

**Educational Secure Data Service**

**Data Ingestion Guide**

**July 1, 2014**

Copyright Notice

© 2014 Educational Secure Data Services  
All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission of ESDS.

**Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Revised by | Description |
| v.1.0 | 06/18/2014 | Laura Heintz | First Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 1](#_Toc390852094)

[1.1 Scope 1](#_Toc390852095)

[1.2 Audience 1](#_Toc390852096)

[1.3 Dependencies, Assumptions & Constraints 1](#_Toc390852097)

[2 ESDS Architecture 2](#_Toc390852098)

[3 ESDS Technology and Ed-Fi Unifying Data Model 3](#_Toc390852099)

[4 Ingestion Overview 3](#_Toc390852100)

[4.1 Tenancies 3](#_Toc390852101)

[4.2 Landing Zones 4](#_Toc390852102)

[4.3 Ingestion Sequence 4](#_Toc390852103)

[4.4 Ingestion Validation 5](#_Toc390852104)

[4.5 Ingestion Job ID and Resource ID 5](#_Toc390852105)

[4.6 Processing an Ingestion Job 6](#_Toc390852106)

[4.7 Ingestion Schemas 6](#_Toc390852107)

[4.7.1 ESDS Extensions to Ed-Fi 6](#_Toc390852108)

[4.7.2 Interchange Schemas 7](#_Toc390852109)

[4.8 Ingestion Security 7](#_Toc390852110)

[4.8.1 Authentication 7](#_Toc390852111)

[4.8.2 Authorization 8](#_Toc390852112)

[4.8.3 Encryption 8](#_Toc390852113)

[5 Preparing Data for Ingestion 8](#_Toc390852114)

[5.1 Preparing the XML Data Files 9](#_Toc390852115)

[5.2 Preparing the Control File 9](#_Toc390852116)

[5.3 Preparing the .zip Archive File 10](#_Toc390852117)

[6 Offline Validation Tool 10](#_Toc390852118)

[6.1 System Requirements 10](#_Toc390852119)

[6.2 Using the OVT 10](#_Toc390852120)

[6.3 Viewing OVT Results 12](#_Toc390852121)

[7 Deleting Data Through Ingestion 12](#_Toc390852122)

[7.1 Action Element Type 12](#_Toc390852123)

[7.2 Delete Actions 13](#_Toc390852124)

[7.3 Forced Deletes 14](#_Toc390852125)

[7.4 Safe Deletes 15](#_Toc390852126)

[7.5 Action Attributes for ActionType = DELETE 15](#_Toc390852127)

[8 Ingestion Logs and Error Handling 16](#_Toc390852128)

[8.1 Ingestion Logs 16](#_Toc390852129)

[8.2 Ingestion Deltas 16](#_Toc390852130)

[8.3 Parser Errors and Warnings 17](#_Toc390852131)

[8.3.1 XML elements are out of order 17](#_Toc390852132)

[8.3.2 XML element not defined in the XSD 17](#_Toc390852133)

[8.3.3 XML element of minimum size "1" cannot be empty 18](#_Toc390852134)

[8.4 Example Ingestion Log 18](#_Toc390852135)

[8.5 Ingestion Log Files 19](#_Toc390852136)

[8.6 Ingestion Errors and Warnings 20](#_Toc390852137)

[9 Ingestion File Formats 33](#_Toc390852138)

[9.1 Ingestion File Conventions 33](#_Toc390852139)

[9.2 Ingestion Control File Format 33](#_Toc390852140)

[9.2.1 Control File: Data File Lines 34](#_Toc390852141)

[9.2.2 Control File: Job Control Lines 35](#_Toc390852142)

[9.2.3 Control File: Example 36](#_Toc390852143)

[9.3 Ingestion Data File Format 36](#_Toc390852144)

[9.4 Ingestion Schemas and Interchanges 36](#_Toc390852145)

[9.5 Ingestion Constraints 37](#_Toc390852146)

[9.5.1 Key Fields 37](#_Toc390852147)

[9.5.2 Ordering Constraints 37](#_Toc390852148)

[9.5.3 EducationOrganization Interchange Constraints 39](#_Toc390852149)

[9.5.4 EducationOrgCalendar Interchange Constraints 39](#_Toc390852150)

[9.5.5 StaffAssociation Interchange Constraints 39](#_Toc390852151)

[9.5.6 StudentAssessment Interchange Constraints 40](#_Toc390852152)

[9.5.7 StudentAttendance Interchange Constraints 41](#_Toc390852153)

[9.5.8 StudentCohort Interchange Constraints 41](#_Toc390852154)

[9.5.9 StudentDiscipline Interchange Constraints 41](#_Toc390852155)

[9.5.10 StudentGrade Interchange Constraints 41](#_Toc390852156)

[9.5.11 StudentEnrollment Interchange Constraints 42](#_Toc390852157)

[9.5.12 StudentParent Interchange Constraints 42](#_Toc390852158)

[9.5.13 StudentProgram Interchange Constraints 42](#_Toc390852159)

[9.5.14 TeacherSchoolAssociation Entity Constraints 43](#_Toc390852160)

[Appendix A: Entity-to-Interchange Schema Mapping 44](#_Toc390852161)

[Appendix B: Interchange Entity Mapping 48](#_Toc390852162)

[Appendix C: Ingestion Natural Keys 51](#_Toc390852163)

# Introduction

The Educational Secure Data Services (ESDS) ingestion system provides a framework that enables the processing of large data sets into the ESDS. It provides ingestion agility through an easy-to-use interface as well as providing reliability and security by leveraging proven technology such as encryption and secure network protocol. Data ingestion is one of the critical ESDS workflows, enabling the movement of massive amounts of data from various sources into the ESDS. The ingestion process provides the ability to perform bulk data intake into the ESDS. Generally, the ingestion process permits a zip file to be uploaded that contains xml files along with a control file. The ESDS efficiently allows for the collecting, aggregating and movement of data from many different sources to a centralized data store and then provides for the transformation and analysis of that data.

## Scope

This Data Ingestion Guide provides the instructions and supporting information you need to deploy ESDS data ingestion. This guide provides an overview of the ESDS ingestion process, file formats, entity mapping, ingestion keys, ingestion logs and error handling.

## Audience

This guide is intended for system operators and administrators responsible for the data ingestion process. True?

## Dependencies, Assumptions & Constraints

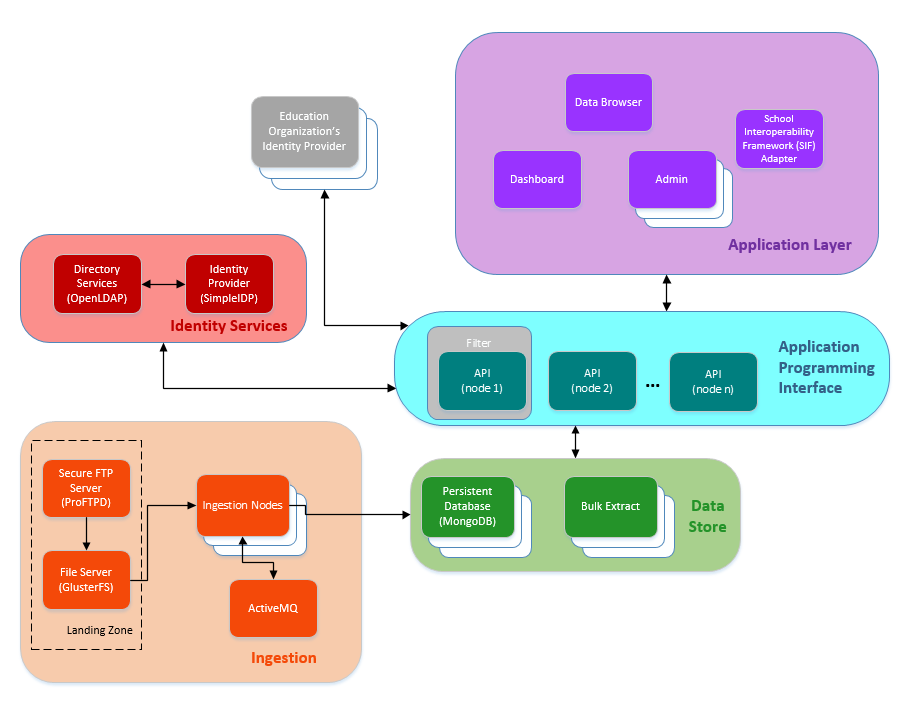
This section provides the known dependencies, assumptions and constraints for the successful ingestion of data into the ESDS.

It is assumed that the knowledge and skill-levels required by an IT Administrator or System Operator in order to successfully perform ingestion include ???.

A positive data ingestion outcome is constrained by the regional or local education agency ???

# ESDS Architecture

This section defines the ESDS architecture. Prior to performing ingestion, it is important to have a basic understanding of the components and processes that make up the ESDS. The sections that follow provide an overview of ESDS technology. ESDS 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 how information flows between the subsystems and components.



Major subsystems shown:

* **Application Layer** - Layer that includes all web-based ESDS applications as well as any third-party web or mobile applications that an education organization adds to its ESDS implementation. A user’s defined role determines the access level to applications.
* **Application Programming Interface** - Layer that applications use to interact with the ESDS. It consists of one or more nodes hosting the REST API.
* **Identity Services** – Layer that displays the integrated identity solution for ESDS. SimpleIDP is the identity provider for ESDS, and the OpenLDAP Directory Services are utilized for the state and local education agencies using the ESDS deployment.
* **Data Store** – Layer that includes the ESDS databases that consist of educational data from the state and local education agencies plus other data necessary for ESDS operation.
* **Ingestion** - Layer dedicated to the process of adding large amounts of data at one time to the ESDS.

The ESDS architecture is a scalable deployment of commodity Linux servers. These servers fulfill the roles of API nodes, ingestion processors, application hosts, data store nodes, and background service hosts. Each part of the system can be horizontally scaled to support the high-levels of concurrency and large data sets that the SDS manages.

# ESDS Technology and Ed-Fi Unifying Data Model

The ESDS technology is based on the Ed-Fi Unifying Data Model. ESDS technology maintains a one-to-one mapping between the two models for the majority of the entities. However, some Ed-Fi entities have been merged into a single entity or split into multiple entities in ESDS.

The ESDS data ingestion subsystem, which handles bulk data additions and updates to the ESDS data store, uses Ed-Fi XML schemas as the primary format for the source data. To handle the differences between the ESDS CEM and the Ed-Fi data model, ESDS provides an Ed-Fi XML schema extension. Deviations from Ed-Fi and the entity-level mapping between ESDS and the Ed-Fi are defined in the *Data Mapping* section of this guide.

# Ingestion Overview

This section defines how ESDS ingests, stores, and handles data.

## Tenancies

ESDS identifies state and local education agencies as *tenancies*. Each tenancy has access to its own set of data, and the same ESDS data store can serve multiple tenancies. In most cases, there exists a hierarchy of tenancies served by the same data store: the state education organizations and the local levels below it (district, county, city, etc.).

When a new tenancy comes on board with ESDS, an existing administrator for that ESDS deployment must create the new tenancy using administration tools. This process is defined in the *Administrator Guide*.

As part of on-boarding a new tenancy, the administrator must, at a minimum, create an administrator account that serves that tenancy. Then, either that new administrator or another authoritative administrator must create the following for that tenancy:

* A landing zone - a group of resources allocated to ingesting data for the tenancy.
* One or more ESDS user accounts with the *Ingestion User* role. If you are ingesting data, you must use the credentials of this account.

With the administrators, landing zones, and ingestion users in place, a tenancy can ingest its first set of data to the ESDS.

## Landing Zones

Each tenancy in the ESDS has one or more landing zones that serve that tenancy. Unlike the ESDS REST API and applications built on it, the landing zone is designed to handle a significant amount of data (thousands of records) as part of a single operation.

From a user's perspective, the landing zone is a secure FTP server used to upload files for data ingestion. An ESDS user with the Ingestion User role can use those user credentials to connect to the landing zone and upload files.

The landing zone detects newly uploaded files and begins the ingestion process. Each file uploaded launches a new ingestion job, and the job ID value contains the name of its corresponding .zip file.

After handling each ingestion job, the landing zone generates a series of log files with messages about the log including status. The log files are named using the job ID and other identifying information. An ingestion user can download these files from the same FTP server connection used to upload the .zip files.

The *Ingestion Logs* section of this guide provides instructions on how to read and interpret the messages in the log files.

## Ingestion Sequence

This section provides the sequence of events that occur when ingesting data to the ESDS.

**Prerequisites:**

When a new tenancy comes on board the ESDS operator creates the tenancy. The ESDS operator will create the first administrator account that serves that tenancy. From that point on, additional administrator accounts can be created by any existing administrator account (tenant administrators). Tenant administrator then creates their own landing zone.

**Sequence:**

1. A .zip archive file is created containing one control file and one or more of XML data files.
2. FTP client is used to upload the .zip archive file to the landing zone for the target tenancy.
3. The ingestion system picks up the .zip archive and creates a corresponding ingestion job.
4. The ingestion system parses and validates the .zip archive.
5. The ingestion system parses and validates the control file within the .zip archive.
6. The ingestion system validates the XML files within the .zip archive.
7. The ingestion system parses those XML files and creates objects from the data.
8. The ingestion system continues to process the data objects until all data is processed and all changes are carried to the ESDS.
9. The ingestion system adds log files for the ingestion job and its individual XML files to the landing zone. The log files contain errors, warnings, statistics, and other status information.

If any part of the ingestion job fails, then some or all of the data may not be included in the ESDS. The *Ingestion Logs* section of this guide provides details on how to determine where failures happened and what to do about them.

## Ingestion Validation

During its validation phases, the ingestion system checks the consistency of the job control file and validates the XML files against the XML schemas for ingestion. The control file format is defined in the section *Ingestion Control* *File Format*. The XML schemas are described in *Ingestion Schemas.*

## Ingestion Job ID and Resource ID

A single ingestion job corresponds to a .zip archive uploaded to a tenancy's landing zone. When ESDS begins processing a new ingestion job, it assigns it a unique identifier called its Job ID. The Job ID contains the name of the ingested file and an epoch time string.

The XML data files within each .zip file are resources processed as part of the ingestion job. As each file is processed, it is identified with a Resource ID corresponding to its file name.

Log files generated by the ingestion system use the Job ID and Resource ID values in their filenames and content.

## Processing an Ingestion Job

When processing an ingestion job, the ingestion system is responsible for generating a changeset to represent the changes to the ESDS data store. This *changeset* is an object that indicates what should change in the data store in order to bring the data up-to-date. It includes both what to add to the data store and what to change about existing data.

While creating the changeset, the ingestion system must resolve entity references in the XML. This includes both entities within the same ingestion job and to previously ingested entities. The matching algorithm for resolving entities relies on a combination of each entity's fields to distinguish between insert and update operations in the ESDS.

For entities with unique tenancy-wide identifiers (Student, Teacher, Staff, and Parent), the stateUniqueId attribute 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. If no match is found, the operation is an insert.

Association entities are handled similarly. 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 is an insert.

## Ingestion Schemas

The structure of the ESDS is organized based on the Ed-Fi Core standard. Ed-Fi Core was created as a unifying logical data model for educational data. The Ed-Fi Core XML schema aligns with the Common Education Data Standards (CEDS). By aligning with Ed-Fi, the ESDS ingestion schema also aligns with CEDS.

The ESDS ingestion schema currently conforms to Ed-Fi Core version 1.0.03. The XML schema description is available for download from *ed-fi.org*. For more information on CEDS, refer to *ceds.ed.gov*.

### ESDS Extensions to Ed-Fi

ESDS has extended the Ed-Fi Core XML schema to achieve the following goals:

* Remove XML IDREF support.
* Constrain entity references to use one pattern of natural keys for referencing entities.
* Remove support for unused entities and for unused attributes of supported entities.
* Add attributes to supported entities.
* Make optional attributes of supported entities mandatory.
* Restructure reference types to use nested references.
* Make identity types within reference objects mandatory.

For ingestion users, these extensions are expressed through a series of XSD files available for download from ESDS. Look for "ESDS ingestion extension schemas" at *https://????/library*.

### Interchange Schemas

Interchange schemas define XML representations of particular data spaces for transport between systems. From an XML perspective, an interchange is a group of entities and associations. Different interchange schemas may be used to reflect different use cases.

ESDS interchange schemas are largely based on the Ed-Fi interchange schemas. ESDS covers most, but not all, entities in the Ed-Fi Core standard. ESDS consolidates some of the Ed-Fi entities for the sake of scalability and performance, both for bulk data ingestion and for data interaction using the REST API.

The interchange schemas along with the entities are listed in the section *Entity-to-Interchange Schema Mapping*. They map to both Ed-Fi and ESDS-Ed-Fi.

Ingestion users specify these interchanges in the control file when preparing data for ingestion. Look for these interchanges at "ESDS ingestion extension schemas" at *https://????/library*.

## Ingestion Security

Security concerns for ingestion include how users are authorized and authenticate with the landing zone and how data is encrypted while uploading files and downloading logs.

### Authentication

The FTP server for the landing zone authenticates each ingestion user against the ESDS directory. The ingestion user signs in to the landing zone from an FTP client using normal authentication (username/password) and secure protocols and ports (SFTP/SSH). These credentials should match one of the ESDS user accounts with the Ingestion User role for that tenancy. These accounts are created and managed by the ESDS administrators for that tenancy.

### Authorization

The ESDS maintains an internal mapping between landing zones, tenancies, and education organizations. An ingestion user role covers one or more tenancies, typically associated with some point in the education organization hierarchy. An ingestion user can only access the landing zones and upload data for the tenancies in which that user has the ingestion user role permission. An administrative account, including an ingestion user, inherits tenancy and education organization associations when that account is created. These associations will not change as a result of any subsequent changes to that information in the ESDS directory. Therefore, if any important identifying information changes for an ingestion user, an administrator for that user's tenancy should delete those ingestion user accounts and create new ones with the desired information. This identifying information, for example, includes a name referenced in the user's username, such as "jsmith" becoming "jwatson".

### Encryption

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 ESDS applications using a secure network protocol.

# Preparing Data for Ingestion

This section provides an overview and instructions to prepare data for ingestion into the ESDS. Data is ingested from a .zip archive file containing one control file and one or more XML data files. Data ingestion operations for ESDS require that the data files are compliant with the ingestion schema as described previously in this guide. For more information refer to the *Ingestion Schemas* section.

When ingesting data, ensure that you ***only*** ingest to your education organization. Ingestion users have the ability to ingest data to any education organizations within their specific tenancies. This can lead to ingestion collisions between education organizations.

## Preparing the XML Data Files

The XML data files must meet the following requirements:

* Must be valid XML files
* Must validate against the ESDS-Ed-Fi ingestion schema as detailed in the *Ingestion Schemas* section.
* Must meet certain constraints to ensure dependencies are processed in the correct order as discussed in the *Ingestion Constraints* section.
* The top-level education organization in the data files, typically a state education agency (SEA), must match the education organization defined when the landing zone was created.

Assemble these files using your preferred processing and editing tools. In most cases, data conversion is necessary to ensure the data is in the right XML schema. ESDS provides assistance to on-board education organizations. This assistance includes facilitating the conversion of data in preparation for ingestion.

For use in validating content throughout the ingestion process, ESDS publishes the ingestion XSDs. While preparing data, you can use your preferred XML validation tools to validate against these XSDs. You may also find the open source validation tool useful, as described in the *Offline Validation Tool* section.

## Preparing the Control File

After preparing the data XML files, create a text file to serve as the control file for an ingestion job. Name the file using the ".ctl" extension. In your preferred text editor, add lines to the control file using the format defined in the section *Ingestion Control File Format* as a convention guide.

There should be one control file for each ingestion job and each job corresponds to a single .zip archive uploaded to the landing zone. Ensure there is a line in the control file for each XML data file included in the job and that there are no lines for files that are not included in the job.

## Preparing the .zip Archive File

The landing zone for ESDS ingestion only accepts .zip archive files. Create a .zip file that meets the following requirements:

* Includes the control file plus the XML data files listed within that control file.
* Has a flat file structure. All files are at the same directory level with no subdirectories.
* Is created using the "Deflate" method for compression.

# Offline Validation Tool

To minimize the risk of handling validation errors during ingestion, your preparation steps should include validating your data files and correcting any invalid content. You can perform this validation using an open source Offline Validation Tool (OVT) provided as a free download from the ESDS Github repository.

The following sections explain how to install and use the OVT.

## System Requirements

The OVT has the following system requirements:

* Java Runtime Environment (JRE) 1.6
* 512 MB available memory
* 50 Mb available space for the tool
* Extra space for validating the XML file that is at a minimum double the size of the XML file you are validating
* Write permission to the home directory of the tool

## Using the OVT

Use the following procedure to set up and run the OVT:

1. In your web browser, open the ESDS Github repository: *https://github.com/???*
2. Click the following sequence to navigate and download the OVT:
3. ingestion-validation
4. bin
5. OfflineValidationTool.zip
6. View Raw
7. Your web browser should begin downloading the file OfflineValidationTool.zip. If necessary, specify where you want to save the file on your local computer.
8. Unzip the file you downloaded (OfflineValidationTool.zip) into a folder on your computer, such as *C:\OfflineValidationTool*. This is the application directory for the tool.   
   The application directory includes the following:

* The lib directory contains the dependent libraries.
* The ingestion-validation-1.0-{version}-SNAPSHOT.jar file is the executable file for the tool.
* The OfflineValidationTool.sh file is a script for running the tool on Unix and Linux platforms.
* The OfflineValidationTool.bat file is a script for running the tool on a Windows platform.

1. Run the script appropriate for your local computer using the corresponding steps below.

**On a Microsoft Windows system:**

1. Open the Run command dialog. This may be in your **Start** menu, or you may need to press the **Windows key + R** to open it.   
   As an alternative, you can also open the Windows command prompt application and enter the command from there.
2. Use the dialog browse and select OfflineValidationTool.bat from the application directory. Then, place your cursor in the text box and add the path to the zip file you want to validate (shown as $ZipFilePath here):

C:\OfflineValidationTool.bat $ZipFilePath

1. Click OK to run the command.

**On a Linux- or UNIX-based system, including a Mac:**

1. Open a terminal application on the system; on a Mac, you can start **Terminal.app** under **Applications/Utilities/**.
2. Make the shell script file executable from the command line:

chmod +x OfflineValidationTool.sh

1. Run the shell script using the path to the zip file you want to validate as an argument to the command (shown as $ZipFilePath here):

./OfflineValidationTool.sh $ZipFilePath

## Viewing OVT Results

Output from the OVT consists of the following:

* A log file that generates information and errors about the validation process located in the input folder alongside the files that you validated.
* Text, located in the console window where you ran the script, indicating any errors you may have encountered while using the tool.

# Deleting Data Through Ingestion

The ingestion system features bulk deletion of data in addition to the ingestion of data. The sections that follow provide the reference and instruction you need to use this delete feature.

## Action Element Type

By default, ingestion operations can be described as an "upsert," meaning an insert or update depending on whether data with the specified key values exists in the data store.

Additional actions on data can be performed by wrapping the XML describing that data with an *Action* wrapper element. Attributes provided for an instance of the Action element describe the action to take on the data specified starting with ActionType.

The following is an example of the Action element type used to delete specified data:

<Action ActionType="DELETE">

<Student>

<StudentUniqueStateId>123456</StudentUniqueStateId>

<StudentIdentificationCode IdentificationSystem="Local" AssigningOrganizationCode="School">

<IdentificationCode>abcde</IdentificationCode>

</StudentIdentificationCode>

<Name>

<FirstName>Test</FirstName>

<LastSurname>User</LastSurname>

</Name>

<!-- ... sample data omitted for brevity ... -->

<LoginId>test.user@example.com</LoginId>

</Student>

</Action>

The following are some additional characteristics of the Action wrapper element:

* This wrapper can be used alongside default ingestion operations without disrupting those insert or update actions. For example, you can use the Action wrapper around entity instances representing records you want to delete while the ingestion job includes data for new and existing records.
* Actions of different ActionType values have different characteristics based on that ActionType. See documentation specific to that ActionType for details.
* Multiple entity instances can be specified within the same Action wrapper, but the actions performed on those instances might not occur as a single operation.

## Delete Actions

One of the actions you can take by adding the Action element type (described in the *Action Element Type* section) is bulk deletion of the specified data:

<Action ActionType="DELETE">

<!-- Data to be deleted, described using the ingestion schema XML. -->

</Action>

Data to be deleted can be specified in one of two ways:

* The entire record, complete with all required fields, expressed as an entity instance in the ingestion XML schema. Though this option requires expressing the entire record, only the identifying information for the record is examined for the delete action. The rest of the information is ignored.   
  For example:

<Action ActionType="DELETE" Cascade="false" Force="true">

<!-- Whole entity case ->

<Student>

<StudentUniqueStateId>1</StudentUniqueStateId>

<StudentIdentificationCode IdentificationSystem="Local" AssigningOrganizationCode="School">

<IdentificationCode>abcde</IdentificationCode>

</StudentIdentificationCode>

<Name>

<FirstName>Jessie</FirstName>

<LastSurname>Mcgowan</LastSurname>

</Name>

<!-- ... additional record data omitted for example... -->

<LoginId>2895@fakemail.com</LoginId>

</Student>

</Action>

* The entity reference type in the XML schema, typically consisting of only the natural key fields.   
  For example:

<Action ActionType="DELETE" Cascade="false" Force="true">

<!-- Reference case -->

<StudentReference>

<StudentIdentity>

<StudentUniqueStateId>1234678</StudentUniqueStateId>

</StudentIdentity></StudentReference>

</Action>

The rules that the ingestion system follows during a delete action can be customized using additional attributes on the Action entity. Two common scenarios are described in the sections that follow, and a list of the attributes and their default settings can be found in the section *Action Attributes for ActionType = DELETE*.

## Forced Deletes

Action attributes:

Cascade= false

Force= true

Forced deletes follow these rules:

* Deleting a record (expressed as an element instance in XML) does not trigger a cascade action. Any other records referring to the target record remain unchanged in the data store.
* If a deleted record has referring "child" data, that data is orphaned in the database except in the cases listed in the section *Action Attributes for   
  ActionType = DELETE.*
* When force deleting an education organization, student data that has an association to it remains unchanged. That association is removed if the studentSchoolAssociation that includes the deleted education organization in its parent hierarchy is also deleted with a forced delete action. However, the student data will still remain. Only the association between the student and the education organization will be removed.

## Safe Deletes

Action attributes:

Cascade = false

Force= false

Safe deletes follow these rules:

* Deleting a record (expressed as an element instance in XML) does not trigger a cascade action. Any other records referring to the target record remain unchanged in the data store.
* If any other records refer to the target record, the delete action fails. The logged error message for a failed delete indicates that child data exists for that record.

## Action Attributes for ActionType = DELETE

The following are the attributes you can use to control delete actions through ingestion. These are attributes of the Action entity in the XML that apply when attribute ActionType = DELETE.

| **Attribute** | **Data Type** | **Default Value** | **Description** |
| --- | --- | --- | --- |
| Cascade | Boolean | False | If this is True, the ingestion system attempts to propagate the delete to referring objects. This eliminates having existing records that refer to a deleted record. |
| Force | Boolean | True | This is only relevant when Cascade = False. In that case, setting Force to False creates a safe delete, which means that the ingestion system will fail and produce an error when one or more other records refer to the target record. If Force is True, referring records remain unchanged, still referring to the deleted record. |
| LogViolations | Boolean | True | This is only relevant when Force = True. In that case, LogViolations = True causes the system to log information for objects that remain in the system (orphaned data), identifying them as integrity violations. This information is posted in the log files for the corresponding ingestion job. |

# Ingestion Logs and Error Handling

The sections that follow define the log files associated with ingestion, list the types of information you will find in those files, and provide information on how to handle the errors and warnings found in those files.

## Ingestion Logs

When ESDS processes an ingestion job, it creates several log files and places those files in the landing zone, accessible for download from the same FTP 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 file names and contents for each log file use the job ID and resource ID values. Within the files, each new log file entry begins with INFO, WARN, or ERROR to indicate the type of message. See the section *Ingestion Errors and Warnings* for a complete list of messages along with instructions on how to handle them.

***Important!*** Error messages imply that the ingestion of the specified data has failed, either as a whole or in part. To complete a successful ingestion, you must resolve these errors.

## Ingestion Deltas

Deltas are records that are not processed by an ingestion job because the data already exists in the database. In other words, the ingestion of the data would not change the values stored in the database. When deltas occur, ESDS records them in the main log file associated with the ingestion job. The deltas appear in the following format:

INFO <number of deltas detected per resource>

## Parser Errors and Warnings

ESDS embeds the Xerces XML parser to validate incoming files against the ESDS-Ed-Fi schema. These Xerces errors and warnings, along with hints on how to resolve them, can be found in the Xerces parser documentation located at <http://xerces.apache.org/>.

A Xerces parser error results in an error for ingestion, too. This means that the associated data is not ingested and follow-up action is necessary to resolve the issue.

The following sections detail common scenarios where a parser error results in an ingestion failure, including information on the incorrect XML causing the error and the correct XML that should resolve the error.

### XML elements are out of order

The ESDS-Ed-Fi schema defines a certain sequence those elements should follow.

|  |  |
| --- | --- |
| **Incorrect XML** | **Correct XML** |
| <Student>  <Name>  <FirstName>Rhonda</FirstName>  <MiddleName>Shannon</MiddleName>  </Name>  <StudentUniqueStateId>  530425896  </StudentUniqueStateId>  ...  </Student> | <Student>  <StudentUniqueStateId>  530425896  </StudentUniqueStateId>  <Name>  <FirstName>Rhonda</FirstName>  <MiddleName>Shannon</MiddleName>  </Name>  ...  </Student> |

### XML element not defined in the XSD

An element not defined in the ESDS-Ed-Fi schema or that is not part of a record cannot be ingested.

|  |  |
| --- | --- |
| **Incorrect XML** | **Correct XML** |
| <Student>  <StudentUniqueStateId>  530425896  </StudentUniqueStateId>  <Name>  <FirstName>Rhonda</FirstName>  <MiddleName>Shannon</MiddleName>  </Name>  <FavoriteFlower>Rose</FavoriteFlower>  ...  </Student> | <Student>  <StudentUniqueStateId>  530425896  </StudentUniqueStateId>  <Name>  <FirstName>Rhonda</FirstName>  <MiddleName>Shannon</MiddleName>  </Name>  ...  </Student> |

### XML element of minimum size "1" cannot be empty

An element defined in the ESDS-Ed-Fi schema to have a minimum size of "1" cannot be left empty when creating or updating the record.

|  |  |
| --- | --- |
| **Incorrect XML** | **Correct XML** |
| <Student>  <StudentUniqueStateId>  </StudentUniqueStateId>  <Name>  <FirstName>Rhonda</FirstName>  <MiddleName>Shannon</MiddleName>  </Name>  ...  </Student> | <Student>  <StudentUniqueStateId>  530425896  </StudentUniqueStateId>  <Name>  <FirstName>Rhonda</FirstName>  <MiddleName>Shannon</MiddleName>  </Name>  ...  </Student> |

## Example Ingestion Log

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 for processing: 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 processing: 0

INFO [file] Grade\_12\_Math\_CCS\_G\_C.xml records not considered for processing: 0

INFO [file] Grade\_12\_Math\_CCS\_G\_SRT.xml (neutralrecord/AssessmentMetadata)

INFO [file] Grade\_12\_Math\_CCS\_G\_SRT.xml records considered for processing: 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 processing: 0

INFO [file] Grade\_12\_Math\_CCS\_G\_C.xml records not considered for processing: 0

INFO [configProperty] dry-run INFO All records processed successfully.

INFO Processed 19 records.

Records that fail XSD validation are not considered for processing and produce corresponding error files. Only records that pass XSD validation are considered for processing. If those records fail later during processing, they are reported as *records failed processing* and produce corresponding error files.

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.

## Ingestion Log Files

Ingestion job log files include the following identifiers in their filenames:

* 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 resource ID (resourceId) refers to the name of a specific file within the job.

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 <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> |

## Ingestion Errors and Warnings

The following table defines the errors and warnings that may appear in the ingestion log files as described in the section *Ingestion Log Files*. Detailed within the table are descriptions of what transpired to generate each error and the required corrective actions.

| **Message Code** | **Message Text** | **Issue and Corrective Action Required** | **Level** |
| --- | --- | --- | --- |
| BASE\_0001 | File {0}: Specified in control file, but not found in zip file. | The named file is listed in the control file, but is not present in the uploaded .zip archive. Recreate the .zip archive to include the file, then try ingesting again. | Error |
| BASE\_0002 | No valid files specified in control file. | All the files specified in the control file are invalid. Be sure the XML files validate in accordance with ingestion documentation. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0003 | No files specified in control file. | The control file does not have any file entries. Edit the control file to correctly describe each file in the .zip archive, then recreate the archive and try ingesting again. | Error |
| BASE\_0004 | File {filename}: File name contains a path. | The indicated file has an invalid file name, possibly being a file path rather than just a file name. Be sure the .zip archive has a flat file structure (files only, no directories), and update the control file to remove any paths in the file names. Then, recreate the archive and try ingesting again. | Error |
| BASE\_0005 | File {filename}: Unknown or no file format specified in the control file. | The control file entry for the indicated file either has no file format or specifies an unrecognized file format. Update the XML and control files as needed to ensure the formats follow the specification in accordance with ingestion documentation. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0006 | Deprecated message code. | | |
| BASE\_0007 | Deprecated message code. | | |
| BASE\_0008 | Deprecated message code. | | |
| BASE\_0009 | No control file found in .zip archive {filename}. | The .zip archive indicated did not contain a control file. Create a control file in accordance with ingestion documentation. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0010 | The .zip archive {filename} contains one or more directories. | The .zip archive has one or more directories under the top-level directory used to form the archive. Be sure the .zip archive has a flat file structure (files only, no directories), and update the control file to remove any paths in the file names. Then, recreate the archive and try ingesting again. | Error |
| BASE\_0011 | Deprecated message code. | | |
| BASE\_0012 | Deprecated message code. | | |
| BASE\_0013 | File {filename}: Not found in the .zip archive. | The indicated file was listed in the control file, but it was not found in the landing zone during ingestion processing. Either remove the entry from the control file, or add the file to those included in the archive. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0014 | File {filename}: Error occurred while reading file from the .zip archive. | The indicated file was listed in the control file, but there was an error while processing the file from the landing zone. If there are other errors, address those first. Otherwise, try ingesting again. | Error |
| BASE\_0015 | File {filename}: Empty file. | The indicated file was specified in the control file, but the file itself is empty. Update the file to have valid, non-empty contents, or remove the file information from the control file. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0016 | Invalid control file entry | The control file contains an entry that does not validate, possibly due to spaces or an incorrect number of arguments on a line. Edit the control file to correct any syntax errors and to format it in accordance with ingestion documentation. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0017 | Deprecated message code. | | |
| BASE\_0018 | File {filename}: Unknown or empty file type specified in control file. | The control file entry for the indicated file either has no file type or specifies an unrecognized file type. Update the XML and control files as needed to ensure the types follow specifications in accordance with ingestion documentation. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0019 | Deprecated message code. | | |
| BASE\_0020 | Could not read .zip archive {filename}. File not found when validating. Possible file damage or corruption. | The specified .zip archive was not found when trying to validate the file. Be sure the file remains unchanged in the landing zone to complete the ingestion process. | Error |
| BASE\_0021 | Could not read .zip archive {filename}. Error reading the file during validation. Possible file damage or corruption. | An error occurred in the ingestion system while attempting to validate the specified .zip archive. Recreate the .zip archive and try ingesting again. | Error |
| BASE\_0022 | Could not read .zip archive {filename}. Unsupported compression method. Use the Deflate compression method to create archive. | The specified .zip archive was created using a compression method other than "Deflate," the most common method for creating .zip files. The ingestion job parses data directly from the archive, and it requires that the "Deflate" method was used to create the file. Recreate the archive using the “Deflate” method and try ingesting again. | Error |
| BASE\_0023 | File {filename}: Not found for XSD validation. | During the XSD validation process, the specified XML file was not found in the .zip archive. Be sure the file exists. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0024 | File {filename}: Problem reading file for XSD validation. | During the XSD validation process, the specified XML file could not be read from the .zip archive. Be sure the file exists. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0025 | Could not read .zip archive {filename}. Error occurred while reading contents. Possible file damage or corruption. | While reading the contents of the .zip archive indicated, the ingestion job encountered an error. Resolve any errors. Then, recreate the .zip archive and try ingesting again. | Error |
| BASE\_0026 | XSD warning: {warning message} | This is an XSD warning for the identified XML file, including the specific line number and column number. XSD validation provides user-facing warnings defined by the SAX parser vendor, Xerces. See Apache Xerces documentation for a list of possible warning messages that could appear here. | Warning |
| BASE\_0027 | XSD error: {error message} | This is an XSD error for the identified XML file, including the specific line number and column number. XSD validation provides user-facing errors defined by the SAX parser vendor, Xerces. See Apache Xerces documentation for a list of possible error messages that could appear here. | Error |
| CORE\_0001 | The target tenant is still on-boarding. Times to complete this process can vary. Please try ingestion again later. | The landing zone used for uploading the .zip archive is for a tenant in the ESDS data store who is still on-boarding that tenancy. This means that ESDS administrators are probably still handling necessary on-boarding tasks that must be complete before ingesting data for that tenancy. Check the on-boarding status with the tenant and ESDS administrators, and try ingesting again when on-boarding is complete. | Error |
| CORE\_0002 | Deprecated message code. | | |
| CORE\_0003 | Failed to parse control file: {error message} | The ingestion job encountered the indicated error when parsing the control file. This is an internal error. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0004 | Failed to process [{record type}] due to preceding errors | The record type indicated was not processed because of errors that occurred previously in the ingestion job. See other log file entries to determine what errors occurred and how to resolve them. | Error |
| CORE\_0005 | Fatal problem saving records to database: Entity {entity type} | The entity type indicated was a point of failure when saving ingested records to the ESDS data store. This is an internal error. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0006 | There has been a data validation error when saving an entity: Error: {type of error}; Entity: {entity type}; Field: {field name}; Value: {field value}; Expected: {expected type} | The indicated element has invalid data or does not validate with the XML schema for ingestion. The reason for failure is specified in the message. Look for additional errors in the job error log to determine which records failed. | Error |
| CORE\_0007 | Entity ({entity type}) reports warning: {warning message} | The warning shown in the message occurred when processing the entity type indicated. | Warning |
| CORE\_0008 | Issue finding key field {key field type} for entity of type {entity type} | The ingestion job encountered an error finding the key field indicated. Update the XML files so that the fields that serve as key fields for that entity type have unique identifying data for each instance of that entity. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0009 | Failed to resolve an ID for entity {entity type}; reference to {reference type} is incomplete because the following reference field is not resolved: {path to the referenced field} | The ingestion job was unable to find a deterministic ID value for the entity indicated because the indicated reference was not found. Update the XML files so that the indicated entity instance follows schema requirements for the entity in accordance with ingestion documentation. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0010 | An entity is missing one or more required natural key fields: Entity: {entity type}; {natural key field name(s)} | The entity instance with the indicated record number is invalid due to the missing key field(s) specified. Update the XML files so that the key fields for all entity instances have a unique, valid combination of values, distinguishing them from other instances of the same entity. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0011 | Internal error occurred. Please contact the operator for assistance resolving this problem. | The ingestion job encountered an error finding a key field value specified for an entity instance. This is an internal error, not something that requires a change in the ingested files. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0012 | Invalid key fields for an entity instance: {error message} | The key fields for an entity instance are invalid, resulting in the error message indicated. Update the XML files so that the key fields for all entity instances have a unique, valid combination of values, distinguishing them from other instances of the same entity. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0013 | Deprecated message code. | | |
| CORE\_0014 | Failed to process .zip archive: {error message} | The ingestion job was unable to process the .zip archive due to the indicated error. If necessary, take action to resolve the indicated error and try ingestion again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0015 | Deprecated message code. | | |
| CORE\_0016 | Deprecated message code. | | |
| CORE\_0017 | Deprecated message code. | | |
| CORE\_0018 | Index verified: {collection name} {index keys}, unique {unique index; Boolean value} | The ingestion job has verified the indicated index. No action is required. | Info |
| CORE\_0019 | Deprecated message code. | | |
| CORE\_0020 | Deprecated message code. | | |
| CORE\_0021 | Deprecated message code. | | |
| CORE\_0022 | Deprecated message code. | | |
| CORE\_0023 | Deprecated message code. | | |
| CORE\_0024 | Deprecated message code. | | |
| CORE\_0025 | Deprecated message code. | | |
| CORE\_0026 | Deprecated message code. | | |
| CORE\_0027 | Error processing ingestion job, data transformation failed: {error message} | The ingestion job failed as described in the message shown. Take corrective action if necessary and try ingestion again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0028 | Deprecated message code. | | |
| CORE\_0029 | Deprecated message code. | | |
| CORE\_0030 | LearningObjective cannot have multiple parent objectives: {parent learning objective(s)} | Each LearningObjective entity instance can have at most one other LearningObjectiveReference. Update the XML files to enforce this requirement. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0031 | Could not resolve child LearningStandardReferences for LearningObjective {parent learning objective} | The LearningObjective entity instance indicated is missing a reference to a LearningStandard instance. Update the XML files so that each LearningObjective instance references a LearningStandard instance. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0032 | Deprecated message code | | |
| CORE\_0033 | Deprecated message code | | |
| CORE\_0034 | Could not resolve LearningObjectiveReference with Objective: {objective value}, AcademicSubject{value}, ObjectiveGradeLevel{value} | The data is missing a LearningObjective instance that matches a combination of objective, academicSubject, and objectiveGradeLevel attributes indicated. These three attributes are key fields for the LearningObjective entity. Update the XML files so that each LearningObjective instance has a unique combination of values for these three attributes. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0035 | No tenant ID provided. Please contact an ESDS administrator or operator for assistance resolving this problem. | The ingestion job encountered an error during a purge operation due to missing tenant ID information. This is an internal error that is not likely resolved just by trying again. Contact the product administrator to manage the issue. | Error |
| CORE\_0036 | Error processing purge job: {error message} | The ingestion job encountered the indicated error during a purge operation. Take corrective action if necessary and try ingesting again. If the error continues, contact the product administrator to manage the issue. | Error |
| CORE\_0037 | Two or more LearningObjectives have duplicate combination of Objective: {value}, AcademicSubject: {value}, ObjectiveGradeLevel: {value} | Multiple LearningObjective entity instances are identified by the combination of key field values indicated. Update the XML files so that there is only one unique LearningObjective instance with this key field value combination. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0038 | Index missing: {collection name} {index keys}, unique: {unique index; Boolean value} | The ingestion job encountered a missing index in the ESDS datastore. This is an internal error that is not likely resolved just by trying again. Contact the product administrator to manage the issue. | Error |
| CORE\_0039 | Deprecated message code. | | |
| CORE\_0040 | Deprecated message code. | | |
| CORE\_0041 | Deprecated message code. | | |
| CORE\_0042 | Deprecated message code. | | |
| CORE\_0043 | Could not find ObjectiveAssessment reference with the ObjectiveAssessmentIdentificationCode {value} | The ObjectiveAssessment entity instance indicated by the ObjectiveAssessmentIdentificationCode was provided as a reference in the data set, but the ingestion job could not find that ObjectiveAssessment instance. | Warning |
| CORE\_0044 | Ignoring child ObjectiveAssessment with ObjectiveAssessmentIdentificationCode {value} since it already exists in the hierarchy. | The ingestion job found the indicated ObjectiveAssessment entity instance as part of the ObjectiveAssessmentReference attribute for another ObjectiveAssessment instance, but determined that it already existed in the ObjectiveAssessment hierarchy. | Warning |
| CORE\_0045 | Deprecated message code. | | |
| CORE\_0046 | Internal error occurred. Please contact the operator for assistance resolving this problem. | The ingestion job failed to find a matching entity instance in the data store. This is an internal error. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0047 | Internal error occurred. Please contact the operator for assistance resolving this problem. | The ingestion job was unable to transform any entity instance. This is an internal error. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0048 | Deprecated message code. | | |
| CORE\_0049 | Deprecated message code. | | |
| CORE\_0050 | Deprecated message code. | | |
| CORE\_0051 | Control file is invalid. Remove all associated files from the landing zone. | The ingestion job has an invalid control file. Using an FTP client, clear files from the landing zone before uploading any more files. Repair the control file in accordance with ingestion documentation. Then, recreate the .zip archive and try ingesting again. | Warning |
| CORE\_0052 | Could not find key field {key field value} for entity of type {entity type}. | The ingestion job could not find the key field indicated for the specified entity type. Update the XML files so that all key fields exist and are valid for each entity. Then, recreate the .zip archive and try ingesting again. | Error |
| CORE\_0053 | Deprecated message code. | | |
| CORE\_0054 | Deprecated message code. | | |
| CORE\_0055 | Deprecated message code. | | |
| CORE\_0056 | Deprecated message code. | | |
| CORE\_0057 | Deprecated message code. | | |
| CORE\_0058 | Landing zone file {filename} is not a zip file | The ingestion job found a file in the landing zone that is not a .zip archive. Remove all files from the landing zone other than valid .zip archives. | Error |
| CORE\_0059 | Failed to onboard tenant. Please contact the operator for assistance resolving this problem. | The ingestion system failed to create the tenant databases needed for successful ingestion. Contact your ESDS administrator to escalate the problem to an ESDS operator. | Error |
| CORE\_0060 | Error processing batch job {job ID} due to: {error message} | The ingestion job encountered the error shown while processing the indicated job. Take corrective action if necessary and try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0061 | Internal error occurred. Please contact the operator for assistance resolving this problem. | The ingestion job encountered an error reading the XML file. This is an internal error. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0062 | Internal error occurred. Please contact the operator for assistance resolving this problem. | The ingestion job encountered an error parsing the schema file. This is an internal error. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0063 | XML file parsing error occurred. Please make sure it is a valid XML file or contact the operator for assistance. | The ingestion job encountered an error parsing the XML files. This is an internal error. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0065 | Access denied to entity type {0}, ID {1}, at depth {2} while safe deleting entity type {3} with ID {4}. | The ingestion job encountered an error during a delete operation. The user does not have the access rights necessary to delete this data. An ESDS operator must grant the user roles with appropriate access rights for this data. | Error |
| CORE\_0071 | Failed to delete entity type {0}, ID {1}. | The ingestion job encountered an error during a delete operation. The deletion of the entity instance failed. This is an internal error. Try ingesting again. If the error continues, contact your ESDS administrator to escalate the problem. | Error |
| CORE\_0072 | Cascade deletes are not currently supported by ingestion. | The ingestion job found a delete request with Cascade option set to "true". Cascade deletes are not currently supported. Please update the XML file to set Cascade to "false". | Error |
| CORE\_0073 | Delete operations for {0} are not currently supported by ingestion. | The ingestion job found a delete request for an entity or reference that is not currently supported for deletes through the ingestion system. Please update the XML file to remove unsupported requests. | Error |
| VALIDATION\_0001 | Invalid input: No control or .zip file found | The uploaded file is neither a control file or a .zip archive. Be sure to create a valid .zip archive with the XML files and control file as described in ingestion documentation. Then, try ingesting again. | Error |
| VALIDATION\_0002 | Validating file: {filename} | The ingestion job has started the validation process for the XML file indicated. | Info |
| VALIDATION\_0003 | Validating file {filename} resulted in errors. | The ingestion job encountered errors while validating the XML file indicated. | Info |
| VALIDATION\_0004 | Validation complete for data file {filename}. | The ingestion job finished validating the XML file indicated. | Info |
| VALIDATION\_0005 | Processing .zip archive [{filename with path}] | The ingestion job has started processing the .zip archive indicated. | Info |
| VALIDATION\_0006 | Processing complete for .zip archive [{filname and path}] | The ingestion job has finished processing the .zip archive indicated. | info |
| VALIDATION\_0007 | Processing control file [{filename and path}] | The ingestion job has started processing the control file indicated. | Info |
| VALIDATION\_0008 | Cannot parse control file | The ingestion job failed to parse its control file. | Error |
| VALIDATION\_0009 | Processing complete for control file [{filename and path}] | The ingestion job has finished processing the control file indicated. | Info |
| VALIDATION\_0010 | Validation for control file resulted in error: {error message} | The ingestion job was unable to validate, and thus parse, its control file. | Error |
| VALIDATION\_0011 | {validation tool name}: Illegal arguments | An unexpected number of arguments were specified. | Error |
| VALIDATION\_0012 | Usage: {validation tool name} [Zip/Control File] | Explains the usage of the tool. | Error |
| VALIDATION\_0013 | Illegal option - directory path. Expecting a Zip or a Control file | The input file specified is a directory. A .zip/.ctl file was expected. | Error |
| VALIDATION\_0014 | {filename} does not exist | The indicated file is not present at the specified physical location. | Error |
| VALIDATION\_0015 | Invalid input: No control/zip file found | The file specified is not a file at all. | Error |
| VALIDATION\_0016 | Usage: {validation tool name} [Zip/Control File] | The input file specified is a directory. Explains the usage of the tool. | Error |

# Ingestion File Formats

The following sections provide a reference for the file formats and constraints used when preparing data for ingestion.

## Ingestion File Conventions

All files uploaded for ingestion should follow these conventions:

* They should be UTF-8 encoded unless otherwise specified.
* Machine-readable timestamps use epoch (UNIX) time. 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.
* Human-readable timestamps use a formatted date/time string.

## Ingestion Control File Format

The control file defines the set of XML data files to be ingested as part of the ingestion job. ESDS expects this file to be a text file with a .ctl file extension (such as control\_file.ctl). Within the control file is a series of lines providing information about each XML data file with additional parameters to control the ingestion job.

### Control File: Data File Lines

For each XML file, the control file must include a line formatted as a comma-separated list of values as shown below:

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

|  |  |
| --- | --- |
| **XML File Component** | **Description** |
| <file format> | The file format in relation to the ingestion system. For ESDS, this would be defined as follows:  edfi-xml  This is the only supported file format. |
| <interchange> | The applicable Ed-Fi interchange name. These interchanges are listed and described in the appendix *Interchange Entity Mapping*. |
| <file name> | The file name. This value should follow the rules below:   * Case insensitive. * Should not contain any relative file paths or operating-system-specific path delimiters (such as "/", "\", or ":") since the .zip archive should have no directories, there is no need for paths. * Should **not** be enclosed in double quotes, a common convention for configuration files. *Exception:* If the file name itself contains double quotes or commas, then it should be enclosed in double quotes, and a backslash ("/") should be used to escape the double quotes that are part of the file name. |

Other rules for XML data file entries:

* Each of the three (3) 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.

**Note:** The file's MD5 hash or checksum for a file may appear as an additional value in each line. This checksum is expressed as 32 hexadecimal digits, and it was a required component in earlier versions of ESDS. However, if it is included in the file, it is ignored during ingestion processing.

### Control File: Job Control Lines

Job-level parameters in a control file direct the job as to how to proceed with parsing and storing the data. Job-level parameters are additional lines that start with the "@" symbol. Parameters that do not require a value are treated like flags ("on" if it appears in the file, "off" if not). Both parameter names and parameter values are case-insensitive.

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

| **Parameter** | **Effect** |
| --- | --- |
| @dry-run | Indicates that the results of the ingestion job should not be written to the persistent data store. This allows for testing the success or failure of an ingestion job, and for handling errors produced by the job, before making permanent changes to the data store. |
| @purge | Deletes all previously ingested data from this tenant. This is the only operation performed by the job that processes it. All other content in 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 its registered applications.  NOTE: This is only used in production mode. |
| @duplicate-detection | Various modes for the duplicate detect/discard optimization (a.k.a. the recordHash optimization a.k.a. the delta-hash optimization):   * [unset]: Normal mode, hash data and compare with recordHash * reset: Like normal, but first purge recordHash of possibly stale data. This will have the performance characteristics of "Day 1" for the current job, and "Day N" for the next job. * disable: Purge the recordHash and also do not hash the incoming data nor insert it into recordHash, leaving recordHash empty. This will have the performance charactersitics of "Day 1" for the current job and the next job. * debugdrop: Treat all entities subject to recordHash based duplicate discarding as if they are in fact duplicates. This will mimic a Day-N fully redundant ingestion, with the side effect that all recordHash processing is skipped. This is in turn useful for measuring the performance impact of recordHash processing. |

### Control File: Example

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

@dry-run

edfi-xml,StudentEnrollment,data.xml

## Ingestion Data File Format

Data files are standard XML following the schema described in the section *Ingestion Schema*. When these files are validated, the validation process includes checking to ensure that these files are valid XML.

## Ingestion Schemas and Interchanges

All data to be ingested must follow the Ed-Fi XML Schema. This schema is based on the Ed-Fi Core Schema (version 1.0.03) for describing educational data. This section provides the general description of the schemas and interchanges.

The ESDS Ed-Fi XML Schema is published as a series of XSD files, compressed into a single .zip archive, available for download from [https:/???/library](https://inbloom.org/library).

The .zip archive contains the "Core" file and the series of "Interchange" files.

For a list of the interchanges and their entity mappings, see the *Appendix A Entity-to-Interchange Schema Mapping* and *Appendix B Interchange Entity Mapping*.

***Important!*** Because of dependencies between entities and attributes in the XML data files, you must also follow a series of constraints described in the section *Ingestion Constraints*.

## Ingestion Constraints

The following sections discuss the ordering constraints and interchange constraints for ingesting data.

### Key Fields

Each entity instance described in the XML data files must have values for its key fields. The natural keys associated with interchanges and entities are listed in *Appendix C* *Ingestion Natural Keys*.

### Ordering Constraints

Some constraints associated with ingestion include managing dependencies when handling data that spans multiple interchanges. Scenarios where these constraints are important comprise:

* Entities within the same interchange that refer to each other and span multiple XML files in the same ingestion job (entity-specific interchange-level).
* A single ingestion job that contains data from multiple interchanges.
* An entity in one ingestion job that refers to entities ingested by a different ingestion job.
* An entity in one interchange in a single ingestion job that refers to entities in other interchanges in the same job.

**Note:** When an entity is created with a reference to an entity that does not already exist, the SDS will create a "phantom entity" to satisfy the referential integrity of the reference. For example, if an Assessment Item is created, but references an Assessment that does not already exist, an Assessment entity will be created containing only an ID and no data. This Assessment will not be usable, but will allow the Assessment Item to be created. If an Assessment is later ingested that matches the phantom entity, the remaining data will be filled in, and the entity will then be usable. Therefore, when an Assessment comes through with a natural key set that matches that of the Assessment Item, the hash will resolve to the same ID.

ESDS technology does not support XML ID referencing for data to be ingested. Any ID/IDREF pairs within an XML ingestion data file will not be resolved during ingestion. Instead, they are either silently ignored except for any "ref=" notation that violates the XML schema for a particular entity. This is reported to the user as a warning.

AssessmentMetadata Interchange Constraints

The AssessmentMetadata interchange supports ingestion of the following entities:

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

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

Entity constraints are listed in the following table:

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

### EducationOrganization Interchange Constraints

The EducationOrganization interchange supports ingestion of the following entities:

* EducationOrganization
* Course
* CompetencyLevelDescriptor
* Program
* ClassPeriod

An EducationOrganization entity provides the authorization context for all ESDS users. Therefore, it is critical that every user is tied to a particular tier within the EducationOrganization hierarchy. During on-boarding for an education organization, a tenancy's administrator(s), corresponding to LEAs within the hierarchy, are identified by an EducationOrganization entity with an OrganizationCategory set to "Local Education Agency". Ingestion users of an LEA must be associated (using a unique staff identifier) to an ingested EducationOrganization with an appropriate OrganizationCategory. For more information about the implementation of identity services with ESDS, refer to the *Administrator Guide*.

### EducationOrgCalendar Interchange Constraints

The EducationOrgCalendar interchange supports ingestion of the following entities:

* Session
* GradingPeriod

### StaffAssociation Interchange Constraints

The StaffAssociation interchange supports ingestion of the following entities:

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

When teachers and staff access data and applications, the SDS 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 SDS 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.

### StudentAssessment Interchange Constraints

The StudentAssessment interchange supports ingestion of the following entities:

* StudentAssessment
* StudentObjectiveAssessment
* StudentAssessmentItem

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

Entity constraints are listed in the following table.

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

### StudentAttendance Interchange Constraints

The StudentAttendance interchange supports ingestion of the following entity:

* AttendanceEvent

### StudentCohort Interchange Constraints

The StudentCohort interchange supports ingestion of the following entities:

* Cohort
* StudentCohortAssociation
* StaffCohortAssociation

### StudentDiscipline Interchange Constraints

The StudentDiscipline interchange supports ingestion of the following entities:

* DisciplineIncident
* StudentDisciplineIncidentAssociation
* DisciplineAction

### StudentGrade Interchange Constraints

The StudentGrade interchange supports ingestion of the following entities:

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

### StudentEnrollment Interchange Constraints

The StudentEnrollment interchange supports ingestion of the following entities:

* StudentSchoolAssociation
* StudentSectionAssociation

### StudentParent Interchange Constraints

The StudentParent interchange supports ingestion of the following entities:

* Student
* Parent
* StudentParentAssociation

When students and parents access SDS data and applications, SDS 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 SDS 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.

### StudentProgram Interchange Constraints

The StudentProgram interchange supports ingestion of the following entity:

* StudentProgramAssociation

### TeacherSchoolAssociation Entity Constraints

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

# Appendix A: Entity-to-Interchange Schema Mapping

This table shows the mapping between a domain type for Ed-Fi entities (rows) and an interchange schema (columns). These values are used when forming control files used for data ingestion.

|  | **AssessmentMetadata** | **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 |  |  |  |  |  |  |  |  |  |  |  |  |
| LearningObjective | X |  |  |  |  |  |  |  |  |  |  |  |  |
| LearningObjective | X |  |  |  |  |  |  |  |  |  |  |  |  |
| Course |  | X |  |  |  |  |  |  |  |  |  |  |  |
| EducationOrganization |  | X |  |  |  |  |  |  |  |  |  |  |  |
| Program |  | X |  |  |  |  |  |  |  |  |  |  |  |
| CompetencyLevelDescriptor |  | X |  |  |  |  |  | X |  |  |  |  |  |
| Staff |  |  | X |  |  |  |  |  |  |  |  |  |  |
| StaffEducationOrg EmploymentAssociation |  |  | X |  |  |  |  |  |  |  |  |  |  |
| StaffEducationOrg AssignmentAssociation |  |  | 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 |  |  |  |  |  |  |  |  |
| StudentObjective Assessement |  |  |  |  | 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 |  |  |  |
| StudentDisciplineIncident Association |  |  |  |  |  |  |  |  |  | X |  |  |  |
| DisciplineAction |  |  |  |  |  |  |  |  |  | X |  |  |  |
| AttendanceEvent |  |  |  |  |  |  |  |  |  |  | X |  |  |
| StudentProgramAssociation |  |  |  |  |  |  |  |  |  |  |  | X |  |
| CourseOffering |  |  |  |  |  |  |  |  |  |  |  |  | X |
| Section |  |  |  |  |  |  |  |  |  |  |  |  | X |

# Appendix B: Interchange Entity Mapping

This table shows the mapping between the ESDS Ed-Fi interchanges and the entities associated with them, both in the Ed-Fi Core Schema and in the ESDS Ed-Fi Schema.

| **ESDS-Ed-Fi Interchange** | **Ed-Fi Entity** | **ESDS Entity** |
| --- | --- | --- |
| Interchange-AssessmentMetadata | LearningStandard | learningStandard |
| PerformanceLevelDescriptor | studentAssessment/performanceLevelDescriptors, studentAssessment/studentObjectiveAssessment/ performanceLevelDescriptors and assessment/assessmentPerformanceLevel/ performanceLevelDescriptors |
| AssessmentItem | assessment/assessmentItem and assessment/objectiveAssessment/assessmentItem |
| Assessment | assessment |
| ObjectiveAssessment | assessment/objectiveAssessment |
| AssessmentPeriodDescriptor | AssessmentPeriodDescriptor |
| AssessmentFamily | assessmentFamily |
| Interchange-Assessment-Metadata  and  Interchange-Student-Grade | LearningObjective | learningObjective |
| Interchange-Education Organization | Course | course |
| EducationOrganization | educationOrganization |
| Program | program |
| ClassPeriod | classPeriod |
| Interchange-Education Organization  and  Interchange-StudentGrade | CompetencyLevelDescriptor | -- |
| Interchange-EducationOrg Calendar | CalendarDate | calendarDate |
| GradingPeriod | gradingPeriod |
| Session | session/schoolSessionAssociation |
| Interchange-MasterSchedule | CourseOffering | courseOffering |
| Section | section |
| BellSchedule | bellSchedule |
| Interchange-StaffAssociation | CredentialFieldDescriptor | staff|teacher/credentials/credentialFieldDescriptor |
| StaffCohortAssociation | staffCohortAssociation |
| Staff | staff/abstractStaff |
| Teacher | teacher |
| StaffProgramAssociation | staffProgramAssociation |
| TeacherSchoolAssociation | teacherSchoolAssociation |
| TeacherSectionAssociation | teacherSectionAssociation |
| StaffEducationOrg AssignmentAssociation | staffEducationOrganizationAssociation |
| Interchange-StudentAssessment | StudentAssessmentItem | studentAssessmentAssociation/studentAssessment Item |
| StudentAssessment | studentAssessment |
| StudentObjectiveAssessment | studentAssessment/studentObjectiveAssessment |
| Interchange-StudentAttendance | AttendanceEvent | dailyAttendance |
| Interchange-StudentCohort | Cohort | cohort |
| StudentCohortAssociation | studentCohortAssociation |
| Interchange-StudentDiscipline | DisciplineAction | disciplineAction |
| DisciplineIncident | disciplineIncident |
| StudentDisciplineIncident Association | studentDisciplineIncidentAssociation |
| Interchange-StudentEnrollment | GraduationPlan | graduationPlan |
| StudentSectionAssociation | studentSectionAssociation |
| StudentSchoolAssociation | studentSchoolAssociation |
| Interchange-StudentGrade | Grade | grade |
| ReportCard | reportCard |
| StudentCompetency | studentCompetency |
| StudentCompetencyObjective | studentCompetencyObjective |
| CourseTranscript | courseTranscript |
| GradebookEntry | gradebookEntry |
| StudentAcademicRecord | studentAcademicRecord |
| StudentSectionGradebook Entry | studentGradebookEntry |
| Interchange-StudentParent | Parent | parent |
| Student | student |
| StudentParentAssociation | studentParentAssociation |
| Interchange-StudentProgram | StudentProgramAssociation | studentProgramAssociation |

# Appendix C: Ingestion Natural Keys

The table below provides entities, the associated ESDS Ed-Fi natural keys and the detailed key fields.

| **Entity** | **ESDS 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 |
| SchoolYear | SchoolYear |
| AttendanceEventCategory | AttendanceEventCategory |
| AssessmentPeriod Descriptor | CodeValue | CodeValue |
| CalendarDate | Date | Date |
| EducationOrgReference | StateOrganizationId |
| ClassPeriod | ClassPeriodName | ClassPeriodName |
| EducationOrganizationReference | StateOrganizationId |
| Cohort | EducationOrgReference | StateOrganizationId |
| CohortIdentifier | CohortIdentifier |
| CompetencyLeve lDescriptor | CodeValue | CodeValue |
| Course | EducationOrganizationReference | StateOrganizationId |
| UniqueCourseId | UniqueCourseId |
| CourseOffering | SchoolReference | StateOrganizationId |
| SessionReference |  |
|  | SessionName |
| CourseReference | StateOrganizationId |
|  | UniqueCourseId |
| CourseTranscript | CourseAttemptResult | CourseAttemptResult |
| CourseReference |  |
|  | 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 |
| 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 | StudentCompetency ObjectiveId |
| CompetencyLevel | CodeValue |
| StudentSectionAssociationReference | StudentUniqueStateId |
|  | StateOrganizationId |
|  | UniqueSectionCode |
|  | BeginDate |
| StudentCompetency Objective | StudentCompetencyObjectiveId | StudentCompetency ObjectiveId |
| StudentGradebookEntry | StudentSectionAssociationReference | StudentUniqueStateId |
|  | StateOrganizationId |
|  | UniqueSectionCode |
|  | BeginDate |
| GradeBookEntryReference | GradebookEntryType |
|  | DateAssigned |
|  | StateOrganizationId |
|  | UniqueSectionCode |
| StaffEducationOrg AssignmentAssociation | 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 | AssessmentItem IdentificationCode |
| StudentCohortAssociation | StudentReference | StudentUniqueStateId |
| CohortReference | StateOrganizationId |
|  | CohortIdentifier |
| BeginDate | BeginDate |
| StudentDisciplineIncident Association | StudentReference | StudentUniqueStateId |
| DisciplineIncidentReference | StateOrganizationId |
|  | IncidentIdentifier |
| StudentObjective Assessment | StudentAssessmentReference | AdministrationDate |
|  | StudentUniqueStateId |
|  | AssessmentTitle |
|  | AcademicSubject |
|  | GradeLevelAssessed |
| ObjectiveAssessmentReference | ObjectiveAssessment IdentificationCode |
| StudentParentAssociation | StudentReference | StudentUniqueStateId |
| ParentReference | ParentUniqueStateId |
| StudentProgram Association | StudentReference | StudentUniqueStateId |
| ProgramReference | ProgramId |
| BeginDate | BeginDate |
| EducationOrganizationReference | StateOrganizationId |
| StudentSchoolAssociation | StudentReference | StudentUniqueStateId |
| SchoolReference | StateOrganizationId |
| EntryDate | EntryDate |
| StudentSectionAssociation | StudentReference | StudentUniqueStateId |
| SectionReference | StateOrganizationId |
| UniqueSectionCode |  |
| BeginDate | BeginDate |
| Teacher | StaffUniqueStateId | StaffUniqueStateId |
| TeacherSchoolAssociation | TeacherReference | StaffUniqueStateId |
| SchoolReference | StateOrganizationId |
| ProgramAssignment | ProgramAssignment |
| TeacherSection Association | TeacherReference | StaffUniqueStateId |
| SectionReference | StateOrganizationId |
|  | UniqueSectionCode |