

Transportation Extract Configuration

Last Modified on 06/04/2024 10:27 am CDT

Tool Search: Data Extract Utility

The information within this guide outlines how to use the Data Extract Utility tool to create a data extract to be used with a transportation vendor. The secure FTP credentials for this process must be obtained from your transportation vendor. These credentials usually include username, password, server (IP Address or server name), port number, and sometimes the path directory.

Creating a New Extract

1. Select the **New** icon. A **Detail Information** section will appear.
2. Enter the **Extract Instance Name**.
3. Choose the other necessary items for adding the extract by populating and marking the appropriate fields, as defined above.
4. Click the **Save** icon when finished. You can modify the extract to determine when it should be generated.

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

- The **Test Extract** button generates the containing information requested by SQL or Ad hoc. Data is not written to the database or stored in any location.
- The **Test Connection** button checks the connection to the entered server by attempting to write a 1-byte file to the location setup. When it is successful, a pop-up displays indicating the connection is good; if it is not successful, it provides the reason the connection failed.
- The **Run Extract** button runs and generates the extract, sends and stores the file where needed (in case of an issue when it is an auto-run extract).

Extracting the Transportation Data

Data Extract Utility ☆ System Settings > Data Utilities > Data Extract Utility

Save Delete New

Extract Instance Name	Delivery Mode	Server	Frequency Scheduled	Extract Mode	Stopped	Last Run
Example extract	PULL			sql	false	
SMB Testing AZ	SMB	addev2022a		sql	false	Export ran successfully
test	SMB	test		sql	false	Error transmitting extract: Invalid object name 'campusMaster.dbo.ConfigProfile'.
Transportation	SFTP	127.0.0.1	daily 08/04/2024 01:00	sql	false	

Detail Information

*Extract Instance Name
Transportation

*File Name
studentfile

File Extension
CSV (.csv)

Column Delimiter
CSV

Character Set
US-ASCII

*Username
SFTP_username

Domain

*Server
127.0.0.1

Port : (Default SFTP=22)
22

Path
/transportation

Delivery Mode
 SMB
 SFTP Legacy (Being Deprecated)
 SFTP
 FTPS
 PULL

SFTP Key Exchange

SFTP Key Exchange Configuration

Append Date to Filename: No Date yyyyymmdd yyyyymmddhhmmssSS

Include header row?

Include double quotes?

Password

SQL Prism AdHoc

*SQL/Prism
EXEC get_transportation

Test Extract Test Connection Run Extract

Data Extract Schedule
 Stop Automated Extract
 Frequency: Daily Start Date: 08/04/2024 Start Time: 01:00

In the SQL/Prism textbox, enter “EXEC get_transportation” or any of the formats listed in the Examples.

This stored procedure contains two parameters that can be passed into it:

1. Date - 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. Example: If prior to the first day of school a district wishes to extract active data for the first day of school the query can be modified. i.e. EXEC get_transportation @date = ‘07/15/2023’ Having the date entered will pull student’s and data that are active as of that date.
2. Include SSN - If no value is passed, the query will include both the students SSN and the primary contacts SSN. If a value of 0 is passed, the query will not return SSN values. Those fields will be null. i.e EXEC get_transportation @includeSSN = 0. A value of 1 will include SSN’s and a value of 0 will return no SSN’s.

Note: The columns will still exist, but that data may not.

A single parameter can be passed, both parameters can be passed, or no parameters can be passed.

Examples

Example	Description
---------	-------------

EXEC get_transportation @date = '07/15/2023', @includeSSN = 0	Will extract all active data as of 07/15/2023 and not include SSN's.
EXEC get_transportation @includeSSN = 0	Will extract all active data for today's date and no SSN's.
EXEC get_transportation @includeSSN = 1	Will extract all active data for today's date and also include SSN's.
EXEC get_transportation @date = '07/15/2023'	Will extract all active data for 07/15/2023 and SSN's.
EXEC get_transportation	Will extract all active data for today's date and SSN's.

Data Fields

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

Field	Description
p.studentnumber AS StudentDistrictID	Local student number
CASE WHEN @includeSSN = 1 THEN i.ssn ELSE NULL END AS StudentSSN	Student SSN
p.stateID AS StudentStateID	Student state ID
CASE WHEN e.enddate < GETDATE() THEN 0 ELSE 1 END AS StudentEnrollmentStatus	0 = No longer enrolled student. 1 = Actively enrolled student.
s.number AS StudentSiteCode	School number
d.number AS StudentDistNum	District number
COALESCE(rm.[name], 'NHR') AS StudentHomeroom	Homeroom room number
i4.lastname AS StudentHomeRoomTeacherLastname	Homeroom teacher last name
i4.firstname AS StudentHomeRoomTeacherFirstname	Homeroom teacher first name
i.lastname AS StudentLastName	Student last name
i.firstname AS StudentFirstName	Student first name
i.middlename AS StudentMiddleName	Student middle name
LTRIM(RTRIM(CONVERT(CHAR(24),i.birthdate,101))) AS StudentBirthdate	Student birthdate

i.gender AS StudentGender	Student gender
e.grade AS StudentGradeLevel	Student grade
i.raceEthnicity AS StudentRace	Student race
t.transportationcode AS StudentTransportationCode	Student current transportation code
c1.email As StudentEmail	Student email
i2.lastname AS StudentPrimaryContactLastName	Primary contact last name
i2.firstname AS StudentPrimaryContactFirstName	Primary contact first name
r.[name] AS StudentPrimaryContactRelationship	Relationship to student
CASE WHEN @includeSSN = 1 THEN i2.ssn ELSE NULL END AS StudentPrimaryContactSSN	Primary contact SSN
c2.email As StudentPrimaryContactEmail	Primary contact email
c2.workphone As StudentPrimaryWorkPhone	Primary contact work phone
a.county AS StudentPrimaryContactCounty	Primary contact county
CASE WHEN a.postOfficeBox = 1 THEN 'P.O. Box ' + a.number	-
ELSE a.number + ' ' + COALESCE(a.prefix + ' ,') + COALESCE(a.street + ' ,') + COALESCE(a.tag + ' ,') + COALESCE(a.dir + ' ,') +	-
COALESCE(CASE WHEN COALESCE(a.apt, '') <> '' AND LEFT(a.apt, 1) <> '#' THEN '#' + a.apt END, '') END AS StudentPrimaryContactStreetAddress	Primary address
a.city AS StudentPrimaryContactCity	Primary city
a.state AS StudentPrimaryContactState	Primary state
a.zip AS StudentPrimaryContactZip	Primary zip code
COALESCE(h.phone, '') AS StudentPrimaryContactPhoneNumber	Primary home phone