

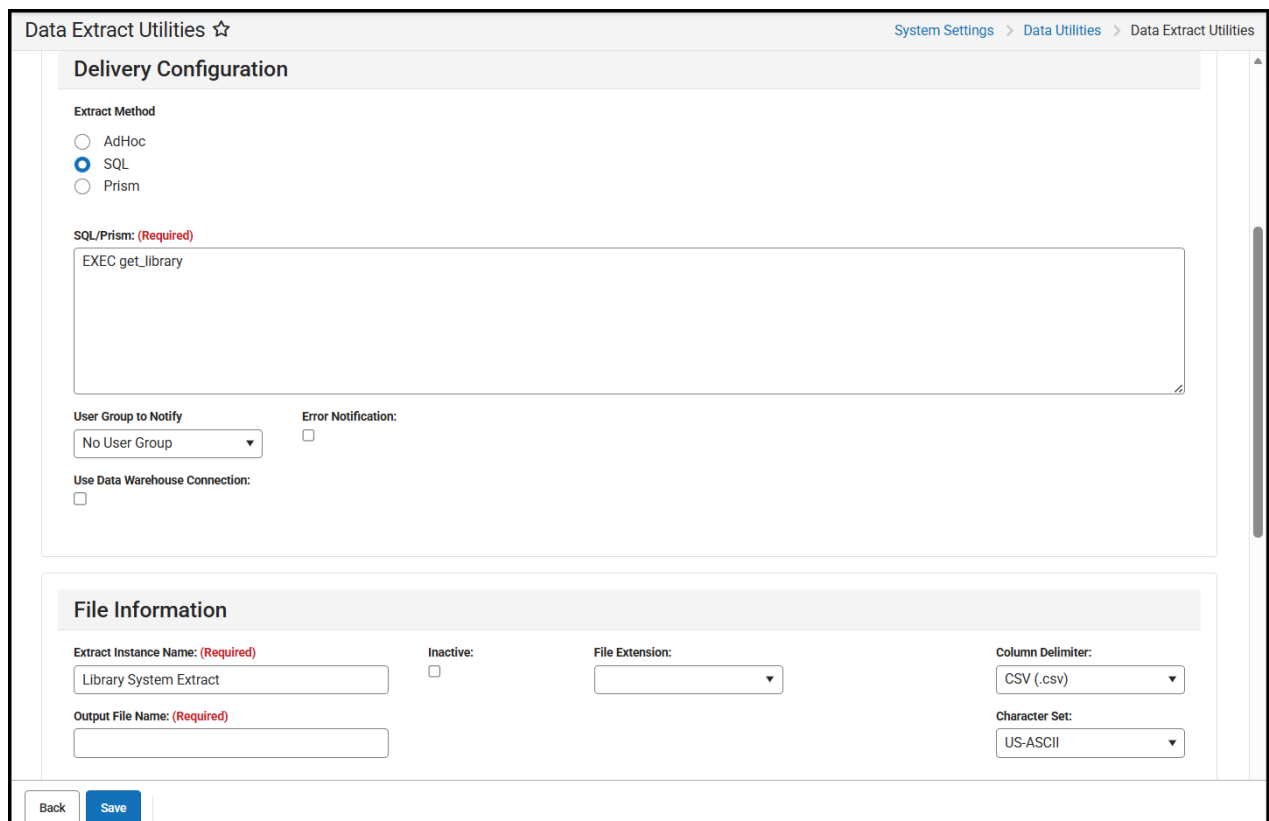
Library System Extract Configuration

Last Modified on 07/24/2026 11:08 am CDT

Tool Search: Data Extract Utilities

This guide outlines how to use the [Data Extract Utilities](#) tool to create a data extract for use with library systems (e.g., Destiny, Follett). This extract provides student demographic, enrollment, homeroom, contact, and address data so the library system can match patrons and manage checkouts by homeroom/grade.

Create the Library System Extract



The screenshot shows the 'Data Extract Utilities' configuration window. The 'Delivery Configuration' section includes options for 'Extract Method' (AdHoc, SQL, Prism), a text area for 'SQL/Prism: (Required)' containing 'EXEC get_library', a dropdown for 'User Group to Notify' set to 'No User Group', an 'Error Notification' checkbox, and a 'Use Data Warehouse Connection' checkbox. The 'File Information' section includes fields for 'Extract Instance Name: (Required)' (Library System Extract), 'Output File Name: (Required)', 'Inactive' checkbox, 'File Extension' dropdown, 'Column Delimiter' dropdown (CSV (.csv)), and 'Character Set' dropdown (US-ASCII). Navigation buttons 'Back' and 'Save' are at the bottom.

1. Select the **New** icon.
2. Select the **Extract option** and click **Submit**. A series of editors will appear.
3. Enter Admin/Vendor Provided Data. The credentials for this process must be obtained from your library system. These credentials usually include username, password, server (IP Address or server name), port number, and sometimes the path directory.
4. Select an Extract Method of **SQL**
5. In the **SQL/Prism** field, enter **EXEC get_library**
 1. This stored procedure accepts a date parameter. This determines which homeroom section to resolve for the student on that specific calendar day. If no date is passed, the query will pull all active data for today's date. If a date is added, it will pull all active data for that date. For example, **EXEC get_library @date = '01/15/2026'** will pull the snapshot for this specific date (useful for verifying homeroom on a past date).
6. Give it an **Extract Instance Name** (i.e., Library System Extract).

7. Determine the **Output File Name**.
8. Fill out the rest of the File Information fields (as applicable).
9. Determine the File Extract Format. Field definitions can be found on the [Data Extract Utilities article](#).
10. Set the Data Extract Schedule.
11. Click the **Save** icon when finished.

At this time, a user can select the appropriate buttons to test the extract, test the database connection, or run the extract. Extracts will display in a new window in the selected format.

- The **Test Extract** button generates the requested information using SQL or Ad hoc. Data is not written to the database or stored anywhere.
- The **Test Connection** button checks the connection to the entered server by attempting to write a 1-byte file to the configured location. If it is successful, a success ribbon is displayed; if it is unsuccessful, a pop-up with the reason the connection failed is displayed.
- The **Run Extract** button runs the extract, sends and stores the file where needed (in case of an issue with an auto-run extract).

Example executions

Example	Description
EXEC get_library	Get today's snapshot with SSNs included (default)
EXEC get_library @includeSSN = 0	Get today's snapshot without SSNs (privacy-safe for sharing with library vendor)
EXEC get_library @date = '01/15/2026', @includeSSN = 1	Get the snapshot for a specific date with SSNs
EXEC get_library @date = '01/15/2026', @includeSSN = 0	Get the snapshot for a specific date without SSNs
EXEC get_library @date = '08/19/2025'	Get a snapshot of the first day of school to verify initial homeroom assignments (08/19/2025 is the school's first day).

How it functions

1. Finds each student's "best" enrollment for the active school year: prioritizes Primary service type (P) over Secondary, then the most recent start date. Excludes summer school calendars and no-show enrollments.
2. Resolves the student's primary household by selecting the household where the student is an active member, and, if available, the non-secondary household with a guardian relationship.
3. Resolves the primary contact/guardian from the chosen household, picks the first guardian in the relatedPair table who is also a current household member.
4. Resolves the mailing address: from the student's active household locations, prefers the primary household's mailing address, then falls back on the secondary flag.
5. Resolves the homeroom section for @date: walks the schedule structure (Roster > Section > Placement > Period > Day) to find the section the student is rostered into on the given date, prioritizing courses flagged as homeroom or homeroom sections, then by period sequence.

Data fields

The data extracted will include the following fields in this order:

Column	Source	Meaning
StudentDistrictID	person.studentnumber	District-assigned student number
StudentSSN	identity.ssn	Student's Social Security Number Returns NULL when @includeSSN = 0
StudentStateID	person.stateID	State-assigned student identifier
StudentEnrollmentStatus	Computed from enrollment.enddate	1 = currently enrolled (end date is today or later), 0 = enrollment has ended
StudentSiteCode	school.number	School's numeric code
StudentDistNum	district.number	District's numeric code
StudentHomeroom	room.name (from resolved section)	Name of the homeroom; defaults to 'NHR' if no homeroom is resolved
StudentHomeRoomTeacherLastname	identity.lastname (teacher)	Homeroom teacher's last name
StudentHomeRoomTeacherFirstname	identity.firstname (teacher)	Homeroom teacher's first name
StudentLastName	identity.lastname	Student's last name
StudentFirstName	identity.firstname	Student's first name
StudentMiddleName	identity.middlename	Student's middle name
StudentBirthdate	identity.birthdate	Formatted MM/DD/YYYY, trimmed
StudentGender	identity.gender	Gender code (typically M/F)
StudentGradeLevel	enrollment.grade	Grade level from the enrollment record
StudentRace	identity.raceEthnicity	Race/ethnicity code
StudentEmail	v_Contact.email	Student's email from the contact view
StudentPrimaryContactLastName	identity.lastname (guardian)	Guardian's last name
StudentPrimaryContactFirstName	identity.firstname (guardian)	Guardian's first name
StudentPrimaryContactRelationship	relatedPair.name	Relationship label (e.g., "Mother", "Father")

StudentPrimaryContactSSN	identity.ssn (guardian)	Guardian's SSN Returns NULL when @includeSSN = 0
StudentPrimaryContactEmail	v_Contact.email (guardian)	Guardian's email
StudentPrimaryWorkPhone	v_Contact.workphone (guardian)	Guardian's work phone
StudentPrimaryContactCounty	address.county	County of the mailing address
StudentPrimaryContactStreetAddress	Computed from address fields	Full street address; handles PO Box vs. standard format (number + prefix + street + tag + dir + apt)
StudentPrimaryContactCity	address.city	City
StudentPrimaryContactState	address.state	State code
StudentPrimaryContactZip	address.zip	ZIP code
StudentPrimaryContactPhoneNumber	household.phone	Household phone number (empty string if null)