

Platform Dev. Guide - Extensibility & Customization

This section describes the basics of how to extend the ODS / API platform to handle your implementation-specific needs. Since the code is open source, you can customize pretty much anything – but this section outlines the intended, built-in extension points.

Required Customizations

The Ed-Fi ODS / API is provided as a basis for an integrated education data solution. That's a pretty tall order. As such, it may not satisfy all of your specific implementation requirements. In fact, we're pretty sure it doesn't. There are a few areas where customization is almost always required.

Unique Identity Systems

The Ed-Fi ODS / API contains an interface for working with an Unique Identity System. There is a naïve implementation provided for the purposes of integration testing, but it is anticipated that installations will have their own system. To integrate with your Unique Identity System, implement the `IUniqueIdentity` interface, and register your implementation in Castle Windsor. The [Unique ID System Integration](#) technical article has details.

Database Partitioning Strategy

Large databases may require partitioning. The default ODS database connection string provider for deployed instances is the "SharedInstance" provider, which provides no database partitioning. This and other providers are pre-packaged into various OWIN startup classes (located in the `EdFi.Ods.Api/Startup` project/directory).

Deciding on a database partitioning strategy is something that should be done prior to deployment as migration to another approach may be difficult after production data is loaded and the system is online.

Database Provider	Partitioning Strategy
Sandbox	One database per application key and secret.
Shared Instance	No partitioning. All applications share one database. Database must be manually "rolled over" before each new school year.
Year Specific	Partitioning by year. Different year data is accessed through changing the URL (e.g., http://website-address/api/v2.1/2015/).

Other custom partitioning strategies are possible and have been used to meet the needs of specific implementations. For example, partitioning by-district-by-year works technically, though this approach results in partitioning the ODS into hundreds of databases.

Security Settings

The ODS / API ships with reasonable default security settings, but every implementer will have specific rules about data access – so developers should always customize the security settings in the solution. The [Security](#) section of this documentation covers the conceptual and how-to material to configure security and data access rules.

Other Customizations

The ODS / API allows for other useful customizations. A few are outlined below.

Extending the Data Model

One of the features that draws implementers to the Ed-Fi ODS / API is the support for extending the data model. Typically, in systems of this nature, the data store, data access code, API surface, bulk data loaders, documentation, and other components would all need to be changed to accommodate changes to the data model. The Ed-Fi ODS / API solution handles most of these tasks through code generation.

The [Extending the ODS / API Data Model](#) section of this documentation has an overview of the process, while detailed how-to articles provide step-by-step instructions by walking through example extensions. See, for example the [How To: Extend the ODS / API - Student Transportation Example](#) and [How To: Extend the Ed-Fi ODS / API - Student Transcript Example](#) articles for a complete walkthrough.

API Composite Resources

Developers' Guide Contents

Find out more about how to develop platforms based on the Ed-Fi ODS / API v2.4:

Composite resources allow platform hosts to create API endpoints that combine and re-shape resources – analogous to the way a database view can join information from multiple tables.

Composites resources have several useful aspects, including the ability to "alias" element names, flatten out resources, and combine multiple standard API resources into representations that can be accessed with a single API call. Composites are simple to set up, and honor the Profile and Claim Set configurations made for the standard API, and so do not introduce any new security concerns.

The [API Composite Resources](#) section of this documentation has an overview and basic how-to information. The [ODS / API Composite Resources Technical Approach](#) technical article has information about how API Composite Resources work under the hood.

Support for Cascading Updates

The ODS SQL Server data store allows for the configuration of cascading updates on entities exposed by the API. This means that if an update is made to a key value in a data row where the key value is a foreign key for existing rows in other tables, then all of the foreign key values are also updated to the new value. This is useful because the ODS is a relational database that relies on referential integrity, but the API surface would be complicated if clients had to manage key updates.

The Ed-Fi ODS / API ships with a core set of cascading updates already in place, but platform hosts can expand or modify the cascading updates supported by the system. See the article [How To: Enable Cascading Updates on ODS / API Resources](#) for details.