

Guidance on Code Values and Descriptors for LEAs

In the Ed-Fi Community today, local education agencies (LEAs) often confront the issue of how to handle code sets — referred to in the technology as "Ed-Fi Descriptors" — in their Ed-Fi ODS / API implementations. Essentially, the issue boils down to this question:

In our implementation of the Ed-Fi platform, do we map to and use the default Ed-Fi Descriptor values (adding to those as necessary), or do we use our own, current values (and ignore or remove the Ed-Fi values)?

To illustrate the nature of the question, we'll consider the following example related to student absences.

An Example: Student Absences

The Ed-Fi data model includes a set of "default" attendance event values that have been refined via field work. These values are included in the ODS / API by default. The default values are:

- In Attendance
- Excused Absence
- Unexcused Absence
- Tardy
- Early departure
- Partial

Technically, these are the default values for AttendanceEventCategory in Data Standard 3.1.

However, we can easily imagine other categories that add more specificity to these, such as "Medically Excused Absence" or "Homebound" or possibly "Service Day." We can also imagine that a LEA may not observe some of the Ed-Fi default values — perhaps the LEA has no general concept of "Excused Absence," but rather have specific sub-classes like "Medical," and "Community Service," or similar. Or, an LEA might not use the concept of "Early departure," instead using a concept of "Partial day."

Options for LEAs

Enumerations are primary classifiers of data and are therefore very important to analytics and operational use cases. The approach an LEA takes to this question matters, but many are uncertain as to best practice with regard to the Ed-Fi ODS / API and related technology.

There is no single "right" answer to this question, rather it should be a judgment based on requirements, environment, and circumstance. The following content summarizes the main approaches the Ed-Fi Alliance sees today in field work. It's offered here to help agencies choose the best path for their organization.

Note that this document also focuses on Student Information System data, where extensive localization of option sets is most common. See the Q & A section at the end of this document for more info on areas where enumeration sets are more standardized.

Generally speaking, the approaches are:

- [Approach 1. Adopt and Extend](#)
- [Approach 2. Use Local Values](#)
- [Approach 3. Use a Hybrid of Values \(Approach 2 + State Descriptors\)](#)

Detail on each approach follows.

Approach 1. Adopt and Extend

Some implementations take an adopt-and-extend approach. In this case, the LEA keeps the default Ed-Fi values but adds any additional Descriptor values that are missing from the Ed-Fi set. If there are any Ed-Fi values that should not be used, these are excluded by external documentation and downstream validations. In this approach, no Ed-Fi default values are removed.

Note that when values are added, they must always be added in the LEA namespace (to provide a technical means of identifying new values) and should be given a definition (for obvious reasons).

What are the Pros and Cons of Adopt and Extend?

PROs	CONs
------	------

Sidebar:
What are Descriptor Namespaces?

External parties will understand many of your Descriptor values, which can enhance plug-and-play interoperability.	Implementations can take longer to get started, as they need more data mapping at the outset.
The work to map local and Ed-Fi values can drive internal conversations about if current values are needed or used.	Mixing and matching sets of values often results in fuzzy or partial matches, creating minor sacrifices related to data semantics and coherence.
	Ed-Fi values may not be immediately obvious to local users — local staff must learn new values.

For those not aware of Descriptor namespaces, a Descriptor has a few parts; among those are:

- code value – what is the actual code that is transmitted?
- definition – how is this value defined?
- namespace – whose value is this?

The code value and definition should be self-evident. However, the namespace is often less understood; the namespace is an indicator of **whose** value this is.

Generally, the namespace is provided in URI format using a domain name under the control of the organization who governs this code, as in "mydistrict.edu".

As an example, all Ed-Fi governed values are in the namespace "ed-fi.org", indicating that these are governed by the Ed-Fi Alliance.

When you create your own Descriptor values, it is imperative that those values be in your namespace. No one but the Ed-Fi Alliance should publish values in the "ed-fi.org" namespace.

Approach 2. Use Local Values

Some implementations elect to use a local-values approach. In this case, the agency adds all of its Descriptor values natively and ignores all Ed-Fi values.

Descriptors added in this approach are always added in the LEA namespace, which avoids confusion with values governed by the Ed-Fi Data Standards. See the sidebar on "What are Descriptor Namespaces" for why this is important).

Also, the default Ed-Fi Descriptor values are generally not actually removed (though this is technically possible). It is generally not a problem to have two sets of values because it is easy to see all the local values and distinguish them from the unused values, by looking at the Descriptor namespace.

What are the Pros and Cons of Use Local Values?

PROs	CONS
Reduces time to start an implementation, as less data mapping is needed.	External parties are less likely to understand the values and semantics. Plug-and-play interoperability will require more work.
Value sets may be more coherent.	There's a missed opportunity with this method. Internal conversations about values can be useful, as can norming with widely used values.
Internal users understand these values, so can work with Ed-Fi data easier.	

Approach 3. Use a Hybrid of Values (Approach 2 + State Descriptors)

Some implementations are choosing to use a mix of values, most commonly a mix of local values and state values. In this case, the LEA keeps the local values in their namespace (e.g., "mydistrict.edu") but adds the additional Descriptor values that are pertinent for state reporting in state namespace (e.g., "mystate.edu").

Note

This is a relatively new pattern in the community and so practice is still evolving.

What are the Pros and Cons of using Hybrid Values?

PROs	CONS
Reduces time to start an implementation, as less data mapping is needed (assuming the number of state Descriptors are limited and most state mappings are already known).	External parties are less likely to understand the values and semantics. Plug-and-play interoperability will require more work.
Value sets may be more coherent.	There's a missed opportunity with this method. Internal conversations about values can be useful, as can norming with widely used values.
Internal users generally understand these values, so can work with Ed-Fi data more easily.	Translation of local definitions and values to state ones may result in some data loss, and therefore in lower quality analytics.

Can enhance the LEA ability to understand impacts of data for state contexts.	
---	--

Q & A

Which approach is right for my agency?

There is no right answer: you should consider the tradeoffs above. For example, if you need to get an implementation into production quickly, the use-local-values approach will provide some advantages for that. However, if your main concern is to integrate with third-party service providers quickly and inexpensively, the first approach may serve your organization better.

Why doesn't the Alliance recommend one approach or the other to ensure that the community is behaving consistently?

The Ed-Fi Alliance approach has always been to follow actual field evidence and success. Over time, as the community learns about and can point to real field work showing what works, we can provide more specific guidance on community practice.

How can a technology standard leave open the questions about allowed enumeration values? Doesn't that make a standard "non-standard"?

Note that specific Ed-Fi API specifications and certifications can and do mandate the use of specific enumeration sets. For example, the Ed-Fi Assessment API requires the use of Ed-Fi default values for most enumerations, but allows a small set to vary. See [Ed-Fi Assessment Outcomes API for Suite 2 Certification#Enumerations](#) for details.

Also, as time goes on and the Ed-Fi Community learns more, the Community will determine where additional collaboration on enumerations can be added to specifications.

Note also that while the ultimate goal of data interoperability is plug-and-play systems, in actuality it takes a long time to get to that point in any industry. Earlier stages before plug-and-play that help unlock data from systems can provide substantial value and help the overall system iterate and improve. The fact that our Community is having this conversation about a very nuanced issue is evidence of substantial past success.

What is "operational context" and how does it relate to these questions?

One concept in the Ed-Fi Community is that all data exchanges are shaped by an "operational context." This idea emerged from the long history of enumerations in Ed-Fi field work, and the observations from that work that there are few truly cross-sector contexts, or at least few truly broad cross-sector contexts.

In using enumerations, we often need to switch contexts. For example, many SIS systems have one set of enumeration values for local district operations, but then when it comes time to report to the state, that same SIS system uses another set of enumeration values, the ones defined by the state. Further, when the state reports on elements of that data to the federal government, it may use yet another set of values for the same concepts. Each of these contexts is an "operational context."

Contexts can also be community-governed and use-case-specific. The Alliance can propose and define contexts that are designed to satisfy specific use cases, such as interoperability of student outcomes data from assessment systems. Such contexts may affect not only code sets, but also the identity of elements. For example, an LEA may have a local school ID, but then the state has a different ID, and the federal government has NCES ID — three identifiers all for the same physical school.

In the ODS / API, the work around "operational context" refers to technical work looking at if data can be translated from one context into another context, with enumeration values and identity attributes defined clearly and mapped automatically.