(2) iFit/iData :

« One Class to hold them all, One Class to handle them »

E. Farhi <farhi@ill.fr>

Object programming vocabulary

An object (or class instance) holds some data named '**properties**'. These are equivalent to **fields** in a structure (storage area for named values):

Structure: *structure.property1*
structure.property2 ...

The object can be sent to **methods**, which are dedicated functions taking objects (and other arguments) as input.

plot(object)
object1 + object2 is equivalent to *plus(object1, object2)*

In **Matlab**, the usual object classes are 'double','character','structure','cell' (as well as a few others).

One single object to hold any data set

The spirit of iFit relies on a single object/class definition: ***iData***

An iData 'object' is a variable which is manipulated just as any number, vector or matrix, and carries additional information (error bars, axes, labels, aliases, …).

To create an iData object, we convert the initial data set into an iData one:

```
>> a = iData( rand(10,15) )
>> b = iData('filename')
>> c = iData(x,y,z)
```

The conversion can be used with file names, vector/matrices, structures, cell arrays.

iData object structure

The MetaData (properties)

Tag	***Unique ID***
Title	Global name of the data set
Source	***Origin*** (file, URL, variable)
Creator	Program that generated the data
User	Author/user ID
Date	Creation date
ModificationDate	Modification date
Command	Log of command history
UserData	User-specified data (opt.)
Label	Data category (opt.)
DisplayName	Name shown in legends/plots (opt.)

The Data and its Visible part

Data	The ***scientific data hierarchy*** as read from the file, URL, variable... May be a single numeric, but also a structure or cell.

Aliases	***Links*** to other object parts, e.g. to the Data hierarchy.

Signal ***Error*** ***Monitor*** ***Axes***	Links to object Aliases or other object parts (e.g. the Data hierarchy) which define the visible part of the object as a scientific information

iData object soft example...

```
>> addpath(genpath('Matlab/iFit/'));
```

Install iFit

iData object soft example...

```
>> addpath(genpath('Matlab/iFit/'));
>> a = iData(peaks)
```

Request to convert the 'peaks' matrix to an iData. 'peaks' is a 49x49 matrix known by matlab.

iData object soft example...

```
>> addpath(genpath('Matlab/iFit/'));
>> a = iData(peaks)
iData: Setting Signal="Data.Signal" with length 2401 in object id541970 "double".
a =  iData object:

    [Tag] [Dimension]                                          [Title] [Last command]
 id541970     [49 49]                          'double "Data Signal"' id541970=setalias(id...
```

'peaks' is stored into a.Data.Signal, which is used as the Signal definition in the object.
The object 'abstract' is displayed.

iData object soft example...

```
>> addpath(genpath('Matlab/iFit/'));
>> a = iData(peaks)
iData: Setting Signal="Data.Signal" with length 2401 in object id541970 "double".
a =  iData object:

    [Tag] [Dimension]                                        [Title] [Last command]
 id541970     [49 49]                    'double "Data Signal"' id541970=setalias(id...
>> plot(a)
```

Request to plot the object

iData object soft example...

```
>> addpath(genpath('Matlab/iFit/'));
>> a = iData(peaks)
iData: Setting Signal="Data.Signal" with length 2401 in object id541970 "double".
a =  iData object:

    [Tag] [Dimension]                                          [Title] [Last command]
 id541970     [49 49]                        'double "Data Signal"' id541970=setalias(id...
>> plot(a)
Warning: iData/getaxis: The 2-th rank axis has not been defined yet (use setaxis).
    Using default value=1:49 in object a id541970 "double".
Warning: iData/getaxis: The 1-th rank axis has not been defined yet (use setaxis).
    Using default value=1:49 in object a id541970 "double".
```

Request to plot the object

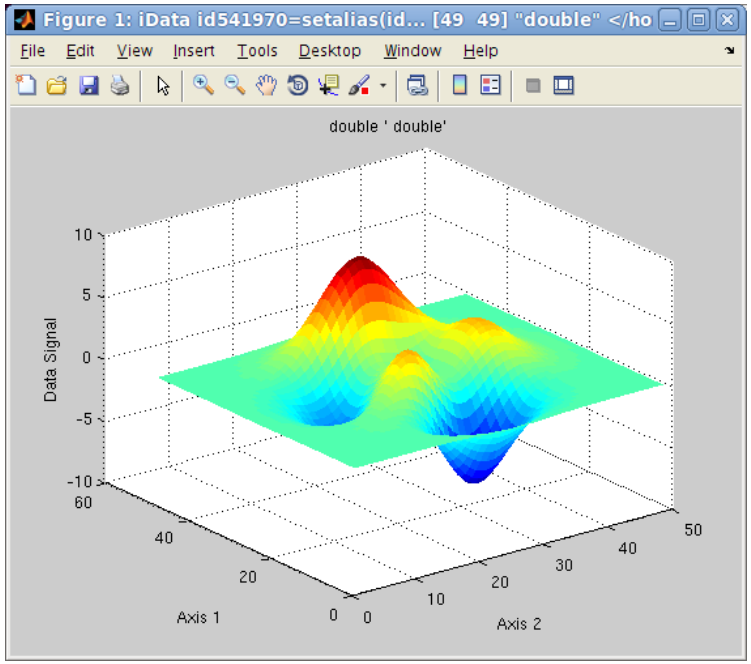
iData object soft example...

```
>> addpath(genpath('Matlab/iFit/'));
>> a = iData(peaks)
iData: Setting Signal="Data.Signal" with length 2401 in object id541970 "double".
a =  iData object:

    [Tag] [Dimension]                                          [Title] [Last command]
 id541970     [49 49]                        'double "Data Signal"' id541970=setalias(id...
>> plot(a)
Warning: iData/getaxis: The 2-th rank axis has not been defined yet (use setaxis).
    Using default value=1:49 in object a id541970 "double".
Warning: iData/getaxis: The 1-th rank axis has not been defined yet (use setaxis).
    Using default value=1:49 in object a id541970 "double".

ans =

  174.0016
```



The object is rendered.

iData object content

Let's now see how is displayed the object:

```
>> a
a =  iData object:

    [Tag] [Dimension]                                    [Title] [Last command]
 id541970     [49 49]                  'double "Data Signal"' id541970=setalias(id...
>> disp(a)
```

Request to display the object structure.
`get(a)` does the same.

iData object content

Let's now see how is displayed the object:

```
>> a
a =  iData object:

    [Tag] [Dimension]                                          [Title] [Last command]
 id541970     [49 49]                          'double "Data Signal"' id541970=setalias(id...
>> disp(a)
a = iData 2D object of size [49  49]:
                  Tag: 'id541970'
                Title: 'double'
               Source: '/home/farhi'
              Creator: [1x122 char]
                 User: 'farhi'
                 Date: '13-Jan-2012 10:51:54'
     ModificationDate: '13-Jan-2012 10:51:54'
              Command: {2x1 cell}
             UserData: ''
                Label: ''
          DisplayName: ''
```

Metadata information.

iData object content

Let's now see how is displayed the object:

```
>> a
a =  iData object:

    [Tag] [Dimension]                                    [Title] [Last command]
 id541970     [49 49]                        'double "Data Signal"' id541970=setalias(id...
>> disp(a)
a = iData 2D object of size [49  49]:
                  Tag: 'id541970'
                Title: 'double'
               Source: '/home/farhi'
              Creator: [1x122 char]
                 User: 'farhi'
                 Date: '13-Jan-2012 10:51:54'
    ModificationDate: '13-Jan-2012 10:51:54'
              Command: {2x1 cell}
             UserData: ''
                Label: ''
          DisplayName: ''
                 Data: [1x1 struct]
```

The object guts !

```
>> a.Data

ans =

    Signal: [49x49 double]
```

Contains a.Data.Signal

iData object content

Let's now see how is displayed the object:

```
>> a
a =  iData object:

    [Tag] [Dimension]                                    [Title] [Last command]
 id541970     [49 49]                        'double "Data Signal"' id541970=setalias(id...
>> disp(a)
a = iData 2D object of size [49  49]:
                  Tag: 'id541970'
                Title: 'double'
               Source: '/home/farhi'
              Creator: [1x122 char]
                 User: 'farhi'
                 Date: '13-Jan-2012 10:51:54'
    ModificationDate: '13-Jan-2012 10:51:54'
              Command: {2x1 cell}
             UserData: ''
                Label: ''
          DisplayName: ''
                 Data: [1x1 struct]

Object aliases:
          [Name]                              [Value]   [Description]
          Signal                         Data.Signal   Data Signal '[6.7e-05 0.0001 0....' [-6.5
           Error                       sqrt(Signal)   Error on Signal
         Monitor                                  1   Monitor (weight) '1'
```

The object aliases.

```
>> a.Signal → a.Data.Signal
>> a.Error → sqrt(Signal)
>> a.Monitor → 1
```

Also displays content overview (min, max, first values)

iData object content

Let's now see how is displayed the object:

```
>> a
a =  iData object:

    [Tag] [Dimension]                                    [Title] [Last command]
 id541970     [49 49]                    'double "Data Signal"' id541970=setalias(id...
>> disp(a)
a = iData 2D object of size [49  49]:
                  Tag: 'id541970'
                Title: 'double'
               Source: '/home/farhi'
              Creator: [1x122 char]
                 User: 'farhi'
                 Date: '13-Jan-2012 10:51:54'
    ModificationDate: '13-Jan-2012 10:51:54'
              Command: {2x1 cell}
             UserData: ''
                Label: ''
          DisplayName: ''
                 Data: [1x1 struct]

Object aliases:
         [Name]                          [Value]   [Description]
         Signal                    Data.Signal   Data Signal '[6.7e-05 0.0001 0....' [-6.5
          Error                  sqrt(Signal)   Error on Signal
        Monitor                             1   Monitor (weight) '1'
Object axes:
[Rank]          [Value]  [Description]
     0           Signal  Data Signal [-6.54664:8.07517] size [49  49] sum=836.197
```

The object axes → aliases.

```
>> getaxis(a, 0) → Signal
>> a{0} is the same
>> a{1}, a{2} not defined → indices
```

Also displays statistics and size.

Setting iData object content: MetaData

The object **properties** can be **accessed** just as a normal structure:

```
>> a.Tag
>> get(a, 'Tag')
```

Similarly, the object properties can be modified

```
>> a.Title = 'scary, hey ?';                    % same as set(a, 'Title', '..')
>> a.DisplayName = 'not so much';
```

Setting iData object content: Aliases

New properties are defined as **aliases**:

```
>> a.ScratchedRecord = 'Data.Signal(1,:)';   % first line of the Signal
>> a.WillNot = [1:10];                       % a new numerical content
```

The **alias definition** may use either a link to an other part of the data, or to an explicit numerical value. Existing aliases are modified.

```
>> set(a,'BuyIt','WillNot');    % BuyIt → WillNot → 1:10
>> a.BuyIt                      % same as get(a, 'WillNot')
     1     2     3     4     5     6     7     8     9    10
```

It is possible to link to an other alias, to any valid expression (possibly making use of the ***this*** symbol to refer to the object itself) or even a link to an other file content (including distant ones through *http:* and *ftp:*).

Warning: with iterative links (links on links), be caution when setting the Alias value, which traverses all aliases.

```
>> a.BuyIt = 1;    % Sets BuyIt → WillNot to 1.
```

Labels can be associated to Aliases (kind of comments):

```
>> label(a, 'ScratchedRecord', 'Monthy Python')
```

Setting iData object content: Signal, Error, Monitor

The ***Signal*** of an object is the object 'value'.

Usually, after importation (remember what you will see next), the *Signal* is set by the load method.

Defaults: largest numerical block (always defined).

The ***Error*** Alias holds the error bar on the *Signal*, and the ***Monitor*** holds the associated weight/counting time.

Defaults: *Error* is $\sqrt{Signal}$ and *Monitor* is 1.

```
>> a.Signal                % extract the Signal Alias (value)
>> get(a,'Signal')         % same as above
>> getalias(a, 'Signal')   % extract the Signal Alias (definition)
>> getaxis(a, 'Signal')    % extract the Signal/Monitor value
>> getaxis(a, 0)           % same as above (rank 0 is Signal/Monitor)
>> a{0}                    % same as above
```

Setting iData object content: Axes

Axes are also *Aliases*, and they are used for e.g. plotting and binary mathematical operators (use 2 arguments such as $a+b$).

Defaults: Signal indices (1:size)

To set the axes (defining automatically corresponding Aliases):

```
>> setaxis(a, 1, linspace(-3,3,49) );   % sets the 1st axis rank (columns, Y)
>> a{2} = linspace(-3,3,49);            % same, with 2nd rank (rows, X)
```

We could have used (defining manually the aliases):

```
>> setalias(a, 'X', linspace(-3,3,49) );
>> a{2} = 'X'; a{1} = 'X';              % the two axes are made equal
```

And then the labels

```
>> xlabel(a, 'This is X'); ylabel(a, 'This is Y');
```

Working with object arrays

Most iData methods support **object arrays**, that is treat objects iteratively.

```
>> a = iData([ ifitpath 'Data/Ag_3_a.edf' ]);
>> b = [ a a' ]; % array of objects
b =  array [1  2] iData object:

Index     [Tag] [Dimension]                                    [Title] [Last command
    1  id542041     [55 71]                      'Ag K ; "Data Signal"' id542041=load
    2  id542042     [71 55]                      'Ag K ; "Data Signal"' id542042=ctra
```

```
>> std(b,1)     % Gaussian width along 1st axis
    6.6155   13.8538
```

```
>> a = iData([ ifitpath 'Data/ILL_*' ]);   % import all files starting with 'ILL'
a =  array [1  4] iData object:

Index     [Tag] [Dimension]                                        [Title] [Last command]
    1  id542048   [30 1024] 'File ILL_D10.dat ILL Data (... "Data ..."' id542048=load(iData
    2  id542049      [57 1] 'align for IN20;File ILL_IN2... "Data ..."' id542049=load(iData
    3  id542051  [340 1024] 'File ILL_IN6.dat ILL Data (... "Data ..."' id542051=load(iData
    4  id542052  [340 1024] 'File ILL_IN6_2.dat ILL Data... "Data ..."' id542052=load(iData
```