**Completion Analysis – All**



Overview. I reloaded this to a test environment. You will use your production environment. There is no select filter. The entire GRADUATES file is read. When you filter and update, you can create a savedlist of GRADUATES keys (*student\_id\*program*) and add a GET.LIST as a select filter.

Columns:

There are quite a few columns from the base (GRADUATES) file and from linked files

GRADUATES:

Ptr (ID)

Student ID

Pgm

Term

Requirements Completed

Grad Honors

Allow Commencement

Comments

Diploma Name

Degree Date

CCD Date

Commencement Date

End pgm on Commencement

Chgdate

STUDENT.PROGRAMS:

Catalog

Eval Status

Eval Date

Pgm Current Status

Pgm Cur Status Date

Hidden:

# These are used as input to scripts and otherwise do not need to be displayed

Raw eval major credit (credit completed/registered)

Raw Pgm GPA

Raw STPR Eval Major Cred

Raw STPR Eval Inst Cred

Script (see below):

Pgm Cred Compl

Num Reqd Cred (not used)

ACAD.PROGRAMS

Required Credit

Reqd Institution Cred

Pgm Title

Division (School)

STUDENTS

First Name

Middle Name

Last Name

Suffix

Scripts:

**Pgm Cred Compl**

/ studentPrograms\_assoc\_stprEvalMajorCredTot is a computed column, that although a scalar value

// returns as an array due to the nature of the way computed columns are handled.

// This variable cannot be used for math or logical operations because it is a pointer.

// We could take the quick way out and use studentPrograms\_assoc\_stprEvalMajorCredTot[0] as

// the variable. Instead we make sure to loop through and build a sum of all returned values (which

// should always be just one value, but you never know.

// SCD

var len = studentPrograms\_assoc\_stprEvalMajorCredTot.length;

var toReturn = 0;

var cmplCred = 0;

for (var i = 0; i < len; i++) {

 cmplCred += studentPrograms\_assoc\_stprEvalMajorCredTot[i];

}

toReturn = cmplCred / 1000.0;

toReturn;

**Cred Needed**

var needCred = 0;

if (pgmCredCompl < reqdCred) {

 needCred = reqdCred - pgmCredCompl;

}

needCred;

**Percent Complete**

var pctCompl = 0.0;

if (reqdCred != 0) {

 pctCompl = (pgmCredCompl \* 100.0) / reqdCred;

}

if (pctCompl > 100) {

 pctCompl = 100

}

pctCompl;

**Pgm GPA**

var len = studentPrograms\_assoc\_stprEvalMajorGpa.length;

var myGpa = 0;

for (i = 0; i < len; i++) {

 myGpa += studentPrograms\_assoc\_stprEvalMajorGpa[i];

}

myGpa /= 100.0;

myGpa;

**Num Reqd Cred**

// This field is used to calculate credit needed and percent complete.

// The source field (studentPrograms\_assoc\_xStprAcprCred.length) looks

// lovely as a display field, but cannot be used in numerical calculations.

// We hide this field so as not to display too many variants on number of

// hours in the program.

// Ideally, we would check for exceptions to the number of hours in

// STUDENT.DA.EXCPTS, but I’m not that good (yet).

// SCD

var len = studentPrograms\_assoc\_xStprAcprCred.length;

var credNeeded = 0;

for (var i = 0; i < len; i++) {

 credNeeded += studentPrograms\_assoc\_xStprAcprCred[i];

}

credNeeded;