

---

# RVDS 3.x with Eclipse IDE

---



<b>Title</b>	<b>Integrated Development Environment Eclipse and RVDS</b>
<b>Keywords</b>	Eclipse, RVDS
<b>Abstract</b>	This is a guide for setting up RVDS development environment on the basis of Eclipse IDE.

Copyright © 2007 Samsung Electronics Co, Ltd. All Rights Reserved.

Though every care has been taken to ensure the accuracy of this document, Samsung Electronics Co, Ltd. cannot accept responsibility for any errors or omissions or for any loss occasioned to any person, whether legal or natural, from acting, or refraining from action, as the result of the information contained herein. Information in this document is subject to change at any time without obligation to notify any person of such changes.

Samsung Electronics Co, Ltd. may have patents or patent pending applications, trademarks copyrights or other intellectual property rights covering subject matter in this document. The furnishing of this document does not give the recipient or reader any license to these patents, trademarks copyrights or other intellectual property rights.

No part of this document may be communicated, distributed, reproduced or transmitted in any form or by any means, electronic or mechanical or otherwise, for any purpose, without the prior written permission of Samsung Electronics Co, Ltd.

The document is subject to revision without further notice.

All brand names and product names mentioned in this document are trademarks or registered trademarks of their respective owners.

**Contact Address:**

Samsung Electronics Co., Ltd.  
San#24 Nongseo-Ri, Giheung\_Eup,  
Yongin\_City, Gyeonggi-do, Korea 449-711

Tel: (82)-(031)-209-2379  
Fax: (82)-(031)-209-6494

Home Page: <http://www.samsungsemi.com>  
Contact us: [youngseok.oh@samsung.com](mailto:youngseok.oh@samsung.com)

MCU Design Team,  
Chipcard & Microcontroller,  
System LSI Division,  
Semiconductor Business  
Samsung Electronics Co., Ltd.

### Revision History

Date	Version	Author	Amendment
Jul.12 2007	0.1	Youngseok Oh	Initial version
Sep.11.2007	1.0	Youngseok Oh	Finished





# 1. Overview and Requirements

## OVERVIEW

This document describes how to setup a development environment of RVDS based on Eclipse tool chain. Since Eclipse is an open-source integrated development environment, a user should install necessary plug-ins and add-ons.

ARM has employed the Eclipse tool chain since RVDS3.x. One of the advantages of Eclipse is free of charge. However, it requires many plug-ins and supplementary programs, for example, CDT and cygwin. In this document, installation procedure of those programs and how to get started will be described.

Additional benefits when using the Eclipse is to work with the Software Configuration Management tool (e.g. ClearCase or CVS) seamlessly. It forces the users to work in the boundary of the configuration tool by means of the systematic approach, so that it is manageable.

## REQUIEREMENTS

- Sun java
- Eclipse IDE for C/C++ Developers
- Cygwin
- RVDS 3.x plug-in

## 2. Installing Sun Java and Eclipse

### SUN JAVA

Eclipse is a set of jar files that can be executed in Java Runtime Environment (JRE). Java is a kind of virtual machine (VM) that interprets java language. The virtual machine allows the Java programs to run independently of the platform. (e.g. windows, linux, microcomputer, etc.) Thus, without JRE, Eclipse is not able to run in the computer. To install JRE, users shall visit sun's website and download Java SE Development Kit (JDK).

#### 1) Install

Firstly, Java shall be installed in the computer to run Eclipse. Users can easily download and install java by visiting sun's website below.

<http://java.sun.com>

Look for Java SE Development Kit (JDK), download, and install it.

#### 2) Check path

It might be necessary to check whether the path to Java home directory is included in the path. (It can be found in environment variables of system dialog box.)

### ECLIPSE IDE FOR C/C++ DEVELOPERS

Eclipse is an open-source integrated development environment. It is often used for Java development; however, there are many plug-ins that allow users to use it in different types of development.

To install Eclipse, visit the website below and download the latest version.

<http://www.eclipse.org/>

goto downloads > download Eclipse IDE for C/C++ Developers

#### 1) Install

Since Eclipse is a compressed file, a user needs to extract it any place in hard disk. (e.g. c:\eclipse)

#### 2) Designate workplace

When Eclipse is executed for the first time, it asks for the work place. It is a directory where the Eclipse projects are located. A user can select any place in hard disk.

## 3. Installing cygwin

### CYGWIN

Eclipse tool chain is just an integrated development environment, but not including development toolset such as make, compiler, linker, assembler, etc. Cygwin provides make file with open-source development suite that includes C, C++, Java, Ada, Fortran, etc. It can be easily incorporated with Eclipse. Cygwin is based on GNU toolset. Originally, GNU tool chain is able to run in any platform, such as, microcomputers like ARM, personal computers, Apple, etc. However, GNU toolset is developed upon Linux environment. Cygwin provides a way to use GNU toolset in windows.

Even if cygwin with GNU tool chain is able to run in ARM, it does not mean development toolset for ARM (compiler, linker, etc.) It shall be separately downloaded and installed later.

Cygwin can be downloaded in the website below.

<http://www.cygwin.com>

#### 1) Install

Clicking on the link to download cygwin pops up a dialog box asking where setup.exe to be stored. This setup.exe is not complete install file, but just setup wizard.

It requires users some procedures to go through. For details, refer to the document link below.

[http://www.embeddedhub.com/development/arm-board/doc/ARM\\_Cross\\_Development\\_with\\_Eclipse\\_version\\_3.pdf](http://www.embeddedhub.com/development/arm-board/doc/ARM_Cross_Development_with_Eclipse_version_3.pdf)

Or, goto the website below. Click Resources > Tools > James P Lynch's GNUARM tool download and installation instructions

<http://www.gnuarm.com/>

From page 30, a detail procedure is described. (Thanks to James P Lynch)

#### 2) Check path

Make sure if path to cygwin/bin is included in the path. Unless otherwise, it shall be included manually.

After completing installation, it might be required to change the version of make file. Make file version 3.81 might have a parsing problem<sup>1</sup> with makefile auto-generated by RVDS. In this case, it is recommended to download 3.80 version make.exe and replace this with 3.81 version make.exe in cygwin/bin.

Note) For those who want to use GNUARM, the document above will be helpful, too. However, it will not be handled in this document, because RVDS is ARM development toolset here.

---

<sup>1</sup> Not every RVDS seems to produce this problem. However, some may do. If make file has problem with recognizing “:” in dependency files (\*.d), it is the parsing problem. It is recommended to replace make file with old version (3.80).

## 4. Installing RVDS plug-in

### RVDS PLUG-IN

To incorporate Eclipse with RVDS, RVDS plug-in shall be installed. It plays a role of creating a project and assisting to manage the properties of project environment. Thus, users does not need to pay attention to creating project makefile.

Visiting the website below, users can download the plug-ins and related documents.

<http://www.arm.com/eclipse/>

The plug-in is formed in compressed file, so that it shall be extracted under Eclipse location. (Eclipse itself and those Plug-ins consist a set of jar files, so that Java Runtime Environment (JRE) shall be installed before execution.)

There are two different versions of plug-ins. One is for RVDS 3.1 version and the other is RVDS 3.0 version.

## 5. Creating RVDS project in Eclipse

Detail of creating a project is described in ARM site. Depending upon the version, users shall refer to appropriate version.

### MIGRATING ADS PROJECT TO RVDS PROJECT

Before starting to modify CSP, tools, documents, and FAQs below will be helpful for users.

1) Tool: SuiteSwitcher

This tool may help users to change the version between ADS and RVDS.

<http://www.arm.com/support/downloads/>

2) Application note: Application note 171 (Migrating from ADS to RVDS 3.0)

3) FAQ: Linker error (8 byte stack alignment error)

This error occurs when a stack alignment conflict is detected in object code. Usually, error messages appear.

- Linker error: L6238E: foo.o(.text) contains invalid call from '~PRES8' function to 'REQ8' function foobar

- Warning: L6306W: 'PRES8' section foo.o(.text) should not use the address of 'REQ8' function foobar

<http://www.arm.com/support/faqdev/1242.html>

<http://www.arm.com/support/faqdev/14269.html>

4) FAQ: stack heap error

This error occurs when `__rt_entry()` function looks for routine that implements stack base and heap base. (`__rt_entry()` is the C library initialization code.)

<http://www.arm.com/support/faqdev/1247.html>

As for 3), it is very simple to have a solution. It is simply to add `PRESERVE8` at the beginning of startup code. (e.g. `csp_boot_ads.s`)

As for 4), it is originally caused because the compiler looks for "main" keyword at entry. CSP make use of `C_Entry()` name, instead.

If the compiler looks for "main", then `C_Entry` in startup code and main routine shall be changed as below. In addition, it requires stack and heap implementation as 4).

- In `csp_boot_ads.s`
  - Change `C_Entry` to `__main` in import part
    - ◆ `import C_Entry` → `import __main`
  - Change `C_Entry` to `__main` in branch part
    - ◆ `bl C_Entry` → `bl __main`
  - Exclude the part for remap (copy) code and data in RAM (call to `CSP_CInit`)
- In main function
  - Include `<rt_misc.h>`
  - Include implementation of `__user_initial_stackheap` (see example)

```
extern unsigned int Image$$SRAM_ZI$$ZI$$Limit;
__value_in_regs struct __initial_stackheap
```

```
__user_initial_stackheap(  
    unsigned R0, unsigned SP, unsigned R2, unsigned SL) {  
    struct __initial_stackheap config;  
    config.heap_base = (unsigned int)&Image$$SRAM_ZI$$ZI$$Limit;  
    config.stack_base = SP;  
    return config;  
}
```

- Add int main(void) function, in which it redirects to C\_Entry() function.

As for 4), the compiler would look for main keyword. However, it does not at times. In case it does not look for main keyword, using C\_Entry does not cause any problem.

## 6. Debugging / Flash Programming with RVD

Instead of AXD, RVD should be used with RVDS. This is because DWARF format of axf file obtained from ADS is different from that of RVDS. (RVDS supports DWARF 3 format)

To avoid this, RVD shall be used or compilation shall be done with an option "--dwarf2".

### **DEBUGGING AND FLASH PROGRAMMING**

Prior to debugging the ARM, it is necessary to define the microcontrollers by means of BCD file (Board/Chip Definition), in which the type and range of memories are declared. (e.g. RAM 0x300000 ~ 0x307FFF.)

In addition to BCD file, the FME (Flash Method) file shall be provided. It defines erase / write / verify protocols according to the flash memory.

In RVD, some of microcontrollers and boards are already included with BCD and FME files. However, S3F4A0K/A1H/A2J have not been supported in RVD so far. Therefore, BCD and FME file shall be created.

The ARM website and document describe how to build BCD and FME file. See:

- Application note 110: Flash Programming with RealView Debugger
- Chapter 5: Programming Flash with RealView Debugger in RVD Target Configuration Guide.)
- AN2007-02: Flash Programming with RVD

## 7. References

ARM website: <http://www.arm.com>

ARM Cross Development with Eclipse, J. P. Lynch

Application note 171: Migrating from ADS to RVDS3.0

Development support FAQ 1242

Development support FAQ: 14269

Development support FAQ: 1247

Application note 110: Flash programming with RealView Debugger

RVD Target Configuration Guide