



Contents

- [1 General](#)
- [2 RDebug::Print format](#)
- [3 Enabling RDebug output in the emulator](#)
- [4 Viewing RDebug output in Carbide.c++](#)
- [5 Viewing RDebug output in Carbide.c++ \(advanced\)](#)
- [6 Viewing RDebug output outside of an IDE](#)
- [7 Capturing RDebug output in epocwind.out](#)

General

RDebug is a useful utility for checking a log of specific messages generated by the running code. This feature is helpful when the breakpoint/step debug tricks' using space has been limited.

To use RDebug, include the header file, `e32debug.h`. (In 2nd Edition SDK, RDebug class declared in `E32SVR.H`)

```
#include <e32debug.h>
```

Then, add the following to any place in your code:

```
// code before log
RDebug::Print(_L("### Log %d %08x"), 5, 0xABCDEF12);
// code after log
```

The output debug message is now viewable with a the [DebugView](#) tool.

This trick is especially useful for tasks with special requirements.

RDebug::Print format

The `RDebug::Print` parameter format is easy because it follows the C `printf` format. The one that most have trouble with is `%S` to print a descriptor. It expects a pointer to a descriptor, so you must use the `&` operator if you are printing a `TBuf`, for example.

```
// Print a HBufC
RDebug::Print(_L("Test string: %S"), hbuf );

// Print a TBuf
RDebug::Print(_L("Test string: %S"), &tbuf );

// useful macro, LINE , evaluates to the current line number in
RDebug::Print(_L("Debug on line %d"), __LINE__);
```

Enabling RDebug output in the emulator

The output of RDebug in emulator is written to:

- A text file, called **EPOCWIND.OUT**. The file is located in the TEMP folder. If you do not know the location of your TEMP folder, check Control Panel | Settings | Advanced | Environment Variables.
- Debugger view of the IDE (see the following sections).

For S60 3rd Edition, there are two options in the `\epoc32\data\epoc.ini` file to enable or disable RDebug output, that is:

- LogToFile
- LogToDebugger

The following code shows the content of `epoc.ini` with LogToFile and LogToDebugger enabled:

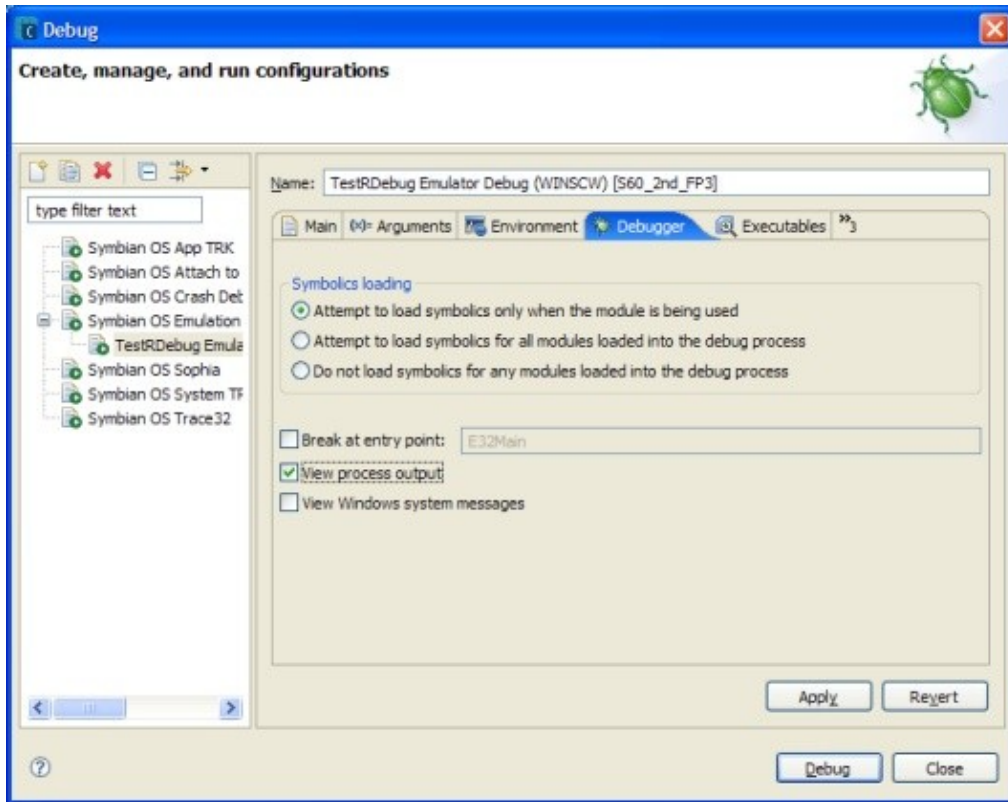
```
LogToFile 1  
LogToDebugger 1
```

Note: If changing the `epoc.ini` is something that you feel comfortable with, then try it. You can also control the same settings through the emulator's window menu. Select Tools -> Preferences. The logging options can be found in the General tab.

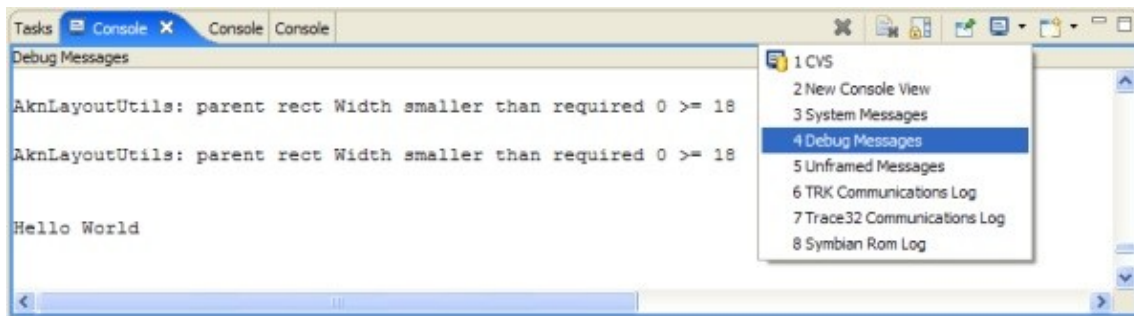
Viewing RDebug output in Carbide.c++

Viewing RDebug output in the Carbide.c++ IDE can be done by enabling "View process output". To enable it, right click project name and select Debug as | Debug.... Click the Debugger tab and enable "View process output" (see picture below).

How_to_use_RDebug



After the "View process output" option has been enabled, debug the project. To display the debug messages, click Open Console and select the Debug Messages menu. The picture below shows the output of RDebug in the Carbide.c++ IDE.

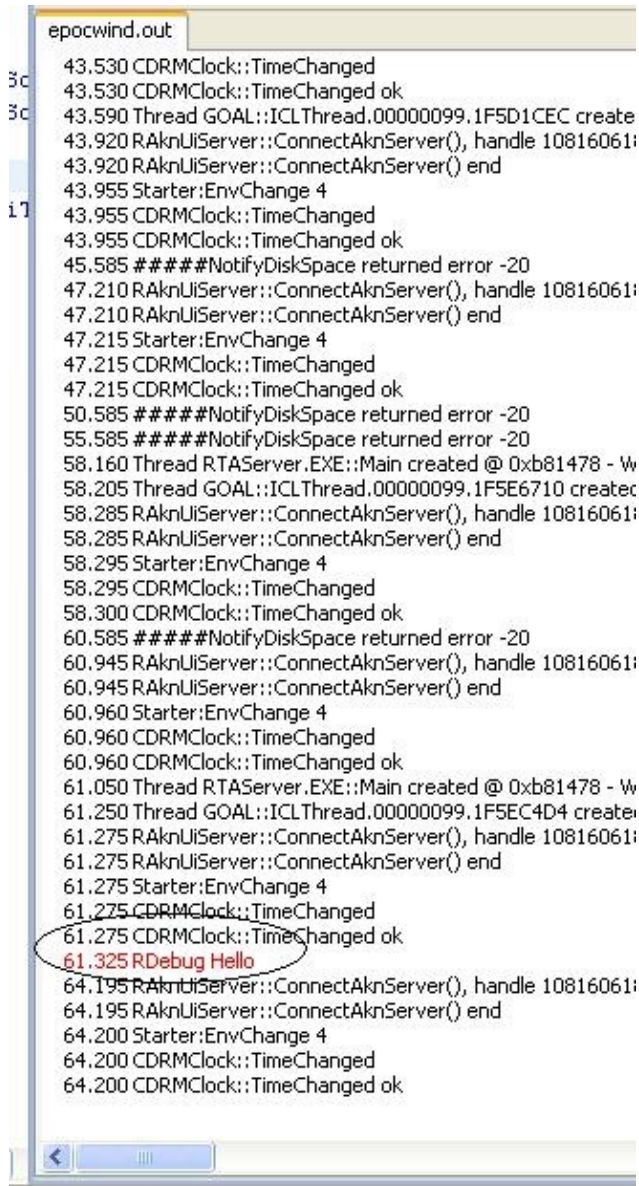


Viewing RDebug output in Carbide.c++ (advanced)

Carbide.c++ is an Eclipse based IDE. The advantage of this is that many existing Eclipse plug-ins can be easily integrated with Carbide.c++. One such example is the [Eclipse Logfile Viewer](#) plug-in, a tool that can dynamically load a log file, parse it according to user defined rules, and then display it in real time with customised formatting.

Plug-in configuration example:

How_to_use_RDebug



```
epocwind.out
3c 43.530 CDRMCLock::TimeChanged
3c 43.530 CDRMCLock::TimeChanged ok
43.590 Thread GOAL::ICLThread.00000099.1F5D1CEC create
43.920 RAknUIServer::ConnectAknServer(), handle 10816061:
43.920 RAknUIServer::ConnectAknServer() end
43.955 Starter:EnvChange 4
17 43.955 CDRMCLock::TimeChanged
43.955 CDRMCLock::TimeChanged ok
45.585 #####NotifyDiskSpace returned error -20
47.210 RAknUIServer::ConnectAknServer(), handle 10816061:
47.210 RAknUIServer::ConnectAknServer() end
47.215 Starter:EnvChange 4
47.215 CDRMCLock::TimeChanged
47.215 CDRMCLock::TimeChanged ok
50.585 #####NotifyDiskSpace returned error -20
55.585 #####NotifyDiskSpace returned error -20
58.160 Thread RTAServer.EXE::Main created @ 0xb81478 - W
58.205 Thread GOAL::ICLThread.00000099.1F5E6710 createc
58.285 RAknUIServer::ConnectAknServer(), handle 10816061:
58.285 RAknUIServer::ConnectAknServer() end
58.295 Starter:EnvChange 4
58.295 CDRMCLock::TimeChanged
58.300 CDRMCLock::TimeChanged ok
60.585 #####NotifyDiskSpace returned error -20
60.945 RAknUIServer::ConnectAknServer(), handle 10816061:
60.945 RAknUIServer::ConnectAknServer() end
60.960 Starter:EnvChange 4
60.960 CDRMCLock::TimeChanged
60.960 CDRMCLock::TimeChanged ok
61.050 Thread RTAServer.EXE::Main created @ 0xb81478 - W
61.250 Thread GOAL::ICLThread.00000099.1F5EC4D4 createc
61.275 RAknUIServer::ConnectAknServer(), handle 10816061:
61.275 RAknUIServer::ConnectAknServer() end
61.275 Starter:EnvChange 4
61.275 CDRMCLock::TimeChanged
61.275 CDRMCLock::TimeChanged ok
61.325 RDebug Hello
64.195 RAknUIServer::ConnectAknServer(), handle 10816061:
64.195 RAknUIServer::ConnectAknServer() end
64.200 Starter:EnvChange 4
64.200 CDRMCLock::TimeChanged
64.200 CDRMCLock::TimeChanged ok
```

Here you can see how the plug-in can be used to highlight the log messages sent by your application, recognised as starting with the keyword **RDebug**.

```
RDebug::Print( _L("RDebug Hello") );
```

When debugging on device with TRK:

Debug configuration -> Debugger tab:

- uncheck View messages between Carbide and debug agent
- check View Program Output

Viewing RDebug output outside of an IDE

It is also possible to view debug output without any IDE or debugger attached. This can speed up launch times for the emulator (for example, if you are using Just-in-time debugging as described in [How to debug with emulator on the fly](#)).

For this, you can use several tools that show Windows debug strings, such as [DebugView](#) from Microsoft. DebugView also has additional features such as highlighting or excluding strings with a particular pattern.

One thing to keep in mind is that the debug output can occasionally come from other processes in your system, not just the Symbian emulator, so with a tool like DebugView you can capture other strings not related to the application you are debugging. Again, filtering can be very helpful here.

Capturing RDebug output in epocwind.out

epocwind.out is a normal text file which is appended by the emulator so you can open it with any text editor. To be able to see the log prints as they come, you can use the [tail](#) freeware program. It is a port of the unix tail program that prints the "tail" of a file.

Create a bat file as follows:

```
tail -f %temp%\epocwind.out
```

This opens a dos-prompt to show the log prints as they come. It can be closed by pressing Ctrl-C.