

Discussion:

Where are names used?

Primitives:

- in selection combos
- in equations
- in the data manager
- in the 'view ...' dialog
- in the edit option in the plot zoom mode rmb menu.

Data Objects:

- in the data manager
- in the edit option in the plot zoom mode rmb menu.

Relations:

- in the data manager
- in the 'curve selection' places (plot dialog, legend dialog)
- in the edit, fit, and filter options in the plot zoom mode rmb menu.

- The names of primitives are entered via the keyboard for labels and equations - which argues for the presence of a short name! The primary goal of the name, elsewhere, is to disambiguate objects for the user.
- In real use, the “best name” for the object can often not be guessed by kst (for the most extreme example, think equations...). So a simple system is probably better than something complex. (“The design had all the disadvantages, and none of the advantages of the more complex design”).
- In most cases, further disambiguation information can be put into the UI without having it actually be the name of the object.
- Typically, automatic names will be based on some trait of the object. In this case, the automatic name should change if the trait is changed.
- Uniqueness needs to be maintained within each group of objects (primitives, data objects, relations).

Names:

Objects will be named

`<descriptiveName>:<typeID><pos>`
eg, INDEX:V1

<descriptiveName>: a short name which, in the common case, has a chance of providing a unique name to the user. eg, the field name for data vectors. It can

either be automatically generated, or set manually by the user through the UI.

<typeID>: A single character which indicates the type of object:

V: Vector
X: Scalar
M: Matrix
T: String (Text)
E: Equation
H: Histogram
S: Spectrum
P: Plugin
G: SpectroGram
C: Curve
I: Image

<pos>: An integer unique to the object, which is permanently set at creation time. It must be unique within each type, but need not be globally unique (in fact, as there could be hundreds of different objects in a session, it is best if it isn't, to keep the numbers down.)

Where the object is referred to by entering from the keyboard (eg, in an equation or a label), it can be called

[<descriptiveName>:<typeID><pos>]

eg, [INDEX:V1]

or

[<descriptiveName>]

eg, [INDEX]

or

:<typeID><pos>

eg, [:V1]

The first case is always unique, but may change (if the name is being automatically generated, and the object's properties change).

The second case may not refer to a unique primitive. In this case, the first one found will be used (or should there be an error?)

The third case is always unique, and won't change.

When the string is regenerated (eg, repopulating the equation dialog) the first

case will always be used.

Automatically generated descriptiveNames:

Slave objects are named

<descriptiveName>/<ID>

eg, GYRO1/Mean

Type	descriptiveName	Slave Vector or Matrix names
dataVector	<fieldname>	NA
dataScalar	<fieldname>	NA
dataMatrix	<fieldname>	NA
dataString	<fieldname>	NA
spectra	<fieldname>	<descriptiveName>/psd <descriptiveName>/f
Spectrogram	<fieldname>	<descriptiveName>/SG
Histograms	<fieldname>	<descriptiveName>/num <descriptiveName>/bin
Images	<fieldname>	NA
Curves	<Y descriptive Name> vs <X descriptive Name>	NA
Equations	<truncated EQ>	<descriptiveName>/y <descriptiveName>/x
Plugins	plugin defined	<descriptiveName>/<plugin defined>

Examples

In these examples, the full name is given. All objects can be uniquely referred to by their suffix only.

i) The second vector created is read from field GYRO1

GYRO1:V2

ii) The 5th vector created is read from field GYRO1, but from a different data source:

GYRO1:V5

iii) The scalar “mean of GYRO1:V2”

GYRO1/Mean:X3

iv) The scalar mean of GYRO1:V5

GYRO1/Mean:X42

v) The mean of the Low Pass filtered output of the spectrum of GYRO1:V5

```
GYR01/psd/LP/rms:X53
```

vi) A curve of the LP filtered output of the spectrum of GYRO1:V5

```
GYR01/psd/LP vs GYRO1/f:C4
```

Tooltips

In exchange for keeping the names short, it is not always clear from the automatic names what they are from (eg, the data source is not in the name - see the examples above). Consequently, the UI will need to give these hints to the user in some other way.

UI elements (eg, vector selectors) which are used to select data objects will have a mouse over tool-tip which will list detailed information about the object that the mouse is over.

For the rather extreme example (v), above, the tool-tip would tell us something like (this format is *not* final!):

```
GYR01/psd/LP/rms:X53 rms
GYR01/psd/LP:P4 Low pass filter
  Vector: GRY01/psd:S1
  cutoff: 0.01
GYR01/psd:S1 Spectrum
  Input: GYRO1:V5
  Sample Rate: 100.16 Hz
  Remove Mean; Apodize Hanning
  Interleaved Average 2^14
  ASD (V/sqrtHz) output
GYR01:V5 Data Vector
  /data/run7.dat
  GYRO1
  f0: 0 N: 200 S:0 A:false
```

Object Names in kst files.

Objects will be referred to by their suffix (eg V2) in kst files, so objects will keep these numbers between sessions. The descriptiveName will not be used.

Translation:

The descriptiveName text and TypeIDs are both subject to translation. However, the typeIDs must be used untranslated in .kst files.