |
| UNKNOWN\_FIELD | Error      UNKNOWN\_FIELD       Entity     ${recordType}       Instance   ${recordNumber}       Field      ${fieldname}       Value      ${fieldValue}       Expected  ${expectedFieldValue} | Unknown parameter specified. | Error |
| REQUIRED\_FIELD\_MISSING | There has been a data validation error when saving an entity       Error      REQUIRED\_FIELD\_MISSING       Entity     ${recordType}       Instance   ${recordNumber}       Field      ${fieldname}       Value      ${fieldValue}       Expected  ${expectedFieldValue} | Missing required parameter. | Error |
| INVALID\_DATE\_FORMAT | There has been a data validation error when saving an entity       Error      INVALID\_DATE\_FORMAT       Entity     ${recordType}       Instance   ${recordNumber}       Field      ${fieldname}       Value      ${fieldValue}       Expected  ${expectedFieldValue} | Parameter with invalid date format. | Error |
| INVALID\_DATATYPE | There has been a data validation error when saving an entity       Error      INVALID\_DATATYPE       Entity     ${recordType}       Instance   ${recordNumber}       Invalid data type for field     ${fieldname}       Value      [${fieldValue}]       Expected  [${fieldValue}] | Parameter with invalid data type. | Error |
| ENUMERATION\_MISMATCH | There has been a data validation error when saving an entity       Error      ENUMERATION\_MISMATCH       Entity     ${recordType}       Instance   ${recordNumber}       Field      ${fieldname}       Value      ${fieldValue}       Expected  ${expectedFieldValue} | Parameter with mismatch enumeration. | Error |
| REFERENTIAL\_INFO\_MISSING | There has been a data validation error when saving an entity       Error      REFERENTIAL\_INFO\_MISSING       Entity     ${recordType}       Instance   ${recordNumber}       Field      ${fieldname}       Value      ${fieldValue}       Expected  ${expectedFieldValue} | Parameter missing referential information. | Error |
| INVALID\_VALUE | There has been a data validation error when saving an entity       Error      INVALID\_VALUE       Entity     ${recordType}       Instance   ${recordNumber}       Field      ${fieldname}       Value      ${fieldValue}       Expected  ${expectedFieldValue} | Bad parameter value or parameter value where no parameter value is expected. | Error |
| INVALID\_CHOICE\_TYPE | There has been a data validation error when saving an entity       Error      INVALID\_CHOICE\_TYPE       Entity     ${recordType}       Instance   ${recordNumber}       Field      ${fieldname}       Value      ${fieldValue}       Expected  ${expectedFieldType} | Selected choice type not a valid option for this field. | Error |
| SELF\_REFERENCING\_DATA | There has been a data validation error when saving an entity       Error      INVALID\_CHOICE\_TYPE       Entity     ${recordType}       Instance   ${recordNumber}       Field      ${fieldname}       Value      ${fieldValue}       Expected  ${expectedFieldType} | Entity contains a reference field that refers to itself. | Error |

### Record-level Persistence Warnings

These are the warnings associated with persisting (permanently saving) the ingested data to the database. The errors associated with persistence are covered in a previous table.

|  |  |  |  |
| --- | --- | --- | --- |
| Message Code | Message | Scenario | Level |
| CORE\_0007 | Entity (${entity type}) reports warning:${message} | Warning | Used |
| CORE\_0028 | [${Number}] attendance events are not processed, because they are not within any school year | Specified number of attendance events were not processed due to missing school year information. | Warning |
| CORE\_0029 | No session found to handle attendance for student: ${studentId} in school: ${schoolId} | Unable to process attendance event for the specified student and school due to missing session information. | Warning |
| CORE\_0041 | Unable to get StudentID within ${StudentAssessmentAttributes} for StudentAssessment transform | Missing *StudentID* within the *StudentAssessment*. | Warning |
| CORE\_0043 | Could not find objective assessment ref: ${ObjectiveAssessmentReference} | Cannot find *ObjectiveAssessmentReference* value. | Warning |
| CORE\_0044 | Ignoring sub objective assessment ${ObjectiveAssessmentReference} since it is already in the hierarchy. | Misplaced *ObjectiveAssessmentReference* value. | Warning |
| CORE\_0045 | Failed to match objective assessment: ${ObjectiveAssessmentReference} for family: ${FamilyHierarchyName}. | Failed to match *ObjectiveAssessmentReference*. | Warning |
| CORE\_0049 | Student with id: ${StudentID} is not associated to any schools. | No school found for that student. | Warning |
| CORE\_0050 | Student with id: ${StudentID} is associated to more than one school. | More than one schools found for that student. | Warning |

# 

# Ingestion Data Dependencies and Constraints

A single ingestion job can contain data from a single interchange or spanning multiple data interchanges. An entity in one job may refer to entities ingested by another job or by another data interchange submitted as part of the same job. Support for XML ID referencing has been removed from ingestion (see caution below).

In all of these cases, referential integrity of data must be preserved. This means that a mandatory field of an entity cannot refer to a non-existing entity. To meet this requirement, ordering constraints are imposed on the overall ingestion process.

Occasionally it may be necessary to split data from a single interchange into two subsets and to ingest them separately to satisfy multiple dependencies.

A *Section* entity, which is part of the *MasterSchedule* interchange, has to refer to a *CourseOffering*, which is also part of the same interchange. We refer to this as an entity-specific interchange-level constraint.

**Caution:** inBloom ingestion no longer supports XML ID referencing.  Any ID/IDREF pairs within an XML ingestion data file will not be resolved. Instead they are silently ignored except for any “ref=” notation which violates the XML schema for a particular entity. Such violations are reported to the user as a warning.

## Interchange Ordering

The following interchange dependencies still exist when ingesting data:

* For related *Assessment* and *StudentAssessment* entities, associated interchanges (*AssessmentMetadata* and  *StudentAssessments*) must be submitted together.
* For *Attendance*-related entities in *EducationOrganization*, *EducationOrgCalendar* and *StudentEnrollment*, all interchanges must be submitted before or with *StudentAttendance*.

## 

## Interchange-level Constraints

All constraints described here apply only to those entities supported by inBloom, which is a subset of those supported by Ed-Fi. You must be sure your data complies with all the “Constraint” items described in this section.

### EducationOrganization

The *EducationOrganization* inBloom-Ed-Fi interchange supports ingestion of the following entities:

* StateEducationAgency
* LocalEducationAgency
* School
* Course
* CompetencyLevelDescriptor
* Program

**Constraints:** An *EducationOrganization* entity provides the authorization context for inBloom users. Thus, it is critical that every inBloom user is tied to a particular tier within the *EducationOrganization* hierarchy. During on-boarding for an education organization, the landing zones adminsitrators set up to correspond to LEAs within the hierarchy, identified by the *LocalEducationAgency* entity. Super Administrators performing ingestion for an LEA must be associated (using a unique staff identifier) to an ingested *LocalEducationAgency* entity. For more information about the implementation of identity services with inBloom, see the inBloom administrator documentation.

### AssessmentMetadata

The *AssessmentMetadata* inBloom-Ed-Fi interchange supports ingestion of the following entities:

* AssessmentFamily
* Assessment
* AssessmentPeriodDescriptor
* ObjectiveAssessment
* LearningObjective
* LearningStandard

**Constraints:**

|  |  |  |
| --- | --- | --- |
| Entity | Referenced Entity | Interchange of Origin |
| AssessmentFamily | AssessmentFamily | AssessmentMetadata |
| Assessment | AssessmentItem, ObjectiveAssessment, AssessmentFamily, Section | AssessmentMetadata, MasterSchedule |
| AssessmentPeriodDescriptor |  |  |
| ObjectiveAssessment | AssessmentItem, LearningObjective, LearningStandard, ObjectiveAssessment | AssessmentMetadata |
| AssessmentItem | LearningStandard | AssessmentMetadata |
| LearningObjective | LearningStandard, LearningObjective |  |
| LearningStandard |  |  |

References to entities within the same interchange must resolve to entities ingested as part of the same or a previous job.

### StaffAssociation

The StaffAssociation inBloom-Ed-Fi interchange supports ingestion of the following entities:

* Staff
* StaffEducationOrgAssignmentAssociation
* Teacher
* TeacherSchoolAssociation
* TeacherSectionAssociation
* StaffProgramAssociation

**Constraints:** When teachers and staff access inBloom data and applications, inBloom must establish correspondence between the users who are signed in (and their roles) and the *Staff* and *Teacher* entities in the data store. Each *Staff* and *Teacher* entity with inBloom access permissions must be associated with identity credentials served by the Identity Provider. Specifically, the unique ID (*StaffUniqueStateId* or *TeacherUniqueStateId*) and *LoginId* must match

the login credentials used to authenticate against the IDP. In addition, the *StaffEducationOrgAssignmentAssociation* must have an *endDate* that has not expired yet.

### StudentParent

The *StudentParent* Ed-Fi interchange supports ingestion of the following entities:

* Student
* Parent
* StudentParentAssociation

**Constraints:** When students and parents access inBloom data and applications, inBloom must establish correspondence between the users who are signed in (and their roles) and *Student* and *Parent* entities in the data store. Each *Student* and *Parent* entity with inBloom access permissions must be associated with identity credentials served by the Identity Provider. Specifically, the unique ID (*StudentUniqueStateId* or *ParentUniqueStateId*) and *LoginId* must match the login credentials used to authenticate against the IDP.

### StudentAssessment

The *StudentAssessment* Ed-Fi interchange supports ingestion of the following entities:

* StudentAssessment
* StudentObjectiveAssessment
* StudentAssessmentItem

**Constraints:**

|  |  |  |
| --- | --- | --- |
| Entity | Referenced Entity | Interchange of Origin |
| StudentAssessment | Student, Assessment | StudentParent, AssessmentMetadata |
| StudentObjectiveAssessment | StudentAssessment, ObjectiveAssessment | StudentAssessment, AssessmentMetadata |
| StudentAssessmentItem | StudentAssessment, StudentObjectiveAssessment, AssessmentItem | StudentAssessment, AssessmentMetadata |

References to entities within the same interchange must resolve to entities ingested as part of the same or a previous job.

### EducationOrgCalendar

The *EducationOrgCalendar* Ed-Fi interchange supports ingestion of the following entities:

* Session
* GradingPeriod

### StudentEnrollment

The *StudentEnrollment* Ed-Fi interchange supports ingestion of the following entities:

* StudentSchoolAssociation
* StudentSectionAssociation

### StudentGrade

The *StudentGrade* Ed-Fi interchange supports ingestion of the following entities:

* StudentAcademicRecord
* CourseTranscript
* GradebookEntry
* StudentGradebookEntry
* ReportCard
* Grade
* StudentCompetency
* CompetencyLevelDescriptor
* LearningObjective
* StudentCompetencyObjective

### StudentProgram

The *StudentProgram* Ed-Fi interchange supports ingestion of the following entities:

* StudentProgramAssociation

### StudentCohort

The *StudentCohort* Ed-Fi interchange supports ingestion of the following entities:

* Cohort
* StudentCohortAssociation
* StaffCohortAssociation

### StudentDiscipline

The *StudentDiscipline* Ed-Fi interchange supports ingestion of the following entities:

* DisciplineIncident
* StudentDisciplineIncidentAssociation
* DisciplineAction

### StudentAttendance

The *StudentAttendance* Ed-Fi interchange supports ingestion of the following entities:

* AttendanceEvent

### TeacherSchoolAssociation

**Constraint:** For every *TeacherSchoolAssociation*, there must be a *StaffEducationOrgAssignmentAssociation* with an *endDate* that has not expired yet.

# 

# inBloom-Ed-Fi Entity to inBloom Entity Mapping

|  |  |  |
| --- | --- | --- |
| inBloom-Ed-Fi Interchange | Ed-Fi Entity | inBloom Entity |
| Interchange-AssessmentMetadata | LearningStandard | learningStandard |
| Interchange-AssessmentMetadata | PerformanceLevelDescriptor | studentAssessmentAssociation/performanceLevelDescriptors and studentObjectiveAssessment/performanceLevelDescriptors and Assessment/AssessmentPerformanceLevel/PerformanceLevelDescriptors |
| Interchange-AssessmentMetadata | AssessmentItem | assessment/assessmentItem and assessment/objectiveAssessment/assessmentItem |
| Interchange-AssessmentMetadata | Assessment | assessment |
| Interchange-AssessmentMetadata | ObjectiveAssessment | assessment/objectiveAssessment |
| Interchange-AssessmentMetadata | AssessmentPeriodDescriptor | assessment/assessmentPeriodDescriptor |
| Interchange-AssessmentMetadata | AssessmentFamily | assessment/assessmentFamilyHierarchyName |
| Interchange-AssessmentMetadata Interchange-StudentGrade | LearningObjective | learningObjective |
| Interchange-EducationOrganization | Course | course |
| Interchange-EducationOrganization | LocalEducationAgency | educationOrganization |
| Interchange-EducationOrganization | Program | program |
| Interchange-EducationOrganization | School | school |
| Interchange-EducationOrganization | StateEducationAgency | educationOrganization |
| Interchange-EducationOrganization Interchange-StudentGrade | CompetencyLevelDescriptor | --- |
| Interchange-EducationOrgCalendar | CalendarDate | calendarDate |
| Interchange-EducationOrgCalendar | GradingPeriod | gradingPeriod |
| Interchange-EducationOrgCalendar | Session | session/schoolSessionAssociation |
| Interchange-MasterSchedule | CourseOffering | courseOffering |
| Interchange-MasterSchedule | Section | section |
| Interchange-StaffAssociation | CredentialFieldDescriptor | staff|teacher/credentials/credentialFieldDescriptor |
| Interchange-StaffAssociation | StaffCohortAssociation | staffCohortAssociation |
| Interchange-StaffAssociation | Staff | staff/abstractStaff |
| Interchange-StaffAssociation | Teacher | teacher |
| Interchange-StaffAssociation | StaffProgramAssociation | staffProgramAssociation |
| Interchange-StaffAssociation | TeacherSchoolAssociation | teacherSchoolAssociation |
| Interchange-StaffAssociation | TeacherSectionAssociation | teacherSectionAssociation |
| Interchange-StaffAssociation | StaffEducationOrgAssignmentAssociation | staffEducationOrganizationAssociation |
| Interchange-StudentAssessment | StudentAssessmentItem | studentAssessmentAssociation/studentAssessmentItem and studentAssessmentAssociation/studentObjectiveAssessment/studentAssessmentItem \* |
| Interchange-StudentAssessment | StudentAssessment | studentAssessmentAssociation |
| Interchange-StudentAssessment | StudentObjectiveAssessment | studentAssessmentAssociation/studentObjectiveAssessment |
| Interchange-StudentAttendance | AttendanceEvent | dailyAttendance |
| Interchange-StudentCohort | Cohort | cohort |
| Interchange-StudentCohort | StudentCohortAssociation | studentCohortAssociation |
| Interchange-StudentDiscipline | DisciplineAction | disciplineAction |
| Interchange-StudentDiscipline | DisciplineIncident | disciplineIncident |
| Interchange-StudentDiscipline | StudentDisciplineIncidentAssociation | studentDisciplineIncidentAssociation |
| Interchange-StudentEnrollment | GraduationPlan | graduationPlan |
| Interchange-StudentEnrollment | StudentSectionAssociation | studentSectionAssociation |
| Interchange-StudentEnrollment | StudentSchoolAssociation | studentSchoolAssociation |
| Interchange-StudentGrade | Grade | grade |
| Interchange-StudentGrade | ReportCard | reportCard |
| Interchange-StudentGrade | StudentCompetency | studentCompetency |
| Interchange-StudentGrade | StudentCompetencyObjective | studentCompetencyObjective |
| Interchange-StudentGrade | CourseTranscript | courseTranscript |
| Interchange-StudentGrade | GradebookEntry | gradebookEntry |
| Interchange-StudentGrade | StudentAcademicRecord | studentAcademicRecord |
| Interchange-StudentGrade | StudentGradebookEntry | studentSectionGradebookEntry |
| Interchange-StudentParent | Parent | parent |
| Interchange-StudentParent | Student | student |
| Interchange-StudentParent | StudentParentAssociation | studentParentAssociation |
| Interchange-StudentProgram | StudentProgramAssociation | studentProgramAssociation |

# 

# inBloom-Ed-Fi Entity Type to Interchange Schema Mapping

The table below provides a mapping between a domain type (rows) and an interchange schema (columns), which supports ingestion of entities of that type.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | AcademicMetadata | EducationOrganization | StaffAssociation | StudentParent | StudentAssessment | EducationOrgCalendar | StudentEnrollment | StudentGrade | StudentCohort | StudentDiscipline | StudentAttendance | StudentProgram | MasterSchedule |
| Assessment | x |  |  |  | x |  |  |  |  |  |  |  |  |
| AssessmentFamily | x |  |  |  |  |  |  |  |  |  |  |  |  |
| AssessmentItem | x |  |  |  |  |  |  |  |  |  |  |  |  |
| AssessmentPeriodDescriptor | x |  |  |  |  |  |  |  |  |  |  |  |  |
| LearningObjective | x |  |  |  |  |  |  | x |  |  |  |  |  |
| LearningStandard | x |  |  |  |  |  |  |  |  |  |  |  |  |
| ObjectiveAssessment | x |  |  |  |  |  |  |  |  |  |  |  |  |
| PerformanceLevelDescriptor | x |  |  |  |  |  |  |  |  |  |  |  |  |
| Course |  | x |  |  |  |  |  |  |  |  |  |  |  |
| LocalEducationAgency |  | x |  |  |  |  |  |  |  |  |  |  |  |
| Program |  | x |  |  |  |  |  |  |  |  |  |  |  |
| School |  | x |  |  |  |  |  |  |  |  |  |  |  |
| StateEducationAgency |  | x |  |  |  |  |  |  |  |  |  |  |  |
| CompetencyLevelDescriptor |  | x |  |  |  |  |  | x |  |  |  |  |  |
| Staff |  |  | x |  |  |  |  |  |  |  |  |  |  |
| StaffEducationOrgEmploymentAssociation |  |  | x |  |  |  |  |  |  |  |  |  |  |
| StaffEducationOrgAssignmentAssociation |  |  | x |  |  |  |  |  |  |  |  |  |  |
| Teacher |  |  | x |  |  |  |  |  |  |  |  |  |  |
| TeacherSchoolAssociation |  |  | x |  |  |  |  |  |  |  |  |  |  |
| TeacherSectionAssociation |  |  | x |  |  |  |  |  |  |  |  |  |  |
| LeaveEvent |  |  | x |  |  |  |  |  |  |  |  |  |  |
| OpenStaffPosition |  |  | x |  |  |  |  |  |  |  |  |  |  |
| CredentialFieldDescriptor |  |  | x |  |  |  |  |  |  |  |  |  |  |
| Student |  |  |  | x |  |  |  |  |  |  |  |  |  |
| Parent |  |  |  | x |  |  |  |  |  |  |  |  |  |
| StudentParentAssociation |  |  |  | x |  |  |  |  |  |  |  |  |  |
| StudentReference |  |  |  |  | x |  |  |  |  |  |  |  |  |
| AssessmentReference |  |  |  |  | x |  |  |  |  |  |  |  |  |
| StudentAssessment |  |  |  |  | x |  |  |  |  |  |  |  |  |
| StudentObjectiveAssessement |  |  |  |  | x |  |  |  |  |  |  |  |  |
| StudentAssessmentItem |  |  |  |  | x |  |  |  |  |  |  |  |  |
| Session |  |  |  |  |  | x |  |  |  |  |  |  |  |
| GradingPeriod |  |  |  |  |  | x |  |  |  |  |  |  |  |
| CalendarDate |  |  |  |  |  | x |  |  |  |  |  |  |  |
| StudentSchoolAssociation |  |  |  |  |  |  | x |  |  |  |  |  |  |
| StudentSectionAssociation |  |  |  |  |  |  | x |  |  |  |  |  |  |
| GraduationPlan |  |  |  |  |  |  | x |  |  |  |  |  |  |
| StudentAcademicRecord |  |  |  |  |  |  |  | x |  |  |  |  |  |
| CourseTranscript |  |  |  |  |  |  |  | x |  |  |  |  |  |
| ReportCard |  |  |  |  |  |  |  | x |  |  |  |  |  |
| Grade |  |  |  |  |  |  |  | x |  |  |  |  |  |
| StudentCompetency |  |  |  |  |  |  |  | x |  |  |  |  |  |
| GradebookEntry |  |  |  |  |  |  |  | x |  |  |  |  |  |
| StudentGradebookEntry |  |  |  |  |  |  |  | x |  |  |  |  |  |
| StudentCompetencyObjective |  |  |  |  |  |  |  | x |  |  |  |  |  |
| Cohort |  |  |  |  |  |  |  |  | x |  |  |  |  |
| StudentCohortAssociation |  |  |  |  |  |  |  |  | x |  |  |  |  |
| StaffCohortAssociation |  |  |  |  |  |  |  |  | x |  |  |  |  |
| DisciplineIncident |  |  |  |  |  |  |  |  |  | x |  |  |  |
| StudentDisciplineIncidentAssociation |  |  |  |  |  |  |  |  |  | x |  |  |  |
| DisciplineAction |  |  |  |  |  |  |  |  |  | x |  |  |  |
| AttendanceEvent |  |  |  |  |  |  |  |  |  |  | x |  |  |
| StudentProgramAssociation |  |  |  |  |  |  |  |  |  |  |  | x |  |
| CourseOffering |  |  |  |  |  |  |  |  |  |  |  |  | x |
| Section |  |  |  |  |  |  |  |  |  |  |  |  | x |

# inBloom-Ed-Fi Natural Keys

|  |  |  |
| --- | --- | --- |
| inBloom Entity | inBloom-Ed-FI natural keys | Detailed Key Fields |
| Assessment | AssessmentTitle | AssessmentTitle |
|  | AcademicSubject | AcademicSubject |
|  | GradeLevelAssessed | GradeLevelAssessed |
|  | Version | Version |
| AssessmentFamily | AssessmentFamilyTitle | AssessmentFamilyTitle |
| AssessmentItem | IdentificationCode | IdentificationCode |
|  |  |  |
| AttendanceEvent | StudentReference | StudentUniqueStateId |
|  | SchoolReference | StateOrganizationId |
|  | EventDate | EventDate |
| CalendarDate | Date | Date |
|  | EducationOrgReference | StateOrganizationId |
| Cohort | EducationOrgReference | StateOrganizationId |
|  | CohortIdentifier | CohortIdentifier |
| CompetencyLevelDescriptor | CodeValue | CodeValue |
| Course | EducationOrganizationReference | StateOrganizationId |
|  | UniqueCourseId | UniqueCourseId |
| CourseOffering | SchoolReference | StateOrganizationId |
|  | SessionReference | StateOrganizationId |
|  |  | SessionName |
|  | CourseReference | StateOrganizationId |
|  |  | UniqueCourseId |
| CourseTranscript | CourseAttemptResult | CourseAttemptResult |
|  | CourseReference | StateOrganizationId |
|  |  | UniqueCourseId |
|  | StudentAcademicRecordReference | StudentUniqueStateId |
|  |  | StateOrganizationId |
|  |  | SessionName |
| DisciplineAction | DisciplineActionIdentifier | DisciplineActionIdentifier |
|  | ResponsibilitySchoolReference | StateOrganizationId |
| DisciplineIncident | SchoolReference | StateOrganizationId |
|  | IncidentIdentifier | IncidentIdentifier |
| EducationOrganization | StateOrganizationId | StateOrganizationId |
| Grade | StudentSectionAssociationReference | StudentUniqueStateId |
|  |  | StateOrganizationId |
|  |  | UniqueSectionCode |
|  |  | BeginDate |
|  | GradingPeriodReference | StateOrganizationId |
|  |  | GradingPeriod |
|  |  | BeginDate |
| GradebookEntry | GradebookEntryType | GradebookEntryType |
|  | DateAssigned | DateAssigned |
|  | SectionReference | StateOrganizationId |
|  |  | UniqueSectionCode |
| GradingPeriod | GradingPeriod | GradingPeriod |
|  | EducationOrganizationReference | StateOrganizationId |
|  | BeginDate | BeginDate |
| GraduationPlan | GraduationPlanType | GraduationPlanType |
|  | EducationOrganizationReference | StateOrganizationId |
| LearningObjective | Objective | Objective |
|  | AcademicSubject | AcademicSubject |
|  | ObjectiveGradeLevel | ObjectiveGradeLevel |
| LearningStandard | LearningStandardId | LearningStandardId |
| ObjectiveAssessment | IdentificationCode | IdentificationCode |
|  |  |  |
| Parent | ParentUniqueStateId | ParentUniqueStateId |
| Program | ProgramId | ProgramId |
|  |  |  |
| ReportCard | StudentReference | StudentUniqueStateId |
|  | GradingPeriodReference | StateOrganizationId |
|  |  | GradingPeriod |
|  |  | BeginDate |
| School | StateOrganizationId | StateOrganizationId |
| Section | UniqueSectionCode | UniqueSectionCode |
|  | SchoolReference | StateOrganizationId |
| Session | SessionName | SessionName |
|  | EducationOrganizationReference | StateOrganizationId |
|  |  |  |
| Staff | StaffUniqueStateId | StaffUniqueStateId |
| Student | StudentUniqueStateId | StudentUniqueStateId |
| StaffCohortAssociation | StaffReference | StaffUniqueStateId |
|  | CohortReference | CohortIdentifier |
|  |  | StateOrganizationId |
|  | BeginDate | BeginDate |
| StudentAcademicRecord | StudentReference | StudentUniqueStateId |
|  | SessionReference | SessionName |
|  |  | StateOrganizationId |
| StudentCompetency | LearningObjectiveReference | Objective |
|  |  | AcademicSubject |
|  |  | ObjectiveGradeLevel |
|  | StudentCompetencyObjectiveReference | StudentCompetencyObjectiveId |
|  | CompetencyLevel | CodeValue |
|  | StudentSectionAssociationReference | StudentUniqueStateId |
|  |  | StateOrganizationId |
|  |  | UniqueSectionCode |
|  |  | BeginDate |
| StudentCompetencyObjective | StudentCompetencyObjectiveId | StudentCompetencyObjectiveId |
|  |  |  |
| StudentGradebookEntry | StudentSectionAssociationReference | StudentUniqueStateId |
|  |  | StateOrganizationId |
|  |  | UniqueSectionCode |
|  |  | BeginDate |
|  | GradeBookEntryReference | GradebookEntryType |
|  |  | DateAssigned |
|  |  | StateOrganizationId |
|  |  | UniqueSectionCode |
| Teacher | StaffUniqueStateId |  |
|  |  | StaffUniqueStateId |
| StaffEducationOrgAssignmentAssociation | StaffReference | StaffUniqueStateId |
|  | EducationOrganizationReference | StateOrganizationId |
|  | StaffClassification | StaffClassification |
|  | BeginDate | BeginDate |
| StaffProgramAssociation | StaffReference | StaffUniqueStateId |
|  | ProgramReference | ProgramId |
|  | BeginDate | BeginDate |
| StudentAssessment | AdministrationDate | AdministrationDate |
|  | StudentReference | StudentUniqueStateId |
|  | AssessmentReference | AssessmentTitle |
|  |  | AcademicSubject |
|  |  | GradeLevelAssessed |
|  |  | Version |
| StudentAssessmentItem | StudentAssessmentReference | AdministrationDate |
|  |  | StudentUniqueStateId |
|  |  | AssessmentTitle |
|  |  | AcademicSubject |
|  |  | GradeLevelAssessed |
|  |  | Version |
|  | AssessmentItemReference | AssessmentItemIdentificationCode |
| StudentCohortAssociation | StudentReference | StudentUniqueStateId |
|  | CohortReference | StateOrganizationId |
|  |  | CohortIdentifier |
|  | BeginDate | BeginDate |
| StudentDisciplineIncidentAssociation | StudentReference | StudentUniqueStateId |
|  | DisciplineIncidentReference | StateOrganizationId |
|  |  | IncidentIdentifier |
| StudentObjectiveAssessment | StudentAssessmentReference | AdministrationDate |
|  |  | StudentUniqueStateId |
|  |  | AssessmentTitle |
|  |  | AcademicSubject |
|  |  | GradeLevelAssessed |
|  |  | Version |
|  | ObjectiveAssessmentReference | ObjectiveAssessmentIdentificationCode |
| StudentParentAssociation | StudentReference | StudentUniqueStateId |
|  | ParentReference | ParentUniqueStateId |
| StudentProgramAssociation | StudentReference | StudentUniqueStateId |
|  | ProgramReference | ProgramId |
|  | BeginDate | BeginDate |
|  | EducationOrganizationReference | StateOrganizationId |
| StudentSchoolAssociation | StudentReference | StudentUniqueStateId |
|  | SchoolReference | StateOrganizationId |
|  | EntryDate | EntryDate |
| StudentSectionAssociation | StudentReference | StudentUniqueStateId |
|  | SectionReference | StateOrganizationId |
|  |  | UniqueSectionCode |
|  | BeginDate | BeginDate |
| TeacherSchoolAssociation | TeacherReference | StaffUniqueStateId |
|  | SchoolReference | StateOrganizationId |
|  | ProgramAssignment | ProgramAssignment |
| TeacherSectionAssociation | TeacherReference | StaffUniqueStateId |
|  | SectionReference | StateOrganizationId |
|  |  | UniqueSectionCode |