Using LYNX to Map SIS Data to Event Names, Categories, and Attributes

Click here to view related articles.

LYNX configuration options offer several ways to represent data from your SIS in 25Live. In some cases, you have the ability to map the same data (such as term codes) into multiple 25Live fields.

The most common areas of overlap are what you can configure for the **name**, **categories**, and **custom attributes** of an event. This page is a guide to CollegeNET's recommendations on how to configure each field.

Event Name

An event's name follows a specific pattern when imported from LYNX. In addition to basic identifying data, you can choose to include several fields for a section:

- Course record number (CRN)
- Term code
- Meeting pattern codes such as MMP (multiple meeting patterns) and XL (cross-listed)

CollegeNET recommends that you enable **all** of these options for your event names, even if you plan on adding some of them as categories or custom attributes later.

This is because keyword searches throughout 25Live look at event names, making it extremely easy to find what you're looking for with as few clicks as possible. If you know a section's CRN you can identify it right away with a keyword search, and if you want to pull up an entire term then all you need is the term code. Additionally, codes like MMP and XL stand out like red flags in an event name, ensuring that you'll notice these special cases in reports and other views.

Event Categories

Depending on your SIS type, LYNX allows you to add several fields as categories using translation tables:

- Academic career code
- Campus code
- Institution code
- Section requirements
- Subterm code
- Term code

CollegeNET recommends that you enable **none** of these options unless necessary. Although they are useful in constructing more advanced queries than keyword searches offer, they carry more administrative overhead. You must create categories in 25Live that match up to each of these fields and keep them updated as your SIS changes.

(For example, creating new term code categories every year.) These will appear throughout the application and may give your category lists a cluttered appearance.

One option to consider is matching several codes from your SIS with the same 25Live category. For example, if all your term codes end in "FA", you could set up translation rules so that they all map to a category named "Fall Semester". This is less specific than using an exact term code, but it alleviates some of the administrative concerns of category mapping.

Event Custom Attributes

LYNX offers a great number of fields that can be added from your SIS as custom attributes:

- Class type
- Term code
- Subject code
- Course number
- Section number
- Level
- Credit hours
- Section type
- Status code
- Campus code
- Subterm code
- Location code
- Institution code
- Instruction method/mode
- Academic career
- Academic department/organization
- Academic group
- Division code
- Section ID
- Part of day code
- Enrollment status

Unlike event categories, administrative overhead is very low for custom attributes. Attributes are automatically created for you when you enable the LYNX option to include SIS data, and they can be deleted if you change your mind and disable them later. Because attributes are containers for data, there is no duplication when codes are different. (For example, all campus codes are added to the same attribute rather than requiring a separate category for each one.) However, any categories used by LYNX will still appear on the total list of all custom attributes used in 25Live.

You have the option of using either the "codes" or "descriptions" from your SIS data. Typically a code is an abbreviation (often in all capital letters) and a description is longer, with more natural language. It's up to you to choose which you prefer to map to your attributes.

CollegeNET recommends that you enable as many of these attributes as would be useful to you, but not to turn on everything without a plan. You don't need to drink from the firehose!

PREVIOUS: Setting Up Translation Tables

UP NEXT: Best Practices for Optimizer Partitions and Preferences