

# Trazer Reference Manual

---



Trazer is a visualization tool that works in conjunction with the RKH framework built in trace facility. Trazer gives the possibility to display selectively the recording of all events of your system, state machines, queues, timers, etc. Trazer helps you to faster troubleshooting especially on complex problems where a debugger is not sufficient, by providing a simple consolidated, human-readable textual output.

Given the RKH cross platform portability, trace data may come from 8, 16, or 32-bits platforms. In order to that Trazer need to be configured to support this diversity of platform and the wide range of RKH framework configurations.

This manual includes:

- **Configuration**
- **Human readable output**
- **Trazer revision history**

---

## Configuration

Before using Trazer you must provides the listed configurations:

- **Command-line parameters** to configure communication link and general options.
- **Configuration file** ( "**trazer.cfg**" ) to setup all target dependencies, such as pointer sizes, signal sizes, etc.
- **Trace event table** trace event table ( "**trazer.evt**" ), to define parseable trace events and custom information.

---

## Command-line parameters

Option	Example	Comments
-h	-h	Help. Prints the summary of options
-q	-q	Quiet mode (no stdout output)
-v	-v	Visualize Traze version and compatibility with RKH framework
-o	-o trazer.txt	Produce output to the specified file
-c	-c COM1 115200 8N1	COM port and baudrate selection. Not compatible with -t, -f
-t	-t 6602	TCP/IP server and port number. Not compatible with -c, -f (Not implemented)

-f -f trace.bin File input selection. Not compatible with -c, -t

## Configuration file

Trazer is designed to work with all possible target CPU, which requires a wide range of configurability. For example, for any given target CPU, Trazer must "know" the size of object pointers, event size, timestamp size and so on. You must to provide this information through "**trazer.cfg**", which content is summarized in the following table:

Parameter	Valid Values	Must match RKH config ( <a href="#">rkhcfg.h</a> )	Comments
TRAZER_SIZEOF_SIG	1, 2, 4	<a href="#">RKH_SIZEOF_EVENT</a>	Event Signal size in bytes.
TRAZER_SIZEOF_TSTAMP	1, 2, 4	<a href="#">RKH_TRC_SIZEOF_TSTAMP</a>	Bytes quantity used by the trace record timestamp.
TRAZER_SIZEOF_POINTER	1, 2, 4	<a href="#">RKH_TRC_SIZEOF_POINTER</a>	Size in bytes of void pointer.
TRAZER_SIZEOF_NTIMER	1, 2, 4	<a href="#">RKH_TIM_SIZEOF_NTIMER</a>	Dynamic range of time delays measured in ticks and expressed in bytes.
TRAZER_SIZEOF_NBLOCK	1, 2, 4	<a href="#">RKH_MP_SIZEOF_NBLOCK</a>	Size in bytes of number of memory block size.
TRAZER_SIZEOF_NELEM	1, 2, 4	<a href="#">RKH_RQ_SIZEOF_NELEM</a>	Maximum number of elements in bytes that any queue can contain.
TRAZER_SIZEOF_ESIZE	1, 2, 4	<a href="#">RKH_SIZEOF_ESIZE</a>	Data type of event size, in bytes.
TRAZER_EN_NSEQ	0, 1	<a href="#">RKH_TRC_EN_NSEQ</a>	Enable/Disable Number of Sequence use in trace stream.
TRAZER_EN_CHK	0, 1	<a href="#">RKH_TRC_EN_CHK</a>	Enable/Disable Checksum use in trace stream.
TRAZER_EN_TSTAMP	0, 1	<a href="#">RKH_TRC_EN_TSTAMP</a>	Enable/Disable Time stamp use in trace stream.

You must ensure that Trazer configuration, match exactly with the target system under test, otherwise will be unable to parse the trace stream, and these errors would be shown:

\*\*\*\*\* Stream Checksum Error

\*\*\*\*\* May be have lost trace info, sequence are not correlatives

## Trace event table

In order to support future events definitions from new RKH releases, and potential user defined traces, each event must be defined into "**trazer.evt**" file to be recognized by Trazer. Each particular event is defined as shown in the table below:

Field	Example	Description
Event	RKH_TRCE_MP_INIT	Event ID, must be according to <a href="#">RKH_TRC_EVENTS</a> definition in <a href="#">rkhtrc.h</a>
Group	MP	RKH service that generate the event.
		Custom Alias for the event

Name      INIT

Comment   Memory Pool Init      Optional descriptive comment.

Trazer user may can change **Alias**, **Comment** and **Group** fields to customize the visualization output. In the instrumented application two particular trace events could be used to better trace visualization, **RKH\_TRCE\_OBJ** and **RKH\_TRCE\_SIG**.

### RKH\_TRCE\_OBJ:

Associates the address of the object, in memory with its symbolic name. Using the **RKH\_TRCR\_RKH\_OBJ()** macro in application source code looks like:

```
...
static int g_status;
static RKHT_T tdll;

(1) RKH_TRCR_RKH_OBJ( &g_status );
(2) RKH_TRCR_RKH_OBJ( &tdll );
```

Explanation:

- (1-2) The **RKH\_TRCR\_RKH\_OBJ()** macro associates the object memory address with its symbolic name. This macro also performs the trace event recording to local stream.

### RKH\_TRCE\_SIG:

Associates the numerical value of the event signal to the symbolic name of the signal. Using the **RKH\_TRCR\_RKH\_OBJ()** macro in application source code looks like:

```
// in the dedicated application header file resides the signal enumeration

enum
{
    ...
    PWR_FAIL,
    PRESS_ENTER,
    ...
};

// frequently, the macro RKH_TRCR_RKH_SIG() is used in the \b main.c file.

(1) RKH_TRCR_RKH_SIG( PWR_FAIL );
(2) RKH_TRCR_RKH_SIG( PRESS_ENTER );
```

Explanation:

- (1-2) The **RKH\_TRCR\_RKH\_SIG()** macro associates the numerical value of the event signal to the symbolic name of the signal. This macro also performs the trace event recording to local stream.

RKH user must generate this particulars trace events as described in RKHTRACE and Trazer will take care of reemplacing the numeric values by the user definition.

## Human readable output

Trazer is a console program that converts the trace stream data in a human-readable format. Following is shown how its output looks like.

Trazer start showing version and compatibility with RKH framework, below are displayed the current Trazer configuration. For this example the trace.bin is used as trace data source.

Each identified trace show:

- **Timestamp**, in CPU ticks. The number of bytes used by the timestamp is configurable by the macro **TRAZER\_SIZEOF\_TSTAMP**. See the **TRAZER\_EN\_TSTAMP** option.

- **Sequence number**, the target component increments this number for every frame inserted into the stream. The sequence number allows the trazer tool to detect any data discontinuities. See the **TRAZER\_EN\_NSEQ** option.
- **System service group name**.
- **Trace event alias**, which is one of the predefined RKH records or an application-specific record. See the **"trazer.evt"** file and **RKH\_TRC\_EVENTS**.
- **Args**, data associated with the event.
- Comment.

Sequences 4, 5 and 6 are examples of **RKH\_TRCE\_OBJ** and **RKH\_TRCE\_SIG** events. After, sequences 16, 17, 32 and 33, show how Trazer identify object address and signal number replacing them by its symbolic representation. In case that no symbolic representation has been defined for a particular object, (null) will be shown, and in same manner the numeric value for signals.

```
TRAZER Visualization Tool V1.0 compatible with RKH V2.0
```

```
Date = Apr 7 2012 01:46:48
```

#### Trace Setup

```
Trace events quantity = 45
TRAZER_SIZEOF_SIG      = 1
TRAZER_SIZEOF_TSTAMP   = 4
TRAZER_SIZEOF_POINTER  = 4
TRAZER_SIZEOF_NTIMER   = 2
TRAZER_SIZEOF_NBLOCK   = 1
TRAZER_SIZEOF_NELEM    = 1
TRAZER_SIZEOF_ESIZE    = 2
TRAZER_EN_NSEQ         = 1
TRAZER_EN_CHK          = 1
TRAZER_EN_TSTAMP       = 1
RKH_TRC_ALL            = 1
RKH_TRC_EN_MP          = 0
RKH_TRC_EN_RQ          = 0
RKH_TRC_EN_SMA         = 0
RKH_TRC_EN_TIM         = 0
RKH_TRC_EN_SM          = 0
RKH_TRC_EN_RKH        = 0
```

```
----- Parsing trace stream from file trace.bin -----
```

```
84 [ 0] MP | INIT           : mp=(null), nblock=16 : Memory Pool Init
84 [ 1] RKH | EPOOL_REG      : epix =1, ssize=64, esize=4
84 [ 2] MP | INIT           : mp=(null), nblock=4  : Memory Pool Init
84 [ 3] RKH | EPOOL_REG      : epix =2, ssize=32, esize=8
84 [ 4] RKH | SYM_OBJ        : obj=0x01091780, sym=&rkheplist[0]
84 [ 5] RKH | SYM_OBJ        : obj=0x01091794, sym=&rkheplist[1]
84 [ 6] RKH | SYM_SIG        : sig=1, sym=ONE
84 [ 7] RQ | INIT           : rq=(null), sma=(null), nelem=4
84 [ 8] SMA | REGISTER       : sma=(null), prio=0
84 [ 9] SM | INIT           : sma=(null), istate=(null)
84 [10] TIM | INIT          : timer=(null), sig=(null)
84 [11] SM | ENTRY_STATE     : sma=(null), state=(null)
84 [12] SM | ENTRY_STATE     : sma=(null), state=(null)
84 [13] SM | ENTRY_STATE     : sma=(null), state=(null)
84 [14] SMA | ACTIVATE       : sma=(null)
86 [15] RKH | ENTER         :
```

```

1509 [ 16] MP | GET : mp=&rkheplist[1], nfree=3 : Memory Pool Get
1509 [ 17] RKH | ALLOC_EVENT : esize=6, sig=ONE
1509 [ 18] RQ | POST_FIFO : rq=(null), nused=1
1509 [ 19] SMA | POST_FIFO : sma=(null), sig=ONE
1509 [ 20] RQ | GET_LAST : rq=(null)
1509 [ 21] SMA | GET_EVENT : sma=(null), sig=ONE
1509 [ 22] SM | DISPATCH : sma=(null), sig=ONE
1509 [ 23] SM | TRANSITION : sma=(null), sstate=(null), tstate=(null)
1509 [ 24] SM | COMP_STATE : sma=(null), state=(null)
1509 [ 25] SM | NUM_EN_EX : sma=(null), nentry=1, nexit=1
1509 [ 26] SM | EXIT_STATE : sma=(null), state=(null)
1509 [ 27] SM | NUM_TRN_ACT : sma=(null), ntrnaction=1
1509 [ 28] SM | ENTRY_STATE : sma=(null), state=(null)
1509 [ 29] SM | CURRENT_STATE : sma=(null), state=(null)
1509 [ 30] SM | DISPATCH_RCODE : sma=(null), retcode=RKH_OK
1509 [ 31] RKH | GC_RECYCLE : sig=ONE
1509 [ 32] MP | PUT : mp=&rkheplist[1], nfree=4
1905 [ 33] MP | GET : mp=&rkheplist[1], nfree=3 : Memory Pool Get
1905 [ 34] RKH | ALLOC_EVENT : esize=6, sig=(null)
1905 [ 35] RQ | POST_FIFO : rq=(null), nused=1

```