


Image Cover Sheet

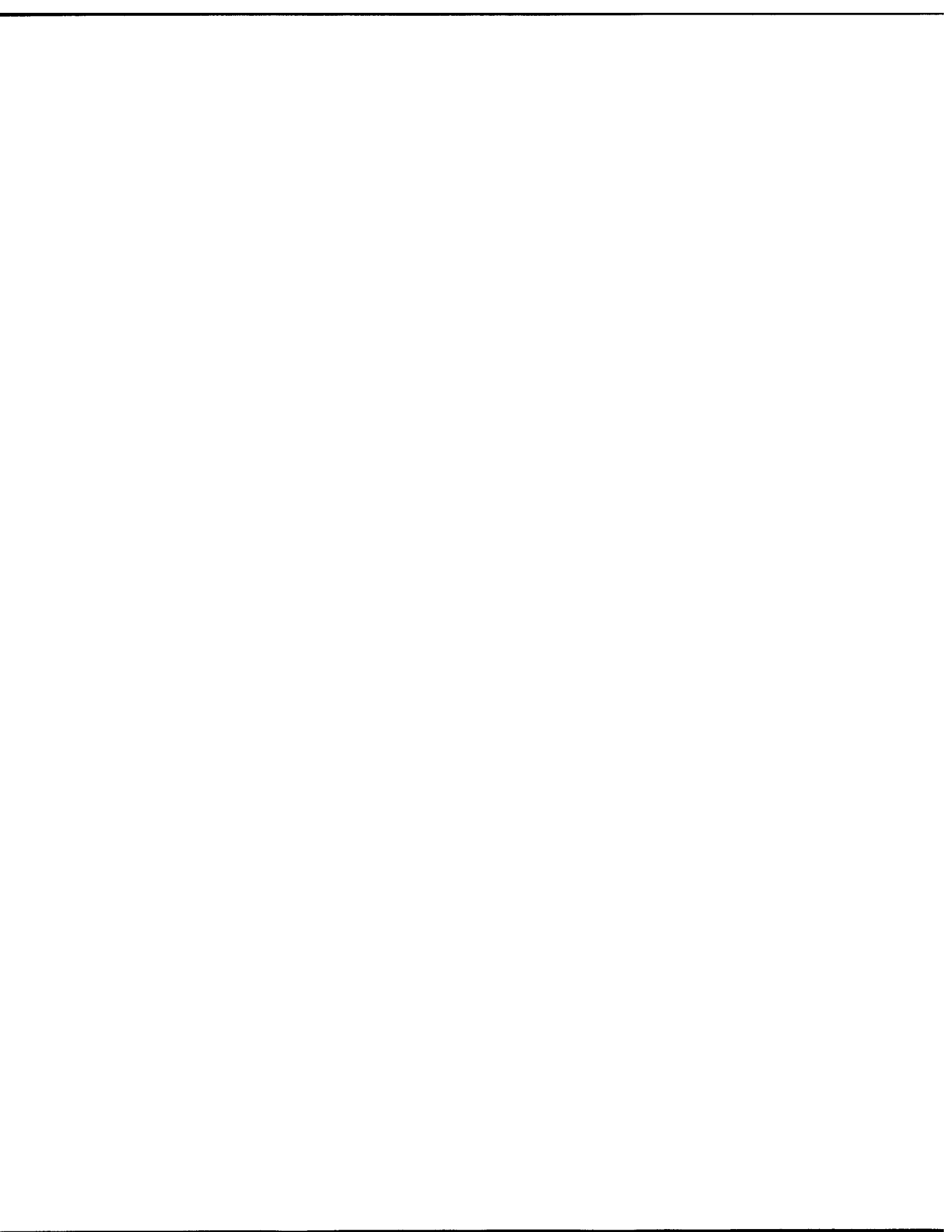
| | |
|---|---|
| CLASSIFICATION UNCLASSIFIED | SYSTEM NUMBER 513094  |
|---|---|

TITLE
An R-Based System for Managing and Analysing Centrifuge Study Data

System Number:
Patron Number:
Requester:

Notes:

DSIS Use only:
Deliver to:



An R-based System for Managing and Analysing Centrifuge Study Data

by

Marc Grushcow

NTT SYSTEMS INC.*
56 Dunblaine Avenue
Toronto, Ontario M5M 2S1

Project Manager:
Mr. William D. Fraser
(416) 635-2064

PWGSC Contract No. W7711-9-7565

on behalf of
DEPARTMENT OF NATIONAL DEFENCE

as represented by
Defence and Civil Institute of Environmental Medicine+
1133 Sheppard Avenue West
Toronto, Ontario, Canada
M3M 3B9

DCIEM Scientific Authority:
Mr. William D. Fraser
(416) 635-2064

10 January 2000

© HER MAJESTY THE QUEEN IN RIGHT OF CANADA (1998)
as represented by the Minister of National Defence

© SA MAJESTE LA REINE EN DROIT DU CANADA (1998)
Défense nationale Canada

RESTRICTIVE LEGEND

GOVERNMENT
OF CANADA

GOUVERNEMENT
DU CANADA

THIS DOCUMENT AND THE INFORMATION IT CONTAINS
IS DISCLOSED IN CONFIDENCE TO:

FOR DEFENCE INFORMATION PURPOSES ONLY, WITH NO
DISCLOSURES TO OTHERS OR ANY OTHER USE (INCLUDING
ANY MANUFACTURING USE) TO BE MADE WITHOUT THE
WRITTEN PERMISSION OF THE CANADIAN DEPARTMENT OF
NATIONAL DEFENCE (via DRDIM 3).

THIS DOCUMENT CONTAINS PROPRIETARY INFORMATION.

1 Introduction

This document describes a system designed to assist the user in managing, analysing and modeling data collected during DCIEM centrifuge studies.

Centrifuge data is digitized in real time and later packed into *jar* files along with an additional file, *dataset.properties*, that maps names to channels and data streams. One *jar* file is created each time a subject is inserted into the centrifuge. There are multiple subjects in each study, and each subject may be inserted multiple times under different or repeated conditions.

During an insertion, a subject will be exposed to a number of *G profiles*. Normally all subjects are exposed to the same profiles in the same order during each insertion, although the time between profiles may be adjusted depending on the subject's performance.

The *cstat* software is designed to let the user:

1. Add insertions (*jars*) to a study.
2. Define names for the *G profiles*.
3. Identify the start and end points for profiles within each insertion.
4. Plot digitized data in a variety of ways using insertion and profile names as identifiers.
5. Display (print) various views of the assembled data.
6. Remove profiles and insertions from the study.
7. Extract data from the *jar* files by specifying channel, profile, and insertion identifiers.

All of these operations let the user work with the information symbolically, without reference to specific channel numbers, file names and start and end times.

2 Installation and Removal

cstat was developed as a standard R package. As such it can be automatically installed and removed by R. Documentation is provided in Rdconv format and is compiled into HTML, L^AT_EX, and man format during installation.

In order to install the package, you must have write access to the R home directory. This is typically `/usr/lib/R`. Insert and/or mount the distribution medium and type

R INSTALL *path/cstat*

eg., R INSTALL /mnt/floppy/cstat.

The package can be removed by typing R REMOVE cstat. Again, you must have read/write access to the R home directory. You should remove an existing cstat package before installing a new version.

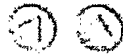
3 Using cstat

Once the package has been installed, you must attach it to your R session when you start. This is done by typing `library('cstat')` at the R prompt. You can then use all the cstat functions and will have access to the documentation.

4 Documentation

The remainder of this document consists of the detailed cstat documentation, printed from their HTML format. The page, `cstat(cstat)`, provides an overview of how to use the cstat functions to amass, display, and extract cstat data.

Centrifuge Statistical Analysis Support Package



| | |
|-------------------------------|--|
| add(cstat) | Add a new insertion to a cstat object |
| addp(cstat) | Add a new profile to a cstat object |
| compatibility(cstat) | Encapsulated class description |
| cstat | Description of the cstat object |
| cstat object variable name | Encapsulated class description |
| current insertion(cstat) | Description of 'current' insertion |
| encapsulated(cstat) | Encapsulated class description |
| getd(cstat) | Get sampled data |
| icreate(cstat) | Create a new cstat object |
| ident(cstat) | Identifies Start and End points for profiles |
| insertion(cstat) | Description of the insertion object |
| plotd(cstat) | Plot cstat data |
| print(cstat) | Print contents of cstat object |
| profile(cstat) | Description of the profile object |
| ProfileTemplate(cstat) | Description of ProfileTemplate object |
| property(cstat) | Description of the property object |
| rm(cstat) | Removes insertion from cstat object |
| rmp(cstat) | Removes profile from cstat object |
| select(cstat) | Makes the specified insertion object 'current' |

Description of the cstat object

Description

cstat is a package designed to support the analysis of data collected from DCIEM centrifuge studies.

Details

The system is predicated on the following assumptions:

The data is stored in **Datatrak** formatted *jar* files. See references for details.

Each *jar* contains a 'dataset.properties' file.

The dataset.properties file identifies a TAG with a value "Gz".

The associated file contains sampled G data for the run.

Other than the above, *cstat* is not sensitive to what physiological data was collected, what channels were used for which signals, or what sampling rates were used. *Cstat* allows users to define the G profiles of interest for a study, add *insertions*, identify relevant time periods in each insertion, and get and plot data for selected channels. The user manipulates all of the data through names that they themselves create and the system manages the mapping of these onto the sampled data.

Using cstat

First create a *cstat object* with the `icreate` (`icreate(cstat)`) function. In doing this you will name the study, and optionally specify names for G profiles and the *decimation factor*. The decimation factor specifies how to subsample the Gz data for display purposes. A *cstat* object is created using standard assignment syntax, eg., `x <- icreate("study", "G5 G7", 100)`. You then perform operations on the object using syntax like `x$select("george")`.

Once this has been done G profiles can be added (`addp(cstat)`) or removed (`rmp(cstat)`) from the study. A profile consists of a name for some part of an insertion, for example "ramp", "G5", "G7", etc. These will be used to identify sections of the record that will be collected for analysis. Profiles can be added and removed at any time.

Each subject is represented by an *insertion*. This is the data that was collected while the subject was in the centrifuge. Insertion data are stored in a *jar* file along with information that describes the contents of the *jar*. Insertions, and thus subjects, are added to the study with the `add` function (`add(cstat)`).

After an insertion has been added, you must identify where the G profiles are in the record. That is, when each profile happened for the particular subject. First, select which insertion you want to deal with (`select(cstat)`). Then identify the profiles using `ident(cstat)`.

At this point, you will have created profiles (eg., ramp, G5, G7), insertions (eg., george, sam, mary), and specified when each profile happened for each subject. You can now plot or retrieve data, for example, the SO2 data for george during the G5 profile. This is done using `plotd(cstat)` and `getd(cstat)`.

Various views of the `cstat` object can be displayed using the `print` method (`print(cstat)`)

Implementation Notes

`cstat` is implemented as a single function that encapsulates both data and methods. Internals are described in `encapsulated(cstat)`. Note that `cstat` relies on R scoping rules in order to implement this data hiding. It will not function properly under S environments.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

References

Gee, Tom et al, *IDEA Datatrac: A Proposed Data Archival System*, Rev 0.2, DCIEM

See Also

`icreate(cstat)`

Encapsulated class description

Description

A `cstat` object contains both the data and methods needed to identify, extract and display centrifuge study data.

Details

`Cstat` objects are created using the `icreate` constructor, eg., `x <- icreate(...)`. Once this has been done, data is accessed and modified through public methods, using syntax like `x$addp("G9")`.

The object itself is a `list` whose elements are data and functions.

`Cstat` objects can be created and stored in any legal R variable. Data is accessed and operated on only through public methods using a syntax like `x$getd()`.

Value

("\$" denotes a method)

| | |
|---------------------------|--|
| <code>name</code> | string variable, contains the name of the study |
| <code>CurInsert</code> | string variable, contains the name of the <i>selected</i> insertion (<code>insertion(cstat)</code>) |
| <code>DcmFct</code> | numeric variable, contains the default decimation factor for this <code>cstat</code> object |
| <code>insertions</code> | The collection of insertion objects (<code>insertion(cstat)</code>) |
| <code>Prof. Templ.</code> | The profile template for all insertions in the object (<code>ProfileTemplate(cstat)</code> <code>profile(cstat)</code>) |
| <code>\$print</code> | Display various parts of the object (<code>print(cstat)</code>) |
| <code>\$add</code> | Add an insertion to the object (<code>add(cstat)</code>) |
| <code>\$rm</code> | Remove an insertion (<code>rm(cstat)</code>) |
| <code>\$select</code> | Select the insertion to use for future operations (<code>select(cstat)</code>) |
| <code>\$ident</code> | Identify profiles for the selected insertion (<code>ident(cstat)</code>) |
| <code>\$addp</code> | Add a profile name to the profile template (<code>addp(cstat)</code>) |
| <code>\$rmp</code> | Remove a profile name from the profile template and all current insertions (<code>rmp(cstat)</code>) |
| <code>\$getd</code> | Get the sampled data for a specific insertion and profile (<code>getd(cstat)</code>) |
| <code>\$plotd</code> | Plot the data for a specific insertion and profile (<code>plotd(cstat)</code>) |

Compatibility

This package definitely works only under R since it relies on R's scoping rules.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

icreate(cstat)

[Package Contents]

Add a new profile to a cstat object

Description

This function adds a new `profile(cstat)` object to the `ProfileTemplate(cstat)` object and to the profile collection of all `insertion(cstat)` objects in `cstat` object.

Usage

```
x$addp(profilename)
```

Arguments

`x` `cstat` object variable name
`profilename` A string naming the new profile object.

Details

Each new profile object has a `null` `start` and `end` time. These must be specified using `ident(cstat)` for each `insertion` in turn.

Value

none.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

`rmp(cstat);print(cstat)`

Examples

```
x$addp("Ramp")
```

Add a new insertion to a cstat object

Description

This function creates a new `insertion(cstat)` object from the specified `jarfile` and adds it to the named `cstat` object.

Usage

```
x$add(name, jarfile, decimationFactor=collectionDefaultdecimationFactor)
```

Arguments

| | |
|--|--|
| <code>x</code> | cstat object variable name |
| <code>name</code> | A string that names the created insertion object. |
| <code>jarfile</code> | Name of the jar file containing the data. |
| <code>decimationFactor</code> | Level of decimation for GZ signal. |
| <code>collectionDefaultdecimationFactor</code> | This defaults to the decimation factor specified when the cstat object was created. |

Details

When instantiating an **insertion** object, this function creates and fills up following inner objects:

Decimatrix object - filled up from the GZ channel by taking one of each `decimationFactor`'s Samples.

Description object - filled up from the parsed `'dataset.properties'` file.

Profiles collection object - filled up from the `cstat`'s `Profiletemplate` object. **NOTE:** The profile template object contains only names for profiles; **start** and **end** points for each profile must be assigned using `ident(cstat)` function.

Value

none.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

References

None

See Also

`rm(cstat); addp(cstat); rmp(cstat); ident(cstat);`

Examples

```
x$add("SlNoSuit", "/export/aero/datatrac/ppslns.jar",100)
```

[Package Contents]

Encapsulated class description

Description

A `cstat` object contains both the data and methods needed to identify, extract and display centrifuge study data.

Details

`Cstat` objects are created using the `icreate` constructor, eg., `x <- icreate(...)`. Once this has been done, data is accessed and modified through public methods, using syntax like `x$addp("G9")`.

The object itself is a `list` whose elements are data and functions.

`Cstat` objects can be created and stored in any legal R variable. Data is accessed and operated on only through public methods using a syntax like `x$getd()`.

Value

("\$" denotes a method)

| | |
|---------------------------|--|
| <code>name</code> | string variable, contains the name of the study |
| <code>CurInsert</code> | string variable, contains the name of the <i>selected</i> insertion (<code>insertion(cstat)</code>) |
| <code>DcmFct</code> | numeric variable, contains the default decimation factor for this <code>cstat</code> object |
| <code>insertions</code> | The collection of insertion objects (<code>insertion(cstat)</code>) |
| <code>Prof. Templ.</code> | The profile template for all insertions in the object (<code>ProfileTemplate(cstat)</code> <code>profile(cstat)</code>) |
| <code>\$print</code> | Display various parts of the object (<code>print(cstat)</code>) |
| <code>\$add</code> | Add an insertion to the object (<code>add(cstat)</code>) |
| <code>\$rm</code> | Remove an insertion (<code>rm(cstat)</code>) |
| <code>\$select</code> | Select the insertion to use for future operations (<code>select(cstat)</code>) |
| <code>\$ident</code> | Identify profiles for the selected insertion (<code>ident(cstat)</code>) |
| <code>\$addp</code> | Add a profile name to the profile template (<code>addp(cstat)</code>) |
| <code>\$rmp</code> | Remove a profile name from the profile template and all current insertions (<code>rmp(cstat)</code>) |
| <code>\$getd</code> | Get the sampled data for a specific insertion and profile (<code>getd(cstat)</code>) |
| <code>\$plotd</code> | Plot the data for a specific insertion and profile (<code>plotd(cstat)</code>) |

Compatibility

This package definitely works only under R since it relies on R's scoping rules.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

[icreate\(cstat\)](#)

[\[Package Contents\]](#)

Description of 'current' insertion

Description

Some insertion(cstat) object in cstat object, with which we going to work with plotd(cstat) or getd(cstat) functions is called *current*. After creating, cstat object has no *current* insertion object. One of the added to cstat object insertion object becomes *current* after selecting it with select(cstat) or ident(cstat) functions. Name of the *current* insertion object can be viewed by print(cstat) or select(cstat) functions.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

References

None

See Also

select(cstat);ident(cstat)

Get sampled data

Description

This function returns a two column matrix of sampled data. The data is extracted directly from the jar file associated with the **current** insertion. The channel/file that is the source of the data is the one associated with the specified *tag*.

Usage

```
y <- x$getd(profilename, channelname)
```

Arguments

- `x` cstat object variable name
- `profilename` A string naming the desired profile object.
- `channelname` The name of the desired data channel to get data from. See `print(cstat)` function with `printall` parameter to see a list of all defined names.

Details

The current insertion is located and the named profile is read from its profiles collection. This in turn yields the *start* and *end* times. The channel name (eg., "BPT") is used to identify the location of the data inside the jar, and an external utility is used to extract the sampled data from *start* to *end*.

Value

2 column numeric matrix object with the sample data from specified channel within the specified by profile time range.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

`ident(cstat)`; `plotd(cstat)`; `print(cstat)`

Create a new cstat object

Description

This function creates an instance of `cstat` object. It will contain no `insertion(cstat)` objects in it and no current `insertion(cstat)` selection. Name member variable will be filled up by 'name' argument. `DcmFct` member variable will be filled up by 'decimationFactor' argument. `ProfileTemplate(cstat)` collection will be filled up according 'prof_descr_str' argument. If this argument is missing or empty, there are no `profile(cstat)` objects in `ProfileTemplate` collection, otherwise each white space separated substring in `prof_descr_str` string will create separate profile object, named by this substring. For example, string "Ramp G1 G2 G3" will create 4 profile objects, named "Ramp", "G1", "G2", "G3". Start and end points for all profile objects in `ProfileTemplate` collection remains undefined.

Usage

```
x <- icreate(name, prof_descr_str, decimationFactor)
```

Arguments

- `name` A string that names the object.
- `prof_descr_str` A string, that describes `ProfileTemplate` profile names.
- `decimationFactor` Default level of decimation for GZ signal.

Details

Created `cstat` object is an `encapsulated(cstat)` object, and `icreate` function is an constructor for this type of object. It give us an instance of such object as return value. See `cstat` object variable name for using it in R.

Value

An instance of created `cstat` object.

Compatibility note

`compatibility(cstat)`

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

References

None

See Also

`addp(cstat);add(cstat)`

[Package Contents]

Identifies Start and End points for profiles

Description

This is an interactive function that uses operator responses to set up or change Start and End points for some or all profiles of the 'current' (or specified) insertion.

Usage

```
x$ident(name)
```

Arguments

- x cstat object variable name
- name An optional string, that names the identified insertion object. If defined, the named insertion object becomes 'current' insertion. If missing, the previously defined 'current' insertion will be processed.

Details

When started, this function prints 'Select menu options from console, please' in the title area of the graphics window that displays the GZ graph and prompts for a menu choice in the R console. In any menu presented, entering 0 on menu prompt will bring you up one level. Entering 0 on first level menu will finish the function. See R(base) menu function for details. First level menu options are:

1:Plot whole time series

This option displays the complete GZ plot, covering the entire insertion.

2:Zoom in

The prompt "Set start point for zoomed region please" is shown in the title area of the graphics window. Respond by placing the mouse at the desired location and pressing the left button. The prompt, "Set end point for zoomed region please" is displayed in the title area. Move the mouse to the desired location and press the left mouse button again. The function will then return to the upper level menu and display the zoomed region only. The operation can be cancelled, usually by pressing the middle mouse button. If this is done, the plot region is not modified. Multiple zoom in operations can be performed.

3:Set Start and End points for profile

This option presents user with second level menu, as follows...

1:Process all profiles

The following is performed for all profiles: If start and end information is available, these points are marked with red vertical lines. You are prompted "Set start point for *profileName*, please." Set the start point as described above. You will then be prompted "Set the end point for *profileName*, please." Repeat the process. You can always interrupt the process, usually with the middle mouse button. It will continue with the next profile until none remain.

2:

Entering option 2 or above, will allow you to process the selected profile. Canceling the operation will bring you back to second level menu.

Value

none.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

addp(cstat); rmp(cstat); print(cstat); select(cstat); plotd(cstat); getd(cstat)

Description of the insertion object

Description

The *insertion* object contains information relating to data collected for one subject during one "ride" in the centrifuge. In general, the same individual can undergo several insertions. Data for each insertion is collected in real time and later packed into a jar file. The jar file is processed by *cstat* and is represented by the insertion object.

Value

The insertion object is a `list` containing the following...

- `name` A string containing the user assigned name for the insertion
- `JarName` A string containing the qualified name of the jar file. It is saved as entered by the user and may or may not be fully qualified. If not, it must locate the jar from the current working directory
- `Descr` A collection of `property(cstat)` objects. This collection represents the parsed contents of the 'dataset.properties' file extracted from the jar
- `Profiles` A collection of `profile(cstat)` objects. This is created as a copy of the profiles template but the *start* and *end* values are filled in when they are identified
- `Gz Data` A 2 column matrix object, containing decimated timestamp/data samples from GZ channel.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

`add(cstat);icreate(cstat)`

Plot cstat data

Description

This function makes a plot of one channel for current insertion(cstat) object. The data is extracted from the insertion's jar file using the start and end times specified in the named profile.

Usage

```
x$plotd(profilename, channelname, insname=CurIns, scr=screen(), ...)
```

Arguments

- `x` cstat object variable name
- `profilename` A string naming of the profile object. Multiple profile names, separated by space, could be specified here like "G1 G3 G2 Ramp". See *Details* for explanation of screen usage in this mode.
- `channelname` A string naming of the desired data channel to plot. See `print(cstat)` function with `printall` parameter to see a list of all defined names.
- `insname` A string naming of the insertion object. Defaults to current insertion.
- `scr` Screen number to plot to. Used together with `split.screen`
- `...` This represents any optional additional parameters, which will be passed to R plot function. For example, `xlim` and/or `ylim` parameters allow you to plot some subrange of profile data.

Details

The current insertion is located and the named profile(s) are read from its profiles collection. This in turn yields the start and end times. The channel name (eg., BPT) is used to identify the location of the data inside the jar, and an external utility is used to extract the sampled data from *start* to *end*. To make a multiplot graph use `split.screen`. To plot to a printer, use `postscript` to redirect output to a file that can then be printed. If one profile specified, the current screen will be used regardless of `split.screen` mode. If multiple profiles specified, the function will check whether or not the current graph device is in `split.screen` mode. If so, it will plot graphs sequentially on the screens in the same window, starting with the "current" screen and returning to the first when the last has been used. If the current graphics device is not in `split.screen` mode, the function will split it into screens (1xN), where N is the number of profiles and plot as specified. When the operation is complete, the graphics device will return to non-split. In `split.screen` mode, you can specify which screen should be considered "current" by specifying the `scr` parameter.

Value

none.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

getd(cstat); ident(cstat); print(cstat)

Examples

```
x$plotd("G2", "BPT") #plot BPT for current insertion for G2 profile
...
split.screen(c(2,3)) #2 rows, 3 columns
x$plotd("ramp G5 G7", "BPT","s1") #first row for insertion s1
x$plotd("ramp G5 G7", "BPT","s2") #second row for insertio s2
```

[Package Contents]

Print contents of cstat object

Description

This function prints contents of whole cstat object or contents of the insertion(cstat) object, specified by the 'name' argument.

Usage

```
x$print(name, printall=FALSE)
```

Arguments

- `x` cstat object variable name
- `name` A string that names the insertion(cstat) object. If missing, function will print contents of whole cstat object .
- `printall` If FALSE (default), prints the short form of contents, otherwise - long form.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

[icreate\(cstat\)](#)

Description of the profile object

Description

A profile object is used to name and describe one of the G profiles that comprise an insertion.

Value

The profile object is a `list` containing the following...

- `Name` A string, containing the name of the profile.
- `start` A numeric, containing the start time of the interval.
- `End` A numeric, contains the end time of the interval

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

`addp(cstat)`; `rmp(cstat)`; `icreate(cstat)`; `print(cstat)`; `ProfileTemplate(cstat)`

Description of ProfileTemplate object

Description

The ProfileTemplate is an collection of profile objects. It lists and names profiles in all insertion objects by acting as template for its profiles collection. Start and End points are undefined for all profiles in ProfileTemplate collection, hence all newly created insertion objects contains only names in its profile collection. All profiles in insertion object need to be identified by ident(cstat) function.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

References

None

See Also

[addp\(cstat\)](#); [rmp\(cstat\)](#); [icreate\(cstat\)](#); [print\(cstat\)](#)

Description of the property object

Description

One property object is created for each channel parsed out of a 'dataset.properties' file.

Value

The property object is a list containing the following...

Name A string containing the channel 'tag'.
Descr A string containing the channel 'name'.
FileName A string object, containing the channel 'data-file-name'.
Rate A numeric containing the channel 'sampling-rate'.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

[add\(cstat\);print\(cstat\)](#)

Removes insertion from cstat object

Description

This function removes the `insertion(cstat)` object, specified by the `'name'` argument from the insertion collection in the `cstat` object.

Usage

```
x$rm(name)
```

Arguments

`x` `cstat` object variable name

`name` A string that names the `insertion(cstat)` object that is to be removed.

Value

none.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

`add(cstat);``icreate(cstat)`

Removes insertion from cstat object

Description

This function removes the `insertion(cstat)` object, specified by the 'name' argument from the insertion collection in the `cstat` object.

Usage

```
x$rm(name)
```

Arguments

- `x` `cstat` object variable name
- `name` A string that names the `insertion(cstat)` object that is to be removed.

Value

none.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

`add(cstat);``icreate(cstat)`

Removes profile from cstat object

Description

This function removes the named `profile(cstat)` object from the `ProfileTemplate(cstat)` and from the all `insertion(cstat)` objects in its `cstat` object.

Usage

```
x$rmp(name)
```

Arguments

- `x` `cstat` object variable name
- `name` A string that names the profile that is to be removed.

Value

none.

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

`addp(cstat);``icreate(cstat);``print(cstat)`

Makes the specified insertion object 'current'

Description

This function makes the named `insertion(cstat)` object the current `insertion(cstat)` object. Certain functions, such as `ident()`, will use the *current* insertion as a default if no other name is supplied.

Usage

```
x$select(name)
```

Arguments

`x` `cstat` object variable name

`name` An optional string, that names the desired insertion object. If missing, `select` prints the name of *current* insertion object.

Value

none.

Compatibility note

`compatibility(cstat)`

Author(s)

Boris Leykekhman Boris.Leykekhman@NTT.ca

See Also

`ident(cstat)`; `print(cstat)`; `current insertion(cstat)`

#513094