

**NAME**

**librtld\_db** - library for run-time linker debugging

**LIBRARY**

Debugging interface to the runtime linker Library (librtld\_db, -lrtltd\_db)

**SYNOPSIS**

**#include** <rtld\_db.h>

*void*

**rd\_delete**(*rd\_agent\_t* \*rdap);

*char \**

**rd\_errstr**(*rd\_err\_e* rderr);

*rd\_err\_e*

**rd\_event\_addr**(*rd\_agent\_t* \*rdap, *rd\_event\_e* event, *rd\_notify\_t* \*notify);

*rd\_err\_e*

**rd\_event\_enable**(*rd\_agent\_t* \*rdap, *int* onoff);

*rd\_err\_e*

**rd\_event\_getmsg**(*rd\_agent\_t* \*rdap, *rd\_event\_msg\_t* \*msg);

*rd\_err\_e*

**rd\_init**(*int* version);

*typedef int*

**rl\_iter\_f**(*const rd\_loadobj\_t \**, *void \**);

*rd\_err\_e*

**rd\_loadobj\_iter**(*rd\_agent\_t* \*rdap, *rl\_iter\_f* \*cb, *void* \*clnt\_data);

*void*

**rd\_log**(*const int* onoff);

*rd\_agent\_t \**

**rd\_new**(*struct proc\_handle* \*php);

*rd\_err\_e*

```
rd_objpad_enable(rd_agent_t *rdap, size_t padsiz);
```

*rd\_err\_e*

```
rd_plt_resolution(rd_agent_t *rdap, uintptr_t pc, struct proc *proc,  
    uintptr_t plt_base, rd_plt_info_t *rpi);
```

*rd\_err\_e*

```
rd_reset(rd_agent_t *rdap);
```

## DESCRIPTION

The **librtld\_db** library provides a debugging interface to the run-time linker (rtld). This library must be used along with **libproc(3)**.

Most library functions take a *rd\_agent\_t* argument. This argument is an opaque structure containing information associated with the current status of the agent.

Before you start using **librtld\_db** you should call **rd\_init()** with the *RD\_VERSION* argument. This initializes the library to the correct version your program was compiled with and provides proper ABI stability.

What follows is a description of what each function.

**rd\_new()** creates a new **librtld\_db** agent. The *php* argument should be the *proc\_handle* you received from **libproc(3)**.

**rd\_reset()** resets your previously created agent.

**rd\_delete()** deallocates the resources associated with the agent.

**rd\_errstr()** returns an error string describing the error present in *rderr*.

**rd\_event\_enable()** enables reporting of events. This function always returns *RD\_OK*.

**rd\_event\_addr()** returns the event address corresponding to the *event* parameter. At the moment we only report events of type *RD\_NOTIFY\_BPT*.

**rd\_event\_getmsg()** returns the message associated with the latest event. At the moment only *RD\_POSTINIT* events are supported.

**rd\_loadobj\_iter()** allows you to iterate over the program's loaded objects. *cb* is a callback of type

**rl\_iter\_f()**.

## RETURN VALUES

Most functions return an *rd\_err\_e* type error. The error codes are described in the header file for this library. You can get the error string using **rd\_errstr()**.

## SEE ALSO

ld(1), ld-elf.so.1(1), ld.so(1), rtd(1), libproc(3)

## HISTORY

The **librtld\_db** library first appeared in FreeBSD 9.0 and was modeled after the same library present in the Solaris operating system.

## AUTHORS

The **librtld\_db** library and this manual page were written by Rui Paulo <[rpaulo@FreeBSD.org](mailto:rpaulo@FreeBSD.org)> under sponsorship from the FreeBSD Foundation.

## CAVEATS

The functions **rd\_event\_enable()**, **rd\_log()**, **rd\_objpad\_enable()** and **rd\_plt\_resolution()** are not yet implemented.