Squirrel Remote Debugger 1.0

Alberto Demichelis

Squirrel Remote Debugger 1.0

Alberto Demichelis

Table of Contents

Chapter 1. The Debugger

This part of the document describes how integrate the squirrel remote debugger in a application.

Overview

SQDBG is a tiny C++ library that allows to esily expose squirrel's VM internal state to an extenal debugger/IDE application. The library is based on TCP/IP and exposes it's functionalities through only 4 public functions.

Integrating the debugger

The debugger is designed to be the non-intrusive, because of this is based on non blocking socket and wont spawn any thread or other OS objects(except for the TCP socket).

The debugger logical flow is extremely simple.

- The application initilizes squirrel's VM.

  Note that for step by step debugging, debug infos generation must be enabled(see `sq_enabledebuginfo()`).

- The application initilizes SQDBG(`sq_rdbg_init()`).

- The application waits for an incoming connection from a client debugger(`sq_rdbg_waitforconnections()`).

- The application updates the debugger.

  This can be done through `sq_rdbg_update()` or by enabling 'autoupdate' while initilizing the debugger.

```
int main(int argc, char *argv[])
{
        if(argc < 2){
                scprintf(_SC("SQDBG error : no file specified"));
                return -1;
        }

        HSQUIRRELVM v = sq_open(1024);
        sqstd_seterrorhandlers(v);

        //!! INITIALIZES THE DEBUGGER ON THE TCP PORT 1234
        //!! ENABLES AUTOUPDATE
        HSQREMOTEDBG rdbg = sq_rdbg_init(v,1234,SQTrue);

        //!! ENABLES DEBUG INFO GENERATION(for the compiler)
        sq_enabledebuginfo(v,SQTrue);

        sq_setprintfunc(v,printfunc);

        const SQChar *fname=NULL;
#ifdef _UNICODE
        SQChar sTemp[256];
        mbstowcs(sTemp,argv[1],(int)strlen(argv[1])+1);
```

```
        fname=sTemp;
#else
        fname=argv[1];
#endif

        scprintf(_SC("SQDBG file =%s\n"),fname);
        if(!rdbg){
                scprintf(_SC("error starting the debugger"));
                return -1;
        }

        //!! SUSPENDS THE APP UNTIL THE DEBUGGER CLIENT CONNECTS
        if(SQ_SUCCEEDED(sq_rdbg_waitforconnections(rdbg))) {
                scprintf(_SC("connected\n"));

                //!!EXECUTES A SCTIPT
                sq_pushroottable(v);
                sqstd_dofile(v,fname,SQFalse,SQTrue);
        }
        //!! CLEANUP
        sq_rdbg_shutdown(rdbg);
        sq_close(v);
        return 0;
}
```

The network protocol

TODO

Chapter 2. API Reference

Debugger

sq_rdbg_init

```
HSQREMOTEDBG sq_rdbg_init(HSQUIRRELVM v, unsigned short port, SQBool autoupdate);
```

creates a new instance of the squirrel remote debugger.

parameters:

HSQUIRRELVM v	the target VM to be debugged
unsigned short port	TCP/IP port to listen at
SQBool autoupdate	enable/disable debugger autoupdate

return: an handle to a squirrel debugger

remarks: if autoaupdate is true the debugger will automatically call sq_rdbg_update() every time a line is executed. Autoupdate is a good choice for applications that do not have a busy loop, however the sq_rdbg_update() performs IO operations and this will impact on the application performances. For applications like games is raccomanded to set autoupdate to false and call sq_rdbg_update() once per frame.

sq_rdbg_shutdown

```
HRESULT sq_rdbg_shutdown(HSQREMOTEDBG rdbg);
```

Suspends the execution of the specified vm.

parameters:

HSQREMOTEDBG rdbg	the target debugger

return: an SQRESULT

sq_rdbg_update

```
HSQREMOTEDBG sq_rdbg_update(HSQREMOTEDBG rdbg);
```

updates the internal state of the debugger. if the parameter 'autoupdate' is set to true at initialization(sq_rdbg_init()), this function is automatically invoked by the debugger everytime a line is executed.

parameters:

HSQREMOTEDBG rdbg the target debugger

return: a SQRESULT (if the result is different than SQ_OK, the debugger wont work again and needs to be reinitialized).

remarks: Autoupdate is a good choice for applications that do not have a busy loop, however the sq_rdbg_update() performs IO operations and this will impact on the application performances. For applications like games is raccomanded to set autoupdate to false and call sq_rdbg_update() once per frame.

sq_rdbg_waitforconnections

```
SQRESULT sq_rdbg_waitforconnections(HSQREMOTEDBG rdbg);
```

waits for incoming connections from a client debugger.

parameters:

HSQREMOTEDBG rdbg the target debugger

return: an SQRESULT

Index

S