

Custom Report Codes:

<http://www.powerschool.com/support/documentation/codes>

List of Tables and Fields: <http://psaddress/admin/home/?ac=structure>

School Info:

^(schoolname)

^([39]SchoolAddress)

^([39]schoolcity), ^([39]schoolstate) ^([39]schoolzip)

Ph: ^([39]schoolphone)

Fax: ^([39]schoolfax)

Stored Grades: used for object report card

^(stored.by.course.period;expression code;store code;what field to pull)

Ex: ^([39]stored.by.course.period;01(A).1;Q1;grade)

^(stored.by.course.period;01(A).1;Q1;teacher_name)

^(stored.by.course.period;01(A).1;Q1;Course_name)

Credit Hours: ^(*credit_hours) **Class Rank:** ^(*class_rank_out_of:gpa)

Custom GPA: ~(*gpa method=" AP honors" term="S1" year="2002")

Honor Roll: ^(*honorroll method="Honor Roll" term="S1" result="level")
^(*honorroll method="Honor Roll" term="S1" result="message")

The Honor Roll DAT returns data based on existing honor roll records.

^(*honorroll method="name" term="abbreviations" grade="grade levels" year="year number" result="data type")

The method parameter is required, and specifies the name of the honor roll method to return (honor roll methods are found at School Setup > Honor Roll). An error will be returned if this parameter is omitted. Note that the value of this parameter must be an exact match to the name of the honor roll method referenced.

The term parameter is also required, and can be passed multiple term abbreviations. An error will be returned if this parameter is omitted.

Example:

^(*honorroll method="High Honors" term="S1")

This would return the level value (see the result parameter, below) for any High Honors honor roll records awarded for the S1 term.

^(*honorroll method="AP" term="T1,T2")

This would return the level value (see the result parameter, below) for any High Honors honor roll records awarded for the T1 and T2 terms.

The grade parameter can be passed one or more comma-separated grade level values. If this parameter is omitted, only records from the current school year will be scanned. For kindergarten and pre-kindergarten students, use numerical values. The kindergarten grade is 0, pre-kindergarten (usually abbreviated "PK") is -1, and so on.

Example:

^(*honorroll method="HS Honors" term="Q2")

This would return the level value (see the result parameter, below) for any HS Honors honor roll records awarded for the Q2 term in the student's current school year.

^(*honorroll method="Principal's List" term="Q2" grade="6,7")

This example would return the level value (see the result parameter, below) for any Principal's List honor roll records awarded during the student's 6th and 7th grades in the Q2 term.

The year parameter is always passed at least one four-digit value if it is used. Use only the first year in a school term that spans multiple years; for instance, since most schools start in the summer, a full year name like "2003-2004" is common. Use only the first, i.e., "2003". If this parameter is omitted from the DAT, honor roll records from the current school year are scanned.

Example:

^(*honorroll method="Scholarship" term="S2" year="2003,2004")

This example would return the level value (see the result parameter, below) for any Scholarship honor roll records awarded for the S2 terms in the 2003-2004 and 2004-2005 school years.

The result parameter defines what is actually printed on reports when using this DAT. The values that can be passed to this parameter are "level", "message", "gpa", "schoolname", and "date". If the result parameter is omitted, "level" is the automatic default. The level value prints the name of the honor roll level met. The message value prints the text message for the honor roll level as defined on the school setup screen. The gpa value prints the actual GPA result that was used to determine the awarding of the honor roll level. The schoolname value prints the name of the school where the honor roll record was recorded. The date value prints the date the honor roll record was awarded (i.e., when the honor roll calculation process was run).

Examples:

^(*honorroll method="Scholarship" term="S2" result="level")

This would print the name of the Scholarship honor roll level awarded to the student for the S2 term of the current year, for example, "State College Fund".

^(*honorroll method="Honors" term="S2" grade="9" result="gpa")

This example would print the calculated GPA that triggered the awarding of the Honors honor roll record for the student's 9th grade in the S2 term, such as "3.92".

^(*honorroll method="Scholarship" term="Q3" result="message")

This would print the text message that has already been defined within the Scholarship honor roll level awarded for the Q3 term of the current year, for example, "Congratulations! You have been awarded the Gold Team scholarship for this year!"

^(*honorroll method="Future Leaders" term="T2" grade="7" result="schoolname")

This prints the name of the school where a Future Leaders honor roll record was awarded for the T2 term of the student's 10th grade, for example, "John Connor Middle School".

^(*honorroll method="AP Award" term="Y1" result="date")

This would print the date the AP Award honor roll award was recorded during for the Y1 term of the current year.

Class Rank: ^(*class_rank_out_of;*gpa method="simple")

The class rank Data Access Tag can be used to return a number of attributes of a student's currently calculated class rank. The asterisk in the ^(*classrank) DAT is required.

Parameter format:

^(*classrank method="method name" result="value" percentiledigits="decimal places")

The method parameter specifies the name of the GPA calculation method for which a rank should be returned. The value of the parameter must be an exact match with an existing method name; if it is mismatched, the DAT will always return the result "Not Ranked". If this parameter is simply omitted, a method named "weighted" will be used (this calculation method is a default supplied by PowerSchool).

The result parameter defines the type of data that will be returned by this DAT. The values that can be passed to this parameter are "gpa", "outof", "percentile", "rank", "rankof", "rankoutof", and "schoolname". Gpa returns the result of the GPA calculation used to determine class rank. Outof returns the total number of students on the class rank list. The percentile value returns the student's rank percentile (see also the percentile digits parameter below). Rank returns the numerical rank of the student, while rankof and rankoutof return the numerical rank and the number of ranked students, separated by the words "of" and "out of", respectively. The schoolname value returns the name of the school where the student earned the class rank. If the result parameter is omitted, "rank" is returned.

The percentile digits parameter can be used to define the number of decimal places to be used when passing the "percentile" value to the result parameter.

Examples:

^(*classrank)

Returns the student's rank value using the weighted class rank method, such as "26".

^(*classrank method="High Honors" result="gpa")

Returns the GPA calculated for the student by the High Honors class rank method used to determine ranking, for instance, "3.92".

^(*classrank method="AP" result="outof")

Returns the total number of students included in the class ranking list as determined by the AP GPA class rank method, for example, "336".

^(*classrank method="Honors" result="percentile")

Prints the student's rank percentile, as determined by the Honors class rank method, for example, "94.12".

^(*classrank method="Honors" result="percentile" percentiledigits="4")

Prints the student's rank percentile, as determined by the Honors class rank method, carrying the result out to four decimal places, like "26.1748".

^(*classrank method="Simple" result="rank")

Returns the student's rank value using the Simple class rank method, for instance, "42".

^(*classrank method="AP" result="rankof")

Returns the student's rank using the AP class rank method and the total number of ranked students separated by the word "of", as in "15 of 206".

^(*classrank result="rankoutof")

Returns the student's rank value using the Weighted class rank method and the total number of ranked students separated by the word "of", as in "4 out of 157".

^(*classrank result="schoolname")

Returns the name of the school where the class rank using the Weighted class rank method was calculated, for example, "Apple Grove High School".

Course listing/Other on transcript list object: *Any field from the Stored Grades Table*

Example: ^(grade_level), ^(course_number), ^(earnedcrhrs)

Decode:

^(decode;^(field);if;then;if;then)

Grade:

^(decode;^(grade_level);9;frosh;10;Soph;11;Junior;12; finally gone)

Gender:

^(decode;^(gender);m;Male;F;Female)

Ethnicity:

^(decode;^(ethnicity);c;Caucasian;I;NativeAmerican;o;other;h;Hispanic;a;asian;p;Polyns esian;f;African American)

Decode PG Final Grades:

^(decode;^(pg.final.grade);A;-;B;-;C;-;D;-;F;-;^(pg.final.grade)

Decode PG Final Grades to Percent:

^(decode;^(pg.final.grade);A;^(pg.final.percent);B;^(pg.final.percent);C;^(pg.final.percent);D;^(pg.final.percent);F;^(pg.final.percent);-)

If/Then:

^(field;if.fieldvalue.xxx.then=)

Ethnicity:^(ethnicity;if.fieldvalue.F.then=African American

Last update

^(pg.final.lastupdate;term)

Add Lunch Balance in Obj Reports: ^(balance2002;+balance2003)

Count using special option: ~(count)

Test Codes:

~(tests;name=ACT_1;which=last;type=num;result=DATE)

ACT 1 - English: ~(tests;name=ACT_1;score=english_1;type=num;result=value)

ACT 1 - Reading: ~(tests;name=ACT_1;score=reading_1;type=num;result=value)

ACT 1 - Math: ~(tests;name=ACT_1;score=math_1;type=num;result=value)

ACT 1 - Science: ~(tests;name=ACT_1;score=science_1;type=num;result=value)

ACT 1 - Composite: ~(tests;name=ACT_1;score=composite_1;type=num;result=value)

How do you report test scores?

The following uses the sample test name SAT. Currently, you may only report the FIRST, LAST, BEST and CURRENT* test.

To report the score of the FIRST SAT test, use this format:

^(tests;name=SAT;score=math;which=first;type=num;result=value)

To report the score of the LAST SAT test, use this format:

^(tests;name=SAT;score=math;which=last;type=num;result=value)

To report the score of the BEST SAT test, use this format:

^(tests;name=SAT;score=math;which=best;type=num;result=value)

***To report the score of a SAT test taken during the CURRENT TERM use this format:

^(tests;name=SAT;score=math;which=term.current;type=num;result=value)

*Note: Regarding the which=term.current parameter: The term selected at the top of the PowerSchool screen has to match the term that the student took the test during (example: if a student had an SAT score that was set with the 2004-2005 full year term, then in

order to pull the proper score with the code, you must select the full year term at the top of your PowerSchool screen, otherwise you may see 0 for the score). As you can see, this is a somewhat dynamic parameter and can be used to report any "current" term.

Issue:

PowerSchool seems to be unable to read multiple SAT score information correctly. When printing a student transcript the information is only accounting for one set of scores. Please advise.

Cause:

When multiple tests have the same "score" name ("MATH" in ACT and SAT) we were seeing a problem getting the correct value pulled. The application is disregarding the "name" code (name=sat, name=act) and just looking at the first record ID of the "score" type specified.

Solution/Workaround:

To ensure the code pulls a value from the correct "score" field, add the code operator: "testscoreID=X" where "X" is the value held in the TestScoreID field within the "StudentTestScore" table. This value is populated with the "ID" field value located with in the "TestScore" table.

The "TestScore" table houses all score types setup in the database, and has a "testID" field that is populated by the value held in the ID field of the "test" table.

Here is the logical path of propagation between the 3 tables:

[Test]ID ---> [TestScore]TestID
[TestScore]ID ---> [StudentTestScore]TestScoreID

Here is a live example in USM trying to find the SAT-Math score:

- TEST (086) table - select all records and do List View
note the test named SAT has an ID of 4
- TESTSCORE (088) - search for testid = 4 and do List View of results
note the ID and Names for the SAT test
example: 9 Reading; 10 English; 11 Math
- TESTSCOREID = 11

Export Test Scores:

StudentID
[01]lastfirst
TestScoreID
[88]testid
[87]test_date
[88]name
NumScore
PercentScore
AlphaScore

Example Test Score Decode: Don't Show Empty Test Scores

```
<b><u>TESTS</u></b>
<tabto 4.92><b>SAT I's</b>
<tabto 5.25><u>Date</u><tabc 6.00><u>Verbal</u><Tabc 6.5><u>Math</u>
<tabto 5.25>^(decode;~(tests;name=SATI-
1;score=any;type=any;which=last;result=date);00/00/00;<>;~(tests;name=SATI-
1;score=any;type=any;which=last;result=date))<tabc 6.00>^(tests;name=SATI-
1;score=verbal;which=first;result=value;format=###)<Tabc
6.5>^(tests;name=SATI-1;score=math;which=first;result=value;format=###)
```

Credit Hours by Credit Type Codes:

```
^(*credit_hours.credit_type.eng)
^(*credit_hours.credit_type.math)
^(*credit_hours.credit_type.sci)
^(*credit_hours.credit_type.soc)
^(*credit_hours.credit_type.pe)
```

Credit Hours by Grade Level:

```
09 Crs: ^( *credit_hours;09) ^( *credit_hours;09;s1)
10 Crs: ^( *credit_hours;10)
11 Crs: ^( *credit_hours;11)
12 Crs: ^( *credit_hours;12)
Total Crs: ^( *credit_hours)
```

Credit Hours Attempted:

```
Total Crs Attempted: ^( *potential_credit_hours)
Credits Attempted 9: ^( *potential_credit_hours;9)
Credits Attempted 10: ^( *potential_credit_hours;10)
Credits Attempted 11: ^( *potential_credit_hours;11)
Credits Attempted 12: ^( *potential_credit_hours;12)
```

Credit Hours Attempted using course listing: ^(translinecredithours)

Teacher Name Codes:

```
Full Name: ^( *period_info;01(A);teacher_name)
Title: ^( *period_info;01(A);teacher_I_title)
First Name: ^( *period_info;01(A);teacher_I_first_name)
Last Name: ^( *period_info;01(A);teacher_I_last_name)
Use period expression to pull period/schedule info:
( *period_info;HR(A) ;teacher_name)
```

Date Formatting: ^(*DA;Dates of Term;format=##.##)

Attendance Codes: Showing period, attendance code, day and date (thanks Ken)

~(dates.of.attendance;01(a);br.code;*all;Q1) Anything before 4.0

~(dates.of.attendance;01;tab.code;*abs;1/1/2003;1/5/2003) Anything before 4.0

~(ATT_DatesofAttendance;01(a);br.code;*all;Q1) If using 4.0

~(ATT_DatesofAttendance;01;tab.code;*abs;1/1/2003;1/5/2003) If using 4.0

This code, as with all other attendance codes pull period attendance. The 01 does indicate the period. As with other codes, the period name must be exactly as you have in period set-up. The br.code indicates that between each date you want a break placed there. The .code extension will preface the date with the attendance code. You could replace it with tab.code to get tab separated dates, *all means to include all attendance codes. You could replace this with one individual code, or with *abs for just codes that equate with absences or *tar for tardies.

Daily Attendance Codes:

- ^ (daily.att.count;school list;att code list;term abbr | { start date;end date })
- ^ (daily.att.points;school list;att code list;term abbr | { start date;end date })
- ^ (daily.att.dates;school list;displayformat;att code list;term abbr | { start date;end date;dateformat })

Attendance Points:

^(attpointsperiod;01(a);Q4) ^(attpointsperiod;02(a);Q4) ^(attpointsperiod;03(a);Q4)

^(attpointsperiod;03(a);Q4) ^(attpointsperiod;04(a);Q4) ^(attpointsperiod;05(a);Q4)

^(attpointsperiod;06(a);Q4)

Roster PDF: Grade Verification: Start Page > reports > report setup > Class Roster

Roster: Grades from PowerGrade

lastfirst\Name\2.5

^(pg.final.grade;Q1)\Grade\5

^(pg.final.grade;Q1)\percent\5

^(att;abs;Q1)\Abs\5

^(att;tar;Q1)\Tar\5

Roster: Grades from Stored Grades Table (Historical Grades)

lastfirst\Name\2.5\L

grade_level\Grade\1.0\L

^(historical;Q1;grade)\Q1\5\c

^(historical;Q1;percent)\Q1\5\c

^(historical;Q1;earnedcrhrs)\EC\5\c

^(att;abs;Q1)\ABS\5\c

^(att;tar;Q1)\TAR\5\c

The **header field** in the Roster Report will accept the following tags:

Anything from the sections table

- ^(Comment)
- ^(Expression)
- ^(Grade_Level)
- ^(LastAttUpdate)
- ^(MaxEnrollment)
- ^(No_of_Students)
- ^(PGVersion)
- ^(Room)
- ^(Section_Number)
- Any other field will work

Anything from the teachers Table

- ^([teachers]email_addr)
- ^([teachers]First_name)
- ^([teachers]Last_name)
- ^([teachers]School_phone)
- ^([teachers]Homeroom)
- ^([teachers]Sched_department)
- ^([teachers]Sched_classroom)
- Any other field will work including all the scheduling fields.

Anything from the Courses

- ^([courses]Credit_hours)
- ^([courses]CreditType)
- ^([courses]Course_Name)
- ^([courses]Sched_department)
- Any other field will work including all the scheduling fields.

Anything from the Terms table

- ^([terms]Name)
- ^([terms]Abbreviation)
- ^([terms]NoOfDays)
- ^([terms]FirstDay)
- ^([terms]LastDay)
- Any other field will work.

Anything from the Schools table

- ^([Schools]Name)
- ^([Schools]Abbreviation)
- ^([Schools]Principal)
- ^([Schools]PrincipalPhone)
- Any other field will work.

Log Object Reports

The New Object Report page, accessible via Start Page > Reports > Report Setup > Object Reports > New, appears the same as in previous releases. The notable difference is the addition of Log to the Table pop-up menu. Selecting Log provide you access to the Log table from which you can select data which you want to include in the report. Proceed in creating the objects report as you have done in previous releases. For detailed page information, visit PowerSchool Online Help.

Note: Although Log object reports are created using the same Objects reporting tool used to create Students and Staff objects reports, you cannot print Log object reports via Start Page > Special Functions > Group Functions > Print Reports as you would for Students and Staff objects reports. Log object reports must be printed via Start Page > Special Functions > Search Log Entries. For more information, see Output Options below.

Note: Selecting Log from the Table pop-up menu provides you access to the Log table from which you can select any field, including custom fields, you want to include in the report. Each field you want to include in the log object report must be specified using the correct syntax in order to successfully display the data. The text that precedes the syntax (separated by a colon) can be modified (or deleted) based on your needs. For example, you could modify Name: ^([01]LastFirst) to appear as Student's Last Name: ^([01]LastFirst).

The following are Log table student fields you can use in the report:

Name: ^([01]LastFirst)
Grade: ^([01]Grade_Level)
SN: ^([01]Student_Number)
DOB: ^([01]DOB)
SSN: ^([01]SSN)
Ethnicity: ^([01]Ethnicity)
Mailing City: ^([01]Mailing_City)
Mailing Street: ^([01]Mailing_Street)
Mailing State: ^([01]Mailing_State)
Mailing Zip: ^([01]Mailing_Zip)
Alert Discipline: ^([01]Alert_Discipline)
State Student #: ^([01]State_StudentNumber)

The following are Log table discipline fields you can use in the report:
Make sure to surround fields with: ^(Field)

^(Entry Author)
Entry Date:
Entry Time:
Title:
Entry Text:
Log Type ID:
Subtype:
Category:

Consequence:
Incident Type:
Action Taken Detail:
Incident Type Category:
Action Taken End Date:
Incident Type Detail:
Student Number: 840001326
Administrator ID:
Likely Injury Flag:
Alcohol Related Flag:
Money Loss Value:
Drug Related:
Offender:
Drug Type Detail:
Police Involved:
Duration Actual:
Reporter:
Duration Assigned:
Reporter ID:
Duration Change:
School Rules Vio Flag:
Duration Notes:
Sequence:
Felony Flag:
Victim Type:
Gang Related Flag:
Weapon Related:
Hate Crime:
Weapon Type:
Hearing Officer:
Weapon Type Notes:
Incident Context:
Incident Date:
Custom:
Incident Location:
Action Date:
Incident Loc Detail:
Discipline Action:

Same Applies to CC Table:

Excel Helper App:

Internet Explorer -> Excel - Macintosh

1. Choose Preferences from the Explorer menu.
2. Click File Helpers.
3. Click Add.
4. Enter PowerSchool Export in the Description field.
5. Enter text/ps-export in the Extension and MIME type fields.
6. Click Browse in the File Type Section.
7. Choose Microsoft Excel.
8. Enter TEXT in the "File type" field.
9. Choose View with Application from the How to Handle pop-up menu.
10. Click OK.
11. Click OK on the Internet Explorer Preferences window.
12. Use Quick Export to export data from PowerSchool.
13. Locate the file you exported on your desktop.
14. Select the file and choose Show Info from the File menu.
15. Choose Open with Application from the pop-up menu.
16. Click the application icon and choose Microsoft Excel.
17. Click Change All.
18. Click Continue.

Balance1: Current Lunch Balance

Balance2: Beginning balance (Displayed on the Lunch Transactions page.)

Balance3: Current Student Fee Balance

Balance4: Beginning Student Fee Balance (Displayed on the Fee Transactions page.)

Bar code Font:

(lunch_id)

Create a Link

Click here to see site