

**Educational Secure Data Service**

**Administrator Guide**

**July 31, 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 | 07/31/2014 | Laura Heintz | First Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 Introduction 1](#_Toc394490824)

[2 ESDS Architecture 2](#_Toc394490825)

[3 Security and Administration Designations 3](#_Toc394490826)

[3.1 Tenant and Realm 3](#_Toc394490827)

[3.1.1 Creating and Managing Realms 4](#_Toc394490828)

[3.2 Federated and Hosted Users 5](#_Toc394490829)

[4 Integrated Authentication: ESDS Technology and Local Directory Services 6](#_Toc394490830)

[4.1 The ESDS Authentication Strategy 7](#_Toc394490831)

[4.2 Federated Authentication Strategy 8](#_Toc394490832)

[4.3 ESDS Authentication Integration Procedure 9](#_Toc394490833)

[4.4 Configure ESDS Technology to work with your IDP 11](#_Toc394490834)

[4.5 Integrated ESDS Application Authentication 12](#_Toc394490835)

[4.5.1 Federated Application Authentication: Sample SAML Exchange 13](#_Toc394490836)

[4.5.2 Federated Application Authentication: Installed (Non-Web) Applications 18](#_Toc394490837)

[4.5.3 Federated Application Authentication: Adding New Applications 18](#_Toc394490838)

[4.6 Effective SDS Permissions for Federated Authentication 18](#_Toc394490839)

[4.7 Mapping Directory Entities to the ESDS 1](#_Toc394490840)

[5 Roles and Permissions 1](#_Toc394490841)

[5.1 Educational Organization Hierarchy 1](#_Toc394490842)

[5.2 Default User Roles and Role Groups 2](#_Toc394490843)

[5.3 How ESDS Determines Permissions 4](#_Toc394490844)

[5.3.1 Default Permissions 6](#_Toc394490845)

[5.3.2 Student Data Permissions 6](#_Toc394490846)

[6 Managing User Roles and Rights 7](#_Toc394490847)

[6.1 Adding a New Role Group 8](#_Toc394490848)

[6.2 Mapping Multiple Local IDP Roles to One ESDS Role 9](#_Toc394490849)

[6.3 Managing Custom Roles for Multiple Realms 10](#_Toc394490850)

[7 Contextual Authorization Rules 10](#_Toc394490851)

[7.1 Global Authorization Rules 10](#_Toc394490852)

[7.2 Global Entities 11](#_Toc394490853)

[7.3 Orphan Entities 12](#_Toc394490854)

[8 Contextual Authorization: Staff 12](#_Toc394490855)

[8.1 Staff Context Relies on Education Organization 12](#_Toc394490856)

[8.2 Staff Context Expires When Association Ends 13](#_Toc394490857)

[8.3 Staff Context with Student Data 14](#_Toc394490858)

[9 Contextual Authorization: Teachers 15](#_Toc394490859)

[9.1 Teacher Entity Must Have Certain Associations 15](#_Toc394490860)

[9.2 Teacher Context Relies on Section, Cohort, or Program 15](#_Toc394490861)

[9.3 Teacher Context Expires When Associations End 2](#_Toc394490862)

[10 Contextual Authorization: Parents 2](#_Toc394490863)

[11 Contextual Authorization: Students 4](#_Toc394490864)

[12 Administrative Options 5](#_Toc394490865)

[12.1 System Tools 5](#_Toc394490866)

[12.2 Application Configuration 6](#_Toc394490867)

[13 Administrative Users 6](#_Toc394490868)

[13.1 Administrative User Roles 6](#_Toc394490869)

[13.2 Administrative Rights 8](#_Toc394490870)

[13.3 Creating a New Administrative User 9](#_Toc394490871)

[13.4 User Account Activation 11](#_Toc394490872)

[14 Data Ingestion 11](#_Toc394490873)

[14.1 How Data Ingestion is Accomplished 11](#_Toc394490874)

[14.2 Ingestion Users 12](#_Toc394490875)

[14.3 Creating and Managing Landing Zones 12](#_Toc394490876)

[14.3.1 Landing Zones for Sandbox Developers 12](#_Toc394490877)

[14.4 Ingesting Data into the SDS 13](#_Toc394490878)

[14.5 Schools Interoperability Framework (SIF) Data 13](#_Toc394490879)

[15 Onboarding Districts 13](#_Toc394490880)

[15.1 Onboarding With Bulk Ingestion 13](#_Toc394490881)

[15.2 Onboarding Districts that use Schools Interoperability Framework Data 14](#_Toc394490882)

[15.3 Onboarding with Minimal Ingestion 15](#_Toc394490883)

[15.3.1 Setting Up a Template File for Onboarding With Minimal Ingestion 15](#_Toc394490884)

[16 Using the Offline Validation Tool 16](#_Toc394490885)

[16.1 System Requirements 16](#_Toc394490886)

[16.2 Using the Offline Validation Tool 16](#_Toc394490887)

[16.3 View the Results 18](#_Toc394490888)

[17 Managing Applications 18](#_Toc394490889)

[17.1 Enabling or Disabling Applications for Education Organizations 18](#_Toc394490890)

[18 Setting Up the Dashboard 20](#_Toc394490891)

[18.1 Configuring the Dashboard 20](#_Toc394490892)

[18.2 Configuring Section Profile Lists 20](#_Toc394490893)

[18.3 Documenting your List Configurations 21](#_Toc394490894)

[18.4 Configuring Student Profiles 24](#_Toc394490895)

[18.4.1 Documenting your Student Profile Configuration 24](#_Toc394490896)

[18.5 Uploading Configurations 25](#_Toc394490897)

[18.5.1 Important Considerations for ESDS Dashboard Configuration 27](#_Toc394490898)

[19 Modifying the Dashboard 27](#_Toc394490899)

[19.1 Modifying a Profile 28](#_Toc394490900)

[19.2 Panels 29](#_Toc394490901)

[20 Data Browser 30](#_Toc394490902)

[20.1 Prerequisites 30](#_Toc394490903)

[20.2 Viewing Data 31](#_Toc394490904)

[20.2.1 Your Education Organization 31](#_Toc394490905)

[20.2.2 Counts 32](#_Toc394490906)

[20.2.3 Ingestion Jobs Status 33](#_Toc394490907)

[20.3 Data Browser Search Bar 34](#_Toc394490908)

[20.4 Navigating the Data Browser 35](#_Toc394490909)

[21 Bulk Data Extraction 36](#_Toc394490910)

[21.1 Previous Year Student Data 36](#_Toc394490911)

[21.2 Previous Year Staff Data 38](#_Toc394490912)

[21.3 Top Level Education Organization Extract 38](#_Toc394490913)

[22 Delegating Permissions to the State Education Organization 39](#_Toc394490914)

[23 Security Event Logs 39](#_Toc394490915)

[23.1 Logged Security Events 40](#_Toc394490916)

[23.2 Permissions for Viewing Security Event Logs 43](#_Toc394490917)

[24 Security Event Notifications 43](#_Toc394490918)

[Glossary](#_Toc394490918) 45

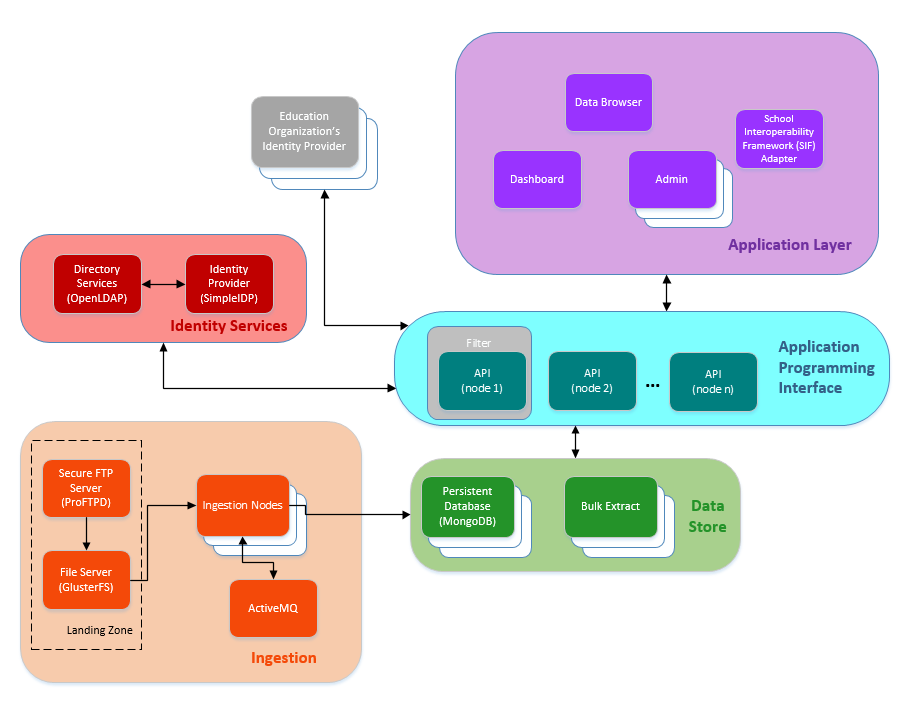
# Introduction

Educational Secure Data Services (ESDS) supports technical administrators at all levels of the Education Organization hierarchy. This document instructs state and local administrators on how to perform day-to-day administrative tasks in the ESDS. These tasks include the following:

* Authentication and authorization
* User roles and permissions
* Creating and managing realms within ESDS that are used to identify authentication boundaries for users.
* Mapping roles between ESDS and an education organization's identity provider.
* Onboarding school data.
* Authorizing applications to use ESDS.
* Configuring the Dashboard application used to view and manage data.
* Performing bulk data extract.
* Delegating local education organization permissions to the state education organization within the same hierarchy.
* Viewing security event logs.

# ESDS Architecture

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

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.

# Security and Administration Designations

To better understand ESDS security and the associated user roles and permissions it is important to understand tenants and realms along with federated and hosted users.

## Tenant and Realm

ESDS administration is often described in terms of a *tenant* and a *realm*.

A [***tenant***](#bookmark152)is a single set of self-consistent data that is logically isolated within the ESDS. A tenant corresponds to an [*Education Organization*](#bookmark112), such as a state, county, or district school system. ESDS was designed so that within a tenant, records can be readily transferred when a student or staff member changes schools, even if it is a transfer between districts in the same state. The highest tenant level that the SDS infrastructure is designed to manage is that associated with a single State Education Agency (SEA).

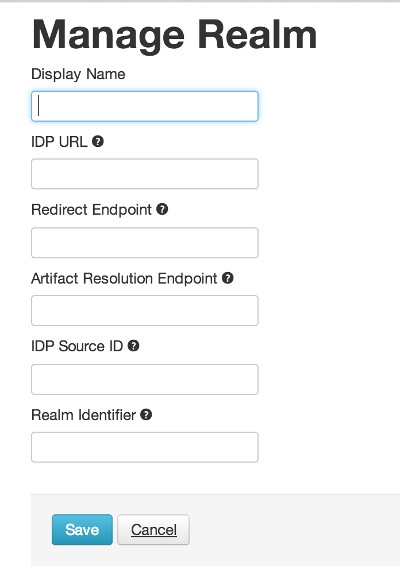
A [***realm***](#bookmark134)is a designation used to describe the context within which a digital user's authentication is valid. In other words, a realm corresponds to the level at which identity services exist. If a district provides its own identity services, that district can serve as a realm. If a state is providing identity services, then the state is the realm. If there is a group of districts within a state that have a shared identity service, then the organization that groups those districts together is a realm. Within the SDS, the highest available realm is an SEA.

**Note:** User IDs, defined by the realm, are only guaranteed to be unique within the context of a single tenant. For example, multiple tenants could have the user ID tsmith, but ESDS does not consider tsmith from one tenant to be the same user as tsmith from another tenant. To avoid data conflicts, a realm should not serve multiple tenants.

### Creating and Managing Realms

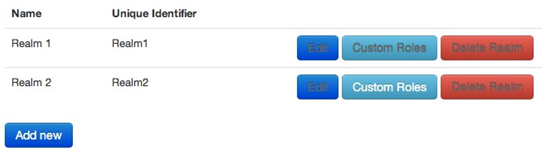
A [*realm*](#bookmark134) serves as an authentication context for a user or application. This means a realm is an Identity Provider (IDP) whose authentication services are integrated with a given ESDS deployment, providing access to data and applications. The highest available realm is a state, though districts also may have their own identity providers.

The Manage Realm tool is available to [*ESDS administrators*](#bookmark147) with a role of [*Realm Administrator*](#bookmark135). To create a realm using this tool:

1. Log in using your Realm Administrator account.
2. Click **Admin**.
3. Click **Manage Realm** under System Tools. If no realms exist, then the **Manage Realm - Create New Realm** page opens as shown below.
4. Enter information in the fields as follows:

* *Display Name* - Enter the name that should appears on the Realm page where the Education Organization users can select and sign in to the IDP. These would be users of the [*federated*](#bookmark116) agency.
* *IDP URL* - Enter the SAML identifier (URL) for the federated agency's IDP.
* *Redirect Endpoint* - Enter the URL to be used for making authentication requests to the federated agency's IDP.
* *Artifact Resolution Endpoint* - Enter the URL to be used for sending artifact resolution requests to the federated agency's IDP.
* *IDP Source ID* - Enter the hex-encoded source ID to be used to identify the federated agency's IDP. This string is matched with the source ID contained in the SAML artifact sent by the IDP. The SAML 2.0 standard contains more details about the IDP Source ID format.
* *Realm Identifier* - Enter the unique value that identifies your realm. This identifier allows applications to send users directly to this realm to sign in, bypassing the realm selection screen.

1. Click **Save**.

The new realm will now appear on the **Realm Selection** page shown below.

1. To create another realm, click the **Add new** button at the bottom of the page and the Manage Realm page opens for you to add another realm.

## Federated and Hosted Users

ESDS user and administrator roles are described as being either *federated* or *hosted*.

A [***federated***](#bookmark116)user is a user that authenticates by using an identity provider that is maintained by an education agency, whether at the state, county, or district level. This IDP, typically a directory service like LDAP, is where the [*realm*](#bookmark134) is defined. Since the IDP is external to ESDS, user roles within the SDS must be customized and mapped to correspond to the user roles in the IDP. This should be accomplished before data is added to the ESDS, whether through [data ingestion](#bookmark109) or through the REST API.

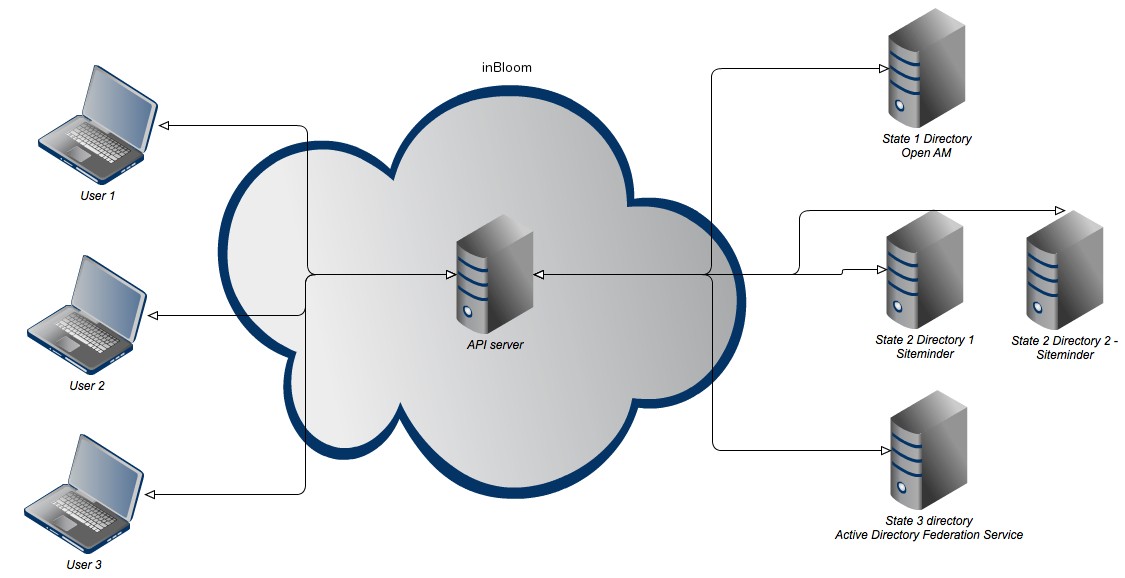
A [***hosted***](#bookmark120)user is a user created in the [*ESDS hosted directory service*](#bookmark148). The hosted directory service is one of the services deployed and managed as part of the ESDS. The role of these hosted users is to serve as administrators for the ESDS and are referred to as [*ESDS administrators*](#bookmark147).

# Integrated Authentication: ESDS Technology and Local Directory Services

ESDS recognizes that an *Education Organization* (EdOrg) will have its own directory services used for user authentication within that organization. The ESDS model aims to integrate authentication to ESDS applications and services with that of other Ed Org resources. By taking this approach, end users can use the same credentials for ESDS resources as for other applications and data in the same Ed Org. This also facilitates integration with any Single Sign-On (SSO) solution already in use by the Ed Org.

The diagram below displays an example of how end users authenticate to ESDS applications and services through this integrated authentication model.

Integrated Authentication Model Overview



ESDS uses the following guiding principles to develop and maintain an integrated authentication solution:

* Only authorized individuals gain access to the information they are authorized to view or change.
* All traffic between ESDS applications and services and the Ed Org's directory services is encrypted.
* All sign-in activities, both successes and failures, are logged by the Ed Org.
* All failed attempts to access data is logged in the ESDS security logs for audit purposes.
* All application and user access to the ESDS REST API requires authentication. No anonymous access is allowed.
* Controls exist to ensure that session identification mechanisms are protected and to ensure that a session cannot be compromised.

Technology used in ESDS for integrated authentication includes the following:

[*Security Assertion Markup Language (SAML) 2.0*](#bookmark144) - ESDS applications and services prompt for user credentials using the SAML protocol, and the identity provider (from the Ed Org) sends a SAML assertion in response. Role and UID information for the user are provided as an attribute of the SAML assertion.

[*OAuth 2.0*](#bookmark130) - ESDS uses the OAuth protocol to authenticate users accessing the ESDS through the ESDS REST API. In the integrated solution, the API handles authentication directly from the identity provider of the Ed Org, bypassing SAML.

The sections that follow describe how integrated authentication works in the ESDS.

## The ESDS Authentication Strategy

ESDS includes its own [*identity provider*](#bookmark121) (IDP) for the purpose of authenticating users to the SDS. However, ESDS also requires that an Ed Org use its own identity provider, typically directory services like LDAP, for managing credentials.

With [*federated authentication*](#bookmark117), users authenticate against the directory service for the education organization. Then, the identity of the user is shared with other ESDS applications through use of the [*SAML*](#bookmark144) protocol. Users can sign in to one ESDS web application, and they are automatically signed in to other web-based ESDS applications operating in the same browser.

The Ed Org’s IDP identifies a [*realm*](#bookmark134) for authentication. ESDS associates each realm with an Ed Org entity in the ESDS. Users associated to that entity, or to a child Ed Org under it, can authenticate using that realm. The staffEdOrgAssignmentAssociation entity establishes that hierarchy in the SDS.

Some possible options for configuration are as follows:

* A state-level education organization has an IDP, and therefore, a single realm associated with that organization. This allows all users in the state to authenticate with this one realm.
* Each local education organization (two or more schools groups together under the state, such a county or district school system) has an IDP, and each administrator at that level sets up a realm in the ESDS to authenticate with that IDP. This would cover authentication for that organization and its schools.
* Two or more local education organizations want to share an IDP and need a shared educationOrganization entity above their individual educationOrganization entities in the ESDS. A realm administrator tied to that parent organization would then set up the realm so that both child organizations could authenticate against.
* Two or more local education organizations want to share an IDP, but need separate realms. The Realm Administrator for each organization sets up a realm pointing to that shared IDP.

## Federated Authentication Strategy

A [*federation*](#bookmark118) as discussed previously in the chapter *Security and Administration Designations* is a group of identity providers that the ESDS identity provider trusts. All communication between these trusted providers is managed using SAML 2.0. The SAML protocol abstracts any unique characteristics of the providers or of the applications using the authentication services. The trust relationship is one-way, meaning that Ed Org directories do not allow ESDS users to sign in to non-ESDS applications and services.

In the federated authentication approach, the Ed Org is responsible for authenticating the end user and managing the authentication credentials. ESDS serves as both of the following:

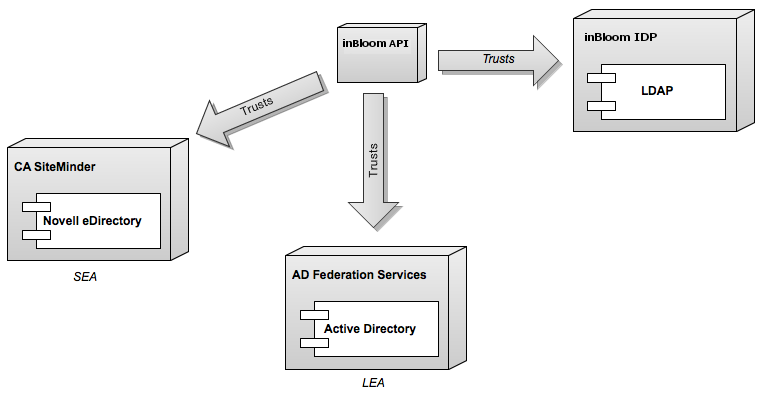
* A native service provider, with applications that require authentication
* A conduit for other service providers (applications) that use ESDS

The following is a summary of the user authentication procedure using this federated approach:

1. When signing in to an application or service in the ESDS, a user is redirected to the education organization's directory or SSO service which collects authentication credentials. ESDS makes this request using SAML 2.0.
2. The identity provider for that education organization authenticates the user and returns a SAML 2.0 assertion to the ESDS service provider indicating that the user provided correct credentials. This SAML assertion includes an attribute with role and unique identifier values from the identity provider.
3. ESDS authorizes the authenticated user to access applications based the following two values:

* **The user's staffClassification value** - In the ESDS, a user is associated with an education organization through its staffEdOrgAssignmentAssociation. Part of the staffEdOrgAssignmentAssociation is the user's staffClassification, which stores the user's role in the education organization. ESDS authentication checks whether a user's staffClassification value matches a role that can authenticate using that realm, either for that education organization, or for one of its child education organizations.
* **How the user's roles in the education organization are mapped to roles in ESDS** - ESDS administrators map user roles as described in the chapter *Managing User Roles and Rights*. This role mapping is required for users to access any applications or data in the ESDS. ESDS authorizes access to applications and data based on this mapping using contextual authorization rules described in the chapter *Contextual Authorization Rules*.

The following diagram shows how federated authentication works in a sample education organization hierarchy. In the example, the state education agency (SEA) and local education agency (LEA) each have different directory services and SSO solutions, and ESDS is configured to trust these identity providers.



Applications also follow certain authentication procedures. These procedures are detailed in the section *Integrated ESDS Application Authentication,* along with what to accept in the request and response data.

The following section provides the steps required to prepare an education organization's directory for federated authentication integration with ESDS.

## ESDS Authentication Integration Procedure

Part of preparing an Ed Org to use ESDS is ensuring that the organization's directory services can support authentication integration with ESDS. The following steps are required before proceeding with integration:

1. Confirm that there are unique identifiers and user roles set for each user listed in the directory, and that they are compatible with ESDS as follows:

* **Unique identifier** - ESDS authorization requires that each user is uniquely identified within the system. This ensures that ESDS can identify all the entities in the ESDS that are associated with the user, such as the current students associated with a teacher. The unique identifier should match the staffUniqueStateId in the data that you will ingest. This will allow us to identify the appropriate users in the ESDS.
* **User role** - ESDS relies on mappings between the user roles in the organization's directory and the roles in the ESDS. On the ESDS side, this includes both the roles in staffEdOrgAssignmentAssociation (assigned to users in the SDS) and the default or custom roles configured by Roles to ensure that ESDS can apply the appropriate permissions for access to records in the ESDS. The following sections provide information on preparing user roles for this mapping procedure:  
  *Default User Roles and Role Groups*  
  and  
  *Managing User Roles and Rights*

1. Prepare the identity provider for the education organization as follows:
2. Verify that the organization's directory service supports SAML 2.0 for passive authentication services.
3. Install and configure a SAML 2.0 [*federation*](#bookmark118). The steps for doing this vary based on the directory service product being used. For example, in a Microsoft Active Directory environment, this may be accomplished with AD FS 2.0. Other SSO solutions that support SAML 2.0, such as SiteMinder (from CA) and OpenAM (from ForgeRock, formerly OpenSSO from Sun), can be used.
4. Confirm an IDP responds with a SAML assertion asserting who the authenticated user is. The response must be signed using a trusted X.509 certificate. The store of trusted certificates that the API has is based on the java 'cacerts' truststore. If the X.509 certificate used by an IDP is not contained within that truststore, then the X.509 certificate must be provided to ESDS. If this is not completed, then all SAML responses provided by the IDP will not be trusted, and no user will be able to authenticate. If a certificate must be provided to ESDS, it might also be necessary to provide intermediate certificates, so that trust can be established between the ESDS API and the X.509 certificate signing the SAML assertion.
5. Verify that the Issuer element of the IDP's SAML assertions is a URL that uniquely identifies the IDP. It should match the IDP URL of the Realm configuration.
6. The X.509 signing certificate may be a wildcard. The wildcard pattern must match the hostname of the SAML Issuer.
7. Ensure that the IDP can be configured to respond to ESDS API's authentication request using SAML Artifact Binding protocol.
8. Confirm the ESDS API supports sending a client certificate with its Artifact Resolution requests to the IDP. If the IDP server wants to verify the ESDS API's certificate, it should be obtained from the ESDS Operator and added to the IDP server's truststore.
9. Verify that the IDP can be set up to send encrypted SAML Assertions to the ESDS API. The ESDS API accepts both encrypted and unencrypted SAML Assertions for both Artifact Binding and POST protocols. The public key that can be used to encrypt assertions is available via the ESDS API's meta- data endpoint.
10. Configure the federation using the following steps:
11. Configure the organization's SAML 2.0 identity provider to enable trust from ESDS. You can use the SAML metadata endpoint to quickly configure the ESDS API as a service provider. The certificate used by ESDS API to digitally sign the artifact resolution request can be obtained from this endpoint. The endpoint can be found at:  
    [$BASE\_URL/api/rest/saml/metadata](file:///C:\Users\Laura%20Heintz\Desktop\AdminGuide\$BASE_URL\api\rest\saml\metadata).
12. Configure the organization's SAML 2.0 identity provider to include the following required user attributes in each SAML assertion response:

* userName - the user's full name
* userId - the unique identifier for that user within the organization
* roles - The roles or group memberships that user has within the organization
* userType (optional) - Staff, Teacher, Student

**Note:**If userType is not passed in the SAML, then the userId is treated as a staffUniqueStateId. This means the user is mapped to Staff through the staffUniqueStateId or to the Teacher through the Teacher's staffUniqueStateId.

1. Using the Manage Realm system tool in the SDS to configure a realm that requires the organization's SAML 2.0 identity provider and endpoint URL.
2. Map the user roles from the education organization to corresponding roles in the SDS. For information and instructions on how to create and maintain these mappings, refer to the chapter *Managing User Roles and Rights*.
3. Prepare for data ingestion by mapping the entities in the ESDS map to attributes within the organization's directory record. This step may take some time to ensure no data is overlooked. Refer to the chapter *Mapping Directory Entities to the ESDS* for the information required to perform this task.

Once these steps are completed, any users whose records are ingested by the data ingestion process, *and* whose user roles in the organization are mapped to roles in ESDS, can authenticate successfully with ESDS applications and services and access data from the ESDS to which they have permission.

## Configure ESDS Technology to work with your IDP

Before you can continue, you must set up your realm in the ESDS system. Refer to the chapter *Creating and Managing Realms* and *Managing User Roles and Rights* for the specific procedures.

## Integrated ESDS Application Authentication

ESDS applications that use the ESDS REST API have an additional authentication layer associated with accessing the ESDS using the REST API. This section provides details to integrate authentication for ESDS applications.

Applications that use the ESDS REST API must use OAuth 2.0. ESDS uses OAuth 2.0 to authenticate the application itself, and the ESDS REST API, specifically, uses OAuth 2.0 to authenticate a user for the application.

Applications using the ESDS REST API authenticate users and authorize each request as follows:

1. When a user signs in to an application using the ESDS REST API, authentication first occurs as described in the previous section *Federated Authentication Strategy*. Included in the SAML request to the education organization, the ESDS identity service includes the ForceAuth attribute. This attribute determines whether the user must authenticate even if there is an existing session with the organization's identity provider. ESDS sets this value to trueif the user does not have an existing ESDS session.
2. After successful authentication, the education organization's identity provider returns the required attributes in its SAML assertion: user name, UID, and roles.
3. The ESDS application receives the SAML assertion and sets up a session for the user. *This session is for one user to one application only.* The education organization's identity provider controls the session timeout value and whether the user has to authenticate to create additional sessions before the first times out.
4. The application calls upon the ESDS OAuth provider to generate an OAuth 2.0 authorization token.
5. The application provides the user's authorization token in HTTP headers when forming calls with the ESDS REST API.
6. Each time a user opens another ESDS application while still signed in to the first, ESDS forms a new SAML request to the education organization. These additional SAML requests, though, have the ForceAuth attribute set to false. The education organization’s identity provider asserts its session control to determine whether that user needs to authenticate again.

### Federated Application Authentication: Sample SAML Exchange

The metadata URL can be used to retrieve the SAML metadata for the ESDS service provider. An organization's identity provider can be configured using this URL, calling the URL at some interval to check whether meta- data about the ESDS service provider has changed. The URL syntax is $BASE\_URL$/api/rest/saml/metadata. The following is an example of a response from this URL:

EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:ds="http:// [www.w3.org/2000/09/xmldsig#"](http://www.w3.org/2000/09/xmldsig) entityID="$ISSUER\_NAME$">

<SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">

<KeyDescriptor use="signing">

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>$CERTIFICATE\_TEXT$</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

<KeyDescriptor use="encryption">

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>

$ENCRYPTION\_CERTIFICATE\_TEXT$

</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</KeyDescriptor>

<AssertionConsumerService index="0" Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP- Artifact" Location="$BASE\_URL$/api/rest/saml/sso/artifact" />

<AssertionConsumerService isDefault="true" index="1"

Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="$BASE\_URL$/api/rest/saml/ sso/post" />

</SPSSODescriptor>

</EntityDescriptor>

The following is an example AuthN request sent by the ESDS service provider to initiate and authentication with an organization's identity provider:

<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" Destination="

ForceAuthn="false" ID="sli-b931632c-9706-4c8e-bd27-0a0a1f8dc408" IsPassive="false" IssueInstant="2012-03-27T17:56:55.486Z" ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Version="2.0">

<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://nxbuild.example.com

</saml:Issuer>

<samlp:NameIDPolicy AllowCreate="true" Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient" SPNameQualifier=" />

<samlp:RequestedAuthnContext Comparison="exact">

<saml:AuthnContextClassRef xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"> urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport

</saml:AuthnContextClassRef>

</samlp:RequestedAuthnContext>

</samlp:AuthnRequest>

The request for artifact binding is different. The following is an example of an artifact binding request sent by the ESDS service provider to initiate and authentication with an organization's identity provider:

[<soap11:Envelope xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/">](http://schemas.xmlsoap.org/soap/envelope/)

<soap11:Body>

<saml2p:ArtifactResolve

Destination="<base url>/idp/profile/SAML2/SOAP/ArtifactResolution" ID="<unique id>"

IssueInstant="2013-11-22T15:28:05.061Z" Version="2.0" xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol">

<saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"><base url>:8080</saml2:Issuer>

[<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">](http://www.w3.org/2000/09/xmldsig)

<ds:SignedInfo>

[<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-](http://www.w3.org/2001/10/xml-exc-)

c14n#"/> sha1"/>

[<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-](http://www.w3.org/2000/09/xmldsig#rsa-)

<ds:Reference URI="<reference URI>">

<ds:Transforms>

[<ds:Transform Algorithm="http://www.w3.org/2000/09/](http://www.w3.org/2000/09/) xmldsig#enveloped-signature"/>

[<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/](http://www.w3.org/2001/10/xml-exc-c14n)

>

</ds:Transforms>

[<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>](http://www.w3.org/2000/09/xmldsig#sha1)

<ds:DigestValue>Digest Value/ds:DigestValue>

</ds:Reference>

</ds:SignedInfo>

<ds:SignatureValue>Signature Value</ds:SignatureValue>

<ds:KeyInfo>

<ds:X509Data>

<ds:X509Certificate>$CERTIFICATE\_TEXT$</ds:X509Certificate>

</ds:X509Data>

</ds:KeyInfo>

</ds:Signature>

<saml2p:Artifact>Artifact ID</saml2p:Artifact>

</saml2p:ArtifactResolve>

</soap11:Body>

</soap11:Envelope>

The following is an example SAML assertion (response) sent to the ESDS service provider. This example shows the attributes that ESDS requires and how those attributes appear as name-value pairs in SAML. Note that the roles attribute (<Attribute Name="roles">) can have multiple values as shown in the example:

samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion" Destination="<base url>/api/

rest/saml/sso/post" ID="<unique id>" InResponseTo="<id of initial saml assertion>" IssueInstant="2013-03-20T19:19:36Z" Version="2.0">

<saml:Issuer>Issuer URL</saml:Issuer>

[<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">](http://www.w3.org/2000/09/xmldsig)

<SignedInfo>

[<CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-](http://www.w3.org/TR/2001/REC-xml-) c14n-20010315#WithComments"/>

[<SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>](http://www.w3.org/2000/09/xmldsig#rsa-sha1)

<Reference URI="">

<Transforms>

[<Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>](http://www.w3.org/2000/09/xmldsig#enveloped-signature)

</Transforms>

[<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>](http://www.w3.org/2000/09/xmldsig#sha1)

<DigestValue>Digest Value</DigestValue>

</Reference>

<SignatureValue>Signature Value</SignatureValue>

<KeyInfo>

<X509Data>

<X509Certificate>X509 Certificate</X509Certificate>

</KeyInfo>

</Signature>

<samlp:Status>

<samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>

<saml:Assertion ID="<unique id>" IssueInstant="2013-03-20T19:19:36Z" Version="2.0">

<saml:Issuer>Issuer URL</saml:Issuer>

<saml:Subject>

<saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent">Saml Name ID</ saml:NameID>

<saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">

<saml:SubjectConfirmationData InResponseTo="<id of initial saml assertion>" NotOnOrAfter="2013-03-20T19:29:36Z" Recipient="<base url>/api/rest/saml/sso/post"/>

</saml:SubjectConfirmation>

</saml:Subject>

<saml:AttributeStatement>

<saml:Attribute Name="userId">

[<saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"](http://www.w3.org/2001/XMLSchema) xmlns:xsi="http:// [www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:type="xs:string">user id</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute Name="sn">

[<saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"](http://www.w3.org/2001/XMLSchema) xmlns:xsi="http:// [www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:type="xs:string">user last name</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute Name="userName">

[<saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"](http://www.w3.org/2001/XMLSchema) xmlns:xsi="http:// [www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:type="xs:string">user full name</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute Name="givenName">

[<saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"](http://www.w3.org/2001/XMLSchema) xmlns:xsi="http:// [www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:type="xs:string">user first name</saml:AttributeValue>

</saml:Attribute>

<saml:Attribute Name="userType">

[<saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"](http://www.w3.org/2001/XMLSchema) xmlns:xsi="http:// [www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:type="xs:string">one of: {staff, student, parent}</ saml:AttributeValue>

</saml:Attribute>

<saml:Attribute Name="roles">

[<saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"](http://www.w3.org/2001/XMLSchema) xmlns:xsi="http:// [www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:type="xs:string">role(s) for user</ saml:AttributeValue>

</saml:Attribute>

</saml:AttributeStatement>

>

### Federated Application Authentication: Installed (Non-Web) Applications

The information provided assumes that the applications that use ESDS are primarily web applications running simultaneously within the same web browser. However, ESDS is not limited to web applications. Software developers can also create mobile and embedded applications that use ESDS. Since these applications require installing some amount of software outside of the ESDS, ESDS refers to these as [*installed applications*](#bookmark124).

**Note:** At this time, mobile and embedded applications must use the same authentication strategy as web applications.

### Federated Application Authentication: Adding New Applications

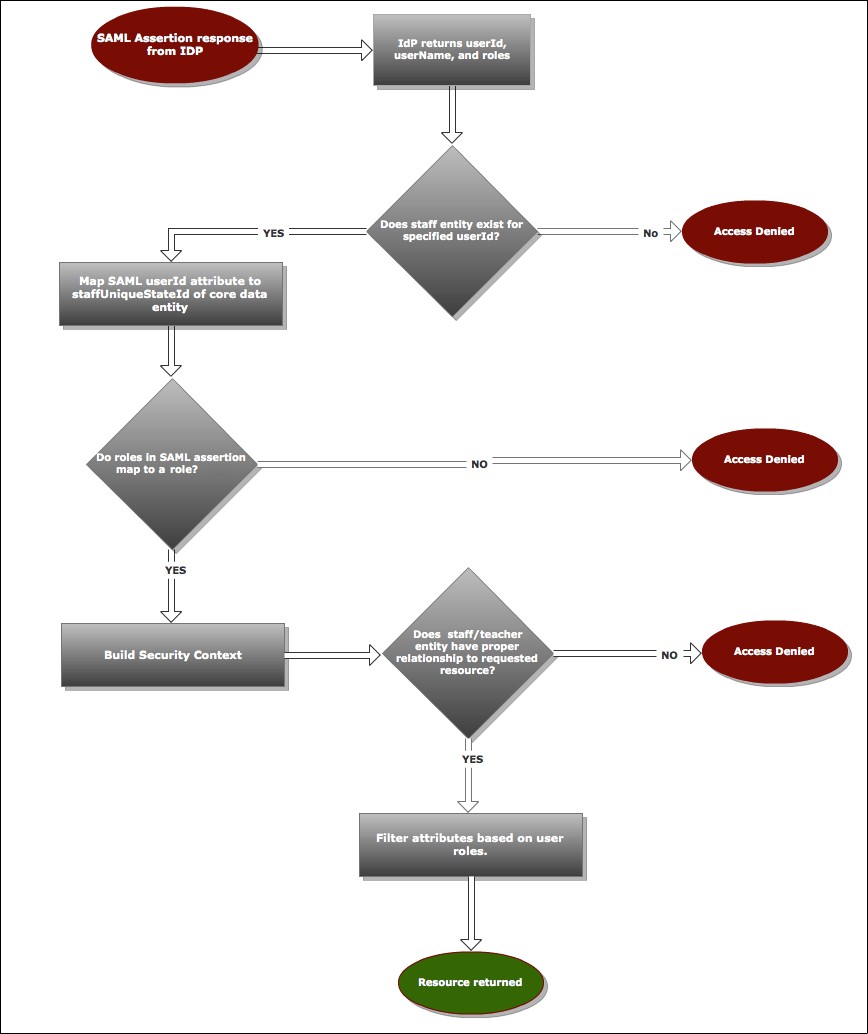
Administrators can add (register) new applications with ESDS. This is necessary when a third party (not ESDS) has developed an application that interacts with the ESDS using the REST API or Java Client Library. As part of registration, ESDS administrators for an education organization must authorize the application to access its data in the ESDS. Successfully registered applications should be able to use the same integrated authentication model described earlier in this Guide.

## Effective SDS Permissions for Federated Authentication

Authorization to protected data, represented by resources in the ESDS REST API, is a multi-step process using ESDS. It starts with determining if any of the roles provided in the SAML assertion correspond to ESDS roles as a result of [*role mapping*](#bookmark141). If there is no match, then no authorization is granted. If there is a match, the ESDS roles determine the basic user permissions and data access.

However, authorization does not stop at the role level. ESDS uses the UID attribute (passed in the SAML assertion) to find the entity within the ESDS corresponding to that user. From there, associations in the data model point to exactly which data the user has permission to read or write. For more information refer to the chapter *Contextual Authorization Rules*.

The following decision tree represents how ESDS calculates these effective permissions to the SDS contents:



## Mapping Directory Entities to the ESDS

The ESDS is built from data provided by an education organization. The preparations and procedures for the original ingestion of data to populate the SDS are detailed in the *Data Ingestion Guide*. Refer to the information presented in that document for specific ingestion procedures.

However, before ESDS can ingest the data, the entities within that data must be mapped to entities in the ESDS. You can use the Ingestion Validation Tool to verify the validity of your data before ingestion. For complete documentation on ESDS entities, attributes, and data types, including a link to the XSD describing the ESDS logical model, refer to the developer documentation on the ESDS website.

# Roles and Permissions

ESDS uses roles and permissions to identify users and applications in order to control access to data.

## Educational Organization Hierarchy

ESDS recognizes an organizational hierarchy consisting of schools, districts, and states as shown in the diagram below. It uses these assumptions to determine data access and permission management.

##### Education Organization Hierarchy



ESDS considers the [*school district*](#bookmark142) as the ultimate arbiter of managing who is able to view or manipulate student data. ESDS assumes that each school belongs to exactly one district and that each district belongs to exactly one state. Data storage and access control for a given student are directly associated with that student's school and, in turn, with that school's district. Default permissions and roles follow this assumption.

While the ESDS is built to facilitate district control over data access, ESDS recognizes that districts may have contractual or regulatory arrangements that enable states to view and administer data and permissions on their behalf. For that reason, ESDS provides for the creation of custom roles if the default roles do not provide adequate permission combinations.

## Default User Roles and Role Groups

ESDS includes a default set of user roles and role groups and permissions associated with them. The roles must map directly to the roles in the education organization’s IDP. If they do not map directly to the IDP roles and permissions, use the Custom Role Mapping Tool as described in the chapter *Managing User Roles and* [*Rights*](#bookmark53). This tool is designed to edit the permissions for the default user roles as well as to create custom roles.

The following table lists the default user roles and role groups along with the default permissions assigned to each. Please note that this information may not apply if your ESDS deployment has modified the default permissions or created custom roles:

##### Default Role Groups, Roles, and Rights

| **Role Group** | | **Roles** | **Context** | **Permissions** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Public Data** | **General Data** | **Restricted Data**  **Student Owned Data** | | | **Student General Data** | | | **Student Restricted Data** |
| Leader | | Leader Principal  Superintendent | Staff | Read | Read, Write | Read none | | | none | | | none |
| Users in the default [*Leader*](#bookmark128) role group can view data related only to those educational organizations to which they are associated. | | | | | | | | | | |
| **Educator** | | Educator  Teacher | Teacher | Read | Read, Write | none none | | | none | | | none |
| Users in the default [*Educator*](#bookmark113) role group can view data only associated with students actively enrolled in current class sections. | | | | | | | | | | |
| **IT Administrator** | IT Administrator  LEA Administrator  School Administrative Support Staff  LEA System Administrator | Staff | Read, Write | Read, Write | | Read Write | | | None | | | none |
| Users in the default [*IT Administrator*](#bookmark125) role group have all the permissions of users with a Leader, plus the abilities to create new users, define custom permissions, and view administrative applications. Additionally, they possess the right to view security events, and to authorize user applications.  The Admin role is on for this group. | | | | | | | | | | |
| **Student** | Student | not applicable | Read | none | | none none | | | Read | | | none | |
| Users in the default [*Student*](#bookmark150) role group can only view public data, Student\_Owned data, and Student\_General data. | | | | | | | | | | |
| **Parent** | Parent | not applicable | Read | none | | | none none | | | Read | |  |
| Users in the default [*Parent*](#bookmark131) role group can only view public data, Student\_Owned data, and Student\_General data. | | | | | | | | | | |
| **Aggregate Viewer** | Aggregate Viewer  Specialist/  Consultant | Staff | Read | none | | | none none | | | none | | none |
| (Reserved for future implementation) Users in the default [*Aggregated Viewer*](#bookmark97) role group can only view public and aggregate data.  The Admin role is on for this group. | | | | | | | | | | |

**Note:** The *isAdminRole* flag can be used by application developers to determine if a user has Admin rights, such as access to administrative applications. By default, the IT Administrator role is the only role with the *isAdminRole* flag marked as true. A user with multiple roles assigned is only considered an administrative user if *all* of those roles have the *isAdminRole* marked as true.

ESDS defines some additional roles for administrators. Collectively, these are referred to as the *ESDS administrator* roles, and they include users who are responsible for the back-end operation of an ESDS deployment.

## How ESDS Determines Permissions

ESDS controls access to an organization's data by calculating [*effective permissions*](#bookmark114). Effective permissions define what the user or application is allowed to do with data. ESDS determines [*effective permissions*](#bookmark114) based on a combination of the following criteria:

**Permissions by role.** Role refers to a single set of associated permissions for a user within the system.

These permissions determine if the user has access to read and/or write specific categories of data in the SDS. However, access to data may be further restricted based on a user's context when attempting to work with ESDS.

*Example:* A school principal given a Leader role may have access to all data for all students within that principal's school database.

**Permissions by education organization.** ESDS uses the role defined in a user's staffEdOrgAs-signmentAssociation value in the data store to determine the roles to apply based on the education organization. This allows a user to have different roles at different education organizations.

*Example:* A principal for School A also teaches at School B. When accessing the data for School A, the user is acting as Principal. When accessing data for School B, the user is acting as Teacher.

**Permissions by context.** The [*context*](#bookmark106) in which a user can access data is a combination of the following criteria:

* *User roles.* This is determined from the entity associated with that user in the data store. Every successfully authenticated education organization user (through federation) is associated with one of the following entities in the data store: Student, Staff, Parent, Teacher. For those with Student and Parent, the entity itself serves as the context for the user. For those with Staff and Teacher, the user's context is defined by the user role. Each staff role in the ESDS system should be assigned a context right of Staff, Teacher, or both. The chapter *Managing User Roles and Rights* covers how to customize the user role structure and map roles to those defined by the education organization identity provider.
* *User relationship to data.* This is determined by the user's relationship to the data through education organization boundaries. For example, a teacher can edit a student's data for her class, but the teacher can only view that student's data for other classes within the education organization.

Contextual authorization rules covered in the chapter *Contextual Authorization Rules* control a user's permissions to data. An example of how this is applied: A school counselor has access to a student record because he has a role with Staff context in the same education organization as the student. His role and associated permissions determine whether he can read or write all, none, or some data to the student’s record. However, it is possible for a user to have context with an entity, but have no permissions to see any of the data on that entity.

ESDS enforces access control at the application level as well. This means a user's access to data could be limited by the permissions for the application used to access data. This is covered in greater detail in the chapter *Managing Applications*.

### Default Permissions

The basic permissions to data in the ESDS are:

* *No access*: the user does not have the ability to access the data.
* *Read access*: the user can view the data, but cannot edit it.
* *Read-write access*: the user can view and edit the data.

As described in the chapter *ESDS Roles and Permissions*, ESDS controls access to an organization's data by calculating [*effective permissions*](#bookmark114). ESDS has created categories for data to make it easier to assign permissions:

**Public** – This [data](#bookmark133) can be viewed by any user with the proper role-based permissions. For example, a school's main phone number is Public data.

**Restricted** – This data can only be viewed by users who have the exact role and permissions that the data requires for access. This information typically is kept at a high level of confidentiality. For example, a student's eligibility for free or reduced price lunch is considered Restricted data.

**General** - By default, all data not defined as Public or Restricted data is considered to be [General data](#bookmark119).

Each role that ESDS recognizes identifies permissions using the categories described in the section *Default User Roles and Role Groups.* For more information about the default roles and how to create custom roles, refer to the chapter *Managing User Roles and Rights*.

Both application and user permissions are assigned during their respective [user registration processes](#bookmark153). User registration is managed by the IT administrator for the ESDS deployment.

### Student Data Permissions

The following permissions apply to access to student data:

* **Student\_General** - Data that can be viewed by a student with any context.
* **Student\_Owned** - Data that can be viewed by a student through self-context.
* **Student\_Restricted** - Data that should not be viewed by a student regardless of context.

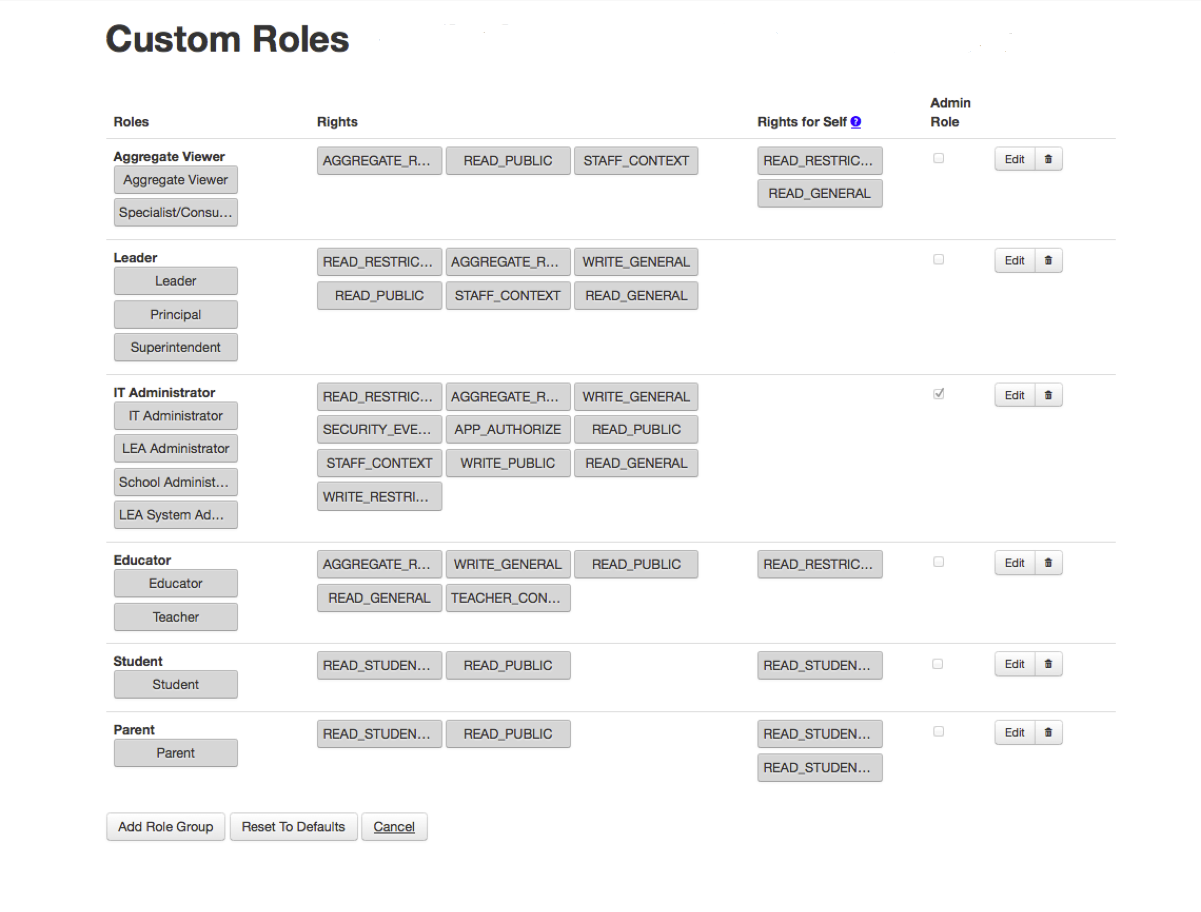
**Note:** These roles are only assigned to the Student and Parent roles by default. They can be assigned to other roles using the Custom Roles tool defined in the chapter *Managing User Roles and Rights*.

# Managing User Roles and Rights

ESDS assigns user roles and rights based on the roles configured in the ESDS. You should ensure that the roles configured in the SDS align with those associated with your education organization. In Production, those user roles are managed by the IDP for the [*education organization*](#bookmark112).

You can create, edit, and delete roles and role groups using the Create Custom Roles tool. For each role group, you can assign one or more roles and select one or more rights to associate with that group.

ESDS provides several default role groups. However, if the roles in your education organization's IDP do not match the roles and rights contained in these role groups, you will need to map new or existing roles from this tool to the roles that IDP.

The following graphic shows the Custom Roles tool with designated roles/rights.

Below is a description of each interface feature shown on the **Custom Roles** tool:

* **Group Name** - Name of the role group to which one or more roles belong. Rights are assigned by group, rather than by role, and one or more roles can be managed under a single group. The default groups are Aggregate Viewer, Leader, IT Administrator, and Educator.
* **Roles** – Names of the roles associated with the group. These should map to the roles within your education organization's identity provider. The default roles are Aggregate Viewer, Leader, IT Administrator, Educator and Student.  
  **Note:** Each role name must be unique across all role groups. In other words, you cannot have two roles with the same name, and the same role cannot occur in two or more role groups.
* **Rights** – Permissions that a group has been granted. Permissions relate to the ability to read and write individual data categories as outlined in the section *Default Permissions*. You can modify the rights for a role group at any time. Note that staff roles also require a context right. That means that each staff role should have one or both of the STAFF\_CONTEXT and TEACHER\_CONTEXT rights. Context rights are not applicable to students and parents.
* **Rights for Self** - This section allows you to assign different rights to a user for Self than they would have for other entities for which they have context. As an example, a teacher can have access to their own private data without also requiring the permission to see the private data for other teachers to which they have context.
* **Admin Role** – This is a role, also referred to as the *isAdminRole* flag, that should only be applied to groups that administer the system. By default, this role belongs to the IT Administrator. A user with multiple roles assigned is only considered an administrative user if all of those roles have the Admin Role selected.
* **Edit/Delete** – This section displays the edit and delete buttons. When you click Edit, the group becomes editable. When you click Delete, the entire row is deleted.
* **Add Role Group** – Select this button to add a new role group and extend the roles beyond the four default roles that come standard in the ESDS. When you click this button, an editable row appears at the bottom of the table.
* **Reset Defaults** – Select this button to reset all roles/rights to the system defaults.
* **Cancel** – Select this button to close the Custom Roles tool.

## Adding a New Role Group

If you want to extend the types of roles beyond ESDS default roles, you must create a new role group. To add a new role group:

1. Log in to ESDS.
2. If you are logging into a Sandbox, click **Administer my Sandbox**.  
   OR  
   If you are logging into Production, click **Admin**.
3. Click **Create Custom Roles** under System Tools.
4. Click **Add Role Group** at the bottom of the page. An editable row appears at the bottom of the table.
5. Enter the **Group Name**. The role group should have a logical name for the roles within the group. For example, School Board Personnel.
6. Enter **New Role Name** for the role that you want to add for the group. When you are satisfied with the name, click the **Add** button.  
   Once added, the role will appear as a row in the Custom Roles tool. You can add multiple roles for a single role group. For example, Superintendent and School Board Member.
7. Select a right to give to the role group.
8. Click Add Right. Once added, the right will appear below the Right Name drop-down menu.
9. You can grant multiple rights to a single role group.
10. If you want the members of the group to have administrative rights, check the **Admin Role** box.
11. When finished, click **Save** to create the role group. The role group appears at the bottom of the table.

## Mapping Multiple Local IDP Roles to One ESDS Role

It is possible to map multiple roles from your local IDP to a single ESDS role. In this case, the user only receives the rights that are included in all of the roles from the local IDP. This prevents users from viewing unauthorized data. For example, if you are a principal in school A and a teacher in school B, you will only have the rights associated with the teacher for both school A and school B.

## Managing Custom Roles for Multiple Realms

Education organizations can manage custom roles independently for multiple realms. If you are administering multiple realms, clicking **Create Custom Roles** directs you to list of the available realms. To manage the custom roles for a particular realm, click the **Custom Roles** button to modify the roles for that realm. After accessing the **Custom Roles** page for the specific realms, modify the custom roles as you would in a single realm environment.

# Contextual Authorization Rules

Context refers to the combined conditions under which a user or application exists relative to the data in the ESDS. This starts with user roles and permissions and extends to other conditions that affect context. This includes the education organization to which a user is assigned, whether an application was granted permission to that education organization, and the date and time a user is attempting to access data.

The ESDS system determines a user's access to data based on that context. A user who has effective permission to access data, with all of these conditions in place, may be said to have "context with" that data. The permissions themselves may be referred to as "contextual access."

Contextual authorization is the set of rules that ESDS uses to determine a user's contextual access. The rules rely on the following user characteristics:

* Each user is associated with a single and unique entity in the ESDS: Staff, Teacher, Parent, or Student.
* Each staff or teacher user has a context of Staff, Teacher, or both based on how the ESDS administrator has configured custom user roles and contexts for the system.

This chapter describes the authorization rules that govern the context-based authorization for the [*federated*](#bookmark116) users of an education organization (those configured in ESDS to authenticate by that organization's IDP) and enforced through the ESDS REST API. These rules are used in conjunction with the permissions associated with a user's role to determine access to data, or the user's [*effective permissions*](#bookmark114). This chapter builds on the previous section *How ESDS Determines Permissions*.

## Global Authorization Rules

While most of the business rules for contextual authorization relate to specific entities or contexts, the following rules apply to all users:

**RULE:** Any context resolution in the ESDS model starts with the context of the user's role in an education organization in the ESDS. Every successfully authenticated user (through federation) must be associated with at least one staff role in the data store in order to access data in ESDS.

**RULE:** [No user from a tenant](#bookmark152) should be able to access any data from another tenant. ***There is no exception to this rule.***

A tenant acts as a permissions boundary. Even if two districts (A and B) belong to the same State, if they are deployed as separate tenants, no users from A will be able to access B or vice versa. In addition, if the State is not in a tenant with any of the districts, state users will not be able to access the district's data.

**RULE:** Every authenticated user in the SDS can access the public information about any education organization that ex[ists in that user's (own) tenancy](#bookmark152).

Education organization information is public. For example, if School A and School B are schools in State S, and State S is on-boarded as a single tenant, then a teacher from School A can access the School B name, address, phone number and any other data considered public.

## Global Entities

Global entities are entities that every user in a tenant has access to within the context of that tenant.

**RULE:** Everyone in a tenant has access to global entities.

An assessment, such as an SAT, is an example of a global entity that is not tied to a specific education organization. Since assessments are considered to be global, they cannot be associated with a context. The following entities are considered to be global:

/assessments

/competencyLevelDescriptors

/courseOfferings

/courses

/educationOrganizations

/gradingPeriods

/graduationPlans

/learningObjectives

/learningStandards

/programs

/schools

/sections

/sessions

/studentCompetencyObjectives

**Note:** These entities are protected with the READ\_PUBLIC and WRITE\_PUBLIC rights.

## Orphan Entities

An Orphan Entity is an [*entity*](#bookmark115) that has no [*association*](#bookmark100) (yet) with other entities. For example, a Student entity was created, but it was not yet associated with an EdOrg.

**RULE:** The creator of an orphan entity has context with and can access that data entity.

Once the creator associates the entity with something else in the system, the creator's ability to view that entity goes back to the regular access rules.

**RULE:** You can only associate two entities if you have context with both entities. Therefore, it is not possible to create a student and then associate that student to a school or section that you do not have context with and cannot access.

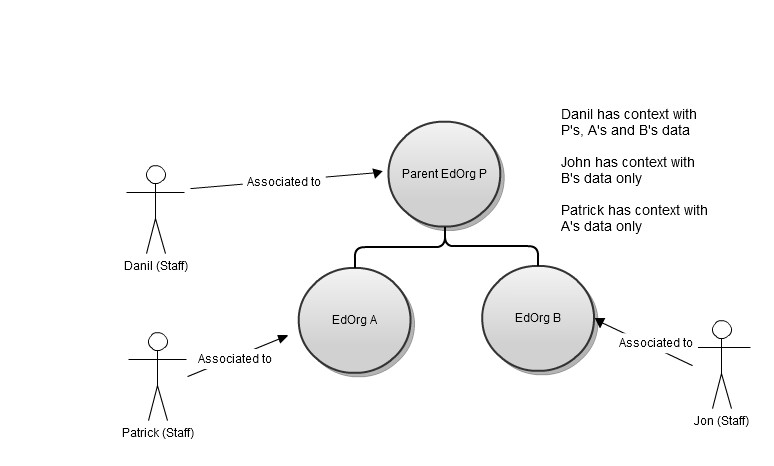
# Contextual Authorization: Staff

The rules defined in this chapter apply specifically to users that are associated with the Staff context configured for user roles.

## Staff Context Relies on Education Organization

**RULE:** If a Staff (user entity) is associated with a certain education organization, then the user has the context necessary to access all the data within or below that education organization within the education organization hierarchy.

Staff can access data within their own education organization and within any education organization that falls under that organization in the hierarchy. The diagram below demonstrates this context relationship between users and education organizations.

Staff and Education Organization Association

**RULE:** [If the Staff (user) has no current association](#bookmark100) with an EdOrg, the user is not authorized to access the data associated with that EdOrg, including the data about that user.

**Note:** This rule applies for both Staff and Teacher.

## Staff Context Expires When Association Ends

**RULE:** Staff A should not have access to Staff B or Teacher B if Staff B or Teacher B is no longer associated with Staff A's EdOrg.

Staff cannot access data for teachers or staff that are no longer associated with their EdOrg.

**RULE:** Staff should not have access to a Student that is not currently enrolled in a School within the Staff's EdOrg, but was enrolled in the past, unless the (system-level configurable) grace period is greater than the time the student is not enrolled at the school.

Staff cannot access data for students that are not enrolled in a school once the grace period for data access has expired. For example, if Jimmy Little was not enrolled in Daybreak High School for the last 3 months, and the grace period is set at 2 months, then the principal of Daybreak High School cannot access Jimmy Little’s data. However, if the grace period was set at 4 months, then the principal would still have access to Jimmy Little's data.

In an example of combined Staff rules, consider this scenario:

* James Does is Principal of Daybreak High School.
* Mark Smith is former Staff (grace period has expired).
* Matt Jones is a former Student (grace period has expired).

In this scenario, James Doe will not be able to access Mark (Staff) or Matt (Student) data.

## Staff Context with Student Data

**RULE:** Staff has context with all of the present data about current students in their EdOrg(s).

**RULE:** Staff has context to historical students’ data as it does to present data. Student historical data is defined as data about the student's past, such as former grades. The Staff rights to that data is based on the Staff rights to the student entity.

While Staff has context to all of the student historical data, the Staff user's rights specify access to the following historical data entities:

* Attendance
* CourseTranscript
* DisciplineAction
* DisciplineIncident
* Grade
* Parent
* ReportCard
* StudentAcademicRecord
* StudentAssessment
* StudentCohortAssociation
* StudentCompetency
* StudentDisciplineIncidentAssociation
* StudentGradebook
* StudentParentAssociation
* StudentProgramAssociation
* StudentSchoolAssociation
* StudentSectionAssociation

# Contextual Authorization: Teachers

The rules defined in this chapter apply specifically to users that are associated with the Teacher context configured for user roles.

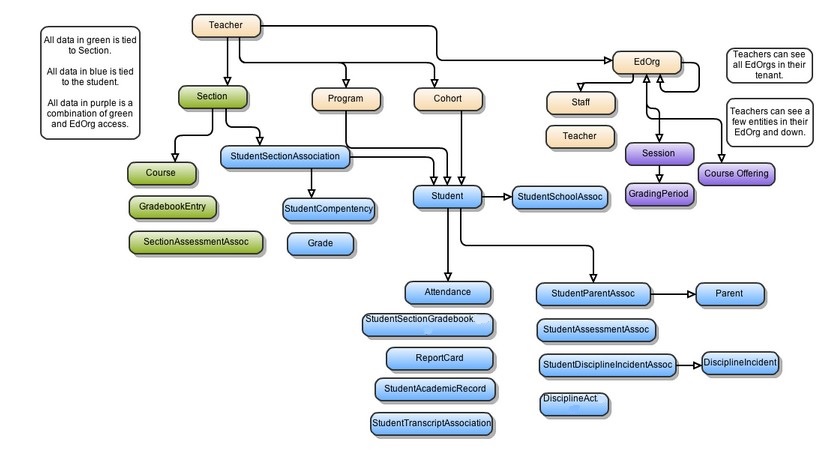
## Teacher Entity Must Have Certain Associations

Part of determining access to data includes confirming that a Teacher has a current TeacherSchoolAssociation. Since the TeacherSchoolAssociation does not have a BeginDate and EndDate, every Teacher must have a StaffEducationOrgAssignmentAssociation to determine if the association to the EdOrg is current. A Teacher without a StaffEducationOrgAssignmentAssociation is considered invalid data.

## Teacher Context Relies on Section, Cohort, or Program

**RULE:** A Teacher has a context to present Student data based on a current Section, Program, or Cohort to which the Teacher is associated (meaning that the EndDate of the association is not in the past).

A Teacher can access data based on current Section, Program, or Cohort associations. The following diagram details the entities involved.

Teacher Security Context Resolution

**RULE:** Teacher has context to historical student data in the same way as present data. Student historical data means current data about the student's past, such as past grades. This rule applies even if the data resides outside the Teacher's EdOrg hierarchy. The Teacher’s rights to that data are based on the rights to the student entity.

While the Teacher has context to this historical data, the Teacher user's rights specify access to the following historical data entities:

* Attendance
* CourseTranscript
* DisciplineAction
* DisciplineIncident
* Grade
* Parent
* ReportCard
* StudentAcademicRecord
* StudentAssessment
* StudentCohortAssociation
* StudentCompetency
* StudentDisciplineIncidentAssociation
* StudentGradebook
* StudentParentAssociation
* StudentProgramAssociation
* StudentSchoolAssociation
* StudentSectionAssociation

The historical data access rule is an exception to the general rule which states that the Teacher's context is within his/her own EdOrg hierarchy.

## Teacher Context Expires When Associations End

**RULE:** Teacher A does not have access to Staff B or Teacher B if Staff B or Teacher B is no longer associated with Teacher A's Ed Org.

Teacher cannot access data for teachers or staff that are no longer associated with his/her Ed Org.

**RULE:** A Teacher does not have access to a Student that is enrolled in the teacher's past Section, Cohort, or Program unless the (system-level configurable) grace period is greater than the time the Section, Cohort, or Program has ended.

The Teacher cannot access a student who was a past, but not current, part of a Section, Cohort, or Program with which Teacher is currently associated. Instead, a Teacher’s access to former students is suspended after a grace period expires. For example, if Linda Kim's sections ended 2 days ago (this means she does not currently teach that section) and the grace period is 10 days, then Linda Kim should still have access to her former student Jimmy Little for another 8 days.

# Contextual Authorization: Parents

The rules in this chapter apply specifically to users that are associated with the Parent entity in the SDS.

**RULE:** Parent has context to all global entities.

**RULE:** Parent does not have write context to any entities except:

* Parent (self)
* Student (Parent’s own child)

**RULE:** Parent has context to the student entity for parent’s own student.

**RULE:** Parent has context to attendances directly associated to a student to which the parent is directly associated.

**RULE:** Parent has context to each calendarDate in associated student’s school.

**RULE:** Parent has context to every cohort that is directly associated to a student to which the parent is directly associated.

**RULE:** Parent has context to every courseTranscript that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every gradebookEntry that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every grade that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every reportCard that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context with parent entity that represents themselves plus any other parent that is directly associated to a student to which the parent is directly associated.

**RULE:** Parent has context to every staff member that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every staffCohortAssociation that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every staffEducationOrgAssignmentAssociation that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every staffProgramAssociation that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every studentAcademicRecord that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every studentCohortAssociation that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every studentCompetency that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every studentGradebookEntry that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every studentParentAssociation that directly references a student to which the parent is directly associated.

**RULE:** Parent has context to every studentProgramAssociation that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to the Name field of a student viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every studentSchoolAssociation that directly references each student to which the parent is directly associated.

**RULE:** Parent has context to every studentSectionAssociation that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every teacherSchoolAssociation that is viewable by a student to which the parent is directly associated.

**RULE:** Parent has context to every teacherSectionAssociation that is viewable by a student to which the parent is directly associated.

**RULE:** Parent does not have context to view any discipline data.

# Contextual Authorization: Students

The rules in this chapter apply specifically to users that are associated with the Student entity in the SDS.

**RULE:** Student has context to all global entities.

**RULE:** Student does not have write context to any entities except:

* studentAssessments
* grades
* studentGradebookEntries
* student (self)

**RULE:** Student has context to his/her own student entity.

**RULE:** Student has context to attendance directly associated to the student, regardless of the date and EdOrg. Student does not need to be currently associated with an EdOrg in order to see own attendances from that EdOrg.

**RULE:** Student has context to each calendarDate in his/her school.

**RULE:** Student has context to every cohort to which they were ever associated.

**RULE:** Student has context to every courseTranscript associated to a StudentAcademicRecord that is directly associated to the student.

**RULE:** Student has context to every gradebookEntry that is directly associated to a section to which the student is associated.

**RULE:** Student has context to every grade that references a studentSectionAssociation for the student.

**RULE:** Student has context to every parent directly associated to him/her.

**RULE:** Student has context to every reportCard directly associated to him/her.

**RULE:** Student has context to every staff member associated with the current: EdOrg, Program, Cohort.

**RULE:** Student has context to every staffCohortAssociation of current cohorts.

**RULE:** Student has context to every staffEducationOrgAssignmentAssociation of current EdOrg.

**RULE:** Student has context to every staffProgramAssociations of his/her current programs.

**RULE:** Student has context to every studentAcademicRecord that is directly associated to the student.

**RULE:** Student has context to every studentCohortAssociation that is directly associated to the student plus every studentCohortAssociations of current cohorts.

**RULE:** Student has context to every studentCompetency that is directly associated with a studentSectionAssociation that directly associates to the student.

**RULE:** Student has context to every studentGradebookEntry that is directly associated to the student.

**RULE:** Student has context to every studentParentAssociation that directly references the student.

**RULE:** Student has context to every studentProgramAssociation that directly references the student plus every studentProgramAssociations of his/her current programs.

**RULE:** Student has context to the name field of other students who share a current section, program or cohort association.

**RULE:** Student has context to every student school association that directly references the student.

**RULE:** Student has context to every student section association that directly references the student plus every studentSectionAssociation of current sections.

**RULE:** Student has context to every teacherSchoolAssociation of current schools.

**RULE:** Student has context to every teacherSectionAssociation of current sections.

**RULE:** Student does not have context to view any discipline data.

# Administrative Options

## System Tools

System tools allow you to configure system settings. Access to system tools is determined by your role and permissions within the ESDS. ESDS provides the following tools to administer the ESDS system and applications. Not all of the tools listed below may be accessible to you.

* Delegate Administration: Allows you to assign permissions to the [Education Organization](#bookmark68).
* Manage Realm: Allows you to create and manage realms.
* Create Custom Roles: Permits you to manage user rights and roles.
* Manage Administrator Accounts: Permits you to setup and manage administrative users.
* Create Landing Zone: Provides for data ingestion into the ESDS.
* Authorize Applications: Allows you to manage applications.
* Change Password: Allows the ability to change the password for an account.

## Application Configuration

Application configuration options are provided depending on your role within the ESDS. You may be able to access configuration options for some approved applications. Not all applications allow configuration access. For more information, refer to *Managing Applications*.

By default, administrators have access to the ESDS Dashboard. For more information, refer to the section *Dashboard Administration*.

# Administrative Users

A *Tenant A[dministrator](#bookmark147)* is a user created in the [*ESDS hosted directory*](#bookmark148). Unlike other ESDS users, Tenant Administrators are responsible for setting up and maintaining an ESDS deployment. These administrative users perform the day-to-day functions of maintaining an Ed Org. This includes adding users, managing roles and permissions, and enabling and configuring applications.

**Notes:**

* Each ESDS deployment will have an initial administrative user account created. The ESDS Operator creates this when setting up a deployment.
* The ESDS Operator can only create Realm Administrators and Ingestion Users if the Ed Org is defined in the database for the tenant.

## Administrative User Roles

The types of ESDS administrator roles for a Production environment are defined in the table below:

| **Role** | **Responsibilities** | **Can Create** | **Created By** | **Accessible System Tools** |
| --- | --- | --- | --- | --- |
| ESDS Operator | Setting up and maintaining an ESDS technology deployment. *Note:* These users are not added to a [*tenant*](#bookmark152) or [*education*](#bookmark112)  *organi*[*zation*](#bookmark112). | Any type of administrative user. | ESDS Operator | Register Application, Account Approval, Manage Administrative Accounts |
| State Education Agency (SEA) Administrator (Tenant Administrator) | Administering state-level instances of ESDS. This includes creating new administrative users for delegated education organizations. | For the SEA:  SEA Administrators, LEA Administrators, Realm Administrators, and Ingestion Users. | ESDS Operator or other SEA Administrators within the same SEA | Authorize Application, Manage Administrator Accounts |
| Local Education Agency  (LEA) Administrator  (Tenant Administrator) | Administering local-level ESDS deployments, such as districts, counties, and cities. Typically, this includes creating new administrative users, and enabling and configuring applications for a district. | For the LEA:  LEA Administrators, Realm Administrators, and Ingestion Users. | ESDS Operator, SEA Administrators or other LEA Administrators within the same LEA | Delegate Administration, Authorize Application, Manage Administrator Accounts |
| Realm Administrator | Administering realms within ESDS. Realm Administrators will also map roles between ESDS and an education organization's identity provider. | Cannot create other administrative users. | ESDS Operator, SEA Administrators, or LEA Administrators | Create Custom Roles, Manage Realm |
| Ingestion User | Uploads data to a specified tenant. | Cannot create other administrative users. | ESDS Operator, SEA Administrators, or LEA Administrators | Create Landing Zone |

## Administrative Rights

Each user has different rights based on a role. The following tables indicate the rights for ESDS administrators and details the rights for each role.

Administrative Rights

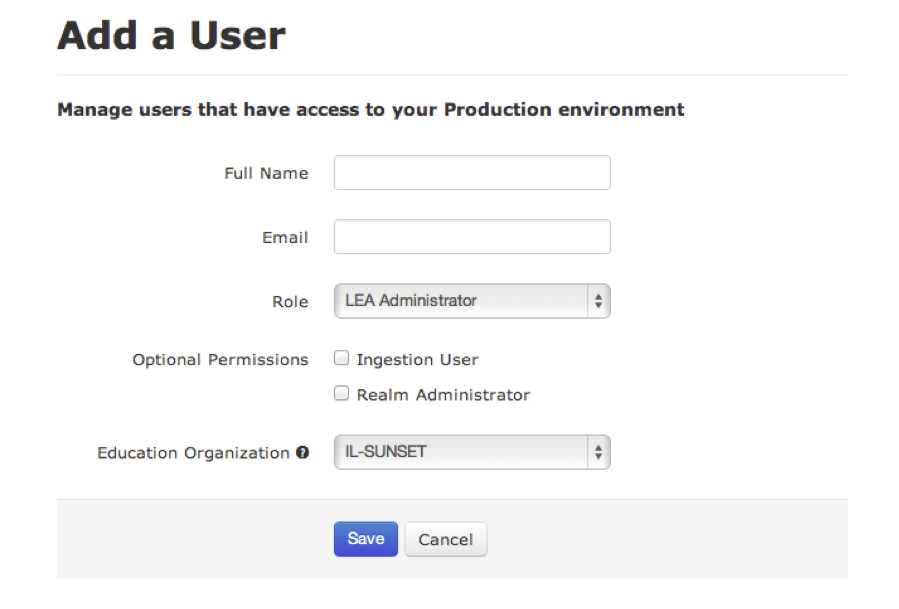
| **Right** | **Abilities** | **Resource** |
| --- | --- | --- |
| CRUD\_REALM | Create, delete, and update realms | RealmResource |
| CRUD\_ROLE | Create, delete, and update custom roles | CustomRoleResource |
| SLC\_APP\_APPROVE | Approve, reject, and unregister applications | ApplicationResource |
| EDORG\_APP\_AUTHZ | Update admin delegation for an LEA, authorize applications for an LEA | ApplicationAuthorizationResource, AdminDelegationResource |
| EDORG\_DELEGATE | View security events if delegated | ApplicationAuthorizationResource, SecurityEventResource |
| DEV\_APP\_CRUD | Create, delete, and update applications | ApplicationResource |
| CRUD\_SLC\_OPERATOR | Create ESDS Operator users | UserResource |
| CRUD\_SANDBOX\_ADMIN | Create Sandbox Admin users | UserResource |
| CRUD\_SEA\_ADMIN | Create SEA Admin users | UserResource |
| CRUD\_LEA\_ADMIN | Create LEA Admin users | UserResource |
| INGEST\_DATA | Create landing zones | UserResource |
| -- | View security events | SecurityEventResource |

Administrator Roles and Associated Rights

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Role** | **Name** | **IDP** | **Association** | **Rights** |
| ESDS Operator | ESDS Operator | ESDS | Global | SLC\_APP\_APPROVE, CRUD\_SLC\_OPERATOR, CRUD\_SEA\_ADMIN, CRUD\_LEA\_ADMIN,  View Security Events |
| SEA Administrator | SEA  Adminis- | ESDS | EdOrg (State), | EDORG\_DELEGATE, |
|  | Administrator | Tenant | CRUD\_SEA\_ADMIN, |
| CRUD\_LEA\_ADMIN,  View |
| View Security Events\* |
| LEA Administrator | LEA Administrator | ESDS | EdOrg (District), Tenant | EDORG\_APP\_AUTHZ, CRUD\_LEA\_ADMIN,  View Security Events |
| Sandbox Administrator | Sandbox Administrator | ESDS | Edorg, Tenant | CRUD\_SANDBOX\_ADMIN |
| Realm Administrator | Realm Admin | ESDS | Realm, Tenant | CRUD\_REALM, CRUD\_ROLE |
| Application Developer | application\_  developer | ESDS | Application, Tenant (for Sandbox) | DEV\_APP\_CRUD  CRUD\_ROLE  (Sandbox only) |
| Ingestion User | ingestion\_user | ESDS | Tenant, EdOrg | INGEST\_DATA |

## Creating a New Administrative User

The Manage Administrator Accounts system tool allows you to create new administrative users in the ESDS that have access to an ESDS deployment. To create a new administrative user:

1. Log in to ESDS using your administrator account.
2. Click **Admin**.
3. Click **Manage Administrator Accounts** under System Tools. The Manage Administrator Accounts page opens.
4. Scroll to the bottom of the page and click the **Add User** button. The **Add a User** page opens.
5. Fill out the form by entering data into the fields as follows:

*Full Name* - Enter the user's first name and last name.

*Email* - Enter the user's email address. All users that you create will receive emails that allow them to activate their accounts.

*Role* - Select a role for the user.   
**Note:**You can add ingestion and realm administration rights for users regardless of the user's primary role.

*Ingestion User* - Select whether or not to give this user the ability to ingest data to ESDS.

*Realm Administrator* - Select whether or not to give this user the ability to administer a realm.

*Tenant* - Add a tenant for the user, if applicable.   
**Note:**This field only appears for Operators that are using this tool and is not displayed in the image above.

*Education Organization* - Add an education organization for the user, if applicable. **Note:**When LEA administrators use this tool, this field will be restricted to their own education organization.

1. Click **Save**. You are returned to the **Manage Administrator Accounts** page and the user will appear in the table. The user will also receive an email that starts the account activation process.

## User Account Activation

Once you have created a new administrative user, that user must activate the account. The process for a user activating an account is detailed as follows:

1. The user receives an email containing a link to the account activation page. The user clicks the link to go to the account activation page.
2. The user enters his/her email address as the user ID. Once the user submits the form, an email is sent to the user that contains a link to the Reset Password page.
3. The user clicks the link in the email to go to the Reset Password page.
4. The user enters a password. Once the page is saved, the user's administrative account is activated. The user will receive a welcome email once the process is complete and the account is active.

# Data Ingestion

[*Data ingestion*](#bookmark109) is the process of uploading new and updated data to the ESDS. Ingestion consumes data from one or more source files placed in an FTP server location and adds the data to the ESDS. There may be applications designed to add, view, and edit one record at a time, but ingestion is designed to manage several dozen to several thousand data records at one time. Ingestion Users have access to the system tools necessary for data ingestion.

## How Data Ingestion is Accomplished

The ingestion process works as follow:

1. Files are prepared for ingestion and placed in a [*landing zone*](#bookmark126). The landing zone is an FTP server location where you can upload files to be ingested to the ESDS.
2. ESDS administrators can set up one or more of these landing zones to serve a single data store. For instructions on how to set up and manage landing zones, see the section *Creating and Managing Landing Zones*.
3. ESDS checks for files in the configured [*landing zone*](#bookmark126).
4. ESDS kicks off the data ingestion process, working through the files located in the landing zone.
5. When ingestion is complete, an ESDS Administrator can confirm that the newly ingested data is available in the ESDS. The ESDS Data Browser is an administration tool designed for this purpose. For more information on using the Data Browser, refer to the chapter *ESDS Data Browser*.

## Ingestion Users

An ESDS administrator does not need to be an Ingestion User to prepare data for ingestion or to use the ingestion validation and conversion tools. However, a user must have the Ingestion User role to ingest data into the ESDS. The Ingestion User creates landing zones and uploads files to any landing zone associated with their tenancy.

Any SEA, LEA, or Realm Administrator can be assigned the Ingestion User role, or an administrator can be designated as an Ingestion User only. For more information about user roles refer to the chapter *Administrative Users*.

An Ingestion User account cannot be moved between tenancies. The relationship between the account and the landing zone for a tenant persists, so an Ingestion User could inadvertently upload data for his new tenant to the old one. Therefore, if an Ingestion User moves to a new tenancy, the old account for the user should be deleted and a new account should be created.

## Creating and Managing Landing Zones

The landing zone is an FTP server location where you can upload files to be ingested into the ESDS. To create and manage landing zones, you need an ESDS administrator account with the [*Ingestion User*](#bookmark123) role.

Use the following steps to a create landing zone:

1. Log in to ESDS with your Ingestion User administrator account.
2. Click **Admin**.
3. Select **Create Landing Zone**.
4. Click **Create**.
5. After you click **Create**, you will receive an email with directions explaining how to ingest data using your new landing zone.

### Landing Zones for Sandbox Developers

Application developers who are using a Sandbox development environment will utilize a different procedure to create landing zones. For documentation about landing zones and data ingestion into a Sandbox environment, refer to the *Developer Documentation* on the ESDS website.

## Ingesting Data into the SDS

Data ingestion starts by making a secure file transfer to a landing zone. To accomplish this, you would upload files using an FTP client to the landing zone configured for the target tenancy. The results of the ingestion job display in messages in a series of log files which you can download from the landing zone.

The uploading and downloading is the easy part of ingestion. The more complex and time-consuming task is preparing the data for ingestion. Data must be provided in XML files compliant with the ESDS-Ed-Fi schema, which is the Ed-Fi Core 1.0.03 schema plus the ESDS extension schema. A control file must list important information about the XML files to be ingested and all the files must be bundled into a single .zip archive for uploading.

The ESDS Data Ingestion Guide provides all the details you need to prepare data for ingestion. It also includes additional information on how ingestion works and how to troubleshoot ingestion jobs based on the log file messages that display.

## Schools Interoperability Framework (SIF) Data

Schools Interoperability Framework (SIF) compliant data must be converted into an Ed-Fi compliant format before it can be ingested into the ESDS. [*CPSI*](http://www.cpsiltd.com/) has partnered with ESDS to provide the help you need converting SIF data. For more information, refer to the *CPSI* website.

# Onboarding Districts

Onboarding is the process of setting up a district to use the ESDS. The onboarding process involves setting up user accounts, ingesting district data into the ESDS, and (if applicable) enabling applications to access the data you have ingested.

Most districts will use the onboarding process described in the following section.

## Onboarding With Bulk Ingestion

Use the following steps for onboarding with bulk ingestion:

1. ESDS Operator creates an SEA Administrator using the Administrative Account Management Tool.
2. SEA Administrator creates a Realm Administrator and an Ingestion User using the **Manage Administrator Accounts** system tool.
3. Ingestion User prepares files for ingestion. For more information on this process, refer to the *ESDS Data Ingestion Guide*.
4. Ingestion User logs into ESDS and creates a landing zone. Refer to the section *Creating and Managing Landing Zones*.
5. Ingestion User ingests the data into the ESDS. The directions to perform this task is located in the email sent after a landing zone is created.
6. Realm Administrator creates a realm. Refer to *Creating and Managing Realms*.
7. SEA Administrator creates LEA Administrator in the local level Education Agency.
8. LEA administrator approves application created for the district to access data. Refer to the chapter Managing Applications for more information.

Once this process is complete, any staff that has an IT Administrator role with rights to ingestion data can log into the system and make API calls through any approved application.

## Onboarding Districts that use Schools Interoperability Framework Data

Districts that use Schools Interoperability Framework (SIF) data also onboard using bulk ingestion. However, SIF data must be converted into an Ed- Fi compliant format before it can be ingested to the ESDS. Districts that use SIF data should work with [*CPSI*](http://www.cpsiltd.com/) to format and ingest their data.

Use the following steps to onboard a district that uses SIF data:

**Note:** Contact [*CPSI*](http://www.cpsiltd.com/) for assistance in formatting and ingesting SIF data.

1. Create an SEA Administrator using the **Manage Administrator Accounts** system tool. Refer to the chapter *Administrative Users* for the procedure.
2. Create an Ingestion User account for CPSI using the **Administrative Account Management Tool**.
3. Work with CPSI to make sure the account has the proper credentials.
4. Create a Realm Administrator using the **Manage Administrator Accounts** system tool.
5. Create a landing zone and share the landing zone information with CPSI. Refer to the section *Creating and Managing Landing Zones*. CPSI will use the Ingestion User account and the landing zone to ingest data into the ESDS on behalf of the SIF district.
6. The Realm Administrator will create a realm using the procedure in the section *Creating and Managing Realms*.
7. SEA Administrator creates LEA Administrator in the local level Education Agency.
8. LEA Administrator approves application created for the district to access data. Refer to the chapter *Managing Applications*.

Once this process is complete, any staff that has an IT Administrator role with rights to ingestion data can log into the system and make API calls through any approved application.

## Onboarding with Minimal Ingestion

Onboarding with minimal ingestion is a process only used to resolve issues a district may have when trying to onboard via bulk ingestion.



Do not follow this process unless you are certain it applies to your district’s circumstances.

The following procedures allows you to onboard with minimal ingestion:

1. The ESDS Operator creates an SEA Administrator using the Administrative Account Management Tool.
2. The SEA Administrator creates a Realm Administrator and an Ingestion User using the **Manage Administrator Accounts** system tool.
3. The Ingestion User modifies provided XML files to match the setup for their district using the information in section *Setting Up a Template File for Onboarding with Minimal Ingestion*.
4. The Ingestion User updates the MainControlFile.ctl file.
5. The Ingestion User logs into ESDS and creates a landing zone according to the process detailed in the section *Creating and Managing Landing Zones*.
6. The Ingestion User ingests the data into the ESDS based on the directions in the email you receive after creating the landing zone.
7. The SEA Administrator creates the LEA Administrator user in the Local Level Education Agency.
8. The LEA administrator approves the application created for the district to access data.

### Setting Up a Template File for Onboarding With Minimal Ingestion

Before data can be ingested, the Ingestion User must download and set up the provided template files. The template files include the following:

* InterchangeEducationOrganization.xml - used to set up the hierarchy of the education organization, including the state and the local systems (district, county, etc.) that fall under the state.
* InterchangeStaffAssociation.xml - used to create the Staff entity for a local level education organization.
* MainControlFile.ctl - defines the set of inbound XML data to be ingested.

Use the following steps to set up the template files:

1. Get the template files from your ESDS Operator.
2. Open InterchangeEducationOrganization.xml and replace the data in the file with the data specific to your organization. For example, you would replace the \_ID of State Level Education Organization\_ with the ID for your state.  
   **Note:** The data you provide in the XML file must exactly match any data for your state and district that is already contained in the SDS. If there are conflicts, the data in the XML file will not ingest properly.
3. Once the file has been updated, be sure to save all of the changes.
4. Open InterchangeStaffAssociation.xml and replace the data in the file with the data specific to the staff in your district.
5. Once the file has been updated, be sure to save all of the changes.
6. Create a zip file that contains the updated InterchangeEducationOrganization.xml, InterchangeStaffAssociation.xml, and MainControlFile.ctl files. For proper ingestion, this zip file must not contain any folders and should only contain those three files.  
   **Note:** You can download and use the [**Offline Validation Tool**](https://github.com/inbloom/ingestion-validation) to validate your zip file. For more information about this tool, refer to the chapter *Using the Offline Validation Tool*.

# Using the Offline Validation Tool

When you ingest data files into the SDS, the ingestion process includes validating your data files. 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 ([*https://github.com/ESDS*](https://github.com/inBloom)).

This chapter explains 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 which is at a minimum double the file size of the XML file that you are validating
* Write permission to the home directory of the tool

## Using the Offline Validation Tool

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

1. In your web browser, open the ESDS Github repository: [*https://github.com/ESDS*](https://github.com/inBloom)
2. Click the following sequence to navigate and download the OVT:

* ingestion-validation
* bin
* OfflineValidationTool.zip
* View Raw

Your web browser should begin downloading the file OfflineValidationTool.zip. If necessary, specify where you want to save the file on your local computer.

1. 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 files:

* 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 below as $Zip- FilePath).

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 Applica- tions/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 below as $Zip- FilePath).

./OfflineValidationTool.sh $ZipFilePath

## View the Results

Output from the OVT consists of the following:

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

# Managing Applications

There are several web applications installed during an ESDS deployment. These applications are developed to use the ESDS. Additional web and mobile applications can be added at any time. However, an application must be registered with your ESDS deployment before it can use the services in that deployment.

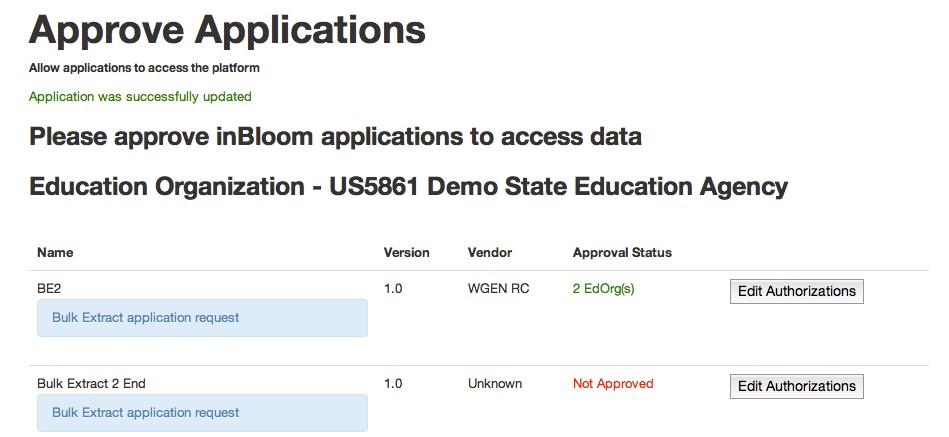
Applications in the SDS are managed by an ESDS Operator as instructed in the *ESDS Runbook*. However, Tenant Administrators must still enable approved applications for use by any educational organization.

## Enabling or Disabling Applications for Education Organizations

The **Authorize Application** system tool allows an ESDS *Tenant* [*Administrator*](#bookmark127) to authorize use of an application by any Ed Org in its education organization hierarchy.

User permissions take precedence in applications. Application users are only able to access Ed Org data in accordance with their user permissions. Because of this, each application must be compliant with the ESDS [*Terms of Use*](http://www.inbloom.org/terms-of-use) and [*Privacy and Information Security Policy*](http://www.inbloom.org/privacy-policy).

Use the following steps to enable or disable applications:

1. Log in to ESDS using your SEA or LEA Administrator account.
2. Go to the Admin URL.
3. Click **Authorize Application** under **System Tools**. The **Approve Applications** page will open as follows:

If you do not see an application that you want to enable in the list of applications, contact your ESDS Operator to have the application registered and approved.

1. To edit authorizations for an application that is listed, click the **Edit Authorizations** button. The **Application Authorization Dashboard** page opens.
2. Use the tree control to select the individual Ed Org(s) that can use the application.
3. Then, click **Save & Update**. The **Approval Status** will be updated on the **Approve Applications** page to reflect the number of education organizations authorized.

# Setting Up the Dashboard

The ESDS Dashboard displays student data for authorized users, such as educators and school, district, and state leaders. Once data is ingested into ESDS, it appears in one of two places on the Dashboard:

* ***Section Profiles***display a listing of each student in a class or section as well as relevant information about that student. The section profiles also allow users to easily sort and filter students based on list criteria.
* ***Student Profiles***display more in-depth information about an individual student. Information includes enrollment history, academic transcript history, and parent contact information.

The sections that follow explain how an IT Administrator configures the Dashboard for the users within a tenancy.

## Configuring the Dashboard

Follow the steps below to configure the Dashboard:

1. Determine your highest priority goals for using ESDS data. This governs your data ingestion plan.
2. Design how you want the data to display for your users. This can be accomplished by reviewing the options for profiles and lists detailed later in this chapter.
3. Complete the configuration template for the Dashboard. If you are unsure about any of the fields, write questions directly in the template.
4. ESDS partners will create the configuration code in JSON. This may take up to several weeks, depending on the complexity of the request.
5. When you receive the configuration code from your ESDS partner, an authorized user can upload the code to the Dashboard application.

## Configuring Section Profile Lists

The Section Profile provides a table of information about students in a class or section. This resembles a teacher's gradebook. The first column is always the student's name, which a user can click to view that student's profile. The second column includes indicators for the student, such as whether the student is an English Language Learner (ELL) or has an Individual Development Plan (IDP). The remainder of the table can include any combination of columns. An example of the Section Profile displays below:



## Documenting your List Configurations

The following steps allow you to set up your list configurations:

1. Open the List of Students Configuration Template spreadsheet.
2. Each tab in the file allows you to enter a different view and each view has its own Microsoft Excel worksheet in the workbook document. The template has several blank worksheets, and you can duplicate additional tabs if needed.
3. Fill in the blue fields at the top of the worksheet. The fields are:

* **View Name** field is the title of the view. It appears as the title in the view dropdown on the Dashboard.
* **Max Grade Range** (inclusive) and **Min Grade Range** (inclusive) fields allow you to enter grade ranges for this view. Leave this blank to show all grade ranges.
* **District** field allows you to enter your school district name (for your reference).

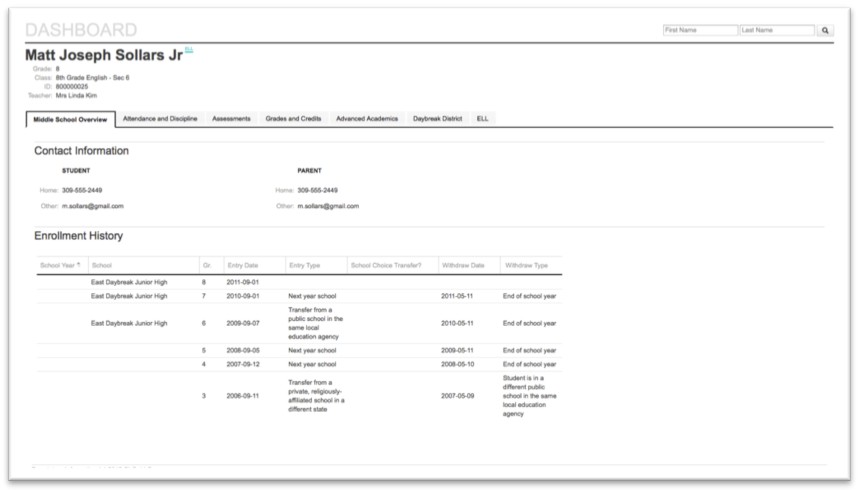
1. Fill in the next block of blue text in the tab. The Name and Indicator columns are there by default and should not change.

* A **Column Category** row precedes each group of similar-type columns.
* The **Column Type** row is the specific column of information; for example, Most Recent Assessment Result.
* **Configuration Options, Line 1** and **Configuration Options, Line 2** are options for the view. For specific information, please see the table below.
* The **Header Line** options allow you to change the heading on that row of the table within the user interface.
* The **Sample Text** row allows you to enter example text for your users.

List View Options

| **Column Type** | **Description** | **Configuration Options** |
| --- | --- | --- |
| **Attendance** | | |
| Attendance Rates | Displays the student's attendance rates. This can be a percentage of days in attendance compared to the total days, or it can be the percentage of days tardy compared to the total days. | * Select the rate type to display: attendance percentage or tardy percentage. * Select the time frame: this school year or only this semester.   (Optional) Indicate the color-coding cut points. |
| Attendance Event Counts | Shows a count of the total number of days a student is absent or tardy. | * Select the attendance count type to display: excused absences, unexcused absences, or all absences. * Select the time frame: this school year, or only this semester (or session).   (Optional) Indicate the color-coding cut points. |
| **Assessment** | | |
| Most Recent Assessment in a Family | Displays the results of the most recent assessment in a family for a student. | * Select an assessment family. * Choose an outcome to display: numeric score or performance level. * (Optional) Provide a scale score and cut points.   Choose to display the depth of assessments within time: most recent, second most recent, etc. |
| Highest Ever | Displays the highest results for an assessment within a given assessment family for a student. | * Select an assessment family. * Choose an outcome to display: numeric score or performance level. * (Optional) Provide a scale score and cut points.   Choose to display the depth of assessments within time: most recent, second most recent, etc. |
| **Grades** | | |
| Most Recent Transcript Grade for a Subject Area | Shows the most recent transcript scores for a completed course in the subject area you are viewing. | * Choose to display the most recent, second most recent, etc. * The transcript grade entries are determined by the selected course. This is not configurable. |
| Most Recent Grade Book Entry | Shows the most recent grade book entry in the subject area you are viewing. | * Choose to display the most recent, second most recent, etc. * The grade book entries are determined by the selected course. This is not configurable. |

## Configuring Student Profiles

Student profiles display in-depth information on an individual student. Profiles consist of a page with one or more tabs. The first panel of each tab (page) is always the core student information and is not configurable. The remainder of the page can contain any combination of panels. An example of a student profile displays below:

### Documenting your Student Profile Configuration

Before the ESDS partner creates the JSON for your Dashboard configuration, you must complete the spreadsheet using the following steps:

1. Open the Student Profile Template spreadsheet.  
   Each tab in the bottom portion of the worksheet's window represents a different page on the student profile. If you need additional tabs to create additional pages, simply duplicate an existing tab.
2. Fill in the blue fields at the top of the worksheet:

* The **Page Name** field displays the name of the page.
* The **Grade Levels** field allows you to enter the grade levels that you want to display.
* The **District** field allows you to enter your school district name for reference.

1. Fill in the fields for the panels that should be included on the page:

* The **Panel Title** appears above the appropriate panel.
* The **Panel Type** field describes the type of panel that must be included.
* The **Panel Configuration** field allows you to modify the panel's properties. **Note:** Several panels are not configurable.

Student Profile Panel Options

| **Panel** | **Description** | **Configuration Options** |
| --- | --- | --- |
| Core  Student Information | Displays the student's name, nickname, State ID, Grade Level, and (in some schools) the student's homeroom and homeroom teacher. | This always appears on student profiles and is not configurable. |
| Current Contact  Information | Displays the names, addresses, phone numbers, and email addresses for a  student. | This is not configurable. |
| Enrollment History | Displays all the schools in which a student has been enrolled. This includes entrance and exit reasons, dates, and grades. | This is not configurable. |
| Assessment History | Displays all of the information for a student's assessments within a given assessment family. | This can be configured to display any node on the assessment family tree. You can also configure this to display all the results for child nodes. |
| Transcript History | Displays the complete student's transcript. This includes all available grade data for each year and semester. The GPA for each semester is available in the summary. | This is not configurable. |
| Attendance History | Displays a calendar that shows the days a student was absent (excused or unexcused) or tardy for the whole school day. | The default number of years included is three (3) years. |

## Uploading Configurations

Once the Dashboard spreadsheet has been completed, it should be sent to the ESDS partner organization contact for your district. The partner generates a piece of configuration code in JSON format. When this is complete, you receive the code in an email. You can then upload this code to your district's Dashboard using the following steps:

You must upload the JSON code as you receive it from the ESDS partner. Do not modify the code before uploading it. Any changes, even seemingly minor ones, present a risk to your district and other district's Dashboards when you upload the code. If you notice a mistake in the JSON code that you receive, work with the ESDS partner to make corrections rather than attempting to change them.

To upload the JSON configuration code:

1. Log in to ESDS.
2. Click Admin.
3. Click ESDS Dashboard under the Application Configuration heading.
4. Paste the JSON configuration code that was emailed to you.
5. Click Save Config.

Sample JSON Configuration Code

### Important Considerations for ESDS Dashboard Configuration

Consider the following when configuring the Dashboard:

* Uploading a new configuration overwrites all previous configurations.
* The configuration changes take effect the next time that a user logs in to the application.
* Users who are on the ESDS Dashboard while configuration changes are uploaded may encounter errors. These errors should clear once the user logs in again.
* Make sure that configurations are secure, including safeguarding documents and emails, so that only authorized people in your district can see them.

# Modifying the Dashboard

The ESDS Dashboard provides users access to data for schools, courses, sections, teachers, and students in a navigable and easy-to-read format.

The ESDS Dashboard is initially established according to the process outlined in the previous chapter, *Setting Up the Dashboard*. After the initial setup, an IT Administrator can modify the layout of the Dashboard using the ESDS Dashboard builder application as detailed in this chapter.

Before continuing with this chapter, it is important that you have an understanding of the following terms:

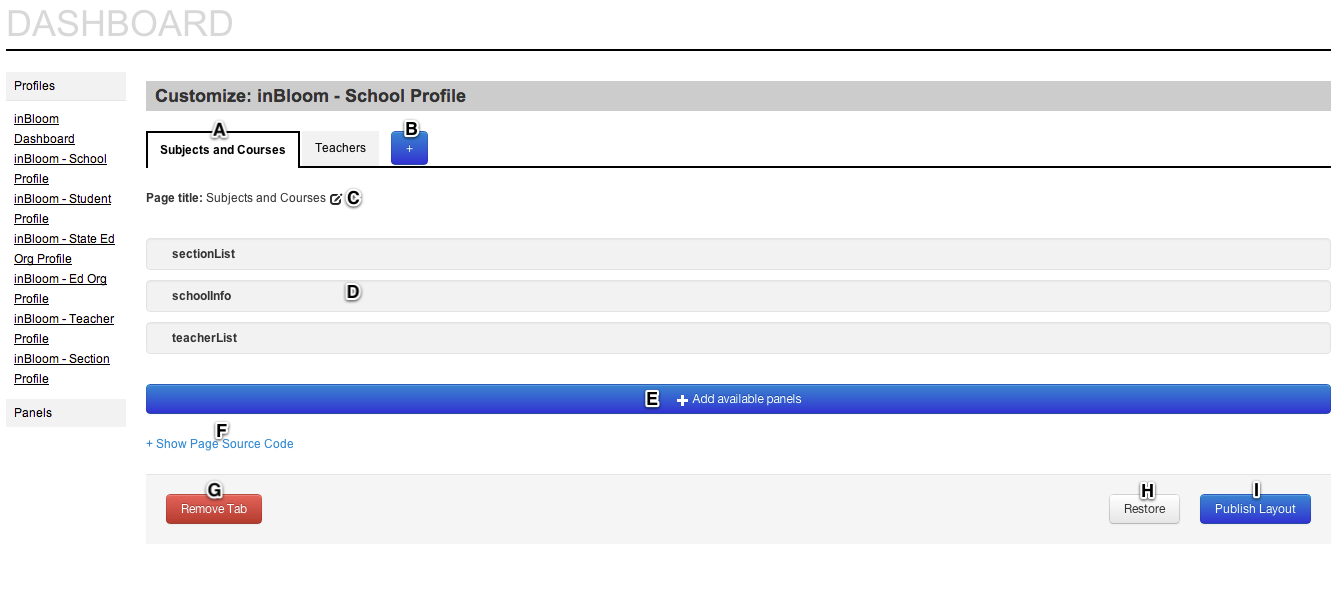
* ***Profile***- A profile is a collection of logically related data organized by scope. The default profiles on the Dashboard are the Dashboard Profile, School Profile, Student Profile, State Education Organization Profile, Education Organization Profile, Teacher Profile, and Section Profile.
* ***Page***- Profiles are made up of one or more pages. Pages help organize the data and appear as tabs on a profile.
* ***Panel***- Contained within each page are one or more panels that display data for the profile. Panels can currently be added or removed from a pre-defined library, established by profile. For example, a Teacher panel can be added to the Education Organization's profile.

For example, a school profile could include pages for teachers and students. Within the student’s page, there may be panels for grades, attendance, and general information.

## Modifying a Profile

Follow the steps below to modify a profile:

1. Log into ESDS.
2. Click on the ESDS Dashboard.
3. In the address bar, add /builder/index.html to the end of the URL.
4. Click the Profiles menu. The menu expands to display the list of available profiles.
5. Click the name of the profile that you want to edit. The profile page appears as shown below:



The callouts on the graphic above are defined as follows:

**A**– Tabs that indicate the individual pages for the profile. In the example in the graphic, the *School Profile* displays tabs for *Subjects and Courses* and *Teachers*. Click a tab to view the page's configuration.

**B**- This button allows you to add a new page to the profile. Once you have added the page, you can add panels that are available for the profile.

**C**- The page title describes the tab as it will appear on the Dashboard profile. Click the button to change the text for the page/tab title.

**D** - The active panels that display. They can be rearranged using the drag and drop method. Hover over the panel to activate the ability to change the order of panels.

**E** - Click the *Add available panels* button to add new panels to the page. When you click the button, a dialog appears that allows you to select from panels available for the profile.

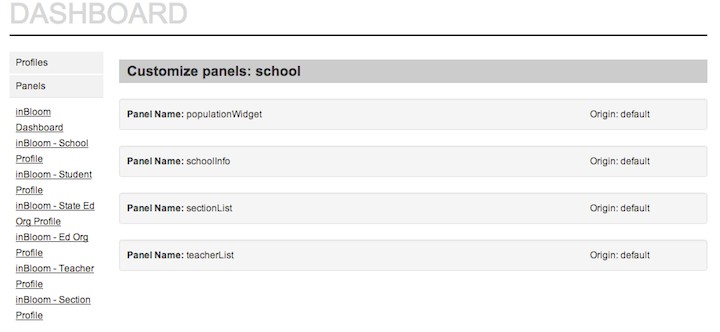
**F**- Click *Show Page Source Code* to display the JSON code for a profile page. This allows you to share configurations with other IT Administrators, as well as make changes to your Dashboard pages in a way not currently supported by the builder interface.

**G**- Click the Remove Tab button to remove the page/tab from the profile.

**H** - Click the Restore button to reset the profile to its most recently saved state.

**I**- Click the Publish Layout button to update the Dashboard with recent changes. You will be prompted to confirm the changes before the profile is updated for your Ed Org.

## Panels

The ESDS Dashboard has a default library of data panels. Each panel can be added or removed from its associated profile. The ESDS Dashboard does not currently support the ability to modify the panel library. An example of Dashboard panels is shown in the graphic below:

# Data Browser

The Data Browser is a data validation tool that provides an IT Administrator the ability to perform consistency and regression checks against data stored in the SDS. Only IT Administrators have access to the Data Browser feature. The information presented in the tool’s interface provides assurance that the data stored is an accurate reflection of what has been loaded and offers the details to troubleshoot any inconsistencies with the loaded data.

The Data Browser shows the number of records loaded into the SDS to ensure the correct number of records are available and if necessary troubleshoot any discrepancies in counts. For example, if a district has 100 schools, but the Data Browser displays only 95 school records, the IT Administrator has the ability to use the Data Browser tool to identify which schools are missing from the data loaded into the SDS.

The Data Browser also provides the status of ingestion jobs. This feature offers the IT Administrator insight into the overall status of ingestion jobs in order to troubleshoot any data discrepancies. For example, when an ingestion job completes, the Data Browser will display the number of records actually ingested compared to the number of records sent to the SDS in the file. Note that ingestion jobs currently in progress do not display on the Data Browser.

In summary, the Data Browser provides an IT Administrator with the following functionality:

* A list of the highest level entities of the district with the ability to drill into each entity and view details.
* A view of counts of entities.
* The ability to easily search for attributes and entities by Ed Org.
* Historical information on ingestion jobs.

## Prerequisites

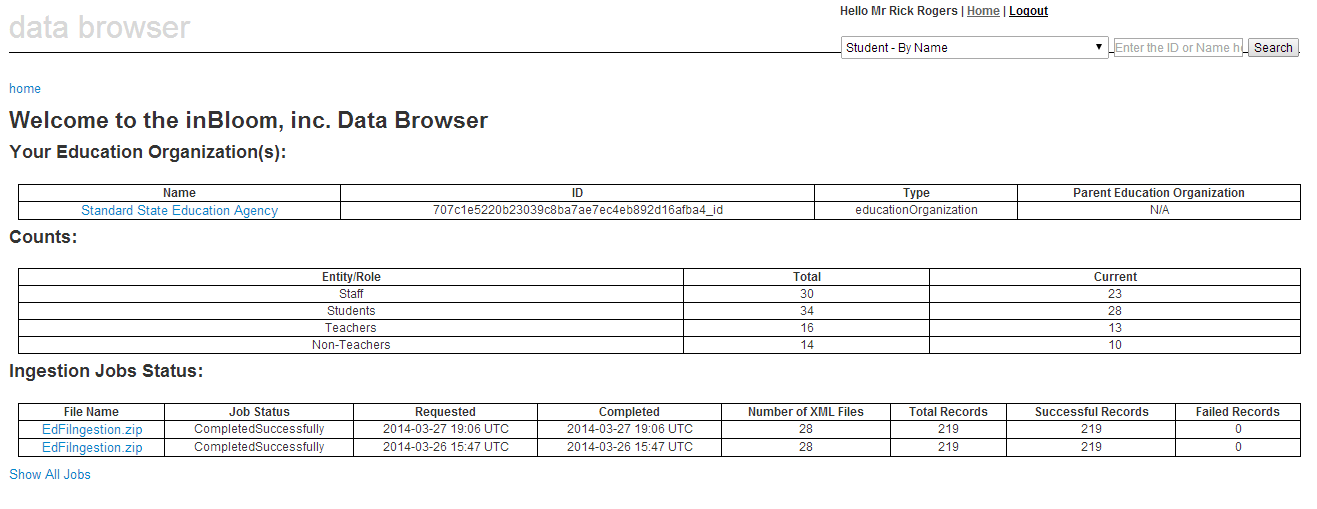
In order to use the ESDS Data Browser, you must have:

* An account with an IT Administrator role access rights
* The Data Browser Application enabled
* Permission*s* to view the appropriate data

## Viewing Data

Follow the steps below to view data using the ESDS Data Browser:

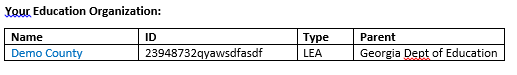
1. Log in to ESDS using your IT Administrator account.
2. Click Admin.
3. Click Data Browser. The Data Browser home page will appear as shown below:



The following sections detail the Data Browser table definitions.

### Your Education Organization

This table displays the highest level Education Organization with which the IT Admin is associated.

****

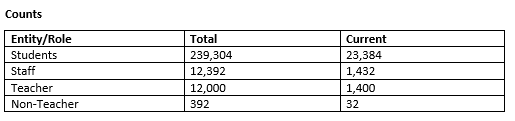
Depending on the IT Administrator and the specific relationship with the Ed Org, this could be at the SEA, the LEA or at the individual school level. It is possible for the IT Administrator to have a relationship with more than one highest level Education Organization, for example with two LEAs. In that case, both LEA would be listed in the table.

The fields in the table are defined as follows:

* **Name:** Name of the Education Organization
* **ID:** The internal ESDS ID for the Education Organization
* **Type:** Education Organization Type
* **Parent:** Education Organization that is listed as the parent for the Education Organization.

### Counts

The counts table provides information on the number of unique entities tied to the Education Organization and its children Education Organizations.



All counts are relevant to the highest level Ed Org to which the IT Administrator has access. As depicted in the graphic above, the counts displayed are the number of unique entities associated with the Ed Org. Counts are provided for Students, Staff, Teachers, and non-teachers.

The fields in the **Entity/Role** column of the Counts table are defined as follows:

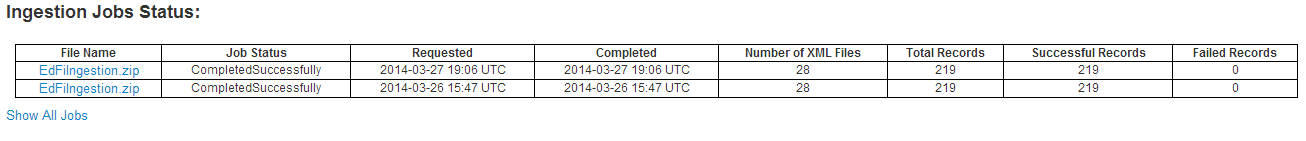
* **Students:** Number of unique Students in the *StudentSchoolAssociation* relative to the Ed Org you are viewing.
* **Staff:** Number of unique Staff members in the *StaffEducationOrganizationAssociation* relative to the Ed Org you are viewing.
* **Teachers:** Number of unique teachers in the *TeacherSchoolAssociation*. The Teacher must also be referenced in the *StaffEducationOrganizationAssociation* for that Ed Org or one of its parent Ed Org.
* **Non-Teacher:** Number of Staff Members minus the Number of Teachers.

The **Total** and **Current** columns of the Counts table are defined as follows:

* **Total Count:** The total number of unique entities that are associated with the Ed Org.
* **Current Count:** The number of unique entities tied to the Ed Org where the current date is between the Begin Date and the End Date of the association.

### Ingestion Jobs Status

The Ingestion Jobs Status table provides information on the latest five ingestion jobs that completed for your Tenant.



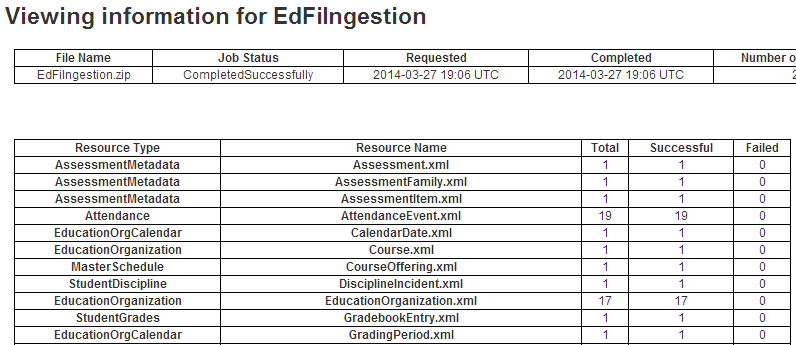
If you have more than one person in your tenant that submits ingestion jobs, they will all display in this table for all IT Administrators. This information is not filtered by Ed Org.

Click on **Show All Jobs** below the table to view information for all completed ingestion jobs.

**Note:** This table does not provide information on ingestion jobs that are in progress. Ingestion log files are not available through the Data Browser. Go to the Landing Zone to view the log files.

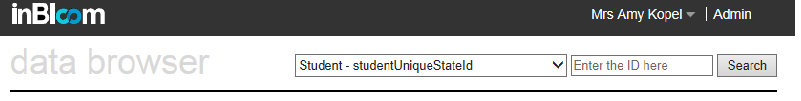
The following fields display in the Ingestion Jobs Status table:

* **File Name:** The name of the uploaded zip file.
* **Job Status:** Overall status of the completed ingestion job (Success/Failed)
* **Requested:** Date/time the ingestion job was submitted.
* **Completed:** Date/time the ingestion job completed.
* **Number of XML files:** The number of XML files in the zip file.
* **Total Records:** The total number of individual records that were part of the ingestion job.
* **Successful Records:** Number of records successfully ingested.
* **Failed Records:** Number of records that failed ingestion.

Clicking on the **File Name** will display a page with more detailed information about the ingestion file and the status of specific files as shown below:

## Data Browser Search Bar

The Data Browser Search Bar allows you to search a for a specific data record.



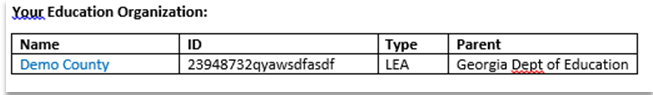
The list below defines attributes that can be used to search by entity:

* **Staff:** StateUniqueID
* **Student:** StateUniqueID
* **Education Organization:** Education Organization Name, StateUniqueID
* **Parent:**  StateUnique

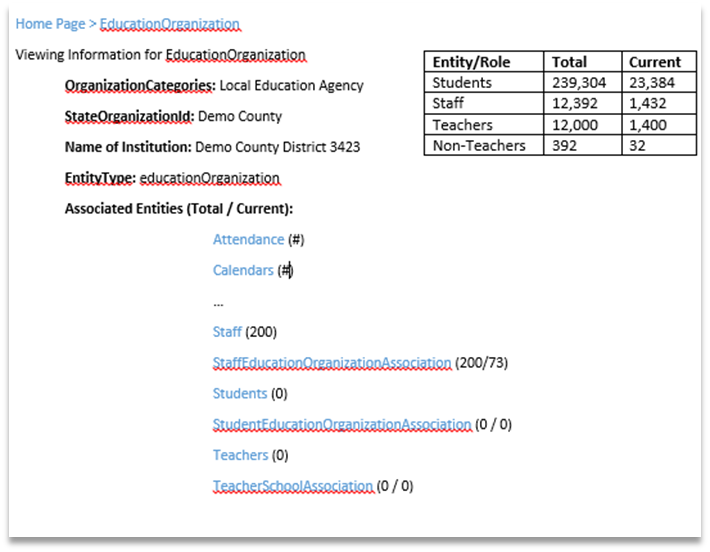
## Navigating the Data Browser

Follow the steps below to explore detailed data for a specific Ed Org.

1. Click the name of the Ed Org under the *Your Education Organization* table, as shown below.



This opens the Education Organization page with attributes about the specific Ed Org, counts for the Ed Org, and links for each entity associated to it with counts. In this example, clicking **Demo County** as shown above will produce the Education Organization Page displayed below.



❷

❶

❶) The Count table displays on each Education Organization page. The Education Organization Count table functions as the Count Table on the home page; however, it is limited to the numbers and data of the Ed Org currently being viewed and any child Ed Orgs tiered below it. For example, an SEA’s Education Organization *page* will display any and all information of the LEAs under its hierarchy.

❷ The Associated Entities is a set of links for each associated entity. Each associated entity is initially displayed with a **#** next to it. By clicking on the **#**, the IT Administrator can find the number of specific entities associated to it.

Most of the entities display with a single number that reflects the total entity count directly associated to it. However, there are a few entities that display two numbers: Total and Current. The logic on the Total and Current counts is the same as in the Count table (evaluate the current counts with the Begin Date and End Date).

2. To view the attributes of an associated entity, click on the entity.

The data returned in the grid can be sorted within the set returned, but not the entire population in the database. The column headers can be used to sort the returned set, but it will not sort the entire population and return all students alphabetically from A-Z. You have the ability to change the number of entities returned by using the counts on the upper right corner of the grid. You can choose the highest number of results to return and then sort those. This will work if you are searching for entities of which the total count is less than 100. For result sets greater than 100, you will need to page through the set to look for a specific item.

# Bulk Data Extraction

Bulk Data Extraction is the process of mining large amounts of data from the ESDS. The Bulk Extract program creates a file for each Ed Org that has an authorized bulk extract application, as well as a public data extract for the tenant. The extract file is available as a full extract (all data) or a delta extract (all data that has changed since the last extract) and contains all data for that Ed Org and its descendants.

The sections that follow detail the different types of bulk data extractions.

## Previous Year Student Data

To provide a clear history of a student's academic past, previous year student data is provided in each Ed Org extract file. Previous year student data is data related to a student from a previous academic year, either from the same Ed Org or from a different one.

The table below specifies date attributes or date attributes of a referenced entity that are used to determine whether an entity is from a previous academic year.

Previous Year Student Dated Attributes

|  |  |
| --- | --- |
| **Entity** | **Attribute** |
| attendance | schoolYear |
| courseTranscript | studentAcademicRecord.schoolYear |
| disciplineIncident | incidentDate |
| disciplineAction | disciplineDate |
| grade | schoolYear |
| gradebookEntry | dateAssigned |
| reportCard | schoolYear |
| studentAcademicRecord | schoolYear |
| studentAssessment | administrationDate |
| studentCohortAssociation | beginDate |
| studentCompetency | studentSectionAssociation.beginDate |
| studentDisciplineIncident Association | disciplineIncident.incidentDate |
| studentGradebookEntry | gradebookEntry.dateAssigned |
| studentProgramAssociation | beginDate |
| studentSchoolAssociation | entryDate |
| studentSectionAssociation | beginDate |

The following entities do not have a dated attribute and appear in the extract file of every education organization to which the student is related:

* parent
* student
* studentParentAssociation

## Previous Year Staff Data

In order to provide a clear history of a staff member's academic past, previous year staff data is provided in each Ed Org extract file. Previous year staff data is data related to a staff from a previous academic year, either from the same Ed Org or from a different one.

The table below specifies date attributes or date attributes of a referenced entity that are used to determine whether an entity is from a previous academic year.

Previous Year Staff Dated Attributes

|  |  |
| --- | --- |
| **Entity** | **Attribute** |
| staffCohortAssociation | beginDate |
| staffEducationOrganization Association | beginDate |
| teacherSchoolAssociation | teacher.staffEducationOrganizationAssociation.beginDate |
| teacherSectionAssociation | beginDate |

The following entities do not have a dated attribute and will appear in extract files of the Ed Orgs to which they have an association:

* staff
* teacher

## Top Level Education Organization Extract

An Ed Org is considered “top level” if it does not have any parent education organization. For example, a State Education Agency can be a top level education organization for a tenant.

Top level Ed Org extract is a special case. The extract contains only the private data that belongs to that education organization. It will not have any data that belongs to its descendant education organizations (unlike other education organization extracts). This is applicable to both full and delta extracts.

**Note:** Changes in the Ed org hierarchy within a tenant may change the contents of an education organization's extract file.

If a top level education organization adopts a parent, it ceases to be treated as top level and will be treated as any other education organization for the purposes of bulk extract. The extracts generated after this update for this education organization will include the data associated with its descendant education organizations.

If an Ed Org loses all of its parent education organizations, it will be treated as a top level education organization. The extracts generated for this education organization after the hierarchal change will cease to include data associated with its descendant education organizations.

# Delegating Permissions to the State Education Organization

A Local Education Organization (such as a district or county school system) can delegate some administrative responsibilities to the state education organization in its hierarchy. For example, the LEA can delegate the right to view security events on the organization's behalf.

The [*LEA Administrator*](#bookmark127) is the role that can delegate these permissions. The recipient of the delegated permissions has the designated [*SEA*](#bookmark143)[*administrator*](#bookmark143)role.

For example, if a District D is part of State S, an [*LEA Administrator*](#bookmark127) for District D can delegate security event viewing permission to State S. In this case, an [*SEA Administrator*](#bookmark143) for State S will be able to view security events on behalf of District D.

Use the following steps to delegate permissions to the SEA:

1. Log in to ESDS using your LEA Administrator account.
2. Click **Admin** to open the Administrator home page.
3. Click **Delegate Administration** under System Tools. The page displays.
4. Choose the permissions to delegate. Reference the *Security Event* chapters to supplement your use of this system tool.

# Security Event Logs

ESDS logs security events for users and applications. The sections that follow cover the events that are logged in the SDS and the permissions ESDS users need to view logged events.

**Note:** Several security events can be viewed using the securityEvent resource in the REST API. For more information about this resource, refer to the securityEvent resource in the *ESDS Developer Documentation*.

## Logged Security Events

Logged security events include the following:

| **Event** | **Origin** | **Log Name** |
| --- | --- | --- |
| Bulk Extract request header pre- conditions failed | API | FileResource |
| Range header doesn't match format | API | FileResource |
| If range is not syntactically valid | API | FileResource |
| Successful request for singlePart- FileResponse | API | FileResource |
| Successful request for multiParts- FileResponse | API | FileResource |
| Logout: status | API | SecuritySessionResource |
| LEA's delegation is enabled/dis- abled | API | AdminDelegationResource |
| Application granted access to EdOrg data | API | ApplicationAuthorizationResource |
| EdOrg data access has been re- voked | API | ApplicationAuthorizationResource |
| Failed to create custom role rights validation failed | API | CustomRoleResource |
| Failed to create custom role unique roles check failed | API | CustomRoleResource |
| Failed to create custom role invalid realm specified | API | CustomRoleResource |
| Failed to create custom role Already exists | API | CustomRoleResource |
| Created custom role with id | API | CustomRoleResource |
| Failed to update realmId | API | CustomRoleResource |

| **Event** | **Origin** | **Log Name** |
| --- | --- | --- |
| Updated Role by id | API | CustomRoleResource |
| Deleted Role by ID | API | CustomRoleResource |
| Realm [name]  {updated,deleted,created} | API | RealmResource |
| SAML Message received from [origin] is invalid | API | SamlFederationResource |
| HttpServletRequest is missing | API | SamlFederationResource |
| [principal id] from tenant [tenant] logged successfully into [applicationDetails] | API | SamlFederationResource |
| {created,updated,deleted} user | API | UserResource |
| Ingestion process started. | Ingestion | ControlFilePreProcessor |
| [file] [filename] considered for processing: [n] | Ingestion | JobReportingProcessor |
| [file] [filename] records ingested successfully: [n] | Ingestion | JobReportingProcessor |
| [file] [filename] records deleted successfully: [n] | Ingestion | JobReportingProcessor |
| Not all records completely processed | Ingestion | JobReportingProcessor |
| Number of records processed | Ingestion | JobReportingProcessor |
| Runtime Exception | IDP | Login |
| Successful Login | IDP | Login |
| Failed Login | IDP | Login |

|  |  |  |
| --- | --- | --- |
| **Event** | **Origin** | **Log Name** |
| Request to stream sample bulk extract | Bulk Extract | BulkExtract |
| Successful request to stream EdOrg data | Bulk Extract | BulkExtract |
| Received request to stream EdOrg data | Bulk Extract | BulkExtract |
| Failed request to stream SEA  public data, missing edOrgId | Bulk Extract | BulkExtract |
| Received request for list of links for all SEAs and LEAs for this user/app | Bulk Extract | BulkExtract |
| Successful request for list of links for all SEAs and LEAs for this  user/app | Bulk Extract | BulkExtract |
| Received request to stream EdOrg delta bulk extract data | Bulk Extract | BulkExtract |
| Failed delta request, missing LEA id | Bulk Extract | BulkExtract |
| Failed delta request, missing date | Bulk Extract | BulkExtract |
| Failed request for EdOrg delta bulk extract data | Bulk Extract | BulkExtract |
| No bulk extract support for: [LEA] | Bulk Extract | BulkExtract |
| No authorized EdOrgs for  application | Bulk Extract | BulkExtract |

## Permissions for Viewing Security Event Logs

The table below lists the permissions assigned to Administrator roles for viewing security event logs.

|  |  |
| --- | --- |
| **Administrator Role** | **Security Events Log Permissions** |
| **ESDS Operator** | View all successful and failed login events through the ESDS IDP that are associated with that user's ESDS instance. |
| **SEA Administrator** | View all successful and failed login attempts for the Administrators at the State level within the same Tenant and associated with the same Ed Org as the user. |
| **LEA Administrator** | View all successful and failed login attempts for the LEA Administrators associated with the same Ed Org as the user. |

# Security Event Notifications

In addition to the logging of security events described in the previous chapter, ESDS sends email notifications about administrative events.

The notification is sent by the ESDS Operator to the appropriate recipients and is delivered to the email address configured in the *sli.support.email* property, part of sli.properties. The sender name and email address are configured in the administration tool's config.yml configuration file: email\_sender\_address and email\_sender\_name.

The table below lists the security events that produce notifications and the administrator roles that would receive those notifications:

Security Event Notifications

| **Event** | **Notification Sent** | | **Notification Recipient** | |
| --- | --- | --- | --- | --- |
| **Sandbox** | **Production** | **Sandbox** | **Production** |
| Application Developer Account  Approved | Yes | Yes | [Application](#bookmark98) [Developer](#bookmark98) | [Application](#bookmark98) [Developer](#bookmark98) |
| Landing Zone Provisioned | Yes | Yes | [Application](#bookmark98) [Developer](#bookmark98) | [Ingestion User](#bookmark123) |
| Application Registration Request Made | Not applicable, registration is automatic | Yes | Not applicable | [ESDS](#bookmark146) [Operator](#bookmark146) |
| Application Registration Request Approved | Not applicable, registration is automatic | Yes | Not applicable | [Application](#bookmark98) [Developer](#bookmark98) |
| SEA Administrator Created | Not applicable | Yes | Not applicable | SEA Ad[ministrator](#bookmark143) |
| LEA Administrator Created | Not applicable | Yes | Not applicable | LEA  Ad[ministrator](#bookmark127) |
| Realm Administrator Created | Not applicable | Yes | Not applicable | Realm Administrator |
| Ingestion User Created | Not applicable | Yes | Not applicable | [Ingestion User](#bookmark123) |
| ESDS Operator Created | Not applicable | Yes | Not applicable | [ESDS](#bookmark146) [Operator](#bookmark146) |

Glossary

Access Context The combined conditions under which a user or application is accessing content in the ESDS. This includes the user role(s), the education organization(s) to which the user is assigned, and the application the user is logged into in the SDS.

Access Token An object that encapsulates a user's authenticated session with an application, including privileges.

Aggregated Viewer Reserved user role

Application Developer An administrator role under the grouping of [*ESDS Administrators*](#bookmark147) that is assigned to software developers who will be using components of ESDS while developing software designed to read and edit data from the ESDS.

Application Registration The process used by an application developer or ESDS administrator to request access to the SDS. This registration process ensures that the application has both credentials to authenticate with ESDS and permission to access ESDS resources (REST API, Data Store, etc.)

Association A rule set in the ESDS which ties together instances of two given en[tities](#bookmark115). An association may exist as an [*entity*](#bookmark115) itself; and, therefore as a [*resource*](#bookmark138) in the REST API), or it may be expressed as a [*direct reference*](#bookmark110) under each entity association.

Attribute The attribute component of the industry standard entity-attribute-value model (EAV) used to describe a data model, such as the model for the ESDS. An attribute is a property or parameter for a given entity. In ESDS, each attribute in the data store corresponds to the name portion of a name/value pair in the request and response bodies used in the REST API.

Authentication The process of verifying that a user is who he/she claims to be by providing a set of accepted [*security credentials*](#bookmark107). For example, to authenticate with a certain system, a user provides a valid username and password that matches his/her user account.

Authorization The process of determining what resources a user has permission to access and what actions that user has permission to perform. After a user authenticatesto a system, the system authorizes him to view specific data.

Callback URL See the term *Redirect URI*.

Client ID An ID that uniquely identifies an application to ESDS. The client ID is granted by the ESDS during the [*application registration*](#bookmark99) process.

Client Secret A value that, in combination with the [*client ID*](#bookmark104), enables your application to authenticate with ESDS. The client secret is granted by ESDS during the *applica*[*tion registration*](#bookmark99) process.

Common Field Field that all REST API resources have, including ID (the unique identifier of the resource), entityType, and metaData.

Context The combined conditions under which a user or application exists relative to data in the ESDS. A context for a given user is expressed as a combination of user role(s), the education organization(s) to which the user is assigned, and the application the user is currently using to interact with the SDS.

Credentials A unique combination of text or digital objects used to gain access to protected content. This is typically a user name and password, although some systems might require rotating key codes or key cards.

Custom Data Data associated with an *entity* but not a part of the ESDS data model. Developers can add this data to store data beyond what is accumulated in the ESDS schema. Refer to the *ESDS Developer Documentation* for information on how to create, query, and maintain custom data.

Data Domain A logical organization of *entities* in the ESDS data model, used to express how a group of entities from the ESDS relate to each other in a given real-world scenario.

Data Ingestion The process used to import (ingest) a large amount of data to the ESDS. This process is performed using an ingestion utility specifically designed to manage large, process-intensive operations. Ingestion bypasses the need for a long series of POST operations using the ESDS REST API. For more information, refer to the *ESDS Data Ingestion Guide*.

Dataset Any set of data from the ESDS identified by having a common feature.

Direct Reference An *attribute* with the purpose of identifying an instance of another entity. For example: A *Section* entity includes a direct reference of *session* which identifies a specific *Session* entity.

Directory A service that manages user identities and user roles and permissions within a specific security boundary. Directory service implementations can vary from a customized OpenLDAP to Microsoft's Active Directory. The ESDS recognizes directory services from an [*education organization*](#bookmark112) as an *au*[*thentication*](#bookmark102) resource to which ESDS administrators can map user rolesand from which ESDS can authenticate users with those mapped roles.

Education Organization A generic term (also known as “Ed Org”) used to refer to a level in the hierarchy of education agencies. An education organization may be a state school system, a county/district school system, or an individual school. It may refer to specific education organizations using terms like state education agency (SEA) for state systems and local education agency (LEA) for county/district systems.

Educator One of the user roles in ESDS. Users with this role can only view data associated with students actively enrolled in the current class sections.

Effective Permissions Access granted to a user to SDS data after combining the user’s assigned role(s), the rules for which roles take priority, and the specific *context* from which that user is operating (such as the application the user is using or the current date and time). This can also apply to applications.

Entity The entity component of the industry standard entity-attribute-value model (EAV) used to describe a data model, such as the model for the ESDS. An entity is the item or object being stored. In ESDS, each entity in the data store also corresponds to a *resource* in the REST API.

ESDS Administrator A user responsible for the day-to-day tasks of managing access and data in a deployed ESDS system. This general term describes two specific ESDS administrator roles: [*LEA Administrator*](#bookmark127) and [*SEA Administrator*](#bookmark143).

ESDS Directory A [*directory*](#bookmark111) contained within ESDS technology used to authenticate users with administrative roles for that ESDS system, such as an [*ESDS*](#bookmark122) *Ad*[*ministrator*](#bookmark122).

Federated An authentication approach in which users authenticate against a directory service that resides outside of the ESDS and is provided by the education organization. The identity of the user is shared with other ESDS applications through the use of the *SAML* protocol. Users can sign in to one ESDS application and are automatically signed in to other ESDS applications operating in the same browser.  
ESDS requires that each federated user map to a *Staff* or *Teacher* user entity in the SDS ESDS which is created during [*data ingestion*](#bookmark109) or through the ESDS REST API.

Federated Agency A general term used when speaking of a federated state, county, or district school system.

Federated Authentication An authentication method in which multiple systems that require user *au*[*thentication*](#bookmark102) are linked together to share a common *directory* of users. Federation is a common means of providing single sign-on (SSO) functionality across multiple services and applications.

Federation Refers to a group of identity providers that the ESDS identity provider trusts.

General Data A category of data identified by ESDS for the purpose of assigning all the same permissions to the data in that category. Any data not identified as [*public data*](#bookmark133) or *re*[*stricted data*](#bookmark139) is considered general data.

Hosted A term describing something that exists and is managed within the ESDS.

Identity Provider A computer system that stores user identity information such as usernames, passwords, and roles, and provides a way for other computer systems to access this data. An identity provider may be a *directory* or a service acting on behalf of one or more directories. The common industry abbreviation *IDP* is usedwhen referring to an identity provider.

Ingestion User One of the [*ESDS administrator*](#bookmark147) roles. This role targets users whose responsibil- ity it is to use landing zones and the ingestion application for [*data ingestion*](#bookmark109) in in- Bloom.

Installed Application Software application that uses the ESDS platform, but requires the end user to install a portion of the software outside of the ESDS. Different from web application, which can run in any supported Web browser without installing additional software (other than, perhaps, third-party browser plugins such as Flash). Examples of installed applications include mobile and embedded applications.

IT Administrator One of the user roles in ESDS. Users with this role have all the permissions of users with the [*Leader*](#bookmark128) roles plus the ability to create new users and to define cus- tom permissions.

Landing Zone An SFTP-accessible directory location that is reserved for files to be ingested using the ESDS data ingestion process. ESDS administrators can set up one or more landing zones to serve a single data store.

LEA Administrator One of the *ESDS administrator* roles. This role targets users who serve as ESDS administrators at the county or district level, below the state level (LEA = Local Education Agency).

Leader One of the user roles in the ESDS. A user with this role can view data related only to those educational organizations to which the user is associated.

Logged Data Field Field that triggers the logging of an event by the ESDS REST API.

Non-Required Attribute An *attribute* that does not require a value in order for the entity to be created or updated in the SDS.

OAuth A protocol for authorization published as an open standard by the Internet Engineering Task Force (IETF). The protocol allows a user to share private resources from one site (web application, network service, etc.) with another site by supplying authentication tokens rather than sharing a user's authentication credentials. ESDS uses the OAuth protocol in combination with [*SAML*](#bookmark144).

Operator An [*ESDS administrator*](#bookmark147) role for assigning permissions to users responsible for deploying ESDS or handling the administrative side of the [application registration](#bookmark99)process.

Parent One of the user roles in ESDS. Users in the default [*Parent*](#bookmark131) role group can only view public data, Student\_Owned data, and Student\_General data.

Permissions The rules that govern a user or application access to data in the ESDS. Basic permissions are no access, read-only, and read-write. ESDS applies these basic permissions to each data point and related data points are designated by categories. Permissions are assigned to these categories rather than to individual data points. Administrators create *roles* to manage permissions to these categories, and users receive permissions based on their roles.

Personally Identifiable Information that can be used, either alone or with other

Information resources, to uniquely identify, contact, or locate a single person. Also known as PII.

Public Data Data that [education organizations](#bookmark112) can make available publicly (outside of the organization) to anyone without designated permissions. An example of this type of data would be a school’s street address.

Realm An ESDS concept used to identify the context of authentication for a user or application. This can be a district that has its own identity service for authentication, or it could be an organization of districts that share an identity service. The highest available realm is a state. Within the ESDS, user IDs are unique only within a [*tenancy*](#bookmark152). Therefore, a single realm should not serve multiple tenants.

Realm Administrator One of the [*ESDS administrator*](#bookmark147) roles for an administrator at the *realm* level.

Redirect URI A target URI within an application to which a user is redirected after successful authentication. This URI is registered with the ESDS during the [*application registration*](#bookmark99) process. Also known as a *Callback URL*.

Required Attribute An *attribute* that must have a value in order for the entity to be created or updated in the SDS.

Resource Under the industry standard representational state transfer (REST) software architecture, this is any meaningful concept around which a user interaction can occur. Resources in the ESDS REST API share [common fields](#bookmark105) such as a unique identifier value. Types of resources include data store entities, event logs, and security administration.

Resource Representation A textual representation of the current state (GET) or intended state (PUT, POST, DELETE) of a resource. The REST API supports JSON and XML representations of resources.

Restricted Data A category of data identified by ESDS for the purpose of assigning all the same permissions to data in that category. Data identified as restricted data is data that [education organizations](#bookmark112) typically must keep confidential, whether by law or for the safety of educators and students.

Role A mapping between a set of permissions and the data to which it applies. A role is typically identified by the title of a group of users that should have that role, such as a *student* or an *educator*.

Role Mapping The process of assigning an authenticated user from the local directory service of an [education organization](#bookmark112) (typically at the state or county/district level) to one of the default roles in ESDS. This creates what ESDS refers to as *federated* *users*.

Sandbox Industry standard term to refer to a non-production environment where software developers can develop and test code. ESDS includes a feature-rich sandbox environment for use in developing applications to use with the ESDS.

School District An [education organization](#bookmark112) in the SDS hierarchy of education organizations.

SEA Administrator A user who serve as ESDS administrator at the state level (SEA = State Education Agency).

Security Assertion An XML-based markup language and associated protocol

Markup Language published by the Organization for the Advancement of Structured Information Standards (OASIS). SAML is an open standard for exchanging authentication and authorization between security domains. The primary goal of SAML is to facilitate creating a single sign-on (SSO) solution between multiple web applications. ESDS uses the SAML language in combination with the *OAuth* protocol.

Security Domain A virtual security boundary shared by multiple computers, services, and applications. Applications and services within a security domain trust a common security token from an [*identity provider*](#bookmark121) that serves that domain.

Security Event Log Data in the ESDS used to record security events, such as events related to signing in and out of the SDS, creating or deleting administrators, and accessing certain restricted data.

Standard Field A field that is a part of a resource representation, as determined by the schema of the resource.

Student User in the default *Student* role group who can only view public data, Student\_Owned data, and Student\_General data.

Tenancy The set of data to which a given user has administrative access based on that user's combination of *roles* and *context*.

Tenant A single set of self-consistent data that is logically isolated within the ESDS.

Unique Identifier An identifying object, typically a string of characters unique among all objects within a given system. The unique identifier for objects in the ESDS are structured similar to the UUID standard (set by the Open Software Foundation), but deviate from that standard by adding to the ID string.

User Registration The process by which an administrator adds new users to the ESDS system, including assigning roles to those users for the purpose of establishing [*effective permissions*](#bookmark114).

Verification Code A code that verifies that the credentials provided by a user have been authenticated against the identity directory that contains that user's account. A verification code enables an application to obtain an [*access token*](#bookmark96).