



## **Conversion Layout Guide**

***Standard Layouts - 2014M.1338 - CO***

***Colorado conversion files***

***08/05/2016***

## Table of Contents

©2011 - 2016 Infinite Campus, Inc. All rights reserved. INFINITE CAMPUS, the INFINITE CAMPUS logo and Transforming K12 Education are registered trademarks of Infinite Campus, Inc. This publication, or any part thereof, may not be reproduced or transmitted in any form or any means, electronic or mechanical, including photocopying, recording, storage in an information retrieval system, or otherwise, by anyone other than Infinite Campus, Inc. without written permission of Infinite Campus, Inc., 4321 109th Avenue NE, Blaine, MN 55449, tel. (651)631-0000, email [info@infinitecampus.com](mailto:info@infinitecampus.com).

# People

## student

Enter one row for each student. You can include students that have graduated or left the district, if desired. Note the householdKey column. For students that live in the same home, if you have a number that you can copy to the householdKey column, the conversion will group them into a Campus household. Since many legacy systems do not have something like family number, districts often copy either the home phone number or street address into the householdKey column. If using the street address it is also a good idea to include the zipcode to ensure two families, that live in different cities but have the same street address, are not put into the same household. That seems to be the best way to get a head start at grouping students and their contacts into households.

Required Columns: studentNum, lastName, firstName, gender, birthDate  
 Unique Key: studentNum  
 Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
stateNum	string (15)			If this state has state-assigned numbers, enter the student number in this column. CA use SSID assigned by the state.
lastName	string (40)	Yes		Enter the last name of this person.
firstName	string (35)	Yes		Enter the first name of this person.
middleName	string (30)			If available, enter the middle name or initial of this person.
suffix	string (50)			A suffix such as Jr., II, III. Refer to the appendix in the Data Migration Guide for a list of codes.
nickName	string (50)			Nickname for the student, such as Bob for Robert.
gender	string (1)	Yes		Refer to the appendix in the Data Migration Guide for a list of codes.
birthDate	date	Yes		Birth date in the format mm/dd/yyyy
ssn	string (11)			Nine digit social security number with no formatting.
comments	string (255)			Additional information pertaining to this person.
hispanicFlag	boolean			Identifies whether this person has Hispanic ethnicity. Refer to the appendix in the Data Migration Guide for a list of codes.
fedRaceAI	boolean			Indicates whether the student has the "American Indian or Alaska Native" federal ethnicity designation.
fedRaceAS	boolean			Indicates whether the student has the "Asian" federal ethnicity designation.
fedRaceBL	boolean			Indicates whether the student has the "Black or African American" federal ethnicity designation.
fedRacePI	boolean			Indicates whether the student has the "Native Hawaiian or Other Pacific Islander" federal ethnicity designation.
fedRaceWH	boolean			Indicates whether the student has the "White" federal ethnicity designation.
dateEnteredUS	date			The date this student moved to the United States
dateEnteredUSSchool	date			The date this student first attended a K-12 school in the United States.
dateEnteredState	date			Date the student first entered a school in the state. This field is used ON some immunization compliance rules.
householdKey	string (100)			The householdKey is used to group students into households within and across buildings in the district. If not used, the conversion will put each student in their own household.
mailingAddress	string (255)			The address where report cards and other school communications will be mailed, in a format like "123 Main St. E". This is most likely the address where the student lives or a PO Box. If this is a PO Box then you should put the address where the student lives into the physicalAddress fields. This information will be moved to Number, Prefix, Street, Tag, Direction, Apt, and PO Box fields during conversion.
mailingCity	string (24)			When submitting an address, always include information in the city, state, and zip fields.
mailingState	string (2)			The 2 character state code.

mailingZip	string (10)			A 5 or 9 digit zipcode.
mailingCounty	string (20)			Some states provide a drop-list and other states use a text field. If your states uses a drop-list then refer to the appendix in the Data Migration Guide for a list of codes.
mailingAddressComments	string (255)			Additional information for this address.
mailingAddressPrivate	boolean			Flag to indicate whether to leave the address out of ad-hoc queries. The default is no.
mailingAddressDistrictNum	string (12)			When you include the district number for this field you can then use Ad Hoc filters to find addresses outside of your district boundaries. Refer to the appendix in the Data Migration Guide for a list of codes.
physicalAddress	string (255)			Only include information in this section when a student has two addresses; one for mailings and the other where they actually reside. These fields should include the address where the student is living. Use the format "123 Main St. E". This information will be moved to Number, Prefix, Street, Tag, Direction, Apt, and PO Box fields during conversion.
physicalCity	string (24)			When submitting an address, always include information in the city, state, and zip fields.
physicalState	string (2)			The 2 character state code.
physicalZip	string (10)			A 5 or 9 digit zipcode.
physicalCounty	string (20)			Some states provide a drop-list and other states use a text field. If your states uses a drop-list then refer to the appendix in the Data Migration Guide for a list of codes.
physicalAddressComments	string (255)			Additional information for this address.
physicalAddressPrivate	boolean			Flag to indicate whether to leave the address out of ad-hoc queries. The default is no.
physicalAddressDistrictNum	string (12)			When you include the district number for this field you can then use Ad Hoc filters to find addresses outside of your district boundaries. Refer to the appendix in the Data Migration Guide for a list of codes.
geoLocationCode	string (40)			District-defined code to identify the address location. The geo fields will be applied in one of two ways. If only the mailing address is included and there is information in the geo fields then the information will be applied to the mailing address. If information is included for both the mailing and physical addresses then the geo fields will be applied to the physical address.
geoLatitude	float (9,6)			Measured in degrees, the north or south placement of an address as it relates to a map or a globe.
geoLongitude	float (9,6)			Measured in degrees, the east or west placement of an address as it relates to a map or a globe.
geoTract	string (5)			Government-defined tract on which the address is located.
geoBlock	string (4)			Government-defined block on which the address is located.
homePhone	string (25)			The home phone number for the student. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
homePhonePrivate	boolean			Yes or no flag for whether to leave the student home phone number out of ad-hoc queries. The default is no.
cellPhone	string (25)			The cell phone number for the student. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
cellPhonePrivate	boolean			Yes or no flag for whether to leave the student cell phone number out of ad-hoc queries. The default is no.
emailAddress	string (100)			Example: joe.smith@aol.com
emailAddressPrivate	boolean			This field determines if this email address will be included or excluded from ad-hoc queries. Submit 1, T, or Y to mark this email address as private and to exclude it from ad-hoc queries.
secondaryEmail	string (100)			Example: joe.smith@aol.com
secondaryEmailPrivate	boolean			This field determines if this email address will be included or excluded from ad-hoc queries. Submit 1, T, or Y to mark this email address as private and to exclude it from ad-hoc queries.
communicationLanguage	string (5)			The default is US English. Refer to the appendix in the Data Migration Guide for a list of codes.
diplomaDate	date			Enter the date the student received their diploma.
diplomaType	string (3)			Refer to the appendix in the Data Migration Guide for a list of codes.
diplomaPeriod	string (3)			Refer to the appendix in the Data Migration Guide for a list of codes.
grade9Date	date			Enter the date when this student first entered the 9th grade. When this field is left blank, the value will be calculated for students with a 9th grade enrollment record.

cohortYearNGA	int (4)			Enter the four digit year the student is expected to graduate. When this field is left blank, the value will be calculated for students with a valid (or calculated) date in the grade9Date field.
postGradLocation	string (2)			Refer to the appendix in the Data Migration Guide for a list of codes.
postGradPlans	string (3)			Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## contact

The contact file should contain one row for each contact person of each student. That means that if a student has three contacts (mother, father, and neighbor), the student number will be on three rows of the spreadsheet. Note the householdKey column. Contacts whose householdKey is the same as the student's are combined into the student's household. As mentioned above, many districts copy each person's home phone number or street address into the family number to help group the people into Campus households. For people who are contacts for multiple students (parents with several children, neighbors listed as contact for several neighborhood children, etc), the person's name will appear on this tab once for each child. If we do not have a way to identify that those multiple contacts are really the same person, the conversion might add the same person into Campus twice. To prevent that, see if your legacy system has a unique number for each person, and enter that number into the contactKey column. For example, you might enter a row with studentNum 25 and contactKey 142, and another row with studentNum 451 and contactKey 142, if students 25 and 451 are siblings, and contact 142 is their parent. The conversion will make sure that all of the rows with the same contactKey are added into Campus as a single person, with many relationships. If the legacy system does not have a unique number for each person, try generating a value for the contactKey column that will uniquely identify each person. One common example is to concatenate the person's first name, last name and home phone number. Another is first name, last name and family number. Example: Jane+Smith+6516310000 or Jane+Smith+1122 This way, only one Jane Smith in family 1122 (or 6516310000) will be added to Campus, even if she is on the contact tab several times for different students. Although the column is called contactKey, it can contain both numbers and letters.

Required Columns: studentNum, lastName, firstName  
 Unique Key: studentNum, contactKey, lastName, firstName  
 Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
contactKey	string (200)		Yes	The district-wide unique identifier for this contact. If left blank, it is possible for the same person to be added to Campus more than once.
lastName	string (40)	Yes	Yes	Enter the last name of this person.
firstName	string (35)	Yes	Yes	Enter the first name of this person.
middleName	string (30)			If available, enter the middle name or initial of this person.
suffix	string (50)			A suffix such as Jr., II, III. Refer to the appendix in the Data Migration Guide for a list of codes.
gender	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.
birthDate	date			Birth date in the format mm/dd/yyyy
ssn	string (11)			Nine digit social security number with no formatting.
comments	string (255)			Additional information pertaining to this person.
relationship	string (40)			The relationship of the contact to the student. (i.e. Mother, Father, Parent, Neighbor, etc.)
relationshipComments	string (255)			Additional information related to the relationship between this person and the student.
contactSeq	int (3)			A number indicating the order to display this person among the student contacts.
guardianFlag	boolean			1 = Yes, 0 = No. Indicates the contact is the legal guardian of the student. CA: Contacts checked as guardian will report in the CALPADS SINF.
mailingFlag	boolean			1 = Yes, 0 = No. Yes indicates that mailings should be sent to this contact.
portalFlag	boolean			1 = Yes, 0 = No. Set to 1 if this contact should see the student information on the portal.
messengerFlag	boolean			1 = Yes, 0 = No. Set to 1 if this is a messenger contact for the student.
privateFlag	boolean			1 = Yes, 0 = No. Used when looking at the household screen on the portal.
householdKey	string (100)			If the householdKey is entered for the contact, and it matches the student householdKey, the contact is added to the student household. If householdKey is left blank or does not match, the student/contact relationship is still recorded, but the contact is NOT added to the student household. If householdKey does not match any student householdKey

				then the contact will have a household created.
streetAddress	string (255)			The "resides at" address for the contact, in a format like "123 Main St. E". This information will be moved to Number, Prefix, Street, Tag, Direction, Apt, and PO Box fields during conversion.
city	string (24)			When submitting an address, always include information in the city, state, and zip fields.
state	string (2)			The 2 character state code.
zip	string (10)			A 5 or 9 digit zipcode.
county	string (20)			Some states provide a drop-list and other states use a text field. If your states uses a drop-list then refer to the appendix in the Data Migration Guide for a list of codes.
geoLocationCode	string (40)			District-defined code to identify the address location.
geoLatitude	float (9,6)			Measured in degrees, the north or south placement of an address as it relates to a map or a globe.
geoLongitude	float (9,6)			Measured in degrees, the east or west placement of an address as it relates to a map or a globe.
geoTract	string (5)			Government-defined tract on which the address is located.
geoBlock	string (4)			Government-defined block on which the address is located.
addressComments	string (255)			Additional information for this address.
addressDistrictNum	string (12)			When you include the district number for this field you can then use Ad Hoc filters to find addresses outside of your district boundaries. Refer to the appendix in the Data Migration Guide for a list of codes.
homePhone	string (25)			The home phone number for the contact if different from household number. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
homePhonePrivate	boolean			Yes or no flag for whether to leave the contact home phone number out of ad-hoc queries. The default is no.
workPhone	string (25)			The work phone number of contact. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
workPhonePrivate	boolean			Yes or no flag for whether to leave the contact work phone number out of ad-hoc queries. The default is no.
cellPhone	string (25)			The cell phone number of the contact. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
cellPhonePrivate	boolean			Yes or no flag for whether to leave the contact cell phone number out of ad-hoc queries. The default is no.
pager	string (25)			The pager number of the contact. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
pagerPrivate	boolean			Yes or no flag for whether to leave the contact pager number out of ad-hoc queries. The default is no.
email	string (100)			Example: joe.smith@aol.com
emailPrivate	boolean			This field determines if this email address will be included or excluded from ad-hoc queries. Submit 1, T, or Y to mark this email address as private and to exclude it from ad-hoc queries.
secondaryEmail	string (100)			Example: joe.smith@aol.com
secondaryEmailPrivate	boolean			This field determines if this email address will be included or excluded from ad-hoc queries. Submit 1, T, or Y to mark this email address as private and to exclude it from ad-hoc queries.
communicationLanguage	string (5)			The default is US English. Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## employee

This file contains the employee's demographic information. Enter one row for each employee in the district. Review notes for the Contact file to determine how to enter the contactKey field.

Required Columns: employNum, lastName, firstName  
 Unique Key: employNum  
 Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
employNum	string (15)	Yes	Yes	The district-wide unique identifier for the employee.
employStateID	string (20)			The state identifier for the employee. CA use SEID.
contactKey	string (200)			If this employee is also in the student contact spreadsheet, and the person had a contact number, also enter the contact number for the person here.
lastName	string (40)	Yes		Enter the last name of this person.
firstName	string (35)	Yes		Enter the first name of this person.
middleName	string (30)			If available, enter the middle name or initial of this person.
suffix	string (50)			A suffix such as Jr., II, III. Refer to the appendix in the Data Migration Guide for a list of codes.
gender	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.
birthDate	date			Birth date in the format mm/dd/yyyy
ssn	string (11)			Nine digit social security number with no formatting.
comments	string (255)			Additional information pertaining to this person.
hispanicFlag	boolean			Identifies whether this person has Hispanic ethnicity. Refer to the appendix in the Data Migration Guide for a list of codes.
fedRaceAI	boolean			Indicates whether the staff has the "American Indian or Alaska Native" federal ethnicity designation.
fedRaceAS	boolean			Indicates whether the staff has the "Asian" federal ethnicity designation.
fedRaceBL	boolean			Indicates whether the staff has the "Black or African American" federal ethnicity designation.
fedRacePI	boolean			Indicates whether the staff has the "Native Hawaiian or Other Pacific Islander" federal ethnicity designation.
fedRaceWH	boolean			Indicates whether the staff has the "White" federal ethnicity designation.
householdKey	string (100)			The householdKey is used to group households within and across buildings in the district. If not used, the conversion will put each employee in their own household.
streetAddress	string (255)			The "resides at" address for the staff, in a format like "123 Main St. E". This information will be moved to Number, Prefix, Street, Tag, Direction, Apt, and PO Box fields during conversion.
city	string (24)			When submitting an address, always include information in the city, state, and zip fields.
state	string (2)			The 2 character state code.
zip	string (10)			A 5 or 9 digit zipcode.
county	string (20)			Some states provide a drop-list and other states use a text field. If your states uses a drop-list then refer to the appendix in the Data Migration Guide for a list of codes.
geoLocationCode	string (40)			District-defined code to identify the address location.
geoLatitude	float (9,6)			Measured in degrees, the north or south placement of an address as it relates to a map or a globe.
geoLongitude	float (9,6)			Measured in degrees, the east or west placement of an address as it relates to a map or a globe.
geoTract	string (5)			Government-defined tract on which the address is located.
geoBlock	string (4)			Government-defined block on which the address is located.
addressDistrictNum	string (12)			When you include the district number for this field you can then use Ad Hoc filters to find addresses outside of your district boundaries. Refer to the appendix in the Data Migration Guide for a list of codes.
addressComments	string (255)			Additional information for this address.
homePhone	string (25)			The home phone number for the employee. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
homePhonePrivate	boolean			Yes or no flag for whether to leave the employee home phone number out of ad-hoc queries. The default is no.



workPhone	string (25)			The work phone number of employee. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
workPhonePrivate	boolean			Yes or no flag for whether to leave the employee work phone number out of ad-hoc queries. The default is no.
cellPhone	string (25)			The cell phone number of the employee. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
cellPhonePrivate	boolean			Yes or no flag for whether to leave the employee cell phone number out of ad-hoc queries. The default is no.
pager	string (25)			The pager number of the employee. All characters that are NOT numbers are removed so you can send this information either formatted or unformatted.
pagerPrivate	boolean			Yes or no flag for whether to leave the employee pager number out of ad-hoc queries. The default is no.
email	string (100)			Example: joe.smith@aol.com
emailPrivate	boolean			This field determines if this email address will be included or excluded from ad-hoc queries. Submit 1, T, or Y to mark this email address as private and to exclude it from ad-hoc queries.
secondaryEmail	string (100)			Example: joe.smith@aol.com
secondaryEmailPrivate	boolean			This field determines if this email address will be included or excluded from ad-hoc queries. Submit 1, T, or Y to mark this email address as private and to exclude it from ad-hoc queries.
communicationLanguage	string (5)			The default is US English. Refer to the appendix in the Data Migration Guide for a list of codes.
districtStartDate	date			The date this employee started working at this district.
districtEndDate	date			The date this employee stopped working at this district.
teachingStartYear	date			The date this employee started working as a teacher in any district.
teachingYearsModifier	int (5)			If the teacher took a leave of absence then enter the number of years they were gone.
licenseNumber	string (10)			Employee folder number, license number or certification number.
ftePercent	int (3)			Enter the full time employment percent for this person.
seniority	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.
educationLevel	string (4)			The highest degree attained by this person. Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Schools And Calendars

## school

Information added to this file defines each school. Add one row for each state reporting entity - this is typically one row per school.

Required Columns: schoolNum, name, type, phone, state  
 Unique Key: schoolNum  
 Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string(7)	Yes	Yes	The district-wide unique number for this school. It will be used throughout the conversion to tie school specific information together. If this number is not the same as the number assigned by the state department of education then you will need to discuss this with your Data Services Engineer.
name	string(40)	Yes		The full name that the school is known by in the district and as defined by the state department of education.
NCESnumber	string(5)			The National Center for Education Statistics defines a number for each school site.
type	string(5)	Yes		
standardCode	string(10)			When using SIF to exchange data between Campus and another SIF-compliant system, this number is a school-specific number for identification and reporting purposes.
CEEBnumber	string(8)			The College Entrance Exam Board number assigned to the school.
courseCatalog	string(50)			A course catalog is a master file of courses used to enforce identical course numbering systems in districts with multiple schools at the same level. For example: multiple high schools having the same course number for Algebra). Smaller districts do not typically need a course catalog and may leave this field blank. If your district uses course catalogs, enter the name of the catalog used by this school.
phone	string(25)	Yes		The main phone number for this school. Used on attendance and behavior letters as well as displaying on the Campus Portal.
fax	string(25)			The main fax number for this school. Used on attendance and behavior letters as well as displaying on the Campus Portal.
email	string(100)			The email address for this school. Used on attendance and behavior letters as well as displaying on the Campus Portal.
dualEnrollment	boolean			This is a legacy field which is no longer used in the Campus application.
url	string(100)			The URL for this school. Used on attendance and behavior letters as well as displaying on the Campus Portal.
principalName	string(80)			Principal full name for this school. This information can be used as mail merge fields on form letters created in the product.
principalTitle	string(50)			The title of the principal. This information can be used as a mail merge field on form letters created in the product.
principalEmail	string(100)			Email address of the principal. This information can be used as a mail merge field on form letters created in the product.
agency	string(50)			This field refers to a separate governing agency that is used by a group of districts for special education services. It may also be used in AYP Reports for accountability information.
address	string(50)			Mailing address. The street address of this school. This information will be used on form letters as well as any state reporting extracts which include address information.
city	string(25)			Mailing address. The name of the city where the school is located. This information will be used on form letters as well as any state reporting extracts which include address information.
state	string(2)	Yes		This information will be used on form letters as well as any state reporting extracts which include address information. Refer to the appendix in the Data Migration Guide for a list of codes.
zip	string(10)			Mailing address. The 5 or 9 character zipcode for the school. This information will be used on form letters as well as any state reporting extracts which include address information.

physicalAddress	string (50)			Physical address if different from the mailing address. The street address of this school.
physicalCity	string (25)			Physical address if different from the mailing address. The name of the city where the school is located.
physicalState	string (2)			Enter information here if it is different from the mailing address. Refer to the appendix in the Data Migration Guide for a list of codes.
physicalZip	string (10)			Physical address if different from the mailing address. The 5 or 9 character zipcode for the school.
comments	string (255)			Notes about the school.
eTranscriptSchoolID	string (9)			The unique identifier assigned to a school by Docufide. This ID number is used to link electronically released eTranscripts to a school's Docufide account. Users must enter a valid eTranscript School ID in order to electronically release eTranscripts.
title1	string (15)			This field is used for federal funding. Refer to the appendix in the Data Migration Guide for a list of codes.
rcci	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.
breakfastProgram	boolean			Indicates whether the school participates in the School Breakfast Program. This field must be selected for some state reports and for the FRAM Verification Final Summary Report.
lunchProgram	boolean			Indicates whether the school participates in the National School Lunch Program. This field must be selected for some state reports and for the FRAM Verification Final Summary Report.
provision	string (1)			Indicates the Provision in which the school is participating. Refer to the appendix in the Data Migration Guide for a list of codes.
provisionBreakfastProgram	boolean			Indicates the Provision is for the School Breakfast Program (SBP).
provisionLunchProgram	boolean			Indicates the Provision is for the National School Lunch Program (NSLP).
provisionBaseYear	string (5)			Indicates the base year in which the Provision began. Refer to the appendix in the Data Migration Guide for a list of codes.
provisionEndYear	string (5)			Indicates the year in which the Provision ended. Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## department

Academic departments can be added to reflect the curriculum for each school. Fill in one row for each department in each school. Courses and teachers can be tied to the departments in other files.

Required Columns: schoolNum, departmentName

Unique Key: schoolNum, departmentName

Constraints: schoolNum must match a row in the school file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string(7)	Yes	Yes	The district-wide unique number for this school.
departmentName	string(50)	Yes	Yes	The name of this department.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## room

Enter one row for each room in each school. Sections can be tied to rooms, and they will appear on the student schedules.

Required Columns: schoolNum, roomNumber

Unique Key: schoolNum, roomNumber

Constraints: schoolNum must match a row in the school file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
roomNumber	string (10)	Yes	Yes	The name or number of this room.
capacity	int (7)			The maximum number of students the classroom will contain in a period. This is used for display purposes only. It is NOT used when scheduling students into sections.
phone	string (25)			The direct phone number to the room.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## calendar

Enter one row for each combination of schools and years. Choose the calendar names carefully. The schoolNum and calendarName are used to link data together from many of the files. Most districts use the initials of the building, plus the year. If you have historical enrollment data to load into Campus (i.e. which school and grade a student was in each year, going back "x" number of years), then include rows in this file to set up calendars for those previous years.

Required Columns: schoolNum, calendarName, startDate, endDate  
 Unique Key: schoolNum, calendarName  
 Constraints: schoolNum must match a row in the school file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The name of this calendar (length = 30) and optionally Schedule Structure name (length = 20). Used throughout Campus in drop down lists. Best practice is to name the calendar with an indication of the year and school name. For example: 10-11 Harrison High or 10-11 HS.
calendarNumber	int (3)			This field is used in some states for state reporting and usually refers to either a track or year type.
startDate	date	Yes		Together, the startDate and endDate define the range of days included in the school year. This should NOT be set to the first day/last day of school for students. There are two options for setting these fields. Option 1: Fiscal year (typically July 1 to Jun 30). Option 2: Staff year (first day teachers report back in the fall to the last possible day teachers report to work).
endDate	date	Yes		The official end of the school year. (see startDate)
summerSchool	boolean			Indicates this calendar includes summer courses.
studentDay	int (5)			This field is used in the Exact ADA calculation. The number of minutes a student is present is divided by this number, resulting in a decimal value. The value entered here should equal the number of instructional minutes the student can be in attendance for one day. If left blank, the default is 360. Not used in CA.
teacherDay	int (5)			The number of minutes in a teacher day. If left blank, the default is 480. Not used in CA.
exclude	boolean			This field is used to exclude the calendar from state reporting extracts. This should only be used if the district is tracking enrollments in a calendar which should not be state reported. If there is a mix of enrollments, where some are state reported and others are not, then this field should be set to 0.
wholeDayAbsence	int (5)			Some states use a threshold model of determining ADA. If a student is absent more than the number of minutes defined in this field, they are counted as absent for the entire day in the Approximate ADA calculation. The default is 240. Not used in CA.
halfDayAbsence	int (5)			Some states use a threshold model of determining ADA. If a student is absent more than the number of minutes defined in this field but less than the Whole Day Absence field, they are counted as absent for half the day in the Approximate ADA calculation. The default is 120. Not used in CA.
schoolChoice	boolean			School Choice allows students to attend any of various participating private and public schools, usually based on a system of vouchers, tax credits and scholarships. These programs are generally intended to give parents more input as to which primary and secondary schools their children attend. Please discuss this option with your trainer or process consultant to determine if you need to include information in this field.
assignmentRequired	boolean			This field is also used to setup School Choice. Please discuss this option with your trainer or process consultant to determine if you need to include information in this field.
comments	string (255)			Notes about the calendar

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## gradeLevel

This table defines the grade levels associated with a school in a year and how data associated with that grade level is reported in terms of attendance, membership and cumulative GPA. Add a row for each grade that is taught, in each calendar. Since a school might change grades from one year to the next, the grade level list must be entered for each calendar. NOTE: The gradeSeq column is required. It is used in the enrollment "roll-forward" process at the end of the year, to promote students from one grade to the next. It is important that the same grade levels have the same sequence number across the whole district. Further, the sequence numbers should be in order from building to building. For example if elementary schools end in 5th grade, and that happened to be sequence number 6, 6th grade in the middle schools should have sequence number 7. Another example - there might be several grade levels for kindergarten, such as KA, KP and KG, for AM, PM and full day kindergarteners. You could put sequence number 1 on all of the kindergarten grade levels, and 2 on 1st grade. That way, all of the kindergarteners will be promoted to 1st grade regardless of which KG code they were in.

Required Columns: schoolNum, calendarName, gradeLevel, gradeSeq, stateGradeLevel

Unique Key: schoolNum, calendarName, gradeLevel

Constraints: The combination of schoolNum and calendarName must match a row in the calendar file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The unique name of this calendar.
gradeLevel	string (4)	Yes	Yes	The name of the grade is how students will be listed in the system and on reports. Grade levels can be text such as KG. Since this is a text field it is recommended to use two digit numbers so grade levels sort in the correct order (10 comes after 09).
gradeSeq	int (3)	Yes		This field defines how students progress through the district. There should be only one grade level sequence through all schools in the district. For example, in a K-12 district, Kindergarten would be sequence 1. This sequence would continue through the grade levels in the elementary school. The lowest grade level in the middle school should continue the sequence defined in the elementary school. (e.g. if a 7-8 middle school, 7th grade is sequence 8)
stateGradeLevel	string (4)	Yes		This field contains the list of grade levels recognized and accepted by the state on state reporting extracts. This allows district to use locally unique names, but still have them mapped to state defined values. Refer to the appendix in the Data Migration Guide for a list of codes.
standardDay	int (5)			This is an override to the field on the Calendar tab. If a grade level has a different number of membership days (every other day kindergarten) or attends school for an amount of time different than other grade levels in the school, enter the correct value for this grade level here.
maxMembershipDays	int (5)			This is an override to the field on the Calendar tab. See standardDay notes.
wholeDayAbsence	int (5)			This is an override to the field on the Calendar tab. See standardDay notes.
halfDayAbsence	int (5)			This is an override to the field on the Calendar tab. See standardDay notes.
schoolChoiceSeatCount	int (10)			This field is used to setup School Choice. Please discuss this option with your trainer or process consultant to determine if you need to include information in this field.
kindergartenCode	string (5)			Some states require information in this field. Refer to the appendix in the Data Migration Guide for a list of codes.
excludeGPA	boolean			Exclude this grade level from GPA calculations.
excludeState	boolean			Exclude this grade level from state reporting.
assignmentExempt	boolean			This field is used to setup School Choice. Please discuss this option with your trainer or process consultant to determine if you need to include information in this field.
standardSIFCode	string (10)			The Standard Code field is used in SIF (Schools Interoperability Framework) to map an irregularly named grade level to a standard grade level. It is best to leave this field blank, unless the school is using integration capabilities that require a SIF or NCES code as directed by Infinite Campus.
excludeFromSIF	boolean			It is best to leave this field unchecked, unless the school is using integration capabilities that require a SIF or NCES code as directed by Infinite Campus.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



## term

Enter one row for each term in the calendar. Usually terms are tied to when report cards come out. Only one set of terms is needed per calendar. For example, report cards are printed at the quarter breaks and semester breaks, just set up the quarters, since the semester report cards can be tied to the end of 2nd quarter. NOTE: If calendars are created for previous school years (see above), terms, periods, etc are not necessarily needed. The student enrollment history can be loaded into Campus with just the previous calendar and gradeLevel records. Terms, periods and other calendar information are needed for prior years only if you will also be loading detailed schedule or attendance data for those years.

Required Columns: schoolNum, calendarName, termName, startDate, endDate

Unique Key: schoolNum, calendarName, termName

Constraints: The combination of schoolNum and calendarName must match a row in the calendar file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The unique name of this calendar.
termName	string (10)	Yes	Yes	The description of the term (up to 10 characters), like "3rd Qtr" or "Q3".
startDate	date	Yes		The first day of classes for students during this term. Term dates should not overlap but it is permissible to have gaps between terms if those days are non-instructional days such as winter break or spring break.
endDate	date	Yes		The last day of classes for students during this term.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## periodSchedule

Periods define the units in a student day for their schedule. Infinite Campus calls these schedules, "Period Schedules". A school needs at least one Period Schedule; if students attend different classes on different days (such as an A/B day rotation), or if periods last different lengths on different days (such as scheduled early release days) multiple period schedules should be created. For a school that does not have rotations, we need one row in this file with a name like "Main" or "Day". For schools with rotating schedules, such as A/B days, enter one row for each schedule. Just enter the names of the standard bell schedules. Do not include schedules for teacher staff meetings, late starts, or parent conferences because this is handled using a different method.

Required Columns: schoolNum, calendarName, scheduleName, scheduleSeq

Unique Key: schoolNum, calendarName, scheduleName

Constraints: The combination of schoolNum and calendarName must match a row in the calendar file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The unique name of this calendar.
scheduleName	string (20)	Yes	Yes	The name of the schedule. (i.e. A, B, M, T, W, Th, F, 1, 2, 3, etc.)
scheduleSeq	int (3)	Yes		The order for this schedule. (i.e. 1 for A Day, 2 for B Day, etc.)

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## period

Each Period Schedule needs at least one period. Enter one row for each period in each Period Schedule. If you have multiple Period Schedules, enter a complete set of periods for each schedule. Schools with A/B days often have the same list of periods and times for both days. We still need the list of periods for each schedule. (You could set up the periods for the A schedule, copy them to another set of rows, and change the scheduleName on the copies to "B".)

Required Columns: schoolNum, calendarName, scheduleName, periodName, periodSeq, startTime, endTime  
 Unique Key: schoolNum, calendarName, scheduleName, periodName  
 Constraints: The combination of schoolNum, calendarName and scheduleName must match a row in the periodSchedule file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The unique name of this calendar.
scheduleName	string (20)	Yes	Yes	The name of the schedule. (i.e. A, B, M, T, W, Th, F, 1, 2, 3, etc.)
periodName	string (10)	Yes	Yes	The name of the period to display on the schedule. (i.e. 1, 01, AM, PM, HR, 4a, 4b, 4c, etc.)
periodSeq	int (3)	Yes		The order this period is listed on the schedule. Make sure you start with 1. A period sequence of zero will generate errors after data conversion.
startTime	time	Yes		The start time for this period in HH:MM 24-hour format.
endTime	time	Yes		The end time for this period in HH:MM 24-hour format.
lunchMinutes	int (3)			The number of minutes taken out of this period for lunch.
nonInstructional	boolean			This field is used to determine if the period should be included when calculating the number of instructional minutes in a day. For example you might have an Activities period at the end of the day used to track students in sports or clubs but it is not considered an academic period so you would set the value to 1.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## day

This information defines what days are school days, if attendance is to be taken by teachers, and which period schedule is used. The conversion program can populate the day table automatically, so you can leave this file blank if desired or you can add the days for your days off such as Thanksgiving, Winter Break, Teacher Conferences. The program creates a day record for every date within each calendar. Weekdays that fall within a term will have the schoolDay, instruction and attendance flags turned on. If a calendar has multiple period schedules (rotations), the program creates one set of day records, and initializes the rotating schedule. After the conversion, you can use the Campus software to change the schoolDay flags for holidays, snow days, etc. If you have a more complicated rotation, or you know that you will have many exceptions to our automated conversion, you can fill in the day file. On each date, include the scheduleName (from the periodSchedule file) matching the rotation that was or will be used on that date.

Required Columns: schoolNum, calendarName, date, scheduleName, schoolDay, instruction, attendance  
Unique Key: schoolNum, calendarName, date  
Constraints: The combination of schoolNum, calendarName and scheduleName must match a row in the periodSchedule file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The unique name of this calendar.
date	date	Yes	Yes	The calendar date that this entry represents, formatted mm/dd/yyyy.
scheduleName	string (20)	Yes		The name of the period schedule that will be used on this date.
schoolDay	boolean	Yes		This field indicates the building is open. Either students are attending classes or it is an in-service day where teachers are in the building. If school is open, and teachers are holding classes and recording attendance, then schoolDay, instruction and attendance are set to 1. If the day is an in-service day, only the schoolDay will be set to 1.
instruction	boolean	Yes		This field indicates teachers will be in the classroom.
attendance	boolean	Yes		This field indicates attendance will be taken.
startTime	time			This information is not typically necessary during data conversion. Please discuss this with your trainer if you think you need to include this information.
endTime	time			This information is not typically necessary during data conversion. Please discuss this with your trainer if you think you need to include this information.
duration	int (5)			Number of instructional minutes on this day. (This is only an override for days that have a different number of minutes than a regular school day.)
dayEventCode	string (20)			This field is generally used for non-school days and is used to identify why there is no school such as a holiday or in-service day. District-specific codes can also be added to the dictionary. Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string (250)			Notes about this day.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Grading Setup

## creditGroup

Credit Groups are used to categorize student scores on the transcript and in the Grad Planner (The Grad Planner shows which types of courses a student needs to complete to graduate.)

Required Columns: creditGroupName, creditTypeName  
Unique Key: creditGroupName, creditTypeName  
Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
creditGroupName	string (50)	Yes	Yes	The name for a group of credit types, such as "HSCredits" or "MSCredits".
creditTypeName	string (50)	Yes	Yes	A "credit type" or "credit category" in which students can earn credit. (i.e. Math, English, Science)
acceptOverflow	boolean			Set to 1 when credit from other types can overflow into this type. (Ex: Electives)
pesName	string (50)			The application only allows two credit group levels; parent and child. The PES Name is only valid ON the child level. PES refers to Post-Secondary Electronic Standards Council which is used as validation for electronic transcripts. PES xml format is a common standard accepted by many colleges and universities. The pesName field should contain the list of acceptable PES course standards. An administrator can then map the curriculum standards to a PES approved curriculum standard. If the administrator fails to map a credit group, a valid transcript will still be generated, but the course in question will be seen as Miscellaneous. (For more information about PES standards, please see the PES website: <a href="http://www.pes.org/">http://www.pes.org/</a> ). Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string (250)			Notes about this item

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## scoreGroup

**SOME NOTES ON GRADING:** There are several places to put grades and scores in the conversion files: \*

- Grades from prior years go in the transcript file. There are no separate grading curves for transcripts. The grade, percent, and gpa values are all entered directly on the transcript record.
- \* Grades for current year classes can be entered in the Score file found after Rosters. Grading curves for these grades are entered in the scoreGroup file. (Score Groups are like grading curves, but they have extra values for calculating credits and GPAs.) Since grades are eventually posted to transcripts, it is not necessary to send grades for the current year's completed terms in the transcript sheet. It works better to send current year grades in the Score file, then post the grades to transcripts after the conversion, or at the end of the year.
- \* Grades for the teacher's gradebook assignments are entered in the Gradebook file. The grading curves are found there too.

**Score Group:** A list of the scores (or letter grades) that can be given, put into logical groupings. You might have a score group for A-F grades, another for Pass/Fail grades. Elementary standards typically use a rubric which is entered in the Rubric file.

Required Columns: scoreGroupName, score, scoreName, scoreSeq, passingScore  
 Unique Key: scoreGroupName, score  
 Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
scoreGroupName	string (30)	Yes	Yes	A name for a group of scores, such as A-F, Pass/Fail, Elementary, etc.
score	string (10)	Yes	Yes	A score that can be given when grading with this score group. (i.e. A+, A, A-, P, N, I)
scoreName	string (30)	Yes		The long description of the score. (i.e. A+, B-, Partial, Incomplete)
scoreSeq	int (5)	Yes		The order to display this score within the group.
minPercent	float (6,3)			The minimum percent needed to achieve this score. (i.e. 90 for an A-, 83 for a B, etc). This is only required if you are using composite grading. Otherwise, the grading scale score maps to the score group score and percents are not used from the score group.
passingScore	boolean	Yes		Enter 1 when the score is a passing score as defined by the district and/or state. Used for color coding in the grade book.
creditCoeff	float (6,3)			Any multiplier of credit for this score attainment. Can be used to increase or decrease the credit value granted for achieving this score. Generally this is 1.
weightedGPAAValue	float (7,4)			The adjusted GPA value for this score, such as 5.0 for an A, 4.0 for a B, etc. Generally applied to AP type courses. If you use bonus points or weighted GPAs, you need to create two score groups. One score group for the weighted and one for the unweighted. **Both the weighted and unweighted columns must be completed for all score groups.
unweightedGPAAValue	float (7,4)			The standard GPA value for this score, such as 4.0 for an A, 3.0 for a B, etc.
bonusPoints	float (7,5)			The number of bonus points associated with this score. (i.e. An A may be worth 4 but a bonus point of 1 would make it a 5.)

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## gradingTask

Grading tasks are items for which a student will receive a grade, such as Quarter Grade, Progress, etc. These tasks are created and then assigned to a course. Once a task is assigned to the course, the teacher has the ability to grade the task. The grading task appears on report cards and in the portal when viewing assignments. To place student grades on a transcript, at least one Grading Task should be marked as Post-To-Transcript.

Required Columns:   taskName  
Unique Key:           taskName  
Constraints:          None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
taskName	string (75)	Yes	Yes	The name of an item that students will receive as a report card grade. (i.e. Quarter Grade, Final Exam, Final Grade, Progress Grade, etc)
taskNumber	string (10)			A number identifying this task.
taskSeq	int (10)			The order for this task to appear in lists.
taskCode	string (25)			Enter a code if your state or district requires one.
postToTranscript	boolean			This field controls which grades are posted to transcripts. Set the value to 1 if scores for this task should appear on transcripts. (ex: usually Final Grades are posted to transcripts, but Progress grades are not).
stateReported	boolean			Identifies this as an item that is included on one or more state reports. This field is not used in CA.
comments	string (250)			Notes about this task

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## rubric

Use this to set up rubrics used on standards.

Required Columns: rubricName, score, passingScore

Unique Key: rubricName, score

Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
rubricName	string(30)	Yes	Yes	A name for a rubric
score	string(10)	Yes	Yes	A score that can be given when grading with this rubric. (i.e. A+, A, A-, P, N, I)
scoreName	string(30)			The long description of the score. (i.e. A+, B-, Partial, Incomplete)
scoreSeq	int(5)			The order to display this score within the group.
passingScore	boolean	Yes		Enter 1 when the score is a passing score as defined by the district and/or state. Used for color coding in the grade book.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



## standard

Setting up standards is an intensive process and there are a lot of variables which impact the "right" way to do it. It is recommended to discuss this with your client exec or trainer before submitting this information.

Required Columns: standardGroupLevel1, standardName, rubricName, abbreviation  
 Unique Key: standardGroupLevel1, standardGroupLevel2, standardGroupLevel3, standardGroupLevel4, standardName  
 Constraints: rubricName must match a row in the rubric file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
standardGroupLevel1	string (50)	Yes	Yes	standardGroupLevel1 is the top level standard such as "1st Grade Standards" or "Work Habits". Every row should include information in this column.
standardGroupLevel1Seq	int (3)			The order in which the standard is displayed.
standardGroupLevel2	string (50)		Yes	standardGroupLevel2 is the second level used for grouping standards. If you include information in this field then you must also include information in standardGroupLevel1.
standardGroupLevel2Seq	int (3)			The order in which the standard is displayed.
standardGroupLevel3	string (50)		Yes	standardGroupLevel3 is the third level used for grouping standards. If you include information in this field then you must also include information in standardGroupLevel1 and standardGroupLevel2.
standardGroupLevel3Seq	int (3)			The order in which the standard is displayed.
standardGroupLevel4	string (50)		Yes	standardGroupLevel4 is the fourth level used for grouping standards. If you include information in this field then you must also include information in standardGroupLevel1, standardGroupLevel2, and standardGroupLevel3.
standardGroupLevel4Seq	int (3)			The order in which the standard is displayed.
standardName	string (75)	Yes	Yes	A short name for the standard. (i.e. "Knows phone number", "Follows directions", etc). It is important to note, this field must be unique throughout this file. (Ex: You cannot include "Follows directions" in two rows where the standardGroupLevel names are different. Instead you would have two rows, one with "Follows directions" and the other slightly different like "Follows directions.").
rubricName	string (80)	Yes		The name of the rubric to use for scoring this standard group.
standardNumber	string (10)			A number identifying this standard. The only place this is visible is in the report card set-up (System Admin > Preferences > reports). It does not display anywhere else.
abbreviation	string (10)	Yes		This is a required field and makes finding the correct standard easier when teachers are grading the standards.
code	string (25)			Enter the pre-defined - 25 character - standard code if necessary (from the state or district)
standardSeq	int (10)			The order for this standard to appear in lists. Although it is possible, sequences should not be duplicated.
postToTranscript	boolean			This field controls which grades are posted to transcripts. Set the value to 1 if scores for this task should appear on transcripts. (not common)
powerStandard	boolean			This field identifies important standards (like those related to NCLB reporting) and is used in generating reports. There is no connection between NCLB and the standards bank. This field is used on the Standards Report (Grading & Standards > Reports > Course Standards) that shows what standards have this box marked, but it does not do anything in the system.
stateReported	boolean			Identifies this as an item that is included on one or more state reports.
oneTime	boolean			Enter 1 for standards that can only be graded once per student. (Ex: Can tie shoes).
comments	string (250)			Additional information about this standard. This information is only shown on the setup screen.
description	string (unlimited)			This is another field for describing this standard and is only shown on the setup screen.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## gradingComment

Grading Comments are designed for use on report cards. Teachers can select from this list of predefined comments when scoring grading tasks. The teachers will see this list of comments if they have chosen to display this option in the teacher preferences. Depending on system preferences, teachers can have the ability to create their own comments.

Required Columns: comment

Unique Key: schoolNum, comment

Constraints: When schoolNum has a value (other than blank or NULL) then schoolNum must match a row in the school file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)		Yes	When this comment is for a particular school, enter the school number here. Leave blank to set up a comment for all schools.
code	string (6)			This code will only display for teachers, it will not be printed on report cards.
comment	string (200)	Yes	Yes	A pre-written comment for teachers to use when grading. This comment can be 255 characters, including spaces, punctuation, etc. Be aware that with a longer comment, it may be truncated at some point when viewing a report card. It is best to keep these to a relatively short length.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Students

## enrollment

Enter one row for each enrollment for each student. If a student changed grades mid-year, or left a school temporarily, you might have two rows with the same studentNum / schoolNum / calendarName combination. The start status and end status codes are often defined by the state. Review the Data Migration Guide for your state's codes. Campus requires an enrollment record for each year that a student attends school (for as many school years back that you plan to convert). We have seen some legacy systems that only have an enrollment record when the student's school changes. For example, a junior high student might have a record in 2001 when they enter 6th grade, then nothing more until 2004 showing the student switching to the high school. Some districts are able to generate the extra rows for the additional enrollments; others ask Campus to do the extra work for them. Talk to your conversion specialist if needed.

Required Columns: studentNum, schoolNum, calendarName, gradeLevel, serviceType, startDate, startStatus  
 Unique Key: studentNum, schoolNum, calendarName, startDate  
 Constraints: The combination of schoolNum, calendarName and gradeLevel must match a row in the gradeLevel file.  
 studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
schoolNum	string (7)	Yes	Yes	The district-wide unique school number that this student enrollment is associated with.
calendarName	string (30)	Yes	Yes	The calendar name for this enrollment.
gradeLevel	string (4)	Yes		The grade level in which the student is enrolled.
serviceType	string (1)	Yes		Indicates the type of enrollment and the intended service the student receives. Refer to the appendix in the Data Migration Guide for a list of codes.
noShow	boolean			Identifies an enrollment record where the student registered but did not show up to school.
startDate	date	Yes	Yes	The date on which this student enrollment began for this grade level. Should be on or after the calendar start date.
startStatus	string (4)	Yes		Enrollment Start Status indicates the reason for beginning the student enrollment, whether entering the school or district or continuing on from another grade level. Refer to the appendix in the Data Migration Guide for a list of codes. CA: Not reported to CALPADS, but important in determining which enrollments do report.
startComment	string (250)			Additional information about this enrollment.
endDate	date			The date on which the student enrollment ended. This is expected to be less than or equal to the calendar end date. Leave blank for the current enrollments.
endStatus	string (4)			End Status indicates the reason for ending the enrollment, such as leaving the school or moving to another grade. Refer to the appendix in the Data Migration Guide for a list of codes.
endComment	string (250)			Additional information about this enrollment.
endAction	string (1)			This field is available for information purposes. It identifies a reason (demote, promote, retain) for ending the enrollment. Refer to the appendix in the Data Migration Guide for a list of codes.
classRankExclude	boolean			When set to 1, this enrollment record, and any grades associated with it, will not be used when calculating class rank. Ex: You might set this to 1 for parochial students enrolled for just a few courses.
teamName	string (40)			For sections taught in teams, the name of a team/schedule group.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## specialEd

Use this file to enter special education information.

Required Columns: studentNum, schoolNum, calendarName, effectiveDate

Unique Key: studentNum, schoolNum, calendarName, effectiveDate

Constraints: The combination of studentNum, schoolNum and calendarName must match a row in the enrollment file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string(15)	Yes	Yes	The district-wide unique identifier for the student.
schoolNum	string(7)	Yes	Yes	The district-wide unique school number that this student enrollment is associated with.
calendarName	string(30)	Yes	Yes	The calendar name for this enrollment.
effectiveDate	date	Yes	Yes	The effective date of the Special Ed program.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## freeReducedEligibility

The USDA Food and Nutritional Services agency requires that districts track student eligibilities for federal reimbursement purposes. Eligibility records are updated when a household applies for benefits, generally at the start of a school year or during the school year. When a household is approved for free or reduced meal benefits, student eligibilities are active for the current school year, plus an additional 30 days into the next school year. Use this file to provide eligibility information.

Required Columns: studentNum, eligibility, certifiedType, startDate, endDate, endYear  
 Unique Key: studentNum, startDate  
 Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
eligType	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.
eligibility	string (1)	Yes		This field indicates the level of meal service benefits awarded to a student/household *Free - student is considered eligible for full meal service assistance and will be allowed free meal items if household income is below 130 percent of the federal poverty guidelines or if he/she qualifies for benefits based on categorical need (see Certified Type, next entry). *Paid - student is not considered eligible for any meal benefits and must pay full price for purchased meal items. *Reduced - student is considered eligible for reduced-priced meal items if household income is between 130 and 185 percent of the federal poverty guidelines. *Non-reimbursable - This option is never assigned or populated as an eligibility for students. It is for the purpose of meal processing and is often used for adult staff members (though it is not necessary for them to be assigned this Eligibility for correct meal processing). Refer to the appendix in the Data Migration Guide for a list of codes.
certifiedType	string (4)	Yes		Determination reason for value displayed in Eligibility field. *Categorical - Student qualifies for individual full meal benefits - eligibility: "Free" *Direct - Student "directly certified" for full meal benefits - eligibility: "Free" *Foster - considered a foster child - eligibility based on his/her income and is likely "Free." *Income - eligibility based on household income and completed app - eligibility: "Free" "Reduced" or "Paid" *Migrant - qualifies for full meal benefits - eligibility: "Free" *Override - district chose to override assigned Eligibility. *Runaway - Student qualifies for RYHA as runaway - eligibility: "Free" *Homeless - qualifies for RYHA as homeless - eligibility: "Free" *Temp/No Income - qualifies for temporary meal - eligibility: "Free" *Declined - Student declined awarded meal benefits *Denied - Student was denied meal benefits *Did Not Apply - Student did not apply for meal benefits. Refer to the appendix in the Data Migration Guide for a list of codes.
startDate	date	Yes	Yes	The first day on which the Eligibility is active.
endDate	date	Yes		The last day on which the Eligibility is active. The eligibility End Date field indicates the last date on which the student is eligible for the eligibility, generally 30 days into the next school year.
endYear	int (4)	Yes		The school year for which the Eligibility applies. (Ex: when the school year starts in 2010 and ends in 2011 you should enter 2011). Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Employees

## employment

This file contains the job information, which building the employee is in, whether the person is a teacher, health worker, counselor, etc. Most districts just enter information for the active year. If you would like to convert the teachers' building history into Campus, enter multiple rows for each teacher, with the appropriate start and end dates.

Required Columns: employNum, schoolNum  
 Unique Key: employNum, schoolNum, startDate, assignmentCode  
 Constraints: When departmentName has a value (other than blank or NULL) then the combination of schoolNum and departmentName must match a row in the department file.  
 employNum must match a row in the employee file.  
 schoolNum must match a row in the school file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
employNum	string (15)	Yes	Yes	The district-wide unique identifier for the employee.
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
departmentName	string (50)			The department that the teacher is assigned to. If entered, must match a name from the department conversion file.
startDate	date		Yes	The date this employee started working at this school.
endDate	date			The date this employment or teaching at this schoolNum ended. Generally not used during conversion but can be included for historical purposes.
type	string (4)			Refer to the appendix in the Data Migration Guide for a list of codes. CA: Types 10, 11, 12, 25, 26, and 27 used in State Reporting.
title	string (30)			The employee job title.
fte	float (7,3)			Percent of time employee is allocated to this assignment.
assignmentCode	string (6)		Yes	CA: Not used in state reporting. States with a list are AZ, IA, MN, MO, OK, and VA. Refer to the appendix in the Data Migration Guide for a list of codes.
teacher	boolean			1 = Yes, 0 = No (default). Determines if this person will appear in the Primary Teacher and Secondary Teacher dropdown lists on Course Sections.
specialED	boolean			1 = Yes, 0 = No (default). Determines if this employee name will appear in the Special Ed Staff dropdown list on the Special Ed > Team Members tab.
program	boolean			1 = Yes, 0 = No (default). There is no logic tied to this field. However, staff members can still be marked accordingly and information can be pulled in Ad hoc Filter Designer if needed.
behavior	boolean			1 = Yes, 0 = No (default). Determines if this person will appear in the Behavior Admin Staff Name dropdown list on student Behavior events and on the Attendance Letter screen.
health	boolean			1 = Yes, 0 = No (default). Determines if this person will appear in the Administered By dropdown field on Health Office Visits.
advisor	boolean			1 = Yes, 0 = No (default). Determines if this person will appear in the Advisor Staff dropdown field on the Special Ed > Team Members tab. Also determines if this person will appear in the Advisor Staff dropdown field on the PLP > Team Members tab.
supervisor	boolean			1 = Yes, 0 = No (default). Determines if this person will appear in the District Assignment editor as an indication they are a supervisor of an employee. If no one in the district has this option selected, the Supervisors dropdown list will not show on the editor.
counselor	boolean			1 = Yes, 0 = No (default). Allows staff to be assigned as a student's counselor on the Fill Counselor tool.
foodservice	boolean			1 = Yes, 0 = No (default). There is no logic tied to this field. However, staff members can still be marked accordingly and information can be pulled in Ad hoc Filter Designer if needed.
excludeReferral	boolean			1 = Yes, 0 = No (default). Creating a District Assignment record adds a person to the Referral Name dropdown list located in Student Information > General > Behavior. Setting this field to '1' removes this person from the list.
approver	boolean			1 = Yes, 0 = No (default). Indicates this person is a designated request processor and has the ability, depending on assigned tool rights, to approve

				or deny self service requests. The staff person will see the requests needing approval in the Process Inbox . In addition to this field, those staff members who will be responsible for approving staff member's My Data requests will also need tool rights to the Request Processor.
framProcessor	boolean			1 = Yes, 0 = No (default). Indicates this person is a designated FRAM processor and has the ability, depending on assigned tool rights, to process Meal Benefits Applications via the Household Applications or Online Applications tools.
supervisorEmployNum	string (15)			If this person has a supervisor and you want to include this information then add the employee number of the supervisor in this field.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## credential

Multiple credentials can be entered for a staff member. The staff member may be highly qualified in two subject areas or may have multiple degrees. Credentials are optional to use for staff information. However, for detailed accountability reporting, Federal EDEN reporting and NCLB data collection, each staff member should have the proper credentials listed. Use this layout for your highly qualified staff. Talk to your Campus team to discuss converting other staff credentials.

Required Columns: employNum, startDate  
Unique Key: employNum, coreSubject  
Constraints: employNum must match a row in the employee file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
employNum	string (15)	Yes	Yes	The district-wide unique identifier for the employee.
startDate	date	Yes		Date on which the credential was given to the staff person.
endDate	date			Date on which the credential expired.
subjectType	string (4)			Refer to the appendix in the Data Migration Guide for a list of codes.
coreSubject	string (3)		Yes	Enter information in this field when subjectType is set to "CORE". Refer to the appendix in the Data Migration Guide for a list of codes.
HoussePassed	boolean			Indication the staff person passed the HOUSSE exam.
HousseDate	date			Date on which the staff person passes the HOUSSE exam.
subjectCompetency	string (4)			Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



# Schedules

## course

Enter the course list for each calendar, one course per row. The course number must be unique within a calendar. You cannot have course "1234" twice in the same calendar, but you can have "1234" in the HS 03-04 calendar, and again in the HS 04-05 calendar. You could also have course "1234" in both the HS 04-05 and MS 04-05 calendars. Take note of the terms, schedules, and periods columns. For a course that meets the whole year, in a calendar with quarters, terms should be set to 4. If a calendar has A and B period schedules, a course that meets on both days should have the value 2 in the schedule column.

Required Columns: schoolNum, calendarName, courseNum, courseName

Unique Key: schoolNum, calendarName, courseNum

Constraints: The combination of schoolNum and calendarName must match a row in the calendar file. When creditGroupName and creditTypeName have values (other than blank or NULL) then the combination of creditGroupName and creditTypeName must match a row in the creditGroup file.

When departmentName has a value (other than blank or NULL) then the combination of schoolNum and departmentName must match a row in the department file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The name of the calendar under which this course is taught.
courseNum	string (13)	Yes	Yes	The unique course number within this calendar for this course.
courseName	string (30)	Yes		The short name for the course.
standardsBased	boolean			1 = Yes, 0 = No. Identifies courses that use only standards.
active	boolean			Enter information here if you want to control which courses are active. If no information is included then, after sections are inserted the conversion will check to see which courses have sections assigned and mark them as active. Courses with no sections will be marked as inactive.
subjectType	string (4)			Used for NCLB- Accountability Reporting. Refer to the appendix in the Data Migration Guide for a list of codes. CA: Used for reporting NCLB Core status to CALPADS.
coreSubject	string (3)			This field is displayed on the screen when the Subject Type field is set to CORE. Only include information in this field when Subject Type is set to CORE. Refer to the appendix in the Data Migration Guide for a list of codes.
stateCode	string (20)			The State Course Code for this course. This code drives certain state reports.
departmentName	string (50)			The name of the department that offers this course. If entered, must match a name from the department tab in the district conversion file.
creditGroupName	string (50)			The name of the credit group, if credit can be earned for this course.
creditTypeName	string (50)			The name of the credit type that can be earned for this course.
priority	int (5)			Used by the scheduling wizard for scheduling priority. Courses with a higher priority will be scheduled first when using the scheduling wizard.
maxStudents	int (5)			The maximum number of students allowed per section. This number indicates how many students will be scheduled into this class with the schedule wizard. Can be overridden for a particular section on the section tab.
terms	int (3)			The number of terms this course meets during the year. (ex: a full year course in a calendar with quarters has terms = 4)
schedules	int (3)			If the school is on a rotating day schedule, enter the number of days this course meets. (the default is 1)
periods	int (3)			The number of periods this course meets during a given day. (usually 1)
gpaWeight	float (6,3)			Multiplier used in GPA calculations. Usually the same as the number of credits earned for the course.
bonusPoints	boolean			1 = Yes, 0 = No. Whether bonus points are factored into the total grade for this course.
inTranscript	boolean			1 = Yes, 0 = No. Whether marks and credits for this course should be posted to the transcript.
required	boolean			1 = Yes, 0 = No. Whether this course is required. This is used for the

				reporting and organizing of courses. There is no logic behind the checkbox.
type	string (2)			Refer to the appendix in the Data Migration Guide for a list of codes.
activityCode	string (3)			Generally used for non-academic courses. Refer to the appendix in the Data Migration Guide for a list of codes.
homeroom	boolean			1 = Yes, 0 = No. Indicates this is a homeroom course. Used in some state reports. CA: Used in PreIDextracts.
requestable	boolean			Setting this field to 1 allows a student to request this course during the registration process.
allowTeacherRequests	boolean			Setting this field to 1 allows a teacher to request this course on behalf of the student during the registration process.
attendance	boolean			1 = Yes, 0 = No. Whether attendance is taken for this course.
comments	string (250)			Additional information regarding this course.
description	string (unlimited)			Enter a brief description if you have it. This information is displayed in the portal for online registration purposes.
addToCatalog	boolean			1 = Yes, 0 = No. Whether to add the course to the course catalog for this school.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## section

A section is an instance of a course when a teacher meets with a particular group of students (a roster) at a particular time on the schedule in a certain room. If any of those items are different (a different teacher with a different roster), another section will be required. Include one row for each section of each course. The section numbers have to be numeric, and have to be unique within their course.

Required Columns: schoolNum, calendarName, courseNum, sectionNum  
 Unique Key: schoolNum, calendarName, courseNum, sectionNum  
 Constraints: The combination of schoolNum, calendarName and courseNum must match a row in the course file.  
 schoolNum must match a row in the employment file.  
 schoolNum must match a row in the employment file.  
 When primaryEmployNum has a value (other than blank or NULL) then primaryEmployNum must match a row in the employment file.  
 When secondaryEmployNum has a value (other than blank or NULL) then secondaryEmployNum must match a row in the employment file.  
 When roomNumber has a value (other than blank or NULL) then the combination of schoolNum and roomNumber must match a row in the room file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The name of the calendar under which this section is taught.
courseNum	string (13)	Yes	Yes	The unique course number within this calendar for this section.
sectionNum	int (4)	Yes	Yes	The section number within the course that uniquely identifies this section. NOTE: This must be a numeric value.
primaryEmployNum	string (15)			The unique employNum of the teacher who is responsible in the classroom.
primaryDistrictAssignment	string (50)			This must match a title in the employment file.
secondaryEmployNum	string (15)			Enter another employNum when you have another teacher who is also assigned to this classroom. In the rare case when there are more than two teachers assigned to one section ask your Data Services Engineer for the Supplemental Teacher file.
secondaryDistrictAssignment	string (50)			This must match a title in the employment file.
teacherDisplayName	string (30)			The name you would like to appear on student schedules and report cards.
roomNumber	string (10)			The room that this section meets in. If entered, must match a name from the room conversion file.
maxStudents	int (5)			The maximum number of students allowed in this section. Overrides the maxStudents value of a course.
lunchCount	boolean			1 = Yes, 0 = No. Whether this section should be used for lunch count.
milkCount	boolean			1 = Yes, 0 = No. Whether this section should be used for milk count.
adultCount	boolean			1 = Yes, 0 = No. Whether this section should be used for adult lunch count.
skinnySeq	int (3)			If the course is scheduled at the same time with another course and the two courses are taken during the same term by the same students (i.e., Choir and Band) in a rotating schedule setup, this field indicates in what order the courses will be listed on a student schedule.
teamName	string (40)			For sections taught in teams, the name of a team/schedule group.
lunchName	string (15)			For sections taught when lunch is served, enter a name for the lunch shift.
homeroom	boolean			Identifies this section as a homeroom. This field can be used in Ad Hoc queries to provide homeroom lists.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## sectionSchedule

This file contains information about when a section meets. Indicate start and end terms, start and end periods and a period schedule. If a section meets for more than one period schedule (such as A days and B days) then enter one row for each period schedule. Otherwise you will enter one row for each section.

**Required Columns:** schoolNum, calendarName, courseNum, sectionNum, startTermName, endTermName, scheduleName, startPeriodName, endPeriodName

**Unique Key:** schoolNum, calendarName, courseNum, sectionNum, startTermName, scheduleName, startPeriodName

**Constraints:**

- schoolNum must match a row in the period file.
- schoolNum must match a row in the period file.
- calendarName must match a row in the period file.
- calendarName must match a row in the period file.
- scheduleName must match a row in the period file.
- scheduleName must match a row in the period file.
- startPeriodName must match a row in the period file.
- endPeriodName must match a row in the period file.
- The combination of schoolNum, calendarName, courseNum and sectionNum must match a row in the section file.
- schoolNum must match a row in the term file.
- The combination of schoolNum and calendarName must match a row in the term file.
- The combination of calendarName and startTermName must match a row in the term file.
- endTermName must match a row in the term file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school.
calendarName	string (30)	Yes	Yes	The name of the calendar under which this course is taught.
courseNum	string (13)	Yes	Yes	The unique course number within this calendar for this course.
sectionNum	int (4)	Yes	Yes	The section number within the course that uniquely identifies this section. NOTE: This must be a numeric value.
startTermName	string (10)	Yes	Yes	The name of the first term that this section meets. If the section meets in multiple terms, such as 1st and 2nd semester, then enter a different value in endTermName
endTermName	string (10)	Yes		The name of the last term that this section meets
scheduleName	string (20)	Yes	Yes	The schedule name of a rotation that this section meets. If the section meets in multiple schedules, such as both A day and B day, there should be multiple rows for the section, one for each schedule.
startPeriodName	string (10)	Yes	Yes	The first period name that this section meets.
endPeriodName	string (10)	Yes		The last period name that this section meets.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## courseTask

This data indicates which grading tasks will be graded in which courses. You can specify from which score group the grades will come, and the credits that students can earn for passing the course. For each course you will add one row for each grading task used by the course.

Required Columns: schoolNum, calendarName, courseNum, taskName, scoreGroupName

Unique Key: schoolNum, calendarName, courseNum, taskName

Constraints: The combination of schoolNum, calendarName and courseNum must match a row in the course file.

taskName must match a row in the gradingTask file.

scoreGroupName must match a row in the scoreGroup file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string(7)	Yes	Yes	The district-wide unique school number.
calendarName	string(30)	Yes	Yes	The name of the calendar under which this course is taught.
courseNum	string(13)	Yes	Yes	The unique number for this course.
taskName	string(75)	Yes	Yes	The name of the task that will be graded in this course.
scoreGroupName	string(80)	Yes		The name of the score group to use when grading this task for this course.
gradebookDefault	boolean			A course can have multiple grading tasks so this field provides the ability to select one of the grading tasks to show first when a teacher uses the gradebook. Enter a 1 to make this the default.
termGPA	float(6,3)			Weight of the grade within the GPA for the term.
credits	float(6,3)			The number of credits earned for a passing grade on this task. For example, a semester course may have a value of .5 while a full year course has a value of 1.
terms	string(10)			Identifies the terms to grade for this the task. Enter one of: "All", "Last", "Semester", a single term name, a combination of term names separated by commas, or "Other". The default is "All". When you use "Other" be sure to discuss with your Data Services Engineer.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## courseStandard

Setting up standards is an intensive process and there are a lot of variables which impact the "right" way to do it. It is recommended to submit a support case requesting consultation with an ICU trainer on how your standards should be set up. If you know how you want to apply your standards to courses then add the information to this file.

**Required Columns:** schoolNum, calendarName, courseNum, standardGroupLevel1, standardName  
**Unique Key:** schoolNum, calendarName, courseNum, standardGroupLevel1, standardGroupLevel2, standardGroupLevel3, standardGroupLevel4, standardName  
**Constraints:** The combination of schoolNum, calendarName and courseNum must match a row in the course file.  
 When standardGroupLevel2 and standardGroupLevel3 and standardGroupLevel4 have values (other than blank or NULL) then the combination of standardGroupLevel1, standardGroupLevel2, standardGroupLevel3, standardGroupLevel4 and standardName must match a row in the standard file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique school number.
calendarName	string (30)	Yes	Yes	The name of the calendar under which this course is taught.
courseNum	string (13)	Yes	Yes	The unique number for this course.
standardGroupLevel1	string (50)	Yes	Yes	The name of the top level group of standards, such as "HS" or "MS".
standardGroupLevel2	string (50)		Yes	The name of the second group level used for grouping standards.
standardGroupLevel3	string (50)		Yes	The name of the third group level used for grouping standards.
standardGroupLevel4	string (50)		Yes	The name of the fourth group level used for grouping standards.
standardName	string (75)	Yes	Yes	The name of the standard that you want associated with this course.
terms	string (10)			Identifies the terms to grade for this standard. Enter one of: "All", "Last", a term name, or "Other". The default is All.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## roster

One row for each class that each student took, or is scheduled to take. The start date should be left blank if the student's start date is the same as the section's start date. Same with the end date.

Required Columns: schoolNum, calendarName, courseNum, sectionNum, studentNum

Unique Key: schoolNum, calendarName, courseNum, sectionNum, studentNum, startDate

Constraints: The combination of studentNum, schoolNum and calendarName must match a row in the enrollment file.

The combination of schoolNum, calendarName, courseNum and sectionNum must match a row in the section file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for this school
calendarName	string (30)	Yes	Yes	The name of the calendar under which this course is taught.
courseNum	string (13)	Yes	Yes	The unique course number within this calendar for this course.
sectionNum	int (4)	Yes	Yes	The section number within the course that uniquely identifies this section. NOTE: This must be a numeric value.
studentNum	string (15)	Yes	Yes	The unique identifier for the student taking this section.
startDate	date		Yes	The date the student first started this class. Leave blank if the student started the class from the beginning. NOTE: If the student dropped and restarted the class there will be multiple rows in this file.
endDate	date			If the student dropped the class, the last date the student attended the class. Leave blank if the student completed the class or the class has not yet started.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## courseRequest

This is an optional file. Include this information if you have not completed assigning your students to their classes. You can also include this information if you would like student requests for the current year.

Required Columns: studentNum, schoolNum, calendarName, courseNum

Unique Key: studentNum, schoolNum, calendarName, courseNum

Constraints: The combination of schoolNum, calendarName and courseNum must match a row in the course file.  
studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
schoolNum	string (7)	Yes	Yes	The district-wide unique school number.
calendarName	string (30)	Yes	Yes	The name of the calendar under which this course is taught.
courseNum	string (13)	Yes	Yes	The unique course number within this calendar for this course.
type	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



# Grades

## reportCard

A report card summarizes a student's progress throughout a school year. Schools can generate report cards for each term within their calendar. Each row in this file will represent one grade that a student received for one grading task in one section. The termName column can be filled in if a student receives more than one grade, such as a quarter grade and semester grade. Note that the taskName has to match a Grading Task from the gradingTask file, and the task should also be tied to the course on the courseTask file.

Required Columns: schoolNum, calendarName, courseNum, sectionNum, studentNum, termName, taskName

Unique Key: schoolNum, calendarName, courseNum, sectionNum, studentNum, termName, taskName

Constraints: taskName must match a row in the gradingTask file.  
The combination of schoolNum, calendarName, courseNum, sectionNum and studentNum must match a row in the roster file.  
The combination of schoolNum, calendarName and termName must match a row in the term file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique school number for this course.
calendarName	string (30)	Yes	Yes	The name of the calendar under which this course is taught.
courseNum	string (13)	Yes	Yes	The unique course number within this calendar.
sectionNum	int (4)	Yes	Yes	The section number within the course number that uniquely identifies this section. NOTE: This must be a numeric value.
studentNum	string (15)	Yes	Yes	The unique identifier for the student taking this section.
termName	string (10)	Yes	Yes	The name of the term from the term spreadsheet that this score was given.
taskName	string (75)	Yes	Yes	Name of the task that this grade is for.
score	string (255)			The letter or number score given to this student. (i.e. A, A-, etc)
percent	float (6,3)			The percent score earned for this course.
comments	string (1200)			Teacher notes specific to this student and this score.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# transcript

A transcript is a historical record of a student's performance in a course. Enter one row for each class that each student took.

Required Columns: studentNum, courseNum, endYear  
 Unique Key: studentNum, courseNum, endYear, actualTerm, additionalKey  
 Constraints: When creditGroupName and creditTypeName have values (other than blank or NULL) then the combination of creditGroupName and creditTypeName must match a row in the creditGroup file.  
 studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
courseNum	string (13)	Yes	Yes	Course number for this transcript entry.
courseName	string (60)			(Recommended) Course name for this transcript entry.
endYear	int (4)	Yes	Yes	The 4 digit end year that this grade was earned in. (For the 05-06 school year, enter 2006). Refer to the appendix in the Data Migration Guide for a list of codes.
gradeLevel	string (4)			(Recommended) The grade level the student was in when this course was taken. i.e. 09, 10, 11, 12.
ncesGrade	string (4)			Reports the National Center for Education Statistics (NCES) grade level of the course. Refer to the appendix in the Data Migration Guide for a list of codes.
repeatCourse	boolean			Indicates student has previously taken this course, usually because of a failing grade.
districtNum	string (12)			(Recommended) Fill in the district number where the course was taken.
schoolNumber	string (7)			(Recommended) Fill in the school number where the course was taken.
schoolName	string (50)			(Recommended) Fill in the school name where the course was taken.
stateCode	string (20)			The state-designated code for this course.
score	string (10)			(Recommended) Letter grade for this course. (A, B, C+, etc.)
percent	float (6,3)			The score as a percentage. This field is not included on the default transcript.
gpaWeight	float (6,4)			(Recommended) Weight value for use in GPA calculations. Usually the same as the number of credits attempted.
weightedGPAValue	float (7,4)			(Recommended) The adjusted GPA value for the letter grade given, such as 5.0 for an A, 4.0 for a B, etc. Generally applied to AP type courses. If this course is a standard course then put the unweighted GPA value in both this field and the unweightedGPAValue field.
unweightedGPAValue	float (7,4)			(Recommended) GPA value for the letter grade given. A=4, B=3, etc.
gpaMax	float (7,4)			(Recommended) The maximum GPA value achievable for this course and score. Used for weighted GPA calculations.
bonusPoints	float (7,5)			Bonus points earned for this course or this score.
creditsEarned	float (6,3)			(Recommended) The number of credits the student earned for this transcript entry. Ex: If the student passed a half-credit course, enter 0.5.
creditsAttempted	float (6,3)			(Recommended) The number of credits that could be earned for this course.
creditGroupName	string (50)			(Recommended) If credit was earned, the name of the credit group from the grading setup file. (i.e. HSCredits)
creditTypeName	string (50)			(Recommended) If credit was earned, the name of the credit from the grading setup file. (i.e. Math, Science, etc.)
date	date			The date of the transcript entry. (mm/dd/yyyy)
actualTerm	int (2)		Yes	The actual term this entry is for. If it is a 2 term course and this grade is for first term, this would be 1.
startTerm	int (2)			(Recommended) The number of the first term this course met. If a 4 qtr year, and this course met 3rd & 4th qtr, this would be 3.
endTerm	int (2)			The number of the last term this course met. If 4 qtr year, and this course met 3rd & 4th qtr, this would be 4.
termsLong	int (2)			(Recommended) The number of terms this course met. If 4 qtr year, and this course met 3rd & 4th qtr, this would be 2.
calendarTerms	int (2)			The total number of terms in the school year, at the school and year that this course was taken.
termStartDate	date			The date this course started.
termEndDate	date			The date this course finished.
comments	string (1200)			General information on transcript entry

additionalKey	string (50)		Yes	Usually left blank. In some rare cases, studentNum+courseNum+schoolYear+actualTerm is not unique. In those cases, enter an additional value here to distinguish the records. This field should only be used after you have verified both records are valid.
---------------	-------------	--	-----	---

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Attendance

## attendanceExcuse

Attendance codes are used to process student attendance. When teachers report attendance, they merely report a student as absent or tardy. An office staff member with proper tool rights can assign an attendance code to the teacher-reported attendance event. Codes can then be used to analyze student-level and school-wide attendance to determine trends and patterns. Enter the list of codes that are used in each calendar.

Required Columns: schoolNum, calendarName, code, description, status, excuse

Unique Key: schoolNum, calendarName, code

Constraints: The combination of schoolNum and calendarName must match a row in the calendar file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for the school.
calendarName	string (30)	Yes	Yes	The name of the calendar that this excuse code is used in.
code	string (4)	Yes	Yes	A short code for this excuse. (i.e. AE, ILL, FT, TU, etc.). This code will be seen in the Campus application and should be descriptive of the attendance event. For example: ILL for an excused absence for illness. Avoid using single-character codes 0, 1, 2, A, B, H, L, N, /, #, or \$. Combinations of these characters are acceptable.
description	string (50)	Yes		The long description of this excuse code. (i.e. Absent Excused, Sick, Field Trip, etc.). The description can be seen by hovering over the code in the student attendance records or in the daily attendance tool of the Attendance module.
status	string (1)	Yes		Must be one of: A=Absent, T=Tardy, E=Early Release. Refer to the appendix in the Data Migration Guide for a list of codes.
excuse	string (1)	Yes		Must be one of: E=Excused, U=Unexcused, X=Exempt, K=Unknown or Unverified. Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## attendance

This file will have one row for each absence by each student. If your conversion date is during the summer, then you will probably not provide this information. For a full day absence, you can leave the periodName column blank. For a period absence, the periodName column will have to be filled in. Since there can be more than one period schedule for a calendar, the conversion will have to know to which schedule the periodName refers (4th hour on Monday might be a different course than 4th hour on Tuesday). The conversion will look up the attendance date in the day table to find out which schedule was used that day.

Required Columns: schoolNum, calendarName, studentNum, date, code

Unique Key: schoolNum, calendarName, studentNum, date, periodName

Constraints: The combination of schoolNum, calendarName and code must match a row in the attendanceExcuse file.

The combination of schoolNum, calendarName and studentNum must match a row in the enrollment file.

When periodName has a value (other than blank or NULL) then the combination of schoolNum, calendarName and periodName must match a row in the period file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique number for the school.
calendarName	string (30)	Yes	Yes	The calendar name by which this calendar is known.
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date for this attendance record.
periodName	string (10)		Yes	The name of the period for this attendance record. (i.e. If 1st period is named Homeroom and the 2nd period is named 01, etc., then enter Homeroom, 01, etc.). If left blank, all periods in the day will be marked with this absence.
code	string (4)	Yes		An excuse code from the attendanceExcuse file.
comments	string (50)			Additional information about this attendance record.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Health

## vaccineMapping

This table is used to link the names of your vaccines to the vaccines available in Infinite Campus. You may have one vaccine that could be matched to more than one vaccine in Campus (such as Hep B). You will need to decide which Campus vaccine to use. To complete this file, list each shot used in your district along with the matching Campus vaccine code listed in the Data Migration Guide appendix.

Required Columns: districtVaccine, campusVaccineCode

Unique Key: districtVaccine

Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
districtVaccine	string(50)	Yes	Yes	Put the name or code of the vaccine from your system into this field. You will use this in the vaccineShot table.
campusVaccineCode	string(20)	Yes		Refer to the appendix in the Data Migration Guide for a list of codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## vaccineShot

Some vaccines require a series of shots. This table provides room for up to six dates for the same vaccine so each shot that a student received will have one row in this file.

Required Columns: studentNum, districtVaccine

Unique Key: studentNum, districtVaccine, vaccineDate1

Constraints: studentNum must match a row in the student file.

districtVaccine must match a row in the vaccineMapping file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
districtVaccine	string (50)	Yes	Yes	The district's name or code of the vaccination the student received. This value must match a value from the districtVaccine column in the vaccineMapping table.
vaccineDate1	date		Yes	The first date the vaccination was received or the waiver approved/recorded.
vaccineDate2	date			The second date the vaccination was received. Do not use this column for exemptions.
vaccineDate3	date			The third date the vaccination was received. Do not use this column for exemptions.
vaccineDate4	date			The fourth date the vaccination was received. Do not use this column for exemptions.
vaccineDate5	date			The fifth date the vaccination was received. Do not use this column for exemptions.
vaccineDate6	date			The sixth date the vaccination was received. Do not use this column for exemptions.
exemptCode	string (1)			CA and WI - refer to the appendix in the Data Migration Guide for a list of codes. All other states - submit your own district codes.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## healthCondition

Used for ongoing health issues and treatments.

Required Columns: studentNum, code, startDate, status, initials

Unique Key: studentNum, code, startDate

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
code	string (10)	Yes	Yes	A code indicating the student condition. Campus is pre-loaded with the ICD9 codes but you can enter your own codes too. Refer to the appendix in the Data Migration Guide for a list of codes.
description	string (60)			A brief description of the student condition. If a valid ICD9 code is entered in the previous column, this column can be left blank.
startDate	date	Yes	Yes	The date when the condition started, or when the district was first notified of the condition.
endDate	date			The date when the district was notified that this condition was resolved.
status	string (15)	Yes		The status of the condition. Refer to the appendix in the Data Migration Guide for a list of codes.
initials	string (10)	Yes		The initials of the person who recorded the condition.
doctorName	string (50)			The name of the doctor, if known.
doctorPhone	string (25)			The telephone number of the doctor, if known.
userWarning	string (30)			Information added to this field will be available for viewing by staff with the proper security settings.
instructions	string (255)			Detailed guidance on how to handle the student if the health condition affects the student during the school day.
comments	string (unlimited)			Additional information related to this record.
treatmentCode	string (10)			A code indicating the treatment for this condition. Campus has a pre-loaded list of treatment codes but you can enter your own codes too. This field is not required, but if left blank, no treatment will be recorded for the condition. Refer to the appendix in the Data Migration Guide for a list of codes.
treatmentDesc	string (24)			A brief description of the treatment. If a valid code is entered in the previous column, this column can be left blank.
treatmentStartDate	date			The date that the treatment was started. Required if treatmentCode is entered.
treatmentEndDate	date			The date that the treatment was stopped for this condition.
treatmentStatus	string (15)			The status of the treatment (required if treatmentCode is entered). Refer to the appendix in the Data Migration Guide for a list of codes.
treatmentInitials	string (10)			The initials of the person who recorded the condition. Required if treatmentCode is entered.
treatmentComments	string (unlimited)			Additional information related to this treatment record.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



## healthVisit

This file only needs to be completed when your go-live date occurs during the school year. In addition, you may not need to complete this file if your health office staff will maintain read-only access to your legacy system. If you do complete this file then create one record for each student's visit to the health office. Infinite Campus provides the ability for multiple complaints, observations and interventions. It also allows multiple medications, treatments, and discharges. If your legacy system contains multiple complaints, observations, interventions, medications, treatments, or discharges then you will need to create additional records in this file.

Required Columns: studentNum, visitDate, visitTime, employNum  
Unique Key: studentNum, visitDate, visitTime  
Constraints: employNum must match a row in the employee file.  
studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
visitDate	date	Yes	Yes	The date the student arrived at the health office. In mm/dd/yyyy format.
visitTime	time	Yes	Yes	The time the student arrived at the health office. In HH:MM format.
employNum	string (15)	Yes		The employNum of the health office staff person who assisted the student. This person should have the Health field checked in the employment file for the school in which this student is enrolled.
recordComplete	boolean			Set this field to 1 when this health office visit has been completed and requires no additional information or modification by health staff.
visitComments	string (unlimited)			Additional information regarding this health visit.
complaintName	string (50)			Infinite Campus comes pre-loaded with a list of complaints. You can use this list or create your own.
complaintComments	string (unlimited)			Additional information regarding this complaint.
dischargeTime	time			The time of day the student left the health office. In HH:MM format.
dischargeName	string (50)			Infinite Campus comes pre-loaded with a list of discharges. You can use this list or create your own.
dischargeComments	string (unlimited)			Additional information regarding this discharge.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## medication

This file provides the ability to store medication information for students. Saved medications will appear in the Medications tab and are available in the Health Office Visit module.

Required Columns: studentNum, schoolNum, calendarName, medicationName, medicationForm, medicationStrength, amtPerDose, dateSubmitted, dosesSubmitted, employNum  
 Unique Key: studentNum, medicationName, dateSubmitted  
 Constraints: The combination of schoolNum and calendarName must match a row in the calendar file.  
 employNum must match a row in the employee file.  
 studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
schoolNum	string (7)	Yes		The district-wide unique number for this school.
calendarName	string (30)	Yes		The name of the calendar in which this medication record is valid.
medicationName	string (50)	Yes	Yes	Enter the name of a medication in this field.
medicationForm	string (50)	Yes		Enter the form this medication takes such as capsule, liquid, etc.
medicationStrength	string (50)	Yes		Enter the strength of this medication.
amtPerDose	float (6,2)	Yes		Enter the amount per dose.
doseMeasurement	string (15)			
dateSubmitted	date	Yes	Yes	The date the medication was received in the health office.
dosesSubmitted	int (5)	Yes		The number of doses available to the student.
employNum	string (15)	Yes		The employNum of the health office staff person who received the medication. This person should have the Health field checked in the employment file for the school in which this student is enrolled.
notificationThreshold	int (5)			If the medication is one that will need to be resupplied as the quantity runs out, enter a value in this field that provides a sufficient cushion as to allow the parent or guardian to bring in additional doses.
rxDate	date			The date the prescription was assigned.
rxNumber	string (25)			Enter the number used by a pharmacy to identify a prescription.
expirationDate	date			
pharmacyName	string (50)			The location where the prescription was filled.
pharmacyPhone	string (25)			The telephone number where the prescription was filled.
doctorName	string (50)			The name of the prescribing doctor.
doctorPhone	string (25)			The telephone number of the prescribing doctor.
directions	string (255)			Directions for use of this prescription.
comments	string (255)			Additional information related to this medication.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Screenings

## childAndTeenScreening

This file is used to document child and teen information.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression over time. Student number combined with date must be unique.
type	string (50)			Indicates the reason for the test. Refer to the appendix in the Data Migration Guide for a list of codes.
location	string (15)			Where the test was performed. Refer to the appendix in the Data Migration Guide for a list of codes.
status	string (15)			The test results. Refer to the appendix in the Data Migration Guide for a list of codes.
healthCare	string (15)			Indicates the parent/guardian having insurance coverage for such screenings. Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## concussionScreening

This file is used to document concussion information.

Required Columns: studentNum

Unique Key: studentNum

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string(15)	Yes	Yes	The district-wide unique identifier for the student.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## dentalScreening

This file is used to document dental information.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression over time. Student number combined with date must be unique.
dentalSealants	string (5)			Refer to the appendix in the Data Migration Guide for a list of codes.
cariesHistory	string (5)			Refer to the appendix in the Data Migration Guide for a list of codes.
untreatedCaries	string (5)			Refer to the appendix in the Data Migration Guide for a list of codes.
stPathology	string (5)			Soft Tissue Pathology. Refer to the appendix in the Data Migration Guide for a list of codes.
malocclusion	string (5)			Refer to the appendix in the Data Migration Guide for a list of codes.
dentalUrgent	string (5)			Refer to the appendix in the Data Migration Guide for a list of codes.
dentalRestorative	string (5)			Refer to the appendix in the Data Migration Guide for a list of codes.
dentalPreventative	string (5)			Refer to the appendix in the Data Migration Guide for a list of codes.
dentalOther	string (100)			Additional dental related information.
dentalWaiverDate	date			The date of the student received the dental waiver.
dentalWaiverReason	string (50)			Refer to the appendix in the Data Migration Guide for a list of codes.
referralDate	date			The date student was referred to a specialist.
dentalTest	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## developmentalScreening

This file is used to document developmental information.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression overtime. Student number combined with date must be unique.
developmentStatus	string (15)			The result of the exam. Refer to the appendix in the Data Migration Guide for a list of codes.
developmentTest	string (15)			The type of test given. Refer to the appendix in the Data Migration Guide for a list of codes.
socialEmotionalStatus	string (15)			Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## earlyChildhoodScreening

This file is used to document early childhood information.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression over time. Student number combined with date must be unique.
provider	string (15)			Indicates where the screening occurred. Refer to the appendix in the Data Migration Guide for a list of codes.
earlyEdExperiences	string (20)			Location student received education before attending school. Refer to the appendix in the Data Migration Guide for a list of codes.
earlyEdReferral	string (20)			Reason for the screening. Refer to the appendix in the Data Migration Guide for a list of codes.
healthCare	string (15)			Indicates the parent/guardian having insurance coverage for such screenings. Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# hearingScreening

This file is used to document the results of hearing tests.

Required Columns: studentNum, date  
 Unique Key: studentNum, date  
 Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression over time. Student number combined with date must be unique.
status	string (50)			Indication of the results of the hearing test. Refer to the appendix in the Data Migration Guide for a list of codes.
hearingType	string (50)			The type of test given. Refer to the appendix in the Data Migration Guide for a list of codes.
otoacousticEmissions	string (25)			Results of the test that measures an acoustic response produced by the inner ear. Refer to the appendix in the Data Migration Guide for a list of codes.
250R	string (3)			Indicates the range at which the student can hear frequencies of sound for the right (R) ear. These are numeric values entered up to 3 digits.
500R	string (3)			Indicates the range at which the student can hear frequencies of sound for the right (R) ear. These are numeric values entered up to 3 digits.
1000R	string (3)			Indicates the range at which the student can hear frequencies of sound for the right (R) ear. These are numeric values entered up to 3 digits.
2000R	string (3)			Indicates the range at which the student can hear frequencies of sound for the right (R) ear. These are numeric values entered up to 3 digits.
4000R	string (3)			Indicates the range at which the student can hear frequencies of sound for the right (R) ear. These are numeric values entered up to 3 digits.
6000R	string (3)			Indicates the range at which the student can hear frequencies of sound for the right (R) ear. These are numeric values entered up to 3 digits.
8000R	string (3)			Indicates the range at which the student can hear frequencies of sound for the right (R) ear. These are numeric values entered up to 3 digits.
tympanometryR	string (25)			Status of the middle ear test and the mobility of the eardrum. Refer to the appendix in the Data Migration Guide for a list of codes.
otoscopicR	string (25)			Status of the auditory canal and ear drum test. Refer to the appendix in the Data Migration Guide for a list of codes.
250L	string (3)			Indicates the range at which the student can hear frequencies of sound for the left (L) ear. These are numeric values entered up to 3 digits.
500L	string (3)			Indicates the range at which the student can hear frequencies of sound for the left (L) ear. These are numeric values entered up to 3 digits.
1000L	string (3)			Indicates the range at which the student can hear frequencies of sound for the left (L) ear. These are numeric values entered up to 3 digits.
2000L	string (3)			Indicates the range at which the student can hear frequencies of sound for the left (L) ear. These are numeric values entered up to 3 digits.
4000L	string (3)			Indicates the range at which the student can hear frequencies of sound for the left (L) ear. These are numeric values entered up to 3 digits.
6000L	string (3)			Indicates the range at which the student can hear frequencies of sound for the left (L) ear. These are numeric values entered up to 3 digits.
8000L	string (3)			Indicates the range at which the student can hear frequencies of sound for the left (L) ear. These are numeric values entered up to 3 digits.
tympanometryL	string (25)			Status of the middle ear test and the mobility of the eardrum. Refer to the appendix in the Data Migration Guide for a list of codes.
otoscopicL	string (25)			Status of the auditory canal and ear drum test. Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



## heightWeightScreening

This file is used to document height and weight information.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression overtime. Student number combined with date must be unique.
height	float (5,2)			Defined as the distance from the student's feet to the top of his/her head. This data is entered in inches. If a student were 6 feet, 2 inches tall, the height would be entered at 74 inches. This field is limited to five characters - two places before the decimal point and two characters after the decimal point (e.g., 74.22)
weight	float (5,2)			Defined as the vertical force exerted by a mass as a result of gravity. This data is entered in pounds. If a student weighs 156 pounds, the entry would read 156.00.
bmi	float (6,3)			The Body Mass Index is a measure of a student's weight in relation to height. This value calculated is based on the student's height and weight. Data is entered in XX.XX format.
bmiPercentile	float (4,1)			A value on the CDC BMI for age growth charts, determined after the calculation of the student's BMI.
bloodPressure	string (7)			This is the force exerted by circulating blood on the walls of blood vessels. Enter values in XXX/XX format.
pulse	string (4)			Value indicates the heartbeats per minute. Enter values in XXXX format.
respiration	string (50)			Indicates the number of breaths per minute.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## leadScreening

This file is used to document lead exposure information.

Required Columns: studentNum

Unique Key: studentNum

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string(15)	Yes	Yes	The district-wide unique identifier for the student.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## scoliosisScreening

This file is used to document scoliosis test results.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string(15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression overtime. Student number combined with date must be unique.
character	string(15)			Indicates the results of the test. Refer to the appendix in the Data Migration Guide for a list of codes.
type	string(50)			Indicates type of the test. Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string(255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## socialEmotionalScreening

This file is used to document social and emotional test information.

Required Columns: studentNum

Unique Key: studentNum

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string(15)	Yes	Yes	The district-wide unique identifier for the student.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## speechScreening

This file is used to document speech information.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string(15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression overtime. Student number combined with date must be unique.
test	string(15)			The type of test given. Refer to the appendix in the Data Migration Guide for a list of codes.
status	string(15)			The test results. Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string(255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## sportsScreening

This file is used to document the date and status of sports physicals

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression overtime. Student number combined with date must be unique.
physicalStatus	string (25)			The result of the sports physical. Refer to the appendix in the Data Migration Guide for a list of codes.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## tuberculosisScreening

This file is used to document tuberculosis information.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression over time. Student number combined with date must be unique.
skinTestDate	date			Date student received the TB skin test.
placeGiven	string (50)			Indicates where the student received the skin test. Refer to the appendix in the Data Migration Guide for a list of codes.
dateRead	date			Date the results of the test were read.
results	string (5)			The result of the skin test.
xRayDate	date			Date the student received an X-ray.
status	string (15)			The results of the test. Refer to the appendix in the Data Migration Guide for a list of codes.
medStartDate	date			If medication is given for the treatment of TB, this field indicates when the medication was first taken.
medEndDate	date			This field indicates when the medication was last taken.
refusedDate	date			The date the student refused to be treated.
freefromCommunicableTB	boolean			Set to 1 when the student is free from TB.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## visionScreening

This file is used to document vision information.

Required Columns: studentNum, date

Unique Key: studentNum, date

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
date	date	Yes	Yes	The date this information was last recorded. Can have multiple entries per studentNum to track progression overtime. Student number combined with date must be unique.
status	string (50)			The status of the condition. Refer to the appendix in the Data Migration Guide for a list of codes.
testType	string (15)			Refer to the appendix in the Data Migration Guide for a list of codes.
acuityR	int (8)			Visual Acuity Right Eye. Entry is limited to five digits.
acuityL	int (8)			Visual Acuity Left Eye. Entry is limited to five digits.
corLens	string (10)			Y: Yes or N: No. Refer to the appendix in the Data Migration Guide for a list of codes.
colorVision	string (25)			BY: Blue/Yellow, P: Pass, or RG: Red/Green. Refer to the appendix in the Data Migration Guide for a list of codes.
coverTest	string (25)			Cover test status. Refer to the appendix in the Data Migration Guide for a list of codes.
externalInspection	string (20)			External Inspection status. Refer to the appendix in the Data Migration Guide for a list of codes.
cornealReflex	string (25)			Corneal Reflex status. Refer to the appendix in the Data Migration Guide for a list of codes.
referralDate	date			The date the student was given a referral for a vision test.
comments	string (255)			Additional information regarding this screening.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



# Behavior

## behaviorIncident

Campus lets you connect several events to one incident. Therefore, this file should just have one row per incident. Note: The incidentID column is needed to link the data from the separate files, particularly when there is more than one event (behaviorEvent) recorded for an incident. If the legacy system does not have a number to use for incidentID, and there will always be just one event per incident, try keeping the rows in all four files in the same order, and use the file row number for the incidentID.

Required Columns: incidentID, incidentDate

Unique Key: incidentID

Constraints: When referralEmployNum has a value (other than blank or NULL) then referralEmployNum must match a row in the employee file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
incidentID	string (50)	Yes	Yes	Unique value assigned to this behavior incident.
submittedDate	date			Date incident was reported. This could be different from the Incident Date.
referralEmployNum	string (50)			The employNum of the person reporting this behavior incident.
title	string (50)			A brief description of the incident. If no title is entered for the incident, the Event TypeName(s) will populate as the incident name.
incidentDate	date	Yes		The date the incident takes place.
incidentTime	time			The time the incident takes place.
damages	float (7,2)			The total cost of all damages that occurred as a result of the incident.
context	string (2)			Refer to the appendix in the Data Migration Guide for a list of codes.
contextDescription	string (100)			Additional information regarding the context.
location	string (15)			
locationDescription	string (100)			Additional information regarding the location.
details	string (unlimited)			Any additional information about the incident, including a description of what occurred.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## behaviorEvent

Campus lets you connect several students to one event. Therefore, this file should just have one row per event.

Note: The eventID column is needed to link the data from the separate files, particularly when there is more than one student (behaviorRole) recorded for an event. If the legacy system does not have a number to use for eventID, and there will always be just one student per event, try keeping the rows in all three files in the same order, and use the file row number for the eventID.

Required Columns: incidentID, eventID, eventName

Unique Key: incidentID, eventID

Constraints: incidentID must match a row in the behaviorIncident file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
incidentID	string(50)	Yes	Yes	Unique value assigned to behavior incident.
eventID	string(50)	Yes	Yes	Unique value assigned to behavior event.
eventCode	string(5)			District defined code.
eventName	string(100)	Yes		Check the appendix in the Data Migration Guide to determine if your state supplies a list of behavior events. If they do then put a code, listed in the appendix, in the eventCode field and leave this field blank. Otherwise put the description in this field and the code in the eventCode field.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## behaviorRole

Put a row in this file for each student involved with each event.

Required Columns: incidentID, eventID, studentNum

Unique Key: incidentID, eventID, studentNum

Constraints: The combination of incidentID and eventID must match a row in the behaviorEvent file.  
studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
incidentID	string (50)	Yes	Yes	Unique value assigned to behavior incident.
eventID	string (50)	Yes	Yes	Unique value assigned to behavior event.
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
role	string (40)			Refer to the appendix in the Data Migration Guide for a list of codes.
demerits	int (5)			
relationshipToSchool	string (4)			Refer to the appendix in the Data Migration Guide for a list of codes.
injury	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.
injuryDescription	string (50)			Provide a brief description of the injury incurred.
medicalServiceProvided	boolean			
comments	string (unlimited)			Additional information related to this record.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## behaviorResolution

A record of the resolution for each student

Required Columns: incidentID, eventID, studentNum, resolutionName, ResolutionAssignDate  
 Unique Key: incidentID, eventID, studentNum, resolutionName, resolutionDate, resolutionTime  
 Constraints: The combination of incidentID, eventID and studentNum must match a row in the behaviorRole file.  
 When employNum has a value (other than blank or NULL) then employNum must match a row in the employee file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
incidentID	string (50)	Yes	Yes	Unique value assigned to behavior incident.
eventID	string (50)	Yes	Yes	Unique value assigned to behavior event.
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
resolutionCode	string (5)			Enter an abbreviation for the resolution.
resolutionName	string (100)	Yes	Yes	A descriptive name for the resolution. Refer to the appendix in the Data Migration Guide for a list of codes.
duration	int (4)			
durationUnits	string (1)			Refer to the appendix in the Data Migration Guide for a list of codes.
status	string (2)			Refer to the appendix in the Data Migration Guide for a list of codes.
ResolutionAssignDate	date	Yes		The date the resolution was assigned.
resolutionDate	date		Yes	Both the resolutionDate and resolutionTime fields are stored in the same field in Campus. Enter this information in these two fields during conversion.
resolutionTime	time		Yes	
ResolutionEndDate	date			The date by which the student must complete the resolution.
ResolutionEndTime	time			The time by which the student must complete the resolution.
durationInSchoolDays	float (5,2)			Use this field when there is no detention assigned. If there is a value in this field then leave Detention Duration, Duration Units, and Status blank.
employNum	string (15)			The employNum of a staff person responsible for this resolution. This person should have "Behavior Admin" checked in the employment file for the school in which this student is enrolled.
comments	string (unlimited)			Additional information about the resolution.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## behaviorDetention

In the Behavior module, students can be assigned to a Resolution Type which is also marked as "detention". This provides the ability to assign the length of time a student must attend a detention session. During data conversion, detention served is applied on a first in/first out basis. This means when a student has multiple behavior records with detention assigned, the oldest behavior record will be assigned detention time served first. If one record in this file provides enough time to cover multiple behavior records then the time is applied to these records. You have many options when submitting this information. You can send one record which covers the entire amount of detention time served. You can provide one record for each day a student serves detention. Or you can provide multiple records for the same day as long as the start times are different.

Required Columns: studentNum, dateServed, checkInTime, checkOutTime

Unique Key: studentNum, dateServed, checkInTime

Constraints: studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
dateServed	date	Yes	Yes	Date student served detention.
checkInTime	time	Yes	Yes	Time the student started detention.
checkOutTime	time	Yes		Time the student finished detention.
comments	string (100)			Information related to this detention record.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Flags

## flag

Student Flags provide a concise way of labeling students for warnings and flag participation. Some options used are for tracking special education students or certain types of graduation students. Flags can be marked in the following ways - flag, state reporting, medical and contact. If a program is marked as flagged, a small notation of a red flag will appear by the student's name when selected for viewing. Each type of flag has a separate user sub-right that controls which user can add the flag to a student. After naming a program, the user can assign one of nine (9) icons to associate with that flag that will appear in the header of the student's information.

Required Columns: flagName  
Unique Key: flagName  
Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
flagName	string (50)	Yes	Yes	Enter a Name for the flag. This name can be up to fifty (50) characters in length, including spaces and punctuation. However, the length of the flag name may adversely affect the display of it when assigned to a student.
code	string (10)			CA and AZ - refer to the appendix in the Data Migration Guide for a list of codes. All other states - submit your own district codes.
description	string (255)			Enter a Description for the program. This description can be up to 255 characters in length, including spaces and punctuation.
flagged	boolean			Enter a '1' in this field if this flag should appear next to the student's name when assigned. If no flag is desired, leave this field blank.
stateReported	boolean			Enter a '1' in this field if the information is state reported.
contact	boolean			Enter a '1' in this field if the information concerns contact outside the school
posDisplay	boolean			Enter a '1' in this field if the flag should appear when food service personnel use the POS
flagImage	string (15)			Entering a value here will display an image next to the flag when it is assigned to a student. Refer to the appendix in the Data Migration Guide for a list of codes..

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## flagParticipation

Flags can be used for many things. They indicate to a user that a student has special information on file in the office or that a student belongs to a special curriculum group that is used to track graduation process. There is no limit to the amount of general flags that can be assigned to a student. However, a student may only be in one curricular flag at a time.

Required Columns: studentNum, flagName, startDate  
Unique Key: studentNum, flagName, startDate  
Constraints: flagName must match a row in the flag file.  
studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
flagName	string (50)	Yes	Yes	All states, enter the flagName included in the Flag conversion file. States NV, OH, and WY can also select from a list of state codes - refer to the appendix in the Data Migration Guide for a list of codes.
startDate	date	Yes	Yes	The date this program starts. CA: Start Dates should be used for all state reported programs other than 101, 127, 135, and 144 (See below).
endDate	date			The date this program ends.
eligibilityStartDate	date			CA: Eligibility Dates should be used for Program codes 101: 504 Accommodation Plan, 127: GATE, 135: Migrant, and 144: Special Ed.
eligibilityEndDate	date			
userWarning	string (255)			The information entered here will appear to users that hover over a flag attached to a student. For example, if a student has a physical education restriction, when hovering, the user will be able to quickly view this user warning.
participationDetails	string (255)			Enter additional information regarding this record.
description	string (unlimited)			This is another area to enter details regarding this record.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Transportation

## bus

Include one row for each bus route used in your district.

Required Columns: busNumber, description

Unique Key: busNumber

Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
busNumber	string (10)	Yes	Yes	The district-wide unique identifier for the bus.
description	string (250)	Yes		A longer description for the bus.
contracted	boolean			Indicates the bus is contracted for use to the district.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



## busRider

One row for each student that rides a bus

Required Columns: schoolNum, calendarName, studentNum, startDate

Unique Key: schoolNum, calendarName, studentNum, startDate

Constraints:

- When inBusNumber has a value (other than blank or NULL) then inBusNumber must match a row in the bus file.
- When outBusNumber has a value (other than blank or NULL) then outBusNumber must match a row in the bus file.
- When lateBusNumber has a value (other than blank or NULL) then lateBusNumber must match a row in the bus file.
- The combination of schoolNum and calendarName must match a row in the calendar file.
- studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique school number.
calendarName	string (30)	Yes	Yes	The name of the calendar that this bus assignment is in.
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
startDate	date	Yes	Yes	Indicates the date the transportation record was entered and/or modified for the student (the first day of riding the bus, the day the student received a parking permit, the day mileage was entered, etc.)
endDate	date			Indicates the date the student's current transportation record was modified and a new record was created.
inBusNumber	string (10)			The number of the bus that will pick the student up.
outBusNumber	string (10)			The number of the bus that will drop the student off.
inTime	time			Time the student will be picked up.
outTime	time			Time the student will be dropped off.
inBusStop	string (30)			Description or number of the bus stop.
outBusStop	string (30)			Description or number of the bus stop.
lateBusNumber	string (10)			The bus number the student will use following after-school activities.
transMiles	float (6,3)			The number of miles the student is transported. Must be numeric if entered.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## parkingPermit

Submit one row per student that has parking permit information.

Required Columns: schoolNum, calendarName, studentNum

Unique Key: schoolNum, calendarName, studentNum

Constraints: The combination of schoolNum, calendarName and studentNum must match a row in the enrollment file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string(7)	Yes	Yes	The district-wide unique school number.
calendarName	string(30)	Yes	Yes	The name of the calendar that this permit is for.
studentNum	string(15)	Yes	Yes	The district-wide unique identifier for the student.
permitNumber	string(15)			The parking permit number for this student.
make	string(40)			Enter the make of the car.
model	string(32)			Enter the model number of the car.
color	string(15)			Enter the color of the car.
plateNumber	string(10)			Enter the license plate number of the car.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Lockers

## locker

Lockers are added to Infinite Campus using this file. You do not need to include a student number if a locker is not assigned to a student. Enter one row for each locker in each school. If more than one student is assigned to a locker then create one row for each student repeating the locker number, combo's etc.

Required Columns: schoolNum, lockerNum

Unique Key: schoolNum, lockerNum, lockerType

Constraints: schoolNum must match a row in the school file.

When studentNum has a value (other than blank or NULL) then studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique school number.
lockerNum	string (10)	Yes	Yes	Unique locker number.
studentNum	string (15)			The number of the student using this locker. Leave blank for unused lockers.
startDate	date			If not used will set to student enrollment start date
endDate	date			If not used will set to student enrollment end date
lockerSerialNum	string (20)			
lockerType	string (3)		Yes	Refer to the appendix in the Data Migration Guide for a list of codes.
location	string (30)			General location of the locker. (i.e. Senior Hall, Science Wing, ...)
gradeLevel	string (4)			The grade level of students that usually use this locker. If left blank, all grade levels for the specified school can use this locker.
reserved	boolean			Indicates this locker should not be assigned using the Locker Wizard.
shared	boolean			Indicates this locker can be shared by multiple students
padlockSerialNum	string (20)			
combo1	string (8)			combo1 through combo6 fields are used to insert locker combinations
combo2	string (8)			combo1 through combo6 fields are used to insert locker combinations
combo3	string (8)			combo1 through combo6 fields are used to insert locker combinations
combo4	string (8)			combo1 through combo6 fields are used to insert locker combinations
combo5	string (8)			combo1 through combo6 fields are used to insert locker combinations
combo6	string (8)			combo1 through combo6 fields are used to insert locker combinations
activeSeq	int (3)			The current sequence number (1-6) of the combination being used for this locker/lock.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Fees

## fee

Fees can be applied district-wide or to a specific school. Include the schoolNum when applying to a specific school.

Required Columns: feeName

Unique Key: feeName, schoolNum

Constraints: When schoolNum has a value (other than blank or NULL) then schoolNum must match a row in the school file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
feeName	string(30)	Yes	Yes	A unique name for this fee. Will be used to connect assignments and payments.
description	string(50)			A longer description of the fee.
feeType	string(4)			Refer to the appendix in the Data Migration Guide for a list of codes.
schoolNum	string(7)		Yes	Leave blank for a district-wide fee, or enter the school number that this fee is used at.
amount	float(6,2)			The amount of this fee. Leave blank for a variable-rate fee. If entered, must be a positive number.
assessOnce	boolean			Yes or No, whether this fee should only be assessed once per year.
accountNum	string(25)			School account number for the fee.
exemptFeeMax	boolean			Fee Maximums limit how much a student and/or household can be charged for fees. For student maximums, this includes all fees assigned to a student. For household maximums, limits are applied to the entire household.
exemptable	boolean			
active	boolean			Determines when a fee is included in lists.
adjustable	boolean			

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## courseFee

Fees are assigned to courses using this file.

Required Columns: feeName, schoolNum, calendarName, courseNum

Unique Key: feeName, schoolNum, calendarName, courseNum

Constraints: The combination of schoolNum, calendarName and courseNum must match a row in the course file.  
feeName must match a row in the fee file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
feeName	string(30)	Yes	Yes	A fee name to be assigned to a course.
schoolNum	string(7)	Yes	Yes	The district-wide unique number for the school.
calendarName	string(30)	Yes	Yes	The calendar name by which this calendar is known.
courseNum	string(13)	Yes	Yes	The course number to be assigned a fee.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## feeAssignment

This file assigns fees and payments to students. Include one row for each student/fee combination.

Required Columns: schoolNum, calendarName, studentNum, feeName  
Unique Key: schoolNum, calendarName, studentNum, feeName, dueDate, courseNum  
Constraints: The combination of schoolNum and calendarName must match a row in the calendar file.  
When courseNum has a value (other than blank or NULL) then the combination of schoolNum, calendarName and courseNum must match a row in the course file.  
feeName must match a row in the fee file.  
studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
schoolNum	string (7)	Yes	Yes	The district-wide unique school number.
calendarName	string (30)	Yes	Yes	The name of the school calendar when the student incurred this fee.
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
feeName	string (30)	Yes	Yes	The unique name for this fee.
feeAmount	float (6,2)			If this is a variable-rate fee, the amount assessed for this student. Leave blank for fixed-rate fees.
dueDate	date		Yes	Date this payment is due.
exempt	boolean			Indicates if a fee is considered exempt and therefore does not need to be paid.
courseNum	string (13)		Yes	Include a unique course number within this calendar when the fee applies to a specific course.
comments	string (255)			Additional information about this fee.
paymentDate	date			The date the payment was made. Required if amountPaid is entered.
paymentAmount	float (6,2)			The amount the student has paid for this fee. Leave blank if no payment has been made.
checkNumber	string (15)			Number of the check used to make the payment. Leave blank if paid by credit card or cash.
creditCardAuthNumber	string (20)			Credit card authorization number when a credit card is used to make the payment. Leave blank if paid by check or cash.
paymentComments	string (50)			Additional information about this fee payment.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Custom Data

## customField

Required Columns: tabName, fieldName, dataType

Unique Key: tabName, fieldName

Constraints: None

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
tabName	string (50)	Yes	Yes	The name of the tab where this information resides. Shorter names are better for display purposes.
fieldName	string (35)	Yes	Yes	The name of the custom field where this information resides. It is recommended to submit names without spaces. You can modify the Display Name of the final conversion.
dataType	string (15)	Yes		The form the field should take ON the screen within the user interface. Refer to the appendix in the Data Migration Guide for a list of codes.
maxLength	int (3)			The maximum number of characters allowed for entry with the custom field.
seq	int (3)			The order in which the items are to be displayed. If no sequence is indicated, values will display in numeric or alphabetic order.
required	boolean			Indicates the custom field will require a value before the page can be successfully saved. In addition, the custom field name will display as red text (instead of black) and will be followed by an asterisk (*) to indicate that it is required.
hidePortal	boolean			When set to '1', indicates the custom field will not be displayed to parents/students ON the Portal.
default Value	string (255)			The value that is populated in the field when a new record is created.
comments	string (100)			Notes about this custom field.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

## customFieldItem

Required Columns: tabName, fieldName, code

Unique Key: tabName, fieldName, code

Constraints: The combination of tabName and fieldName must match a row in the customField file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
tabName	string (50)	Yes	Yes	The name of the tab where this information resides. Should match a value in customField.
fieldName	string (35)	Yes	Yes	The name of the field where this information resides. Should match a value in customField.
code	string (15)	Yes	Yes	An abbreviation for this item.
name	string (100)			The long description for this item
seq	int (3)			The order in which the items are to be displayed.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)



## customData

Required Columns: tabName, fieldName, studentNum

Unique Key: tabName, fieldName, studentNum, date

Constraints: The combination of tabName and fieldName must match a row in the customField file.  
studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
tabName	string (50)	Yes	Yes	The name of the tab where this information resides. Should match a value in customField.
fieldName	string (35)	Yes	Yes	The name of the field where this information resides. Should match a value in customField.
studentNum	string (15)	Yes	Yes	The district-wide unique identifier for the student.
value	string (256)			The information to be stored.
date	date		Yes	Enter a date here when a student can have more than one value for the specified field. NOTE: Do not enter a date in this field when the data type for the custom field is 'date'. Put this type of information in the 'Value' field of this file.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)

# Security

## userAccount

This file can be used to create staff accounts and student accounts. When adding staff accounts be sure to include the employNum from the Employee file. When adding student accounts be sure to include the studentNum from the student file.

Required Columns:    userName

Unique Key:            userName

Constraints:           When employNum has a value (other than blank or NULL) then employNum must match a row in the employee file.  
                             When studentNum has a value (other than blank or NULL) then studentNum must match a row in the student file.

[Back to TOC](#)

Name	Type	Required	Unique Key	Description
employNum	string(15)			The district-wide unique identifier for the employee.
studentNum	string(15)			The district-wide unique identifier for student.
userName	string(50)	Yes	Yes	The name that this employee or student will use to access campus.
password	string(40)			This user's initial password.

Use Alt+Left Arrow to Go Back

[Back to TOC](#)