

Connecting LYNX Universal to Your SIS

LYNX Installation Steps:

- [Overview](#)
- [Planning Your Installation](#)
- [Planning Your Implementation](#)
- [API Integration](#)
 - [Workday](#)
 - [Ethos \(Banner/Colleague\)](#)
 - [Anthology Student](#)
- [Database Integration](#)
 - [Banner \(Oracle\)](#)
 - [Campus Solutions \(Oracle\)](#)
 - [Campus Solutions \(MSS\)](#)
 - [Colleague \(MSS\)](#)
 - [Colleague \(UniData\)](#)
 - [Other \(Oracle/MSS\)](#)
- [Installing the LYNX-APP](#)
 - [Windows](#)
 - [Linux](#)
- [Testing Your LYNX Connection](#)



NOTE: Custom Development

CollegeNET can't assist in designing or troubleshooting the methods your institution uses to connect your SIS DB to the LYNX Universal Data Tables. This effort is considered custom development and is solely the responsibility of your team.

This is the most time-intensive step. Expect to go through numerous rounds of testing and validation.

In This Article:

- [Field Mapping From SIS to LYNX](#)
- [Populate LYNX Data Tables](#)
- [Posting Changes to LYNX Data Tables and your SIS](#)

Your technical implementation team members will need to populate the LYNX Universal data tables [created earlier](#) with SIS data. You will also need to keep the tables updated as changes are made.

Your goals are to create the following:

1. Completed field mapping template
2. Script to populate SIS data fields into their matching LYNX data fields
3. Method to post SIS course changes to LYNX data tables
4. Method to post LYNX data table changes to your SIS

Field Mapping From SIS to LYNX

To define how your SIS data fields map to LYNX data fields, your SIS expert(s) must complete the LYNX Universal Data Mapping Template found on [SIS Data - Universal Tables and Fields](#) (*authentication required*) under "Mapping Template". This template will define how SIS fields match up LYNX fields and serves as a point of reference for the next steps and future troubleshooting.

Choose Tables to Map

The LYNX Universal data fields are extensive in order to meet the needs of a broad customer base. Not all tables in the LYNX schema need to be mapped. Before you start, you should identify which tables you want to map to your SIS.

Tables are marked as Required and Optional. CollegeNET recommends mapping as many tables and fields as possible to take advantage of the maximum functionality of LYNX and 25Live. Optional tables hold additional data associated with optional fields. (See Choose Fields to Map below.)

Required?	Category	Tables	Explanation
Required	Building Codes	BUILDING	Rooms in your SIS can either be identified by a unique facility ID or by a combination of building code and room number. This table is necessary if you choose the latter option.
Required	Classes	<ul style="list-style-type: none"> • MEETING • SECTION • STATUS • SUBJECT • TERM 	All these tables are required in order to bring section data from your SIS to 25Live.
Required	Locations	LOCATION	This table is required in order to assign rooms to classes.

Required?	Category	Tables	Explanation
Required	Instructors	<ul style="list-style-type: none"> • CONTACT • CONT_EMAIL • SEC_CONTACT • EMAIL_TYPE 	All of these tables are required if you plan to import instructor data on your classes.
Required	Exams	EXAM	This table is required if you plan to import exams from your SIS to 25Live. It is not needed if you will build exams in LYNX using the Exam Creation functionality.
Required	Cross-Listing	<ul style="list-style-type: none"> • XL_GROUP • XL_MEMBER 	These tables are necessary if you wish to designate sections as cross-listed, allowing them to be double-booked in the same place at the same time. Otherwise, these tables are optional.
Optional	Additional Class Information	<ul style="list-style-type: none"> • ACAD_CAREER • CAMPUS • DAY_PART • DEPARTMENT • INSTITUTION • INSTRUCTION • MEET_TYPE • REGION • SEC_TYPE • SUBTERM 	<p>These tables are hold reference codes used by optional fields in other tables. They are not otherwise necessary. (See Choose Fields to Map below for more information.)</p> <p>For example, if you choose to include a campus code in your section data, you will need to map corresponding data to the CAMPUS table.</p>
Optional	Instructor Roles	INSTR_ROLE	This table allows you to add specific roles for instructors on a course, such as Primary Instructor or Teaching Assistant.

Required?	Category	Tables	Explanation
Optional	Schedule25 Optimizer	<ul style="list-style-type: none"> MEET_PARTITION MEET_REQUIREMENT PARTITION REQUIREMENT 	<p>If you plan to use the Schedule25 Optimizer for classroom placement, you will find MEET_REQUIREMENT and REQUIREMENT useful. These allow you to indicate that a section must be placed in a room with specific classroom features.</p> <p>The MEET_PARTITION and PARTITION tables allow you to indicate a section's placement preference for specific areas of campus, but CollegeneT provides additional options downstream of LYNX that perform this functionality better. These tables are only recommended for customers converting from a previous interface that made use of identical features.</p>

Choose Fields to Map

Once you have chosen which tables you will map to your SIS, you will need to choose which fields in each table to map. These are marked as Required, Recommended, or Optional.

Note that some Optional and Recommended fields require the use of additional tables. For example, if you include Institution Codes on your SECTION table, then you must also map the fields in the INSTITUTION table.

Fields	Notes
Required	You must map all of these fields for LYNX to function.
Recommended	CollegeneT strongly recommends mapping all of these fields.
Optional	<p>You should consider mapping these fields in order to provide maximum information and functionality to your users. It will be easier to do it now than to incorporate them later.</p> <p>Ask your functional users which optional fields they want to track in 25Live. The Description column in the table details includes situations in which users might find each field helpful.</p>

Populate LYNX Data Tables

Once your template is complete, write a script that populates SIS data fields into their matching LYNX data fields.

At a minimum, demonstrate a proof of concept by moving at least one complete course with all the required fields from your SIS into LYNX Data tables. If you're successful, by the end of this step you will have loaded an entire term's worth of course data without error.

Here are some suggestions for successfully populating the LYNX data tables:

- A SQL script is probably easiest, but you can use any method that reliably populates the tables.

- Populate data based on the DDL of the table (i.e. data map).
- As you work through populating each field, keep your field mapping template updated.

Follow these steps to verify the successful population of the LYNX Data Tables:

1. Query the LYNX Database and verify that all required LYNX data fields are filled in.
2. Run the "E" script (from the [table creation](#) step earlier) as your LYNX user to return a count of entries in various tables. Review the results with the following criteria in mind:
 1. Do any of the tables have zero entries? If so, was that intentional? If not, review your methods for populating and retest after correcting the issue.
 2. Do you get an error when accessing any of the tables? If so, review your security settings for the LYNX user and retest after adjusting.
 3. Do the numbers seem appropriate? For example, if you have five campuses, the 'campus' table should show five entries. If you have three status codes the 'status' table should show three. Check these fields to see if any of them appear wrong.
 4. If you aren't sure of the results of this script, email your CollegeNET Technical Analyst for assistance.
3. Save the results of this query to provide to your CollegeNET Technical Analyst later.

Posting Changes to LYNX Data Tables and your SIS

You need to ensure that data in LYNX and your SIS are kept in sync. You are free to choose any of these common methods for keeping the data in sync:

- Database triggers on source SIS data tables
- SQL Agent Job to automate updates
- PowerShell or another scripting language to perform SQL action
- Database replication



If there are issues during your initial testing, **DO NOT TRUNCATE** the LYNX Data tables. This will cause data to be removed, without triggering delete actions to be sent to the CollegeNET LYNX web service.

Whatever process is used must perform updates to the LYNX Data tables through the use of **INSERT**, **UPDATE**, or **DELETE** actions. Not updating the tables with one of these methods can cause changes to not be registered properly by the LYNX triggers. If you decide to use a batch process instead of database triggers for updates, make sure you decide on an update frequency of at minimum one minute. Because LYNX checks for updates every minute, updates to the LYNX tables should not be more frequent than that.

To be a bidirectional interface, your SIS needs to update LYNX and vice versa. Develop methods to address updates in both directions:

SIS Updates LYNX

Set up a method to post SIS course changes to the LYNX Data Tables. For example, if a time change is made to a course that has already been sent to LYNX, that time change must also be made in the LYNX Data Tables, or if you build a new course in your SIS, that course must be added to the LYNX Data Tables.

Here are some suggestions for updating LYNX Data Tables when data changes are made:

- In addition to updating data in the LYNX Data Tables, Update field last_mod_dt to the current date and time for troubleshooting purposes.
- Never change the data that makes up the primary key columns. If you have to change a value that is part of the primary key, delete the existing record before inserting a new record.
- For updates to existing records in the LYNX Data Tables, merge the changes instead of deleting and re-inserting the records. Unchanged records should not be updated - **see warning below**.



Hyperguard

In an effort to deal with automated processes in a SIS that generate millions of "null" changes, LYNX detects and does not capture changes when they exceed a certain threshold for an individual table during a single synchronization cycle. Updating unchanged records in the LYNX Universal schema can increase the risk of this occurring which would result in any non-null changes in that cycle being skipped.

LYNX Updates SIS

Set up a method to post changes in LYNX data tables to your SIS. At this time, only limited fields receive updates from your hosted Series25 environment—just location assignments on sections and exams.

For example, if a 25Live academic scheduler assigns a location to the ACCT 101 course, LYNX (via the LYNX-APP and stored procedures) updates the corresponding record in the LYNX MEETING data table and sets the SIS_READY flag to 1. It's the job of the method you develop to recognize that the flag is set to 1, make the appropriate update in your SIS table, then re-set the flag to 0.

Further details:

- When a location assignment is exported from 25Live for a section, the assignment is updated in the LYNX **MEETING** table.
- When a location assignment is exported from 25Live for an exam, the assignment is updated in the LYNX **EXAM** table.
- Both the MEETING and EXAM tables have a column called **SIS_READY**. LYNX sets the SIS_READY flag to **1** to indicate that the location assignment needs to be updated in the SIS for those tables only.
- Your process must update the location assignment in the SIS and reset the **SIS_READY** flag to **0**, as well as update the last_mod_dt field to the current date and time (for troubleshooting purposes).

To verify that your LYNX and SIS update methods are working properly:

1. Make a change to a course in your SIS that exists in both your SIS and LYNX, then verify that the associated LYNX data record has been updated correctly.
2. Add a new course to your SIS, then verify that a new LYNX data record was created for that course.
3. Use SQL to update a course record on the LYNX MEETING or EXAM table and set that record's SIS_READY field, then verify that your SIS update method updated your SIS with the change.