ProviewR

OPEN SOURCE PROCESS CONTROL

RTT

User's Guide

99 02 18

Introduction

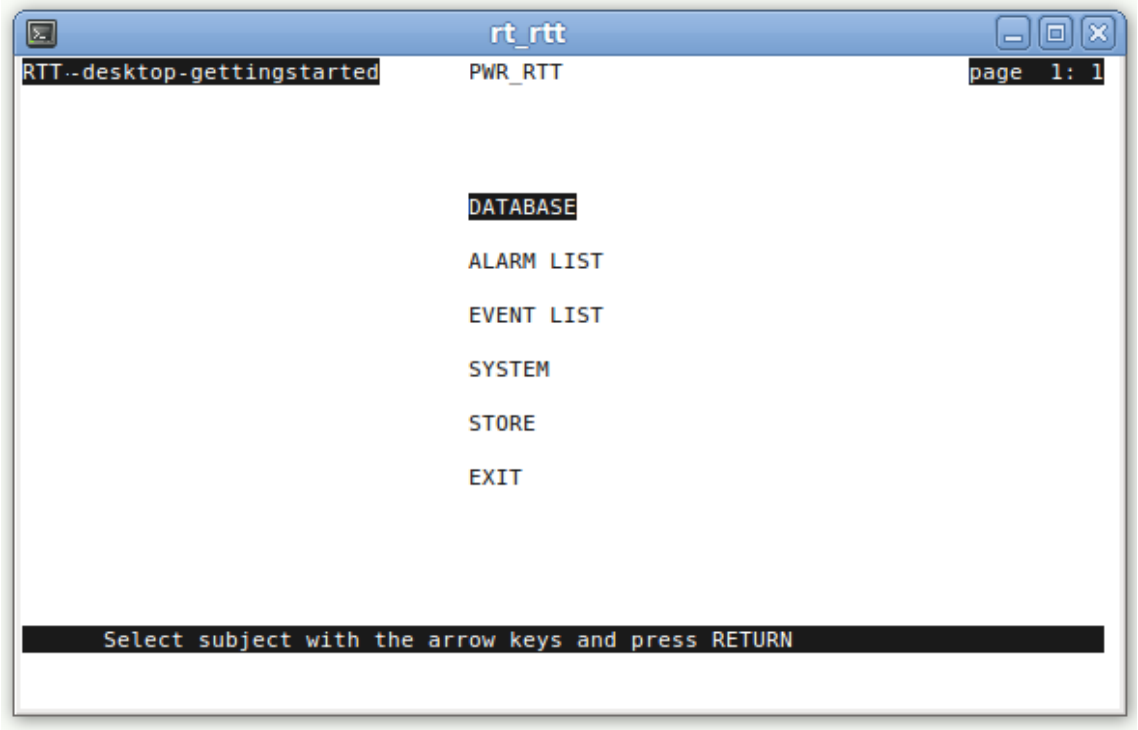
rt_rtt is a tool for troubleshooting a ProviewR system in a VT100 compatible terminal or terminal window. It can also be used to a very simple HMI.

rt_rtt contains

- a tool to navigate in the plant and node hierarchies of the database and show the values of the object attributes.
- a number of commands to show objects, crossreferences, logg attributes values, create/remove objects etc.
- possibility to execute scripts.
- a number of system pictures to show status for I/O devices, nethandler, grafcet sequences etc.
- an editor to build menues and pictures for operators and maintenance.
- A function keyboard for operator communication.

rt_rtt has in most situations been replaced by its X window inheritor rt_xtt, but it still has som advantage in viewing information about qcom and the pools. It is also very useful in embedded systems where X windows is not available, and at remote troubleshooting over a slow network connection.

Starting rt_rtt



rt_rtt is started with the command

```
$ rt_rtt
```

rt_rtt will login to the runtme environment with privileges fetched from
- username and password supplied as arguments to rt_xtt.
- if no arguments is supplied, with the privileges specified in the DefaultXttPriv argument of the Security object.
- if the attribute SetOpsysUser in the Security object is set, the username equals the current Linux user.
- otherwise a login dialog is viewed where username and password is specified.

Arguments

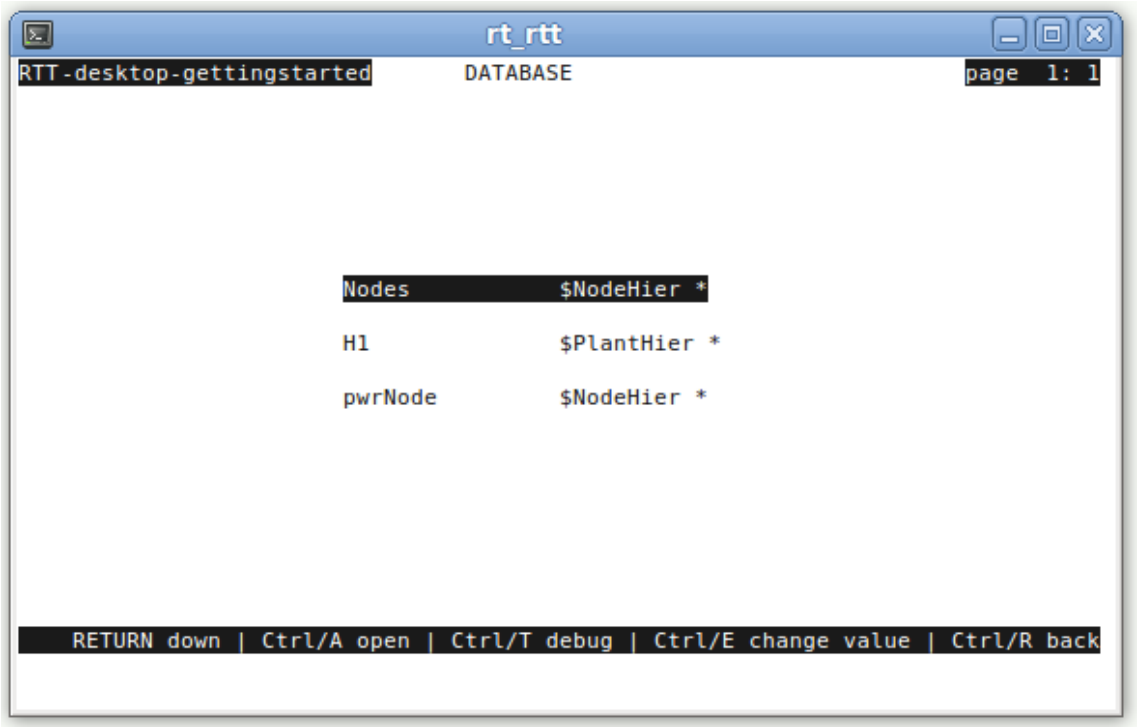
Arguments can be added to the start command.

Argument	***Description***
1	Username
2	Password
3	Rtt configuration object of class RttConfig. The default configuration object is named RttConfig and positioned below the node object.
4	A startup script file.
5	Main menu title.

Navigation

Key	***Function***
Arrow keys	Navigate in the menu and select an item.
Return	Execute a function or display a child menu if there is one.
Ctrl/A or F1	Execute some function if specified.
Ctrl/T or F2	Execute some function if specified.
Ctrl/E or F3	Set the value of the current item.
Ctrl/R or F4	Go back to previous menu.
Page Down or Ctrl/F	Go to next page of the menu.
Page Up or Ctrl/D	Go to previous page of the menu.
Ctrl/Z	Back to root menu.
Ctrl/W	Redraw the picture.
Ctrl/H	Help.
Delete	Delete an item in the menu.
Ctrl/B	Command mode.
Ctrl/K	Acknowledge last alarm.
Ctrl/L	Show or hide description attribute.

Database



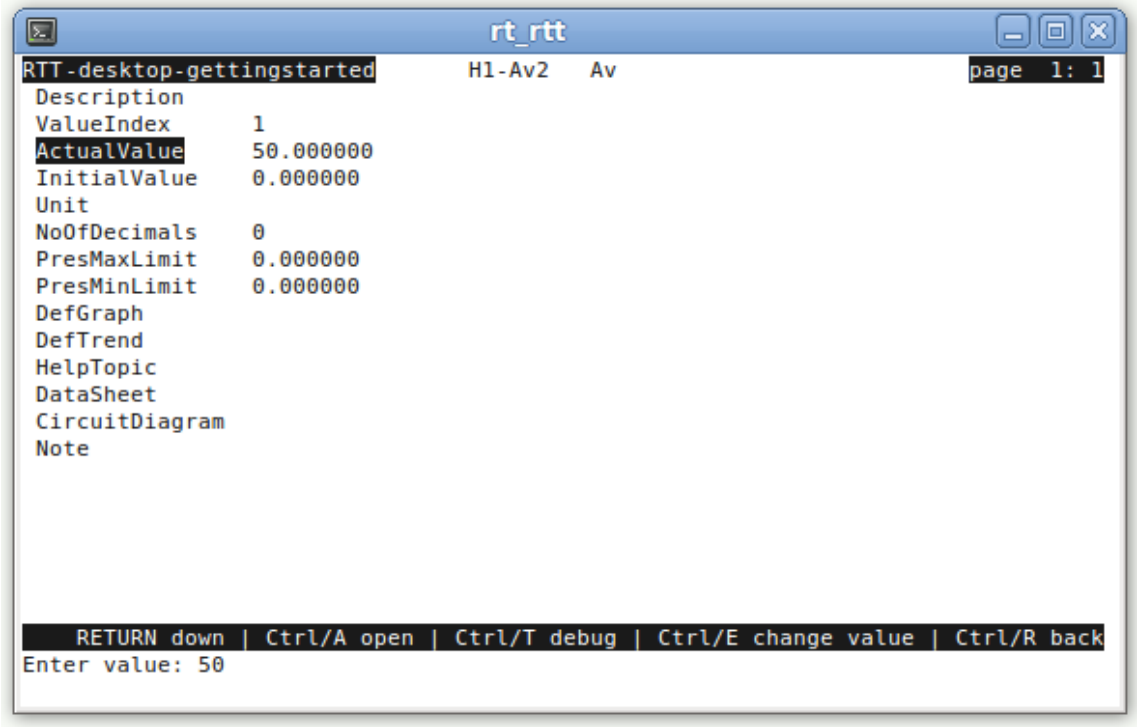
The database menu item shows the content of the ProviewR realtime database, i.e. the plant and node hierarchies. All objects in one level in the object tree is viewed in one picture, and the picture can be divided into several pages. An object is opened by selecting the object pressing Ctrl/A. This will show the attributes of the object and the attribute values.

If an object has children, this is marked with an asterisk '*'.

You navigate in the picture with the arrow keys, moves down in the object tree with the Return key and up with Ctrl/R.

Key	***Function***
Arrow keys	Show the attributes of the object. Navigate in the menu and select an object.
Return	Display the children in the object hierarchy of the object.
Ctrl/A or F1	Show the attributes of the object.
Ctrl/T or F2	Debug the children of the object.
Ctrl/E or F3	Set the value of an attribute.
Ctrl/R or F4	Go back to previous menu.
Page Down or Ctrl/F	Go to next page of the menu.
Page Up or Ctrl/D	Go to previous page of the menu.
Ctrl/Z	Back to root menu.
Ctrl/W	Redraw the picture.
Ctrl/H	Help.
Delete	Delete an item in the menu.
Ctrl/B	Command mode.
Ctrl/K	Acknowledge last alarm.
Ctrl/L	Show or hide description attribute.
Ctrl/V	Insert selected object into collection picture.
Ctrl/N	Show collection picture.

Show object



The content of an object is displayed by selecting the object and pressing Ctrl/A. The attributes and the attribute values are viewed.

The value of an attribute is changed with the function 'Change value' (F3).

Array attributes are marked with an asterisk '*'. These are opened with Ctrl/A to view the content.

Functions in the object view.

Key	**Function**
Arrow keys	Select an attribute.
Ctrl/A or F1	If the attribute value is a referens to another object, the content of the object is viewed.
Ctrl/E or F3	Change attribute value.
Ctrl/R or F4	Go back to previous menu.
Page Down or Ctrl/F	Go to next page.
Page Up or Ctrl/D	Go to previous page.
Ctrl/Z	Back to main menu.
Help	Help.
Ctrl/B	Command mode.
Ctrl/V	Insert selected attribute into collection picture.
Ctrl/N	Show collection picture.
Ctrl/K	Acknowledge last alarm.

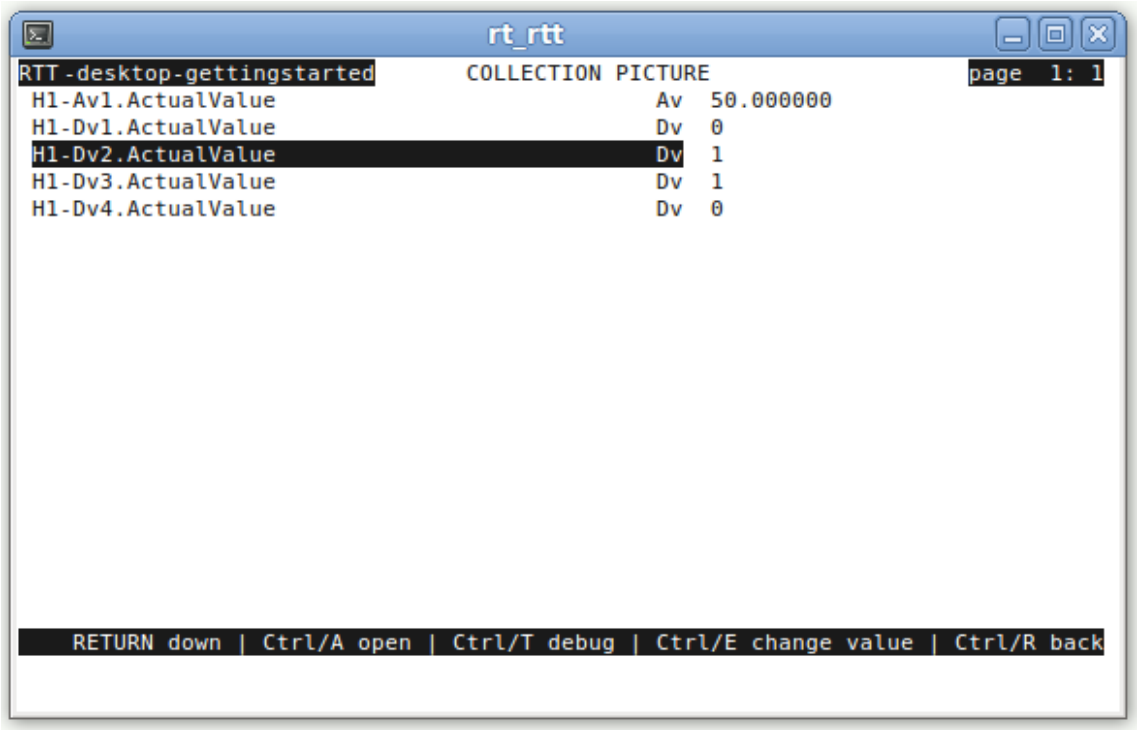
Debug picture

The debug picture shows the value of the debug attribute for all children of the selected object.

Functions in the debug picture.

Key	Function
Arrow keys	Select an object.
Return	Show the children of the selected object.
Ctrl/A or F1	Open the selected object.
Ctrl/E or F3	Change value of selected attribute.
Ctrl/R of F4	Go back to previous menu.
Page down or Ctrl/F	Go to next page.
Page up or Ctrl/D	Go to previous page.
Ctrl/Z	Back to main meny.
Help	Help
Delete	Remove a menu entry.
Ctrl/B	Command mode.
Ctrl/V	Insert selected attribute into collection picture.
Ctrl/N	Show collection picture.
Ctrl/K	Acknowledge last alarm.

Collection picture



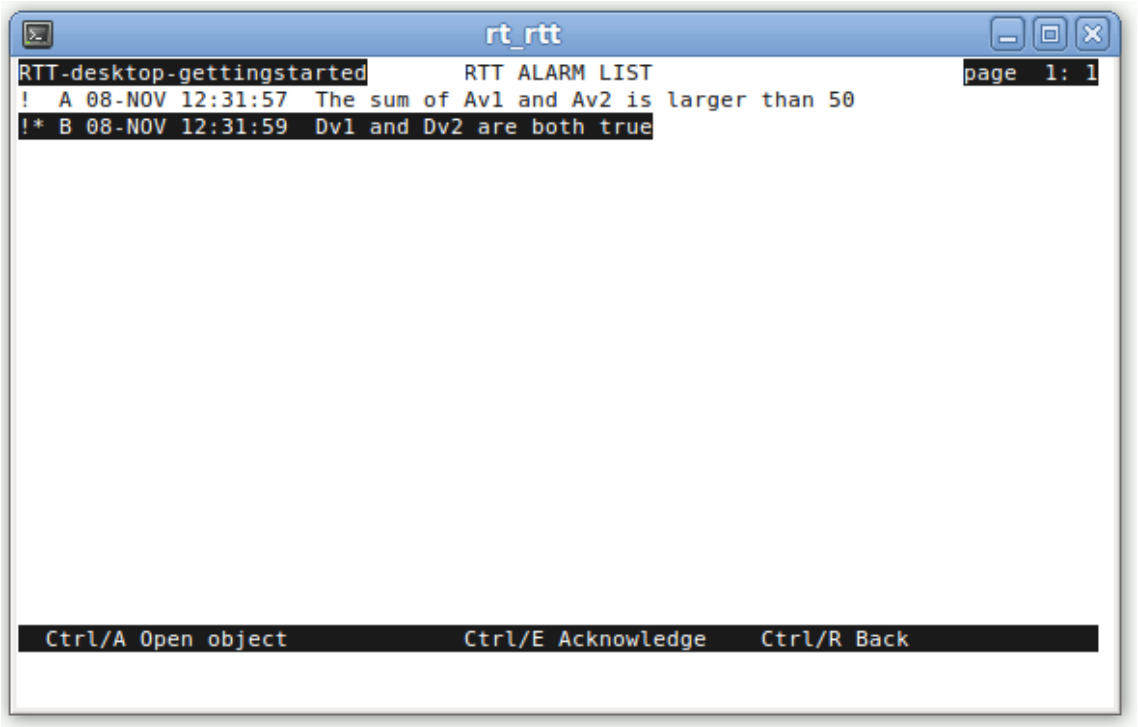
The collection picture makes it possible to collect a number of interesting attributes and display them in one picture. Insert in attribute by selecting the attribute in the object hierarchy and press Ctrl/V. If you select an object without specifying the attribute, the debug attriubte for the object is inserted.

The collection picture is viewed with Ctrl/N, or with the command 'show collection'.

The collection picture is emptied with the command 'collection clear'. Individual attributes is removed with the Delete key.

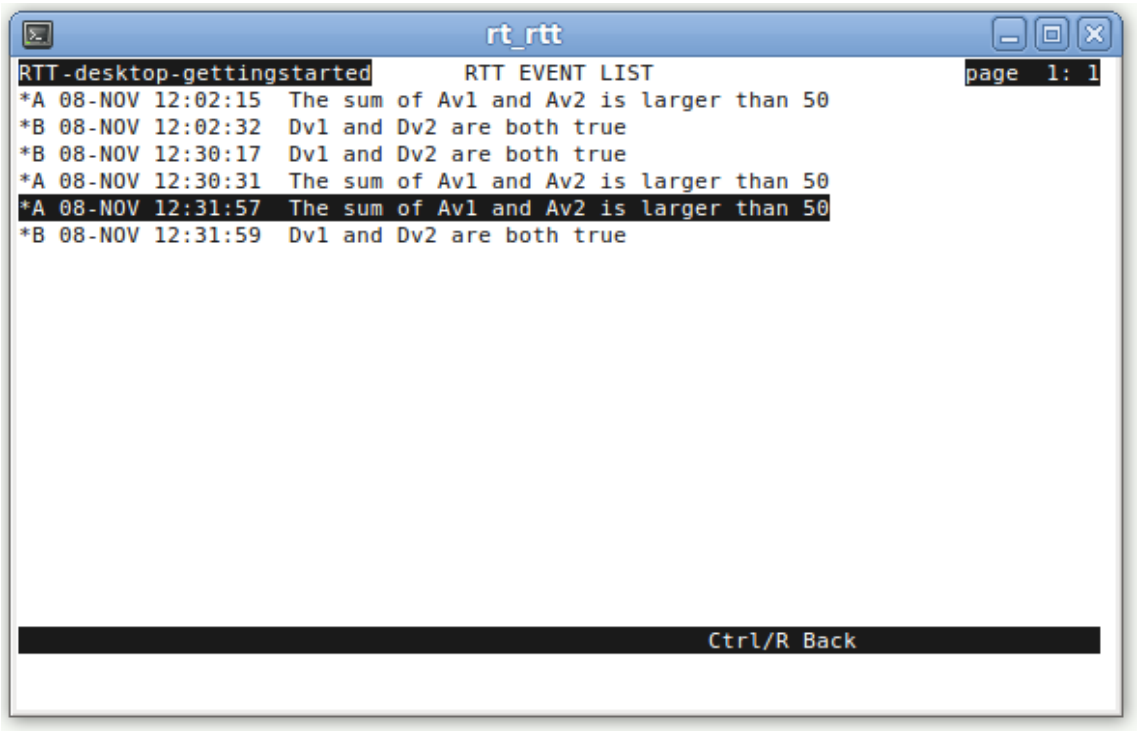
The functions in the collection picture is the same as in the debug picture.

Alamlist



In the alarm list prevailing an unacknowledged alarms are viewed. Unacknowledged alarms are marked with a '!' and prevailing alarm with a '*'. Only alarms from the plant parts stated in the EventSelectList of the RttConfig object are viewed. Also the alarm priority (A, B, C or D), the alarm time and alarm text is viewed.

Eventlist



The events from the plant parts stated in EventSelectList are viewed in the event list.
Alarm events are marked with the priority (*A, *B, C or D), info messaged with 'I', return events with 'r' and acknowledge events with 'a'. Also the event time and text is displayed.

System pictures

Rtt contains a number of pictures displaying information about the system. The pictures i found below the System item in the menu.

SHOW SYSTEM

Not implemented in Linux.

PLCPGM

A list of all plcpgm on the node is displayed.

GRAFCET

The grafcet picture shows the plcpgms containing grafcet sequences. If the grafcet sequence is active, i.e. the InitSteps are not active, this is indicated with the text 'Active'. Up to four sequences can be viewed simultaneously by marking plcpgms with the Return key. All the marked plcpgms are viewed in the grafcet monitor.

DEVICE

The device picture shows all I/O cards on the node, with errorcounters and addresses.

The channels of a card is viewed by selecting the card and pressing the Return key. The value of the channel, and possible test, inversion and conversion flags are also displayed for the channel. Furthermore the signal object of the channel is viewed. By selecting the CngMode field with the arrow keys, and pressing Ctrl/A the channel description, or channel identity is showed instead of the signal object.

The channel picture for Do and Ao channels, can be set in a special signal-test mode, by setting the SignalTestModeOn attribute in the setup menu (command 'setup'). This mode makes it possible to force the value of analog and digital outputs. The test flag for a Do or Ao channel is set with Ctrl/T, and the test value is changed with Ctrl/A (Do) or Ctrl/E (Ao). The SignalTestModeOn has to be set before the channel picture is opened.

PLCTHREAD

PlcThread shows info for the plc treads of the node, priority, loop count, scantime etc.

PID

PID shows all PID controllers on the node. For each controller the object name, force (On/Off), mode (Casc/Auto/Man), processvalue, setvalue and output value is viewed.

LOGGNING

From the logging picture it is possible to log attribute values to a file. The logging function contains ten entries, each entry handles one log file.

Here is how to proceed to start a logging:

- Collect the attributes that are to be logged into the collection picture.
- Enter the logging picture, and select a free entry with PageUp/PageDown.
- Set scan time for the logging by selecting 'Time' with the arrow keys and change the value with Ctrl/E. The time is in ms.
- The filename can be changed, if another name than the default name is desired.
- Set type of logging in 'Type'. Type can be 'Cont' or 'Event'. 'Cont' will log the values continously with the specified scantime. 'Event' will log a value when it is changed.
- Select 'Insert' and press Ctrl/A to insert the attributes into the collection picture.
- Select 'Start' and press Ctrl/A to start the logging. The text 'ACTIVE' is flashing when the logging is started.
- Stopp the logging by selecting 'Stop' and press Ctrl/A.
- Examine the loggfile by selecting 'Show File' and pressing Ctrl/A.

For more information, see the Logging chapter and the logging command.

NETHANDLER

The nethandler menu contains items for pictures about the nethandler.

SHOW NODES

The picture 'SHOW NODES' shows communication links to other nodes.

These parameters are viewed in the picture:

Parameter	**Description**
Name	Nodenamn.
Os	Operating system.
Hw	Hardware platform.
Link	The link can be Up, Active, Connected, Down or Local
Time up	Time when the link was established.
Sent	Number of sent messages.
Rcvd	Number of received messages.

SHOW SUBCLIENT

Shows the subscription client, i.e. prenumerations the current node has requested from other nodes. Note the number of unknown subscriptions, if this is larger than zero when all nodes are present in the system, it indicates subscriptions to nonexisting objects.

SHOW SUBSERVER

Shows the subscription server, i.e. subscriptions other nodes have requested from the current node.

QCOM

Shows Qcom information.

QCOM NODES

Show Qcom links to other nodes.
By selecting a node and pressing Return, more info about this node is showed, e.g. the round trip time and acknowledgement timeout indicating network problems.

QCOM APPLICATIONS

Shows a list of server and application processes attached to Qcom.
By selecting an application and pressing Return, a list of the Qcom queues of the application is showed.

SHOW POOL

Shows the system pools. By selecting a pool and pressing Return, the segments of the pool is viewed.

Store

Under STORE in the menu, stored collection pictures and script files are displayed.

To store a collection picture:

- Collect a number of attributes into the collection picture.
- Enter the collection picture and write the command (Ctrl/B) pwr_rtt> store/collect sto1
- The collection picture is now stored under STORE / STO1. Enter the Store menu, select STO1 and press Return to restore it.

Logging

The logging utility makes it possible to log attribute values to a file.
Up to ten logfiles can be opened simultaneously and in every logfile 100 attributes can be logged. The initialisation of the logging is made by setting data to a log table. Each logfile is handled by an entry in the log table. An entry is handled from the LOGGING picture, and with PageUp and PageDown keys you shift between entries.

The log entries can also be handled by the commands 'logging create', 'logging set' and 'logging delete'.

A condition attribute can be specified for a logging entry. In this case the logging is only performed when the condition attribute is true.

There are two types of logging, either the attribute values are continously logged with a specified cycle time, or an attribute is logged everytime the attribute value is changed.

Continous logging

The value of the attributes in the entry are logged every cycle. The value and the time since start is printed into the file. The file can be read by Excel for graphic presentation.

Event logging

An attribute in the entry is logged if the value of the attribute is changed.
The time of the change and the new value is printed into the logfile.

Commandfiles in rtt

Rtt contains a command language that is used for example to store collection pictures and logging entries. You can modify these files in a text editor, or write your own command procedures.

Create a command file

A command file consists of comment lines and command lines. You can also define symbols and perform simpel arithmetics. Conditional execution and jumps are also possible, see chapter Script. Here is described how to create or modify collection and debug pictures.

Comment lines beginns with exclamation mark in first position.

Commands to handle a collection picture:

- collect clear
- collect /name='objectname'
- show collect
- add debug /name='objectname2' [/class= /hierarchy=]
- add parameter/parameter='attribute'/name='objectname3' [/class= /hierarchy=]

Commands to build a debug picture:

- debug /name='objectname1' [/class= /hierarchy=]
- show parameter/name=...[/class= /hierarchy=]
- add debug /name='objectname2' [/class= /hierarchy=]
- add parameter/parameter='attribute'/name='objectname3' [/class= /hierarchy=]

A debug picture is created with the commands 'debug object' or 'show parameter' and is enlarged wih the commands 'add debug' and 'add parameter'.

Execute a command file

A command file is executed by a '@' sign followed by the filename.

Rtt configuration

Configuration object

The configuration object is of class RttConfig. The default object is named RttConfig and placed under the node object. If another configuration object should be used, this is specified at rtt start in the the fourth argument.

See Object Referens Manual for information about the RttConfig object.

Alarm list configuration

To see the alarm and eventlists in rtt, the EventSelectList in the RttConfig object has to be filled in.

In RttConfig object you also set suitable data of the attributes AlarmAutoLoad, AlarmMessage, AlarmBeep, AlarmReturn and AlarmAck.

Interpretation of transbuffers

In the remtrans picture you can see the communication buffers for the remtrans interpreted as a c struct. This requires that the struct name and the file where the struct is defined is specified in the RemTrans object (attributes StructName and StructFile). See command 'show object /type' for more info.

Create menues

Menues can be added to the rtt menu with the commands 'create menu' and 'add menu'. With these commands you can for example build menues that shows specific objects in the database. The script below shows plates in the length interval 10 – 15 m.

```
string name;
string attr;
float length;
int  first

name = GetChild("VKV-Plates");
first = 1;
while ( name != "")
  if ( GetObjectClass(name) == "END_Plat")
    attr = nname + "Length";
    length = GetAttribute( attr);
    if ( length >= 10 && length < 15)
      if ( first)
        create menu/title="Plates 10-15 m"/text="'name'"/obj="'name'"
      else
        add menu/text="'name'"/object="'name'"
      endif
    endif
    first = 0;
  endif
  name = GetNextSibling(name);
endwhile
```

Crossreferences

Rtt can show crossreferences for signalas and other objects. It is also possible to search for references in arithm objects containg code.

Crossreferences are showed with the command 'crossreference'.

The crossreference information in not present at runtime, it has to be generated in the development environment. This is done in the Configurator with the command 'create cross'. The reference information is store in three files on $pwrp_load, rtt_crr_'volumeid'.dat, rtt_crro_'volumeid'.dat and rtt_crrc_'volumeid'.dat. These are distributed by the distributor to the process station.

Arithm-kod

The objects CArithm, DataArithm, AArithm and DArithm contains c-code where for example c-functions can be called. With the crossreference command you can search for arithm objects that contain calls to a specific function. This is done with the command

```
pwr_rtt> crossref /function=
```

The result is a list of the arithm objects where the function is called, also the the rows in the code where the call is made are displayed.

You can also search for an arbitrary string in the arithm-code with

```
pwr_rtt> crossref /string=
```

Commands

The command input field is opened with Ctrl+B.

add

This command adds objects to the current picture.
Allowed qualifiers are:

```
pwr_rtt> add debug /name= /class= / hierarchy=
pwr_rtt> add parameter/parameter= /name= /class= /hierarchy=
```

add parameter

This command adds objects to the current picture, and displays the value of the parameter of the objects.
Object that fits in the class, name and hierarchy description are displayed on the screen.

```
pwr_rtt> add parameter /parameter= /name= /class= /hierarchy=
```

/parameter=	the name of the parameter to be displayed
/name=	the name of the object. Wildcard is allowed. The name can be written without '/name='. Ex show object *di*
/class=	display objects of this class
/hierarchy=	display objects below this object in the hierarchy. If no object is given the hierarchy of the current picture is default (show object /hierarchy).

add debug

This command adds objects to the current picture, and displays the value of the debug parameters of the objects.
Object that matches in the class, name and hierarchy description are displayed on the screen. Only objects with a defined debugparameter are displayed.

```
pwr_rtt> debug object /name= /class= /hierarchy= /global
```

/name=	the name of the object. Wildcard is allowed. The name can be written without '/name='. Ex show object *di*
/class=	displays objects of this class
/hierarchy=	displays objects below this object in the hierarchy. If no object is given the hierarchy of the current picture is default (show object /hierarchy).
/global	Search is also perfomed in mounted volumes of remote nodes.

add menu

Add a menu item to a menu.
A limited selection of meny items types is supported.
Syntax:

```
pwr_rtt> add menu /text= [/object=] [/command=]
```

/text	Text of the menu item
/object	Name of object an object. The menuitem will have the characteristics of an object. pf1 will open the object, return will show the children and pf2 will debug the children.
/command	The menu item is of type rtt command. The rtt-command that will be executed if pf1 is activated should be added.

Example

Create a menu item of type menu

```
pwr_rtt> add menu/text=”SubMenu”
```

Create a menu item of type object

```
pwr_rtt> add menu/text=”SomeObj”/obj=Motor-Enable
```

Create a menu item of type command

```
pwr_rtt> add menu/text=”SomeCmd”/command=@mycmd
```

alarm

Send an alarm, show the alamlist, show or print the eventlist.

alarm send

Send an alarm with the specified alarm text and alarm priority.

```
pwr_rtt> alarm send /text="..." /priority=
```

`/priority` can be A,B,C or D (default A).

alarm print

Print the eventlist on a file.

```
pwr_rtt> alarm print/file= [/notext] [/noname]
```

The alarmlist has to be loaded before alarm print. The alarmlist will be loaded when the alarm or eventlist is entered.

alarm show

Show the alarmlist.

```
pwr_rtt> alarm show [/user=][/acknowledge][/return][/bell][/maxalarm=]
           [/maxevent=]
```

alarm list

Show eventlist.

```
pwr_rtt> alarm list [/user=][/acknowledge][/return][/bell][/maxalarm=]
                      [/maxevent=]\
```

class

Show the class definition.

/name
A class name.

/volume
Shows all classes in the specified class volume.

collect

Handles the collection picture.

collect

Add the selected object to the collection picture.
If the object is collected from a hierarchy menu, the debugparameter will be displayed, if a parameter is collected in a 'show object' menu the selected parameter will be displayed.

```
pwr_rtt> collect /name=
```

/name	The object name kan include the parametername, otherwise the debug parameter is collected.

collect clear

Clear the collection picture. The collection picture has to be active.

```
pwr_rtt> clollect clear
```

collect show

Show the collection picture, i.e. attributes collected by the collect function (accelerator Ctrl/N).

```
pwr_rtt> collect show
```

create

create object

Create an object.
Objects can only be created in dynamic volumes.

```
pwr_rtt> create object /name= /class=
```

/name	name of the object including hierarchy
/class	class of the object

Example

```
pwr_rtt> create object /name=Noder-Noden-RttUser /class=User
```

create menu

Create a new menu with one menu item. More items kan be added with the command 'add menu'. A limited selection of meny items types is supported.

```
pwr_rtt> create menu /title= /text= [/object=] [/command=]
```

/title	Title of the menu
/text	Text of the menu item
/object	Name of object an object. The menuitem will have the characteristics of an object. Ctrl/A will open the object, Return will show the children and Ctrl/T will debug the children.
/command	The menu item is of type rtt command. The rtt-command that will be executed if Ctrl/A is activated should be added.

Example

```
  pwr_rtt> create menu /title=\"My menu\" /text=\"More menues\"
  pwr_rtt> create menu /title=\"Some objects\" /text=\"Obj1\"
           /object=rt-obj1
  pwr_rtt> create menu /title=\"Cmd menu\" /text=\"cmd1\"
           /command=@cmd1
```

crossreference

Crossreference shows where a signal or object is referred in the plc code by objects as GetDp, StoDp, GetData etc. The crossreference function can also search for strings, e.g. functions and classes, in the code of Arithm objects. The crossreference information has to be generated in the development environment, by the command 'create cross' or by setting the option Crossreferens in the configurator setup.

```
pwr_rtt> crossreference
```

`/file` Name of the crossreference file. Optional.

`/name` Signal or object to show crossreferences for. Wildcard is allowed. Default the selected object.

`/function` Name of a function, class or attribute that is used in the code in AArithm, DArithm, CArithm or DataArithm. The difference between /function and /string is that /string allows all characters as delimiters, when /function only allows digits or letters.

```
pwr_rtt> crossreference /function= [/brief]
           [/case_sensitive]
```

`/brief` Used at search in Arithm objects. /brief only shows one row of code or each arithm object.

`/case_sensitive` Used at search in Arithm objects. Specifies that the search is case sensitive. When case sensitive is used, the function or search string should be enclosed by quotes.

Example

```
pwr_rtt> cross /function=”MyFunc” /case_sensitive
```

Search for a string in AArithm, DArithm, CArithm or DataArithm code.

```
pwr_rtt> crossreference /string= [/brief] [/case_sensitive]
```

`/string` Search for a string in Arithm object code.

debug

Display the value of the debug parameter of one or many objects.

```
pwr_rtt> debug object /name= /class= /hierarchy =
pwr_rtt> debug children /name=
pwr_rtt> debug signals /name=
```

debug signals

Display the signals and the value of the signals referenced in a plcpgm
or a window.
Allowed qualifiers are:

```
pwr_rtt> debug signals /name= /file=
```

/name	The name of a plcpgm or a window object. Wildcard is allowed. If the object is a plcpgm all the signals in the plcpgm are displayed, if a window all the signals in the window are displayed. Default value is the selected object.
/file	Name of the listfile of the plcmodule list. Default $pwrp_lis/pwrp_plcmodulelist.plis.

Example

```
pwr_rtt> debug signals
```

debug objects

Display the value of the debug parameter of one or many objects
Object that matches in the class, name and hierarchy description are displayed
on the screen. Only objects with a defined debugparameter is displayed.

```
pwr_rtt> debug object /name= /class= /hierarchy= /global
```

/name=	the name of the object. Wildcard is allowed. The name can be written without '/name='. Ex debug object *di*
/class=	displays objects of this class
/hierarchy=	displays objects below this object in the hierarchy. If no object is given the hierarchy of the current picture is default (show object /hierarchy).
/global	Search is also perfomed in mounted volumes of remote nodes.

debug children

Display the value of the debug parameter of the children of an object.
Only objects with a defined debugparameter is displayed.

```
pwr_rtt> debug children /name=
```

/name=	the name of the object. The name can be written without '/name='.

define

Define symbols and assign values to symbols.
The symbol can have an integer, float or string value.

System symbols

There are a number of symbols defined by the system.

rtt_os	Current operating system.
rtt_platform	Current platform.
rtt_time	Current time.

Value assignment

The assigned value can be an integer, float or string.

Examples
Float assignment

```
define num 3.22
define num 3.22e-12
```

Integer assignment

```
define i 0
```

String assignment

```
define some_str "This is a string"
```

Assignment of the value of another symbol

```
define a1 'a2'
```

Concatenate two string symbols

```
define a1 "MOTOR"
define a2 " START"
define a3 `a''a2'

sho sym a3
%RTT-I-MSG, Symbol a3 = MOTOR START
```

Fetch attribute values from the database

You can fetch an attribute value from the database by writing # before the attribute name and surround the name with apostrophes.

Example

```
define a '#rt-rtt-vax_vms-dv1.actualvalue'
sho sym a
%RTT-I-MSG, Symbol a = 0
```

```
if '#rt-rtt-vax_vms-dv1.actualvalue' eq 0
  say/text="Dv1 is 0"
endif
```

delete

delete object

Delete an object.
Only objects in dynamic volumes can be deleted.

Note that the current database view is not updated with removed objects. The object hierarchy has to be closed and opened again to show the new view.

```
pwr_rtt> delete object /name=
```

`/name`	name of the object including hierarchy

Example

```
pwr_rtt> delete object /name=N2-UGN-PLÅTAR-Mtrl1
```

directory

Display files. Wildcard can be used.
If no file or directory is specified, the files in the current default directory will be displayed.

```
pwr_rtt> directory 'filespec'
```

exit

Terminate rtt.

help

Show helptexts.

If the command is given without parameter, a list of available helptexts are displayed. By selecting a subject an pressing Return the text for the subject is shown.

```
pwr_rtt> help
```

```
pwr_rtt> help 'subject'
```

learn

Learn makes it possible to store a command and key sequence. The key sequence is stored in a command file of type rtt_com. The sequence is repeted by executing the stored command file.

```
pwr_rtt> learn start /file=
pwr_rtt> learn stop
```

Example

```
pwr_rtt> learn start /file=motorstart
```

...
A command and key sequence.
...

```
pwr_rtt> learn stop


pwr_rtt> @motorstart
```

learn start

Start the learn session.
The following commands are stored in the specified file.

`/file` Name of command file where the commands are stored.

Example

```
pwr_rtt> learn start /file=dosering
```

learn stop

Terminates the learn session.

logging

The logging function makes it possible to logg database attributes to a file. There are two types of logging, continous logging where the values are stored with a specifiec frequency, and event logging where a value is stored when it is changed.

```
pwr_rtt> logging create /insert/time= /parameter= /file= /condition=
           /type= /buffer_size=
pwr_rtt> logging set /insert /entry= /time= /parameter= /file=
          /condition /type= /buffer_size=
pwr_rtt> logging delete /entry= /parameter=
pwr_rtt> logging start /entry=
pwr_rtt> logging stop /entry=
pwr_rtt> logging show /entry=
```

Example
The attributes to logg are first collected to the collection list. The 'logging create/insert' inserts the attributes of the collection list to the current logging entry.

```
pwr_rtt>logging create /insert/file=3.9::test.dat
pwr_rtt>logging start /entry=1
pwr_rtt>logging stop /entry=1
```

logging create

Create an entry in the logg table. One entry keeps information on logging on one loggfile. 10 entries can exist simultaneously. The 'logging create' command supplies an entry number that is specified in the following commands regarding the logging.

`/insert` Inserts the current attributes in the collection list into the logging entry.

`/time` Cycletime in ms

`/file` File specification for the file where the logging is stored.

`/parameter` Name of attribute that is to be logged.

`/condition` Name of an attribute of type boolean for conditional logging. The logging in only performed when the conditions attributes is true.

`/type` Logging type. Continous is the default type.

- EVENT when an attribute value is changed, the time and new value is written to the logfile.
- CONTINOUS the value of all attributes in the entry are written every cycle if the condition attributes is true.

`/buffer_size` Before the logging is written to file it is store in a buffer in the memory, and when the buffer is full, it is written to file. The buffersize can be increased when logging fast and t ime critical sequences to avoid interruption when the write is preformed.

`/priority` Priority during the logging.

`/intern` When the buffer is emptied, the logging is terminated.

logging set

Set or modify properties of a logging entry.

`/insert` Inserts the current attributes in the collection list into the logging entry.

`/time` Cycletime in ms

`/file` File specification for the file where the logging is stored.

`/parameter` Name of attribute that is to be logged.

`/condition` Name of an attribute of type boolean for conditional logging. The logging in only performed when the conditions attributes is true.

`/type` Logging type. Continous is the default type.

- EVENT when an attribute value is changed, the time and new value is written to the logfile.
- CONTINOUS the value of all attributes in the entry are written every cycle if the condition attributes is true.

`/buffer_size` Before the logging is written to file it is store in a buffer in the memory, and when the buffer is full, it is written to file. The buffersize can be increased when logging fast and t ime critical sequences to avoid interruption when the write is preformed.

`/priority` Priority during the logging.

`/intern` When the buffer is emptied, the logging is terminated.

logging delete

Remove an entry in the logging table, or an attribute in an entry. If /parameter is omitted, the entry is removed.

`/entry` Entry number.
`/parameter` Attrbute that is to be removed from the entry.

logging start

Start the logging for the specified entry.

`/entry` Entry number.

logging stop

Stop the logging for the specified entry.

`/entry` Entry number.

logging store

Store the properties of a logging entry, or of all logging entries to a rtt commandfile. The logging etries can be restored by executing the commandfile.

```
pwr_rtt> logging store /entry=1/file=temperature_logg
```

`/entry`	Entry to be stored. By default the currently displayed entry.
`/file`	Name of the commandfile where the entry are stored.
`/all`	Store all logging entries in the commandfile.

logging show

Show the properties for a logging entry.

`/entry`	Entry number.

login

Login as a ProviewR user with username and password.
The login command will open the login frame which promps for username and password.

```
pwr_rtt> login
```

page

Show the specified page.

```
pwr_rtt> page 'pagenumber'
```

monitor grafcet

Display the active steps and orders of a grafcet sequence.
Select a PlcPgm object that contains a grafcet sequence. The name of the active steps and orders will be displayed continously. The monitor can also be started from the 'SHOW GRAFCET' picture.

```
pwr_rtt> monitor grafcet
```

plcscan

This command set the execution of the plc window on or off.
If no hierarchy object i given, the selected object is chosen as hierarchy object.

```
pwr_rtt> plcscan /on [/hierarchy=] [/all]
pwr_rtt> plcscan /off [/hierarchy=] [/all]
```

`/on`	set execution on
`/off`	set execution off
`/all`	set execution of all plc windows in the system or in the node
`/hierarchy=`	set execution of all windows below this object in the hierarchy. If no object is given the hierarchy of the current picture is default (show object /hierarchy).
`/global`	Search is also perfomed in mounted volumes of remote nodes.

print

Print the current picture to a file or printer:
- print the content of a picture or menu to a file.
- print a string into file.
- print a copy of the terminal to a printer.
- print the attributes and values of the current picture toa rtt script-file
  so that the values can be restored when executing the script.

Print the content of the current picture or menu into a file.

```
pwr_rtt> print /file= /append /psize= /tsize=
```

/file	filename. Default extention is .lis
/append	write in an existing file.
/psize	number of characters for parameter name. Default 25
/tsize	number of characters the text. Default 15

Print a copy of the terminal to a printer.

```
pwr_rtt> print /terminal
```

Print a string to a file.

```
pwr_rtt> print /text= /append
```

Print the attribues and values of the curren picture to a rtt script-file, so that the values can be restored when executing the script

```
pwr_rtt> print /restore /file=
```

/file	filename. Default extention is .rtt_com

say

Write a text in the terminal window.

`/text` Text to write.

Example

```
say /text="Now I'm writing a text in the terminal window"
```

search

Search for a string in the current picture.
The search starts from the selected object and the search direction is forward. If no searchstring is specified the last specified searchstring is used.

```
pwr_rtt> search 'searchstring'
```

setup

Show the settings of the current rtt session.

Platform	The current platform.
ConfigureObject	The rtt configure object for the current session.
DefaultVMSNode	The DECNET-address for a VMS node where crossreference-files, logging-files etc will be opened.
DescriptionOn	If TRUE the content of the description-attribute will be diplayed.
DefaultDirectory	The default directory for displaying files with the commands 'directory' and 'show file'.
ScanTime	The rtt scantime for update of pictures, alarmlists etc.
AlarmMessage	If TRUE the last not acknowledged alarm will be displayed in the message row (row 23).
AlarmReturn	Return events will be shown in the event-list (this must be set before the eventlist is loaded).
AlarmAck	Acknowledge events will be shown in the event-list (this must be set before the eventlist is loaded).
SymbolFileName	The name of the symbolfile where symbols can be defined.
Verify	If TRUE the commands executed in commandfiles will be displayed.
SignalTestModeOn	If TRUE the pictures for Do- and Ao-channels will contain functions to set testvalues.

```
pwr_rtt> setup
```

show

Display information about the system
Allowed qualifiers are:

```
pwr_rtt> show object /objdid=
pwr_rtt> show object /name= /class= /hierarchy=
pwr_rtt> show object /name= /type= /file=
pwr_rtt> show parameter/parameter= /name= /class= /hierarchy=
pwr_rtt> show hierarchy
pwr_rtt> show class /name= /volume=
pwr_rtt> show signals /name=
pwr_rtt> show conversion [/name=]
pwr_rtt> show invert [/name=]
pwr_rtt> show system
pwr_rtt> show nodes
pwr_rtt> show subcli
pwr_rtt> show subsrv
pwr_rtt> show error
pwr_rtt> show plcpgm
pwr_rtt> show grafcet
pwr_rtt> show device
pwr_rtt> show plcthread
pwr_rtt> show pid
pwr_rtt> show logging
pwr_rtt> show nmps
pwr_rtt> show remnodes
pwr_rtt> show remtrans
pwr_rtt> show time
pwr_rtt> show clock
pwr_rtt> show alarm
pwr_rtt> show event
pwr_rtt> show symbol
pwr_rtt> show user
```

show nodes

Display the node known by the nethandler.
Data for the node is:
- node name.
- PAMS/DMQ group number.
- ProviewR node number.
- link (Up: nethandler link, Istrt: PAMS/DMQ link, Down: no link)
- upcount
- Time up. Time since nethanler link was up.
- Sent. Messages sent.
- Rcvd. Messages received.

show subcli

Display the subscriptions served by other systems.

show subsrv

Display the served subscriptions.

show error

Display devices that has an errorcounter greater than zero.

show plcpgm

Display the plcpgm's of the system.
Supported actions:

RETURN	Go down in the object hierarchy of the plcpgm.
PF1	Show the signals referenced by the plcpgm code.
PF2	Debug the plcpgm.

show grafcet

Display the plcpgm's containing grafcet sequences.

If the text 'Active' is displayed in front of the plcpgm name the grafcet sequence activity is not in the InitStep.

To monitor grafcet sequences do this:

- mark the plcpgm by positioning the cursor on the plcpgm and press return. Four plcpgm's can be monitored at the same time.

- Then press PF2 to start the grafcet monitor.

The grafcet monitor shows the active steps and the active orders of the grafcet sequence. (By giving the orders proper names it will be easier to examine the sequence from the monitor).

show device

Display the device of the system.

show plcthread

Display the PlcThread objects of the system.
The PlcThread object contains statistic.
The following attributes of the PlcThread objects is displayed:

- Object name.	
- Count	Loopcount for the thread.
- Scantime	Configured scantime for the thread.
- Mean Scantime	Mean scantime.
- Coverage	Ratio between start and stop of execution of the thread, and scantime. The value is in percent.
- Count_1_8	Number of loops where the coverage is less the 1/8
- Count_1_4	Number of loops where the coverage is less the 1/4

show pid

Display the pid objects of the system.
Select a pid object and press RETURN to open the pid object picture of the pid.

show logging

Show logging displays the logging entries, and allows you to set parameters for the logging function, start and stop the logging.
Do this steps to create a logging

- put the attributes you want to log in the collections picture. Max number of attributes in eatch entry is 100. Only the 10 first is show in the logging picture. If you want to see all, use the 'More' button.
- display the collection picture at least once
- enter the logging picture
- replace the filename if you aren't satisfied with the default name.
- choose logg type (EVENT or CONT).
- select 'INSERT' and press PF1. The attributes of the collection picture will now be diplayes in the picture.
- select 'START' and press PF1. The logging will now start.
- select 'STOP' when it's time to terminate the logging.
- select 'SHOW FILE' to examine the loggfile.
- the logging setup can be stored on disk with the 'Store' button, and can be restored later with the 'Re' button.
- if another logging is to be don simultaneously, select another logging entry with 'Next' or 'Previous' and start the logging in this entry.

See the logging function for more information

show nmps

Display the content of the NMps cells in the system.
The following data are shown in the picture:

- Name of the cell.
- Fu: Function of the cell.
- Si: Size. Max number of data objects.
- To: Current number of data objects in the cell.
- Nu: Index of the displayed data object.
- Last name segment of the data object.

Select a cell or a data object with the arrow keys.
RETURN diplays the object picture of the selected cell.
Ctrl/A or PF1 shows the selected data object.
Ctrl/T or PF2 expands/collapses the cell. Show all or the first data objects of the selected cell

In the cell object picture you can examine the content of the cell and set attributes.

show remnode

Display the RemNode objects.

The following data are shown in the picture:

- Name of the RemNode.
- Transport type.
- Description.

If the 'Change Mode' button i activated (PF1) the Address will be displayed.

show remtrans

Display the RemTrans objects.

The following data are shown in the picture:

- Name of the RemTrans.
- DataValid.
- Direction (send or receive).
- TransCount, number of transaktions.
- TransTime, the time of the last transaktion.
- ErrorCount, number of errors.
- Status, status of the last transaktion.

show object

Display information about objects in the database

```
pwr_rtt> show object /objdid=
pwr_rtt> show object /name= /class= / hierarchy= /global /maxobjects=
pwr_rtt> show object /name= /type= /file= n
```

Show an object with a specified objid.

```
pwr_rtt> show object /objdid=
```

`/objid=`	displays the object with the specified objid. Objid should be specified on the format 'A.B.C.D:E'.

Show objects that fits in the class, name and hierarchy description.

```
pwr_rtt> show object /name= /class= / hierarchy= /global /maxobjects=
```

`/name=`	the name of the object. Wildcard is allowed. The name can be written without '/name='. Ex pwr_rtt> show object *di*
`/class=`	displays objects of this class
`/hierarchy=`	displays objects below this object in the hierarchy. If no object is given the hierarchy of the current picture is default (show object /hierarchy).
`/global`	Search is also perfomed in mounted volumes of remote nodes.
`/maxobjects`	Max number of objects that will be shown. Default is 300.n

Show an object and interpret the content as a struct specified in a c-includefile.

```
pwr_rtt> show object /name= /type= /file=
```

`/name=`	the name of the object.
`/type=`	the name of the struct.
`/file=`	the name of c-includefile where the struct is defined.

show parameter

Display the value of a parameter of one or many objects
Object that fits in the class, name and hierarchy description are displayed on the screen.

```
pwr_rtt> show parameter /parameter= /name= /class= /hierarchy= /global
           /maxobjects=
```

`/parameter=`	the name of the parameter to be displayed
`/name=`	the name of the object. Wildcard is allowed. The name can be written without '/name='. Ex show object *di*
`/class=`	displays objects of this class
`/hierarchy=`	displays objects below this object in the hierarchy.

	If no object is given the hierarchy of the current picture is default (show object /hierarchy).
/global	Search is also perfomed in mounted volumes of remote nodes.
/maxobjects	Max number of parameters that will be shown. Default is 300.

show class

Show class definition objects.

Display one class (/name) or all classes in a classvolume (/volume).

```
pwr_rtt> show class /name=
pwr_rtt> show class /volume=
```

/name=	name of a class (without pathname).
/volume=	name of a classvolume.

show hierarchy

Display the root objects in the object hierarchy.

show signals

Display the signals referenced in a plcpgm or a window.

Allowed qualifiers are:

```
pwr_rtt> show signals /name= /file=
```

/name	the name of a plcpgm or a window object. Wildcard is allowed. If the object is a plcpgm all the signals in the plcpgm are displayed, if a window all the signals in the window are displayed. Default value is the selected object.
/file	Name of the listfile of the plcmodule list. Default pwrp_lis:pwrp_plcmodulelist.plis.

Example

```
pwr_rtt> show signals
```

show step

Display the active steps on the current node.
Displays all active steps or active steps below an specified object.

Allowed qualifiers are:

```
pwr_rtt> show step /hierarchy= /initstep
```

/hierarchy=	displays active steps below this object in the hierarchy. If no object is given the selected object or the hierarchy of the current picture is default (show step /hierarchy).
/initstep	The steps and the initsteps are shown. Otherwise only the steps are shown.

Example

```
pwr_rtt> show step
```

show conversion

Display io conversion of a input channel.
The channel can be specified by name (/name qualifier) or by selecting the channel, or the connected signal.

```
pwr_rtt> show conversion [/name=]
```

/name	Name of a channel object, or the connected signal object.

show invert

Display io invert of a channel.
The channel can be specified by name (/name qualifier) or by selecting the channel, or the connected signal.

```
pwr_rtt> show invert [/name=]
```

/name	Name of a channel object, or the connected signal object.

show file

Show files.
If no filespecification is given, the rtt commandfiles in the default directory is shown.
To show a picture or execute a commandfile, select the picture or file and press RETURN.

```
pwr_rtt> show file
pwr_rtt> show file *sto*.rtt_com
```

show alarm

Show the alarm list.
The same function as 'alarm show'. No qualifier is allowed.

```
pwr_rtt> show alarm
```

show event

Show the event list.
The same function as 'alarm list'. No qualifier is allowed.

```
pwr_rtt> show event
```

show symbol

Show the value of a symbol or show all the defined symbols.

```
pwr_rtt> show symbol 'symbol_name'
pwr_rtt> show symbol [/all]
```

show user

Show the current user with name and privileges.

```
pwr_rtt> show user
```

store

Store a collection picture, a debug picture or the defined symbols in a file.
The format of the file is rtt command file format.

Store symbols.
If no file is specified the symbolfile-name in the rttconfig-object will be used.

```
pwr_rtt> store /symbols
```

Store a picture.

```
pwr_rtt> store /file= [/collect]
```

`/file`	filename in which the picture will be stored.
`/collect`	The picture will be restored in the collection picture. If omitted the picture will be restored as a debug picture.

The picture stored in the file /file='filename' will be restored
with the command

```
pwr_rtt> @'filename'
```

set

List of set commands:

```
pwr_rtt> set parameter/name=/value=
pwr_rtt> set pwrp_alias
pwr_rtt> set [no]alarmmessage
pwr_rtt> set [no]alarmbeep
pwr_rtt> set conversion [/name][/on][/off]
pwr_rtt> set invert [/name][/on][/off]
pwr_rtt> set dotest [/name][/on][/off]
pwr_rtt> set testvalue [/name][/on][/off]
pwr_rtt> set clock
pwr_rtt> set time
pwr_rtt> set [no]description
pwr_rtt> set scantime/time=
pwr_rtt> set default
pwr_rtt> set [no]message
pwr_rtt> set [no]draw
```

set pwrp_alias

Executes the settings in the file pwrp_alias.dat.

```
pwr_rtt> set pwrp_alias
```

set alarmmessage

Last not acknowledged alarm will be shown in the message line.

```
pwr_rtt> set alarmmessage
```

The eventlist has to be loaded.

set alarmbeep

A sound signal will occur if there is a a not acknowledged alarm.

```
pwr_rtt> set alarmbeep
```

The eventlist has to be loaded.

set parameter

Set the value of a parameter.

```
pwr_rtt> set parameter/name=/value=
```

`/name`	full name of parameter ('objectname'.'parametername')
`/value`	the new value of the parameter.

Example

```
pwr_rtt> set par/name=sh-fsdn-motorstart.actualvalue/value=1
```

set conversion

Set io conversion of a input channel.
The channel can be specified by name (/name qualifier) or by selecting the channel, or the connected signal.

```
pwr_rtt> set conversion /on [/name=] [/on] [/off]
pwr_rtt> set conversion /off [/name=] [/on] [/off]
```

/name	Name of a channel object, or the connected signal object.
/on	Set conversion on.
/off	Set conversion off.

set invert

Set io invert of a channel.
The channel can be specified by name (/name qualifier) or by selecting the channel, or the connected signal.

```
pwr_rtt> set invert /on [/name=]
pwr_rtt> set invert /off [/name=]
```

/name	Name of a channel object, or the connected signal object.
/on	Set invert on.
/off	Set invert off.

set dotest

Set io test on a do channel.
The value of the do will be fetched from the do testvalue, witch can be set or reset by the 'set testvalue' command
The channel can be specified by name (/name qualifier) or by selecting the channel, or the connected signal.

```
pwr_rtt> set dotest /on [/name=]
pwr_rtt> set dotest /off [/name=]
```

/name	Name of a channel object, or the connected signal object.
/on	Set dotest on.
/off	Set dotest off.

set testvalue

Set testvalue on a do channel.
The testvalue will be set as output if the dotest is on (use the 'set dotest' command)
The channel can be specified by name (/name qualifier) or by selecting the channel, or the connected signal.

```
pwr_rtt> set dotest /on [/name=]
pwr_rtt> set dotest /off [/name=]
```

/name	Name of a channel object, or the connected signal object.

/on	Set testvalue to true on.
/off	Set testvalue to false.

set description

Display or hide the description attribute in the object hierarchy.
If description mode is set off, the name and class of an object will be displayed. If description is set on, the description attribute of the object will also be displayed if there is one.
The action will only affect new hierarchy-pictures, not the already created.

pwr_rtt> set description	display description.
pwr_rtt> set nodescription	hide description.

set file

All print commands, where a file is not specified, will be written to the file specified in the 'set file/on' command. Also messages and command will be written to the file.

/on	Start of set file session.
/off	Stop the file session.
/name	File specification.
/message	Rtt messages will be written to the file.
/command	Executed rtt commands will be written to the file.

Example

```
set file/on/name=”$pwrp_lis/set_file_example.txt”/message/command
print/text="Start of set file example..."
...
set file/off
quit
```

set scan

Set the scan time of rtt. This is the time rtt will update a picture, scan the alarmlist etc.
Time in seconds.

```
pwr_rtt> set scan/time=
```

set default

Set the default directory for displaying files with the 'show file' and
the 'directory' commands.
pwr_rtt> set default 'directory'

set message

Enable and disable of rtt messages.
Termination of commandfile mode will always enable rtt messages.

```
pwr_rtt> set message
```

```
pwr_rtt> set nomessage
```

set draw

Display result of the actions performed in a script on the screen.
The normal condition when executing a script is that nothing is displayed on the screen until the script is terminated. Then the current picture is redrawn. With 'set draw' command, all actions will be displayed on the screen as if they were preformed in an interactive mode. 'set draw' will also supress the redrawing of the picture when at termination of the script.

'set nodraw' will return to the concealed mode.

```
set draw
set nodraw
```

top

Display the first page and select the first item.

```
pwr_rtt> top
```

view

Show the content of a text-file.

```
pwr_rtt> view /file='filespec'
```

wait

Wait a specified time or until the plc program is started.
This command is for rtt command files.

```
pwr_rtt> wait /plcpgm
pwr_rtt> wait /time=
```

`/time`	Time in seconds.
`/plcpgm`	Wait until the plcprogram is loaded and initalized.

Example

```
wait /plcpgm
set parameter/parameter=sh-fsdn-börv_varvtal.actualvalue/value=5
exit
```

Script

A script-file will be executed from the command-line with the command
pwr_rtt> @'filename'n
For more information about the rtt script syntax use command 'HELP SCRIPT'