

Appendix A: Trait Data Type for KDSmart version 2.0.43

B.Pearce 2016-04-12

The *DataType* in each Trait definition specifies the values that may be recorded in each *Sample* for subsequent transfer to the database. If you load a *Trial* from a CSV file and the name of the *Trait* has not yet been defined in *KDSmart* the *DataType* defaults to *Text*.

Data Type	Description	Data Entry Style	Example Validation Rule
TEXT	No constraint on what may be entered.	A text input field with a full keyboard.	No rule required
CATEGORICAL	Value is constrained to be one from a list of allowed values. Translations to other languages are supported.	A list of the allowed values is presented and the user touches the value desired.	Example: CHOICE (BLUE YELLOW RED) There must be at least two values provided in the “pipe” () separated list. You may also provide a description for each value. This is done using the form value : description This means that you cannot use the colon (:) character in a value.

Data Type	Description	Data Entry Style	Example Validation Rule
DATE	Value is a date.	A calendar is presented from which to choose a date.	May be left blank or specified as <code>date</code>
ELAPSED_DAYS	<p>Value is the number of days since the Trial's <i>Planting Date</i>.</p> <p>The range in the calendar is constrained by the start date and an optional upper limit.</p> <p><i>KDSmart</i> stores the values as a date but presents the value as a number when required.</p>	<p>A list of days “around” the current date is presented and the user touches the desired value or chooses “<i>other</i>” to use a calendar to pick a date outside this range.</p> <p>The choices in the list (<code>-3d</code>, <code>-2d</code>, <code>-1d</code>, <code>Today</code>, <code>+1d</code>, <code>+2d</code>, <code>+3d</code>) may be presented in different languages depending on locale support.</p>	<p>May be left blank or specified as <code>elapsed_days</code></p> <p>To enforce a maximum value, use: <code>elapsed_days_max=NNN</code></p>

Data Type	Description	Data Entry Style	Example Validation Rule
INTEGER	<p>Integer numeric values in a range specified.</p> <p>The range is specified as a lower and upper bound and whether or not the bounds are included in the range.</p>	<p>If the number of values is “small” then entry is similar to the <i>Categorical</i> data type. Otherwise, a numeric keypad is presented (excluding a decimal point key).</p> <p>The changeover from <i>Categorical</i> style to numeric keypad may be altered in the <i>Settings</i> screen.</p> <p>For Example: <code>RERANGE(1..5)</code> presents as a list of choices: 1, 2, 3, 4 but <code>RANGE(1..500)</code> presents a keypad.</p>	<p>Specify the range of values as:</p> <p style="text-align: center;"><code>RANGE(min..max)</code></p> <p>or use <code>LERANGE</code>, <code>RERANGE</code>, <code>BERANGE</code> to exclude the left, right or both limits respectively.</p> <p>Example: <code>RERANGE(-1..5)</code> means the accepted values are -1, 0, 1, 2, 3, 4</p> <p>NOTE: If you use a custom keyboard, this may affect the appearance of the numeric keypad.</p>
DECIMAL	<p>Specify a lower and upper bound and whether or not the bounds are included in the range and the number of digits of precision that will be recorded.</p>	<p>Entry is done using a numeric keypad that includes the decimal point.</p> <p>NOTE: If you use a custom keyboard, this may affect the appearance of the numeric keypad.</p>	<p><code>RANGE(min..max)</code></p> <p>or use <code>LERANGE</code>, <code>RERANGE</code>, <code>BERANGE</code></p> <p>However, <code>min</code> and <code>max</code> must now contain a decimal point and at least one digit following the point.</p> <p>So: <code>BERANGE(1.0..2.99)</code> means accept values for <code>x</code> where</p> <p style="text-align: center;"><code>1.00 < x < 2.99</code></p> <p>(the greater number of digits after the “.” for min/max is the “precision” desired)</p>

Data Type	Description	Example Validation Rule
<p>CALC</p> <p>Data entry is not permitted for this type and Traits with this data type CANNOT be scored during Data Collection.</p>	<p>Calculated or Derived Trait,</p> <p>The validation rule has either one or two components (with a comma separating them if required):</p> <ul style="list-style-type: none"> The first component provides a formula that may reference other <i>Traits</i> (by the <i>Trait Name</i>), using the operators and functions listed below. If present, the second component indicates the number of decimal digits to retain in the computed result. <p>The functions available are in a table below.</p> <p>Arithmetic operators are:</p> <p style="padding-left: 40px;">* / % (the last is the modulo operation)</p> <p style="padding-left: 40px;">+ - (unary minus is also supported)</p> <p style="padding-left: 40px;">^ ** (for exponentiation)</p> <p>Note that the comparison operators:</p> <p style="padding-left: 40px;">< <= = != > >=</p> <p>and the logical operators:</p> <p style="padding-left: 40px;">&& </p> <p>evaluate to either 1 or 0 representing true or false respectively. This is relevant for the if(cond, a, b) function.</p>	<p>A simple calculated or derived trait is:</p> <p style="padding-left: 40px;">CALC(max(PH_CM/5,1))</p> <p>This calculation references a <i>Trait</i> named PH_CM, dividing the value by 5 and returning the maximum of the division or 1, whichever is the greater.</p> <p>So a value of 17 for PH_CM gives a result of 3</p> <p>Alternatively,</p> <p style="padding-left: 40px;">CALC(max(PH_CM/5,1) , 1)</p> <p>will retain the result with one decimal place.</p> <p>In this case the value of 17 for PH_CM results in 3.4</p>

Functions available for CALC data type

Function Name(s)	Description	Example
<code>sin cos tan</code> <code>asin acos atan</code>	Trigonometric functions	
<code>sinh cosh tanh</code>	Hyperbolic functions	
<code>abs</code> <code>round floor ceil</code> <code>sqrt exp ln log</code> <code>sign</code>	Unary mathematical functions: ROUND, FLOOR and CEIL return an integral portion of the input. LN is the natural logarithm LOG is the logarithm to base 10	<code>round(1.5)</code> returns 2.0 <code>round(1.4)</code> returns 1.0 <code>floor(1.5)</code> returns 1.0 <code>ceil(1.5)</code> returns 2.0
<code>pow min max</code>	Binary mathematical functions	<code>pow(2,3)</code> returns 8.0 <code>pow(3,2)</code> returns 9.0
<code>rnd</code> <code>deg(radians)</code> <code>rad(degrees)</code>	Other functions: <code>rnd(a)</code> returns the value of a multiplied by a random number satisfying the constraint <code>RERANGE(0..1.0)</code> i.e. a value x where: $0.0 \leq x < 1.0$ <code>deg</code> and <code>rad</code> convert between degrees and radians.	<code>rnd(10)</code> will return a value y satisfying: $0 \leq y < 10$
<code>if(expr , a , b)</code>	Conditional expression: If the value of expr is 1 , the result is the expression a otherwise the result is expression b	<code>if(PH_CM <= 5 , 1 , 2)</code> evaluates to 1 if the value of the trait PH_CM is 5 or less otherwise to 2