Introduction to CodeSourcery G++ tool-chain

2016/10

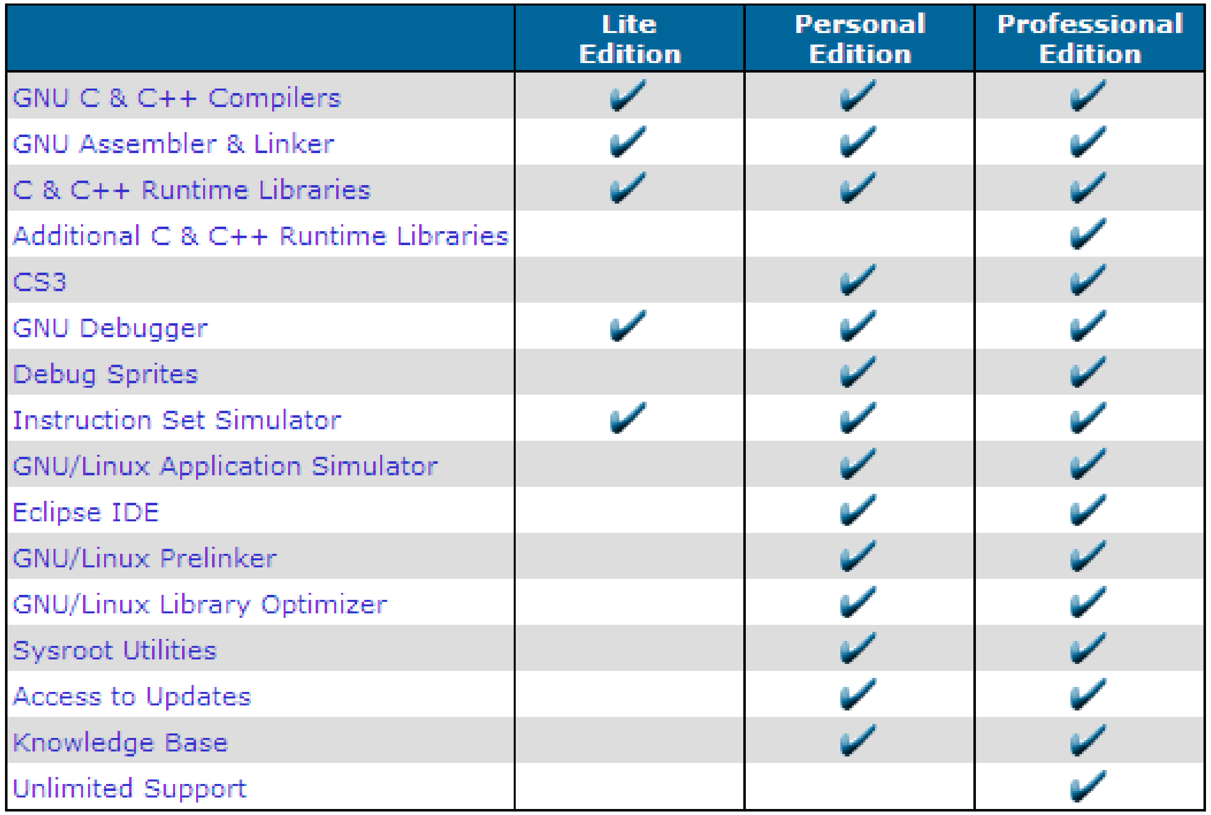
Outline

- What is the CodeSourcery G++ tool-chain
- How to get it
- How to install
- Basic for compilation
- Basic for simulation
- Basic for pure assembly

What is the CodeSourcery G++ tool-chain?

- The ARM tool-chain (GCC based)
- It’s free software
- CodeSourcery, in partnership with ARM, Ltd., develops improvements to the GNU Toolchain for ARM processors and provides regular, validated releases of the GNU Toolchain.

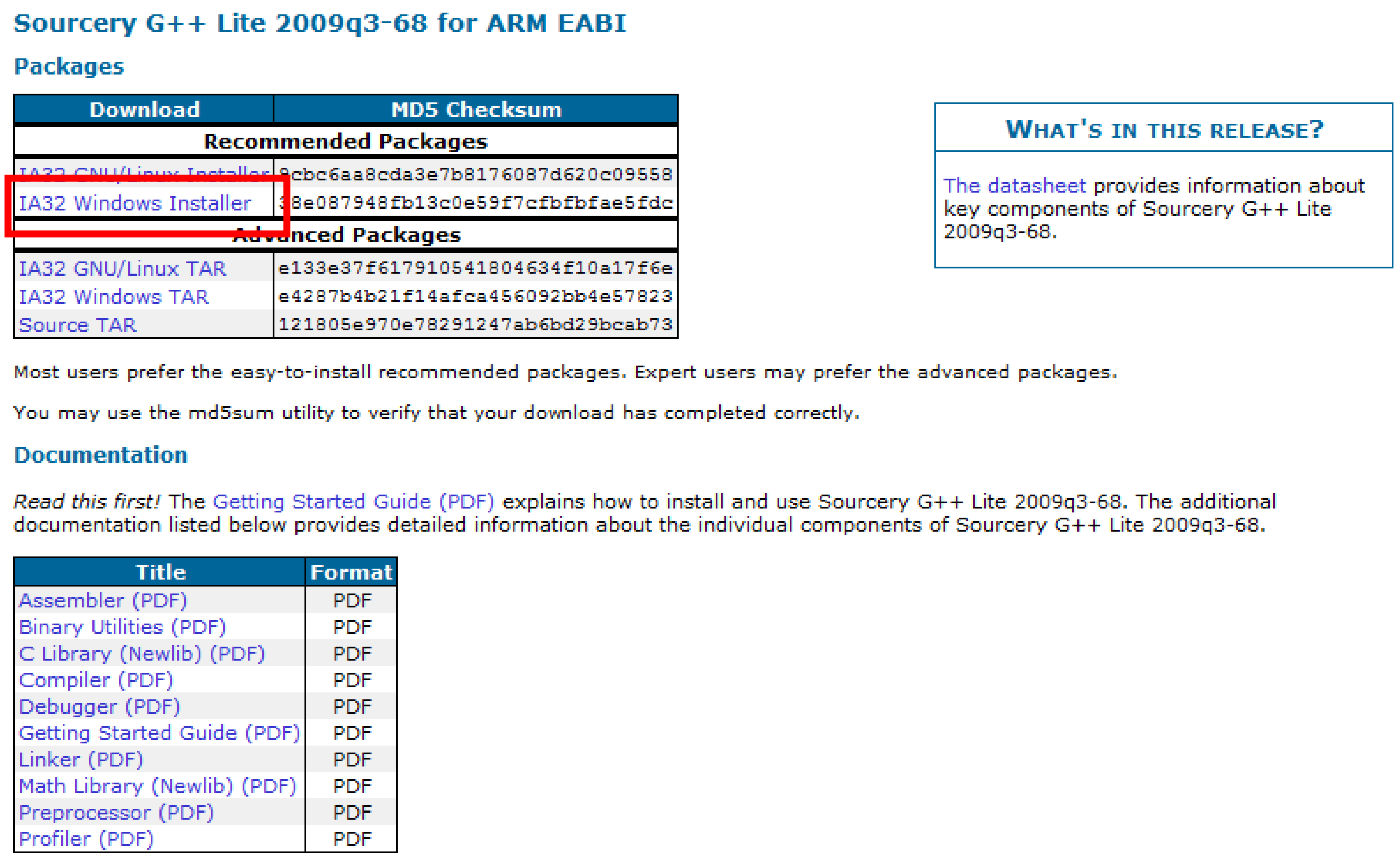
What is the CodeSourcery G++ tool-chain?



	free	$399	$2799
	Lite Edition	**Personal Edition**	**Professional Edition**
GNU C & C++ Compilers	✓	✓	✓
GNU Assembler & Linker	✓	✓	✓
C & C++ Runtime Libraries	✓	✓	✓
Additional C & C++ Runtime Libraries			✓
CS3		✓	✓
GNU Debugger	✓	✓	✓
Debug Sprites		✓	✓
Instruction Set Simulator	✓	✓	✓
GNU/Linux Application Simulator		✓	✓
Eclipse IDE		✓	✓
GNU/Linux Prelinker		✓	✓
GNU/Linux Library Optimizer		✓	✓
Sysroot Utilities		✓	✓
Access to Updates		✓	✓
Knowledge Base		✓	✓
Unlimited Support			✓

How to get it

- http://www.codesourcery.com/sgpp/lite/arm/portal/release1033



Sourcery G++ Lite 2009q3-68 for ARM EABI

Packages

Download	MD5 Checksum
Recommended Packages	
IA32 GNU/Linux Installer	9cbc6aa8cda3e7b8176087d620c09558
IA32 Windows Installer	38e087948fb13c0e59f7cfbfbfae5fdc
Advanced Packages	
IA32 GNU/Linux TAR	e133e37f617910541804634f10a17f6e
IA32 Windows TAR	e4287b4b21f14afca456092bb4e57823
Source TAR	121805e970e78291247ab6bd29bcab73

Most users prefer the easy-to-install recommended packages. Expert users may prefer the advanced packages.

You may use the md5sum utility to verify that your download has completed correctly.

WHAT'S IN THIS RELEASE?

The datasheet provides information about key components of Sourcery G++ Lite 2009q3-68.

Documentation

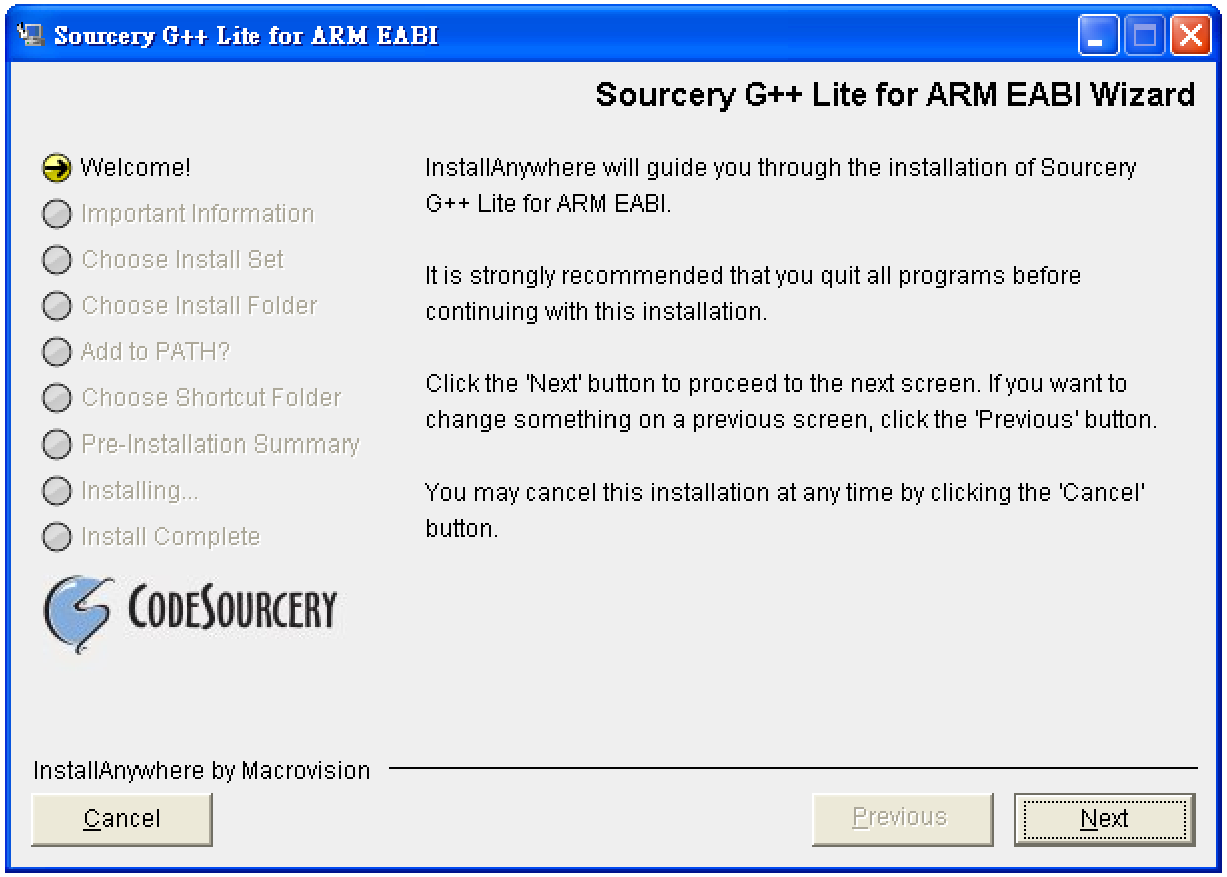
Read this first! The Getting Started Guide (PDF) explains how to install and use Sourcery G++ Lite 2009q3-68. The additional documentation listed below provides detailed information about the individual components of Sourcery G++ Lite 2009q3-68.

Title	Format
Assembler (PDF)	PDF
Binary Utilities (PDF)	PDF
C Library (Newlib) (PDF)	PDF
Compiler (PDF)	PDF
Debugger (PDF)	PDF
Getting Started Guide (PDF)	PDF
Linker (PDF)	PDF
Math Library (Newlib) (PDF)	PDF
Preprocessor (PDF)	PDF
Profiler (PDF)	PDF

How to install

- You must be administrator.
- According to the following slides to install.

Step1



Sourcery G++ Lite for ARM EABI Wizard

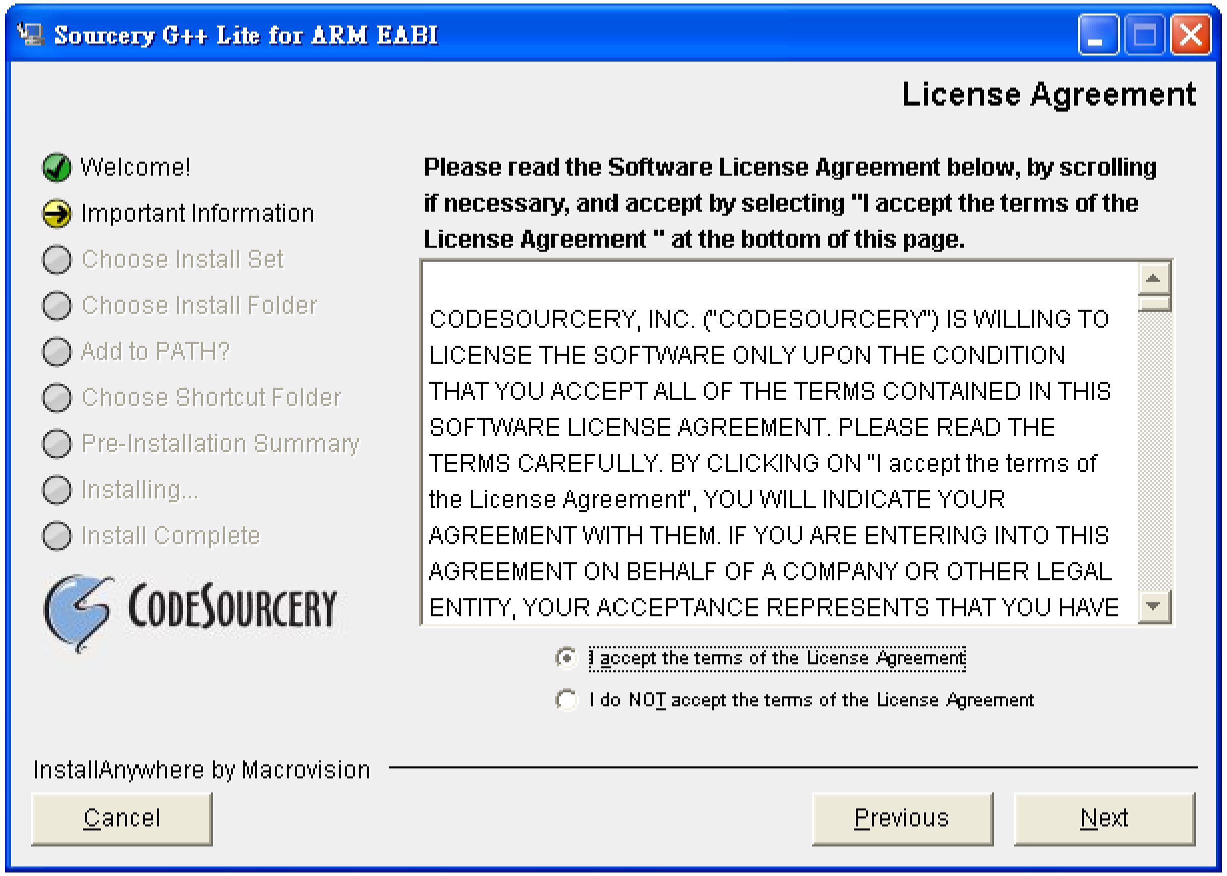
InstallAnywhere will guide you through the installation of Sourcery G++ Lite for ARM EABI.

It is strongly recommended that you quit all programs before continuing with this installation.

Click the 'Next' button to proceed to the next screen. If you want to change something on a previous screen, click the 'Previous' button.

You may cancel this installation at any time by clicking the 'Cancel' button.

Step2



Sourcery G++ Lite for ARM EABI

License Agreement

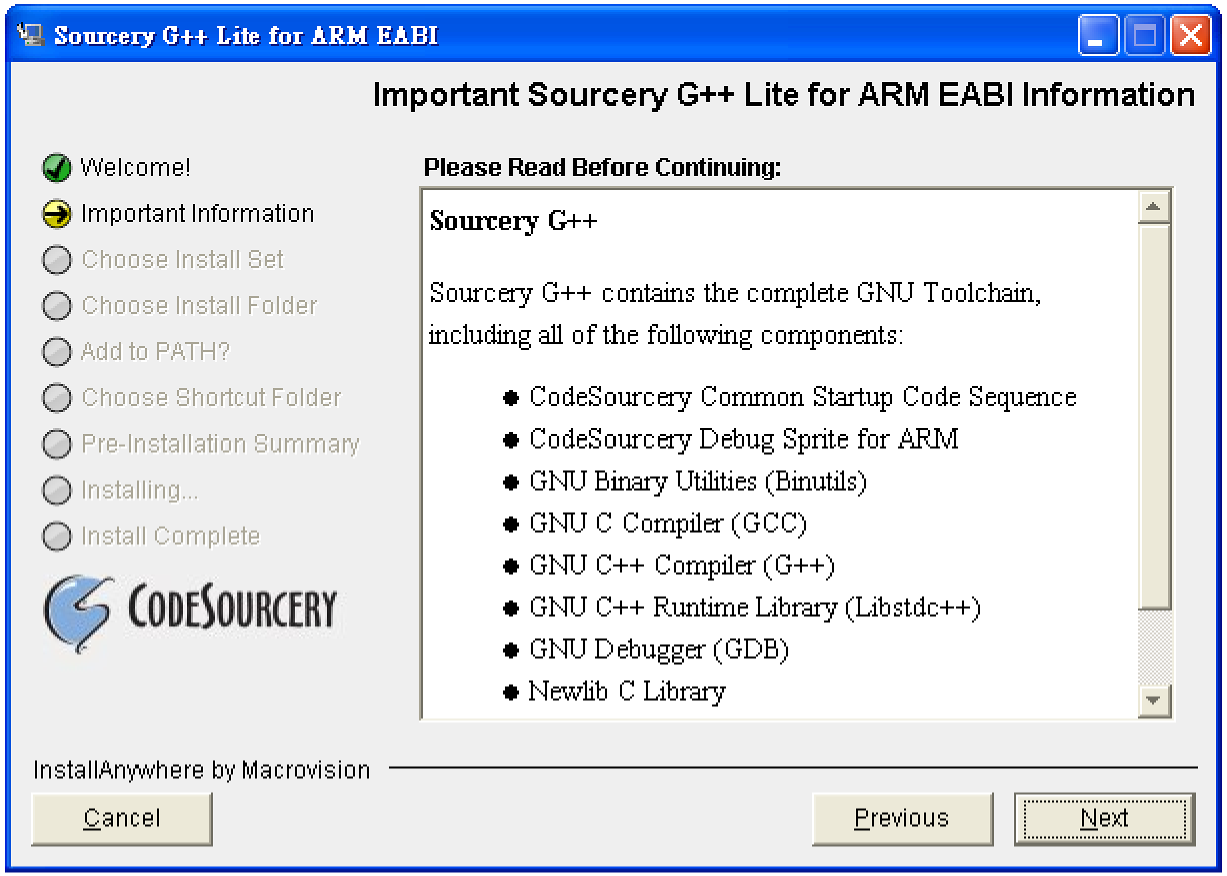
- ✓ Welcome!
- ➔ Important Information
- Choose Install Set
- Choose Install Folder
- Add to PATH?
- Choose Shortcut Folder
- Pre-Installation Summary
- Installing...
- Install Complete

Please read the Software License Agreement below, by scrolling if necessary, and accept by selecting "I accept the terms of the License Agreement " at the bottom of this page.

CODESOURCERY, INC. ("CODESOURCERY") IS WILLING TO LICENSE THE SOFTWARE ONLY UPON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS CONTAINED IN THIS SOFTWARE LICENSE AGREEMENT. PLEASE READ THE TERMS CAREFULLY. BY CLICKING ON "I accept the terms of the License Agreement", YOU WILL INDICATE YOUR AGREEMENT WITH THEM. IF YOU ARE ENTERING INTO THIS AGREEMENT ON BEHALF OF A COMPANY OR OTHER LEGAL ENTITY, YOUR ACCEPTANCE REPRESENTS THAT YOU HAVE

[✓ I accept the terms of the License Agreement]
I do NOT accept the terms of the License Agreement

Step3


Sourcery G++ Lite for ARM EABI

Important Sourcery G++ Lite for ARM EABI Information

Please Read Before Continuing:

Sourcery G++

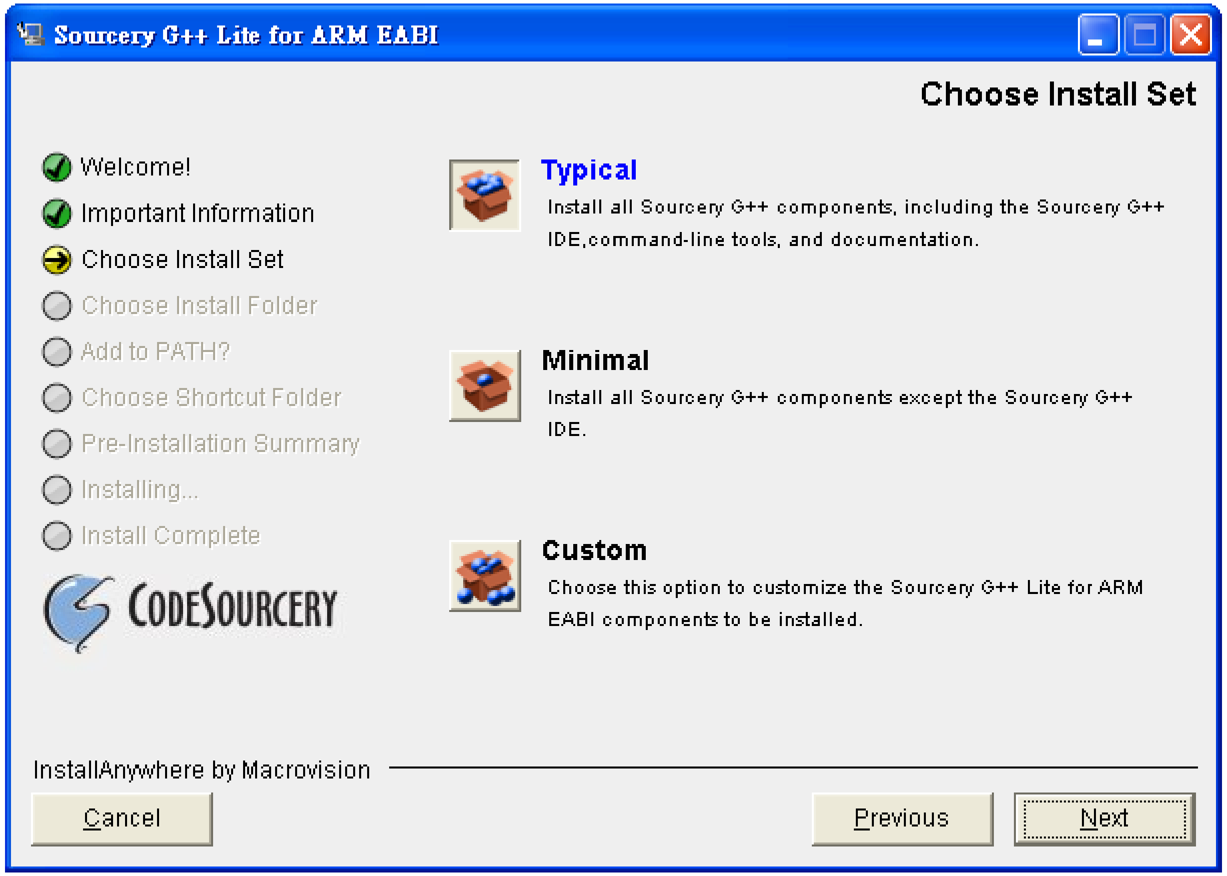
Sourcery G++ contains the complete GNU Toolchain, including all of the following components:

- CodeSourcery Common Startup Code Sequence
- CodeSourcery Debug Sprite for ARM
- GNU Binary Utilities (Binutils)
- GNU C Compiler (GCC)
- GNU C++ Compiler (G++)
- GNU C++ Runtime Library (Libstdc++)
- GNU Debugger (GDB)
- Newlib C Library

InstallAnywhere by Macrovision

Step4

- Choose Typical option



Sourcery G++ Lite for ARM EABI

Choose Install Set

- ✓ Welcome!
- ✓ Important Information
- ➔ Choose Install Set
- ○ Choose Install Folder
- ○ Add to PATH?
- ○ Choose Shortcut Folder
- ○ Pre-Installation Summary
- ○ Installing...
- ○ Install Complete

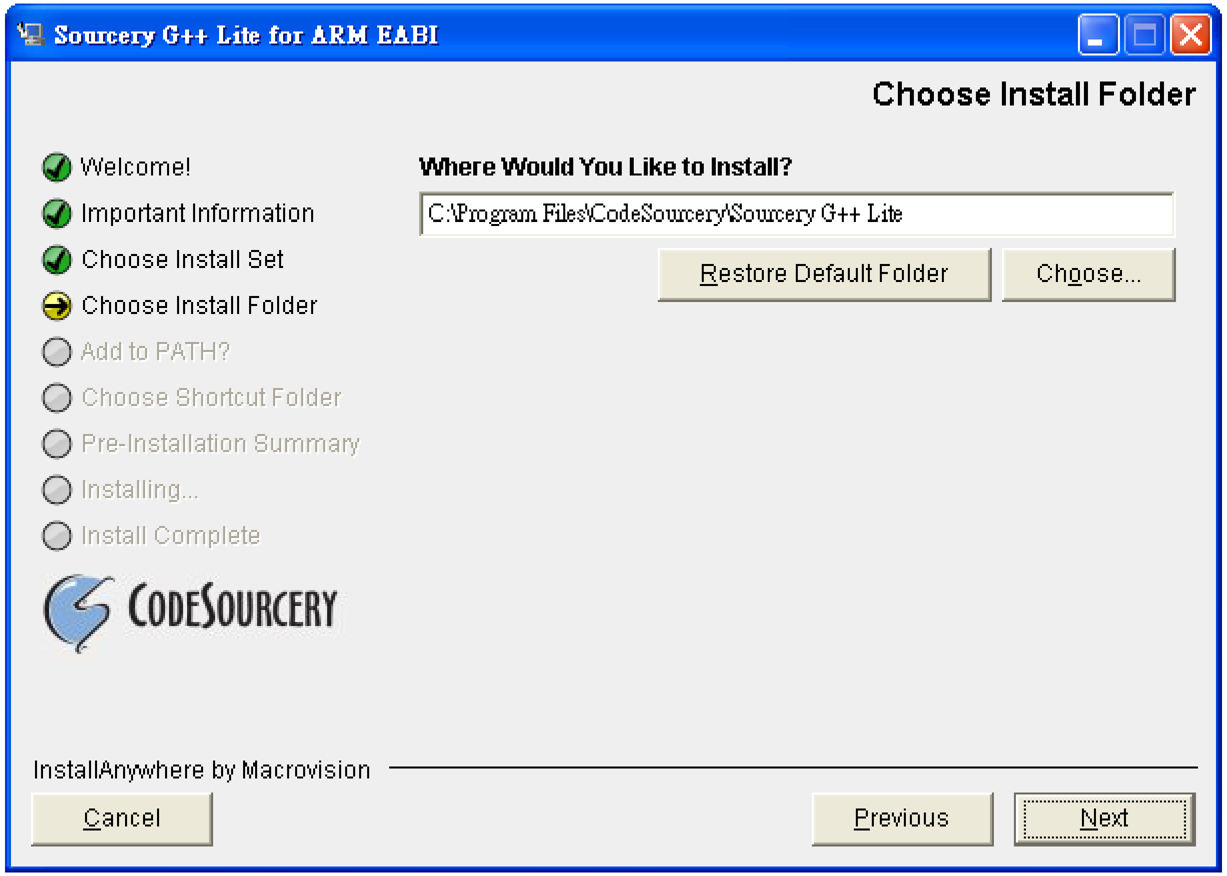
Typical
Install all Sourcery G++ components, including the Sourcery G++ IDE,command-line tools, and documentation.

Minimal
Install all Sourcery G++ components except the Sourcery G++ IDE.

Custom
Choose this option to customize the Sourcery G++ Lite for ARM EABI components to be installed.

InstallAnywhere by Macrovision

Step5


Sourcery G++ Lite for ARM EABI

Choose Install Folder

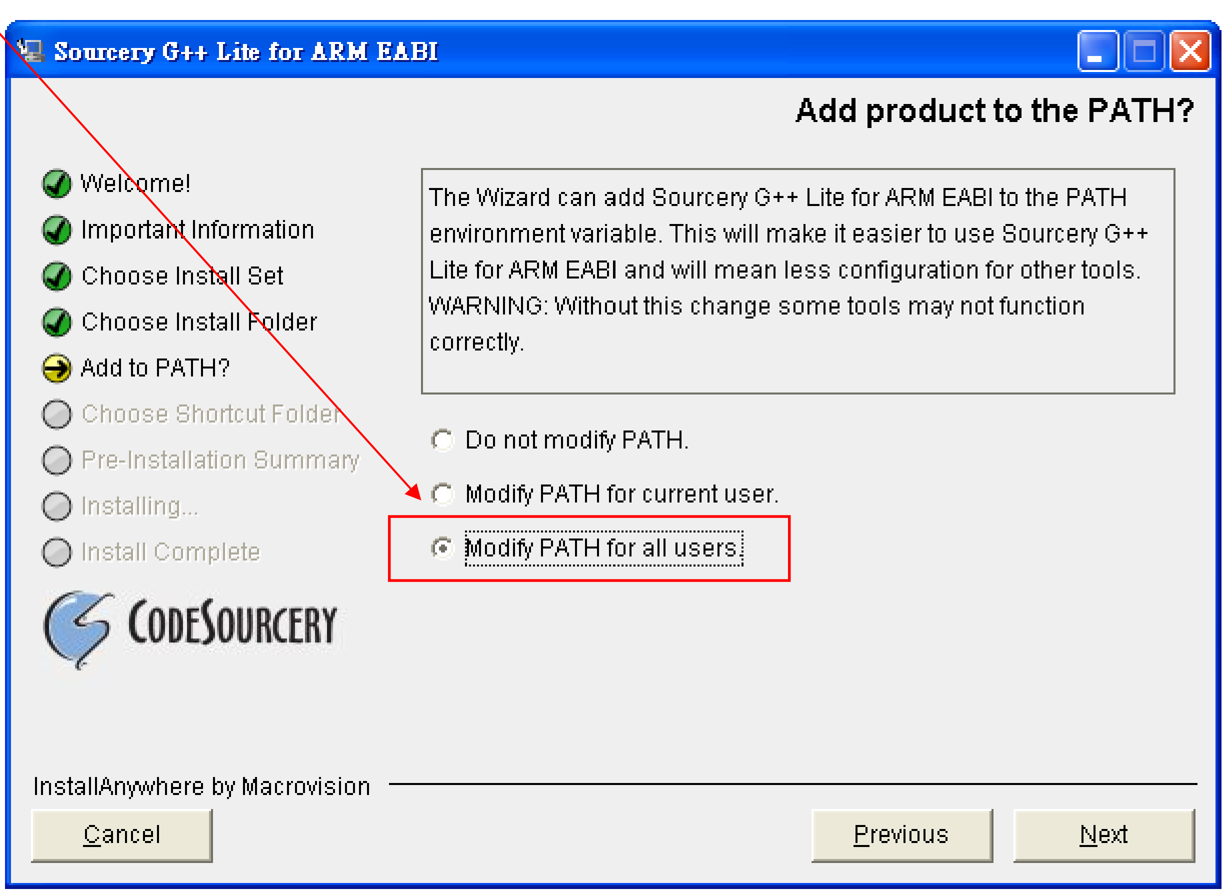
- ✓ Welcome!
- ✓ Important Information
- ✓ Choose Install Set
- ➔ Choose Install Folder
- ○ Add to PATH?
- ○ Choose Shortcut Folder
- ○ Pre-Installation Summary
- ○ Installing...
- ○ Install Complete

Where Would You Like to Install?

C:\Program Files\CodeSourcery\Sourcery G++ Lite

Step6

- Choose this


Sourcery G++ Lite for ARM EABI

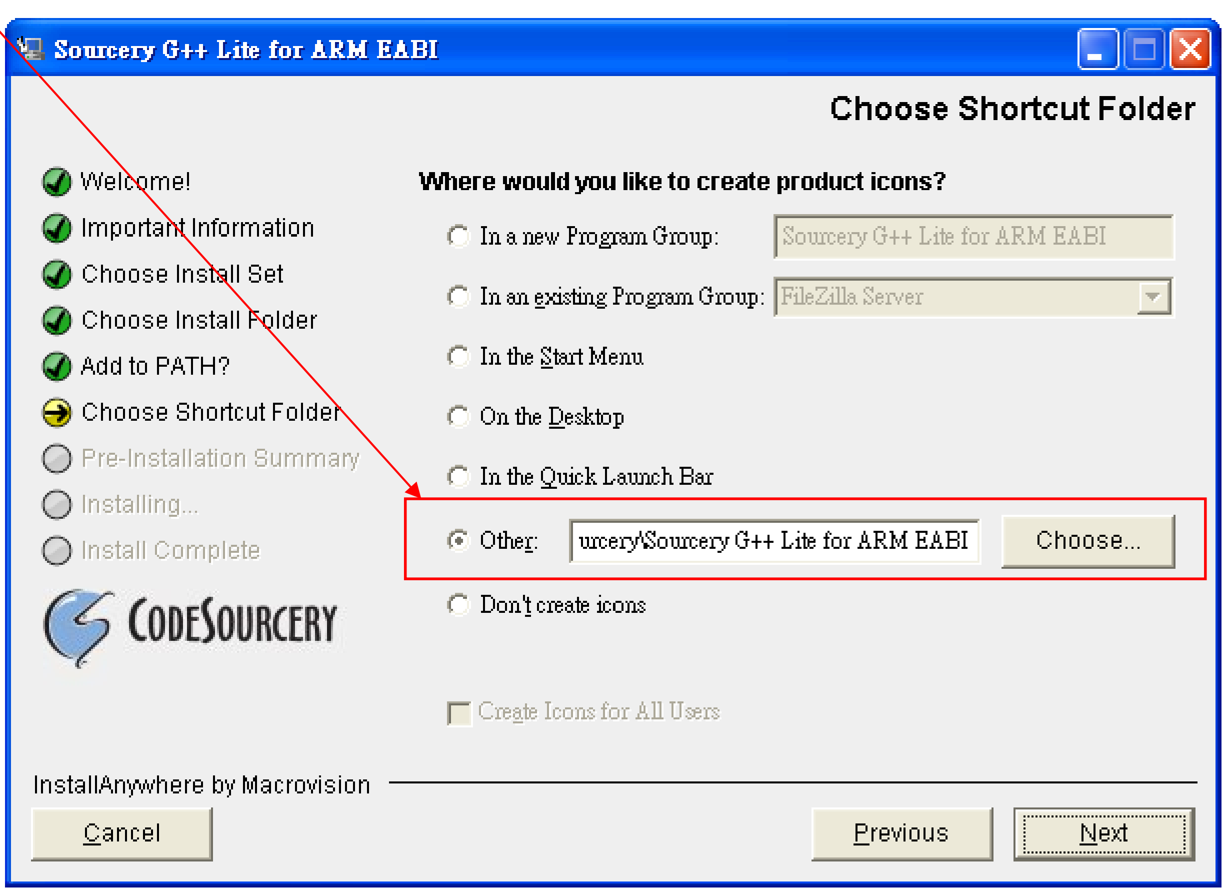
Add product to the PATH?

The Wizard can add Sourcery G++ Lite for ARM EABI to the PATH environment variable. This will make it easier to use Sourcery G++ Lite for ARM EABI and will mean less configuration for other tools. WARNING: Without this change some tools may not function correctly.

- Do not modify PATH.
- Modify PATH for current user.
- [✓ Modify PATH for all users.]

Step7

- Choose this



Sourcery G++ Lite for ARM EABI

Choose Shortcut Folder

Welcome!
Important Information
Choose Install Set
Choose Install Folder
Add to PATH?
Choose Shortcut Folder
Pre-Installation Summary
Installing...
Install Complete

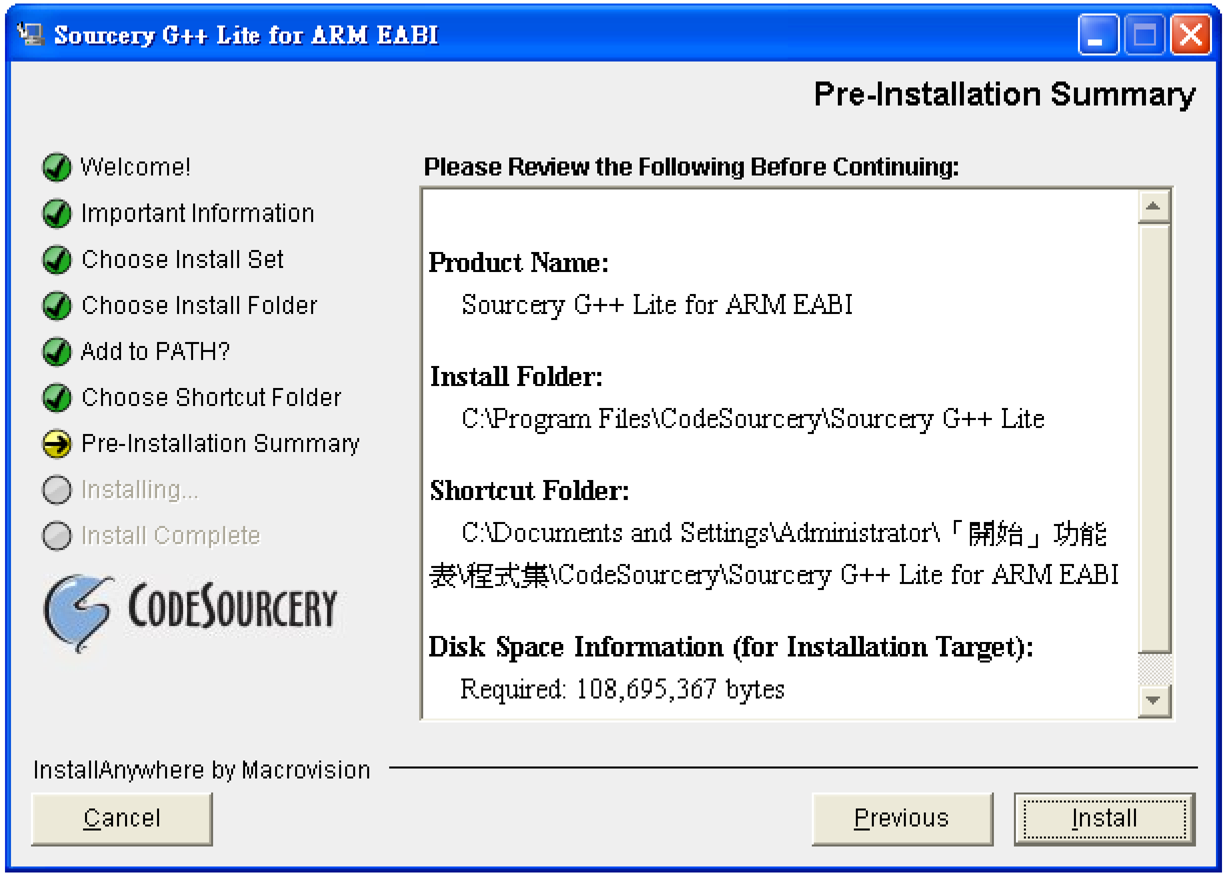
Where would you like to create product icons?

- In a new Program Group: Sourcery G++ Lite for ARM EABI
- In an existing Program Group: FileZilla Server
- In the Start Menu
- On the Desktop
- In the Quick Launch Bar
- [✓ Other:] urcery\Sourcery G++ Lite for ARM EABI
- Don't create icons

Create Icons for All Users

InstallAnywhere by Macrovision

Step8


Sourcery G++ Lite for ARM EABI

Pre-Installation Summary

- ✓ Welcome!
- ✓ Important Information
- ✓ Choose Install Set
- ✓ Choose Install Folder
- ✓ Add to PATH?
- ✓ Choose Shortcut Folder
- ➔ Pre-Installation Summary
- ○ Installing...
- ○ Install Complete

Please Review the Following Before Continuing:

Product Name:
Sourcery G++ Lite for ARM EABI

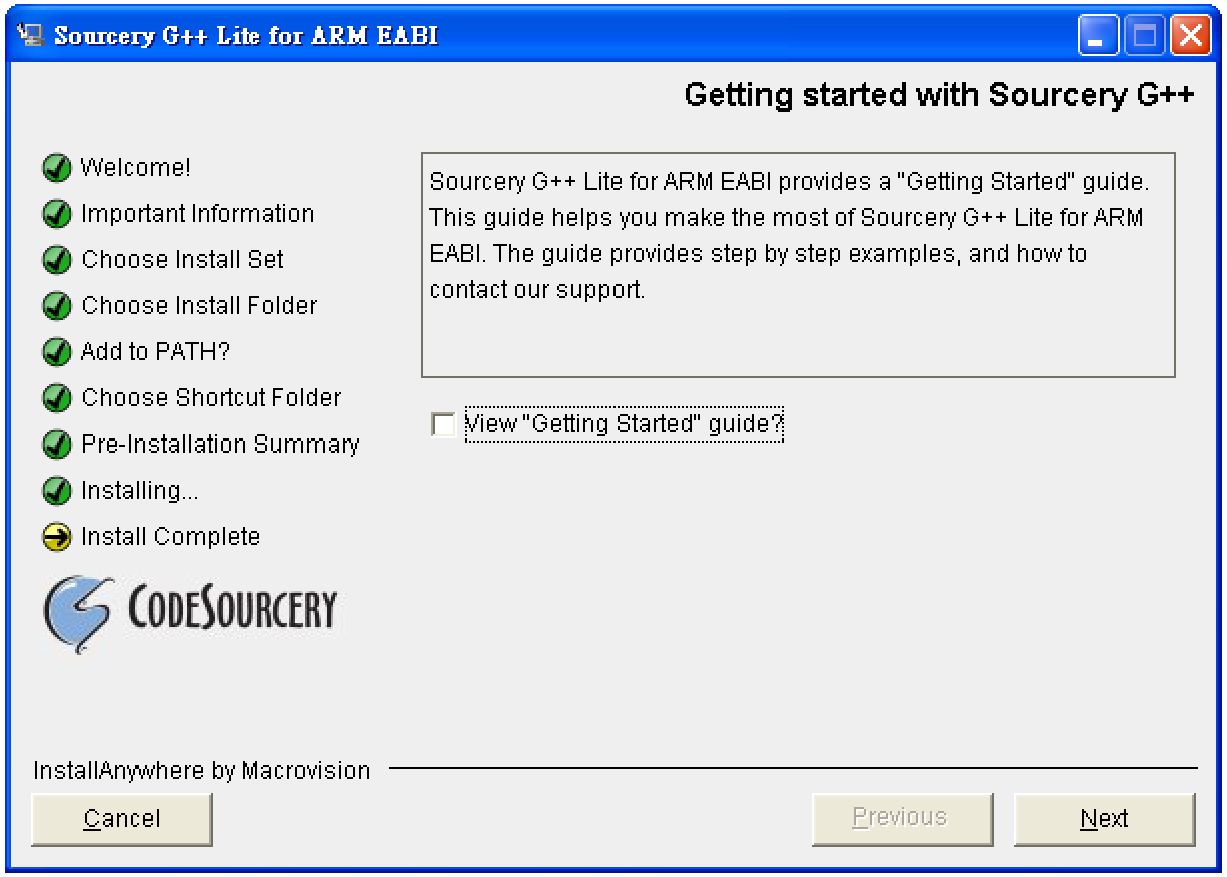
Install Folder:
C:\Program Files\CodeSourcery\Sourcery G++ Lite

Shortcut Folder:
C:\Documents and Settings\Administrator\「開始」功能表\程式集\CodeSourcery\Sourcery G++ Lite for ARM EABI

Disk Space Information (for Installation Target):
Required: 108,695,367 bytes

InstallAnywhere by Macrovision

Step9



Sourcery G++ Lite for ARM EABI

Getting started with Sourcery G++

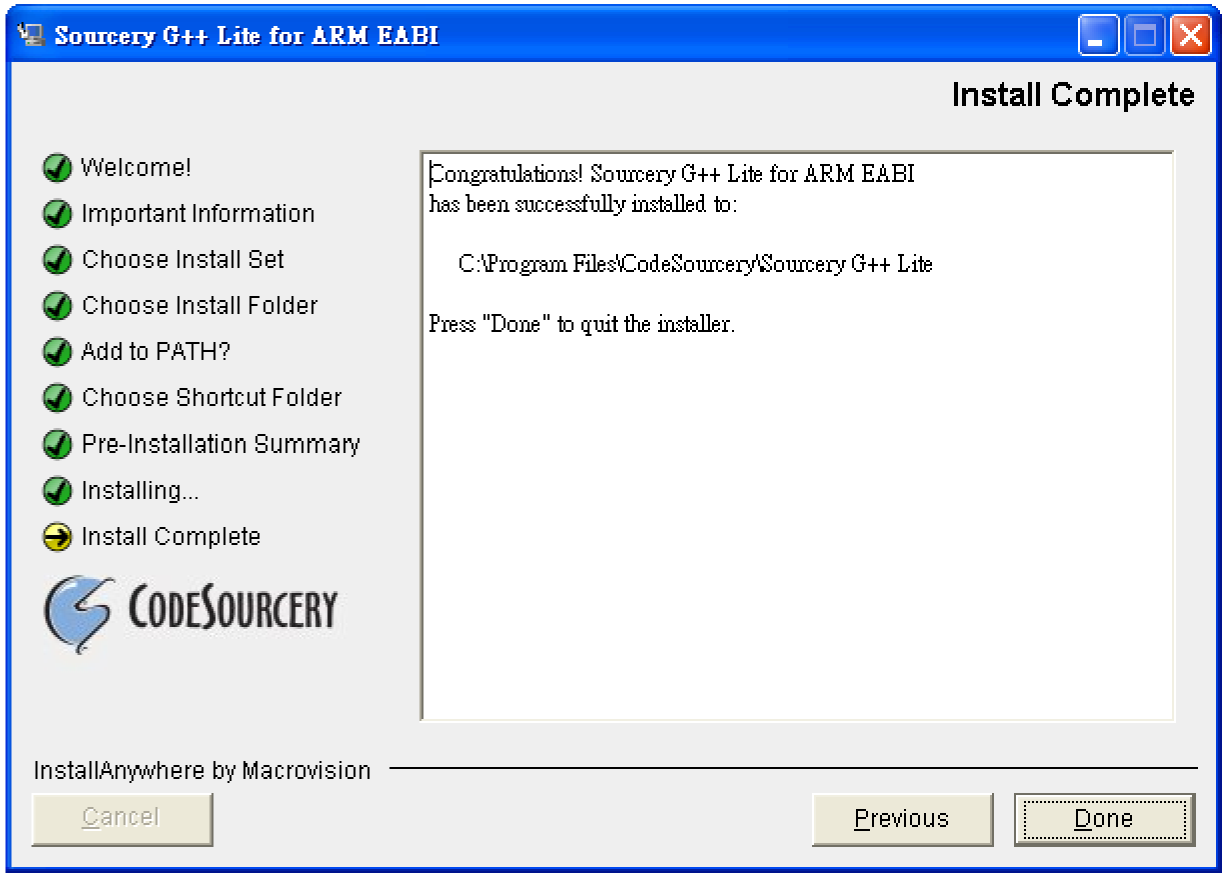
- ✓ Welcome!
- ✓ Important Information
- ✓ Choose Install Set
- ✓ Choose Install Folder
- ✓ Add to PATH?
- ✓ Choose Shortcut Folder
- ✓ Pre-Installation Summary
- ✓ Installing...
- ➔ Install Complete

Sourcery G++ Lite for ARM EABI provides a "Getting Started" guide. This guide helps you make the most of Sourcery G++ Lite for ARM EABI. The guide provides step by step examples, and how to contact our support.

☐ View "Getting Started" guide?

InstallAnywhere by Macrovision

Step10



Congratulations! Sourcery G++ Lite for ARM EABI
has been successfully installed to:

C:\Program Files\CodeSourcery\Sourcery G++ Lite

Press "Done" to quit the installer.

Step11

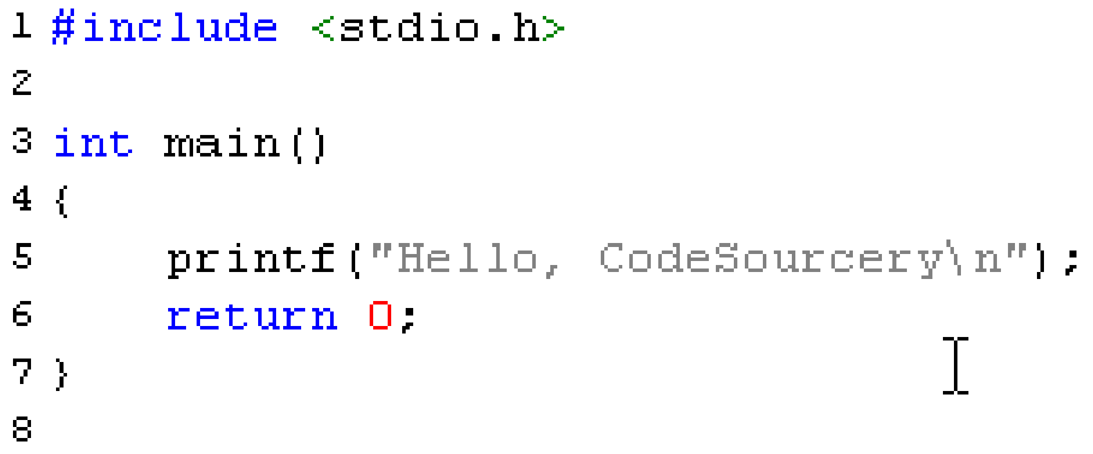
- reboot

Outline

- What is the CodeSourcery G++ tool-chain
- How to get it
- How to install
- Basic for compilation
- Basic for simulation
- Basic for pure assembly

Basic for compilation

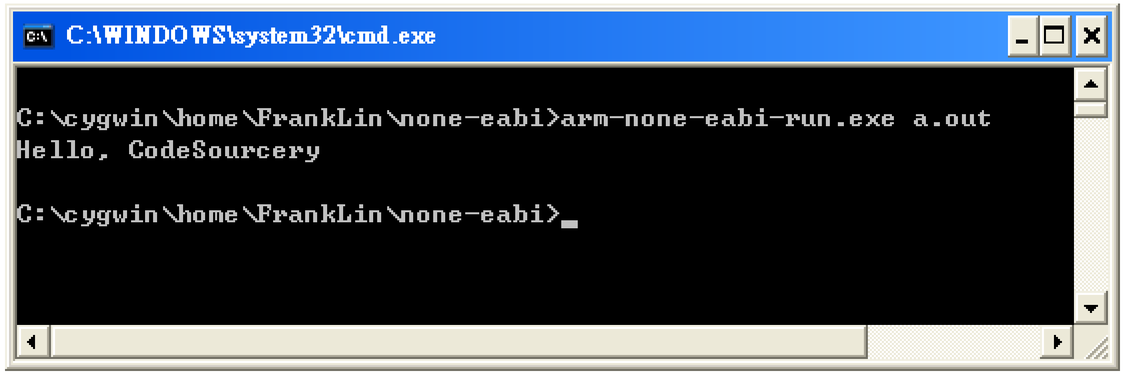
- arm-none-eabi-gcc.exe -T generic-hosted.ld test.c



```c
#include <stdio.h>

int main()
{
    printf("Hello, CodeSourcery\n");
    return 0;
}

```

Basic for simulation

- arm-none-eabi-run.exe a.out



```
C:\cygwin\home\FrankLin\none-eabi>arm-none-eabi-run.exe a.out
Hello, CodeSourcery

C:\cygwin\home\FrankLin\none-eabi>_
```

Outline

- What is the CodeSourcery G++ tool-chain
- How to get it
- How to install
- Basic for compilation
- Basic for simulation
- Basic for pure assembly

Outline

- What is the CodeSourcery G++ tool-chain
- How to get it
- How to install
- Basic for compilation
- Basic for simulation
- Basic for pure assembly

Introduction to GAS for arm

- GAS is Gnu ASsembler
- The following slides would introduce how to write a pure assembly code in CodeSourcery G++.

The assembly example

- Please refer to example.S
- You can use this file as the skeleton and debug environment of your homework.

assemble and run

- assemble
  - arm-none-eabi-gcc.exe -T generic-hosted.ld example.S
- run
  - arm-none-eabi-run.exe a.out

Basic assembly language syntax

- label: instruction ; comment

- Loadin a constant to register
  - ldr ri, =0x12345678
  - ldr ri, =symbol

Some useful assembler directives

- .align
- .global
- string
- number
- comment
  - @this is a comment
- label
  - LABEL0:

.align

- Pad the location counter (in the current subsection) to a particular storage boundary
- It is aligned power of 2
- For example, aligned 4 byte
  - .align 2

.global

- Make the symbol visible to ld
- At least we must have a global symbol called “main” because we use the “generic-hosted.ld” for our linker script.

How to define a string

- .ascii
  - It assembles each string (with no automatic trailing zero byte) into consecutive addresses.
  - for example: .ascii “Hello world\n\0”
- .asciz
  - .asciz is just like .ascii, but each string is followed by a zero byte.
  - for example: .asciz “Hello world\n”

How to define a number

- .byte
- .short
- .word
- Multiple number is separated by comma
  - for example: .byte 0x31, 0x32, 0x33, 0x34
- If you don't use .align, assembler would compact each number.

How to use a constant in expressions

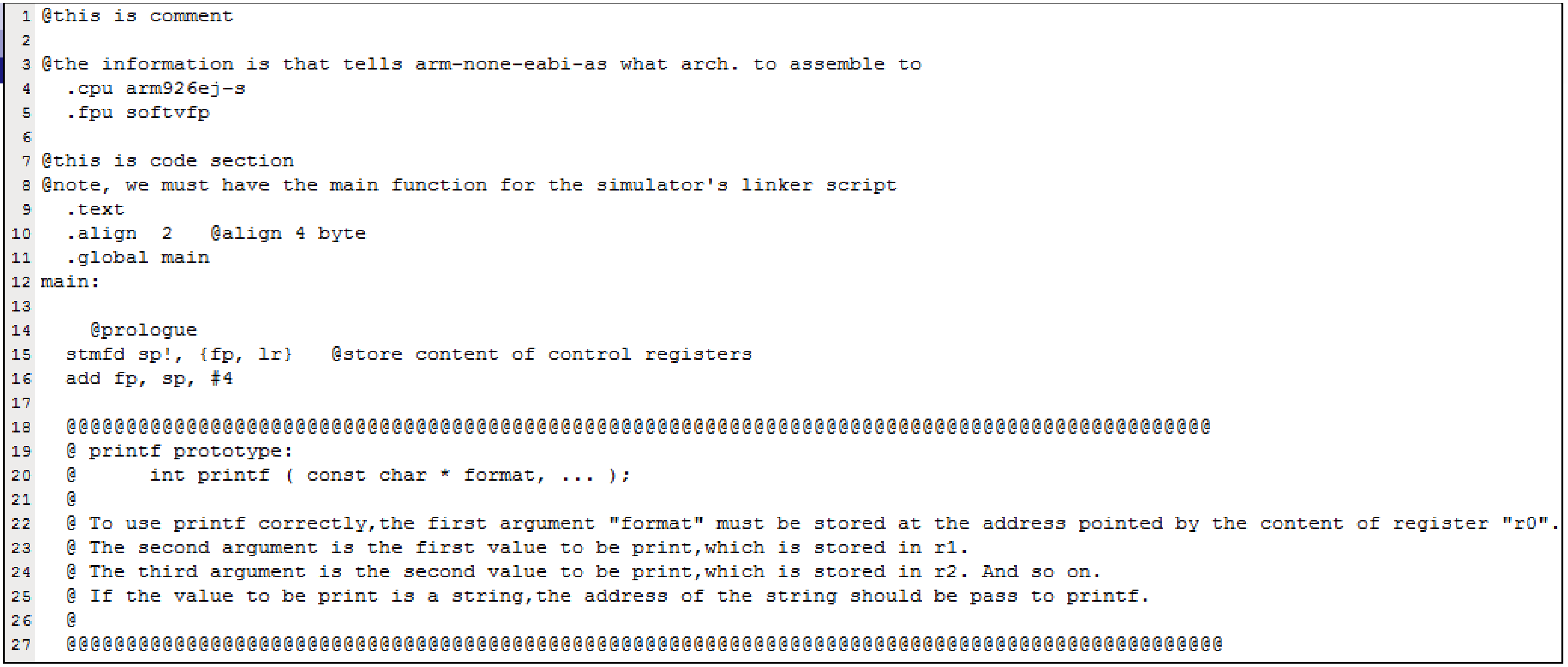
- mov r0,#74 ;
  - decimal number 74
- mov r0,#0x4A ;
  - hexadecimal number 0x4A (0X4A and 0x4a are also OK)
- mov r0,#0112 ;
  - octal number 0112 (leading ‘0’)
- mov r0,#0b1001010 ;
  - binary number 0b1001010 (0B1001010 is also OK)
- mov r0,#'J' ;
  - character constant "J" (preferred syntax)
- mov r0,#'J ;
  - character constant "J" (alternative syntax)

Misc

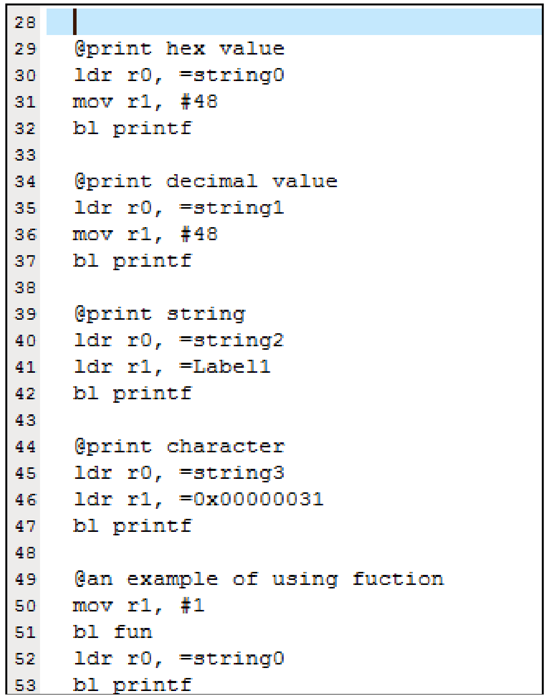
- .set symbol, expression
  - .set CONST, (5*8)+2
  - .equ CONST, 0x2A
  - CONST = 0b00101010
- <register_name> .reg <register_name>
  - acc .reg r0;
  - add acc, r2 , #2
- .space <number_of_bytes> {, <fill_byte>}
  - .space 30

Reference

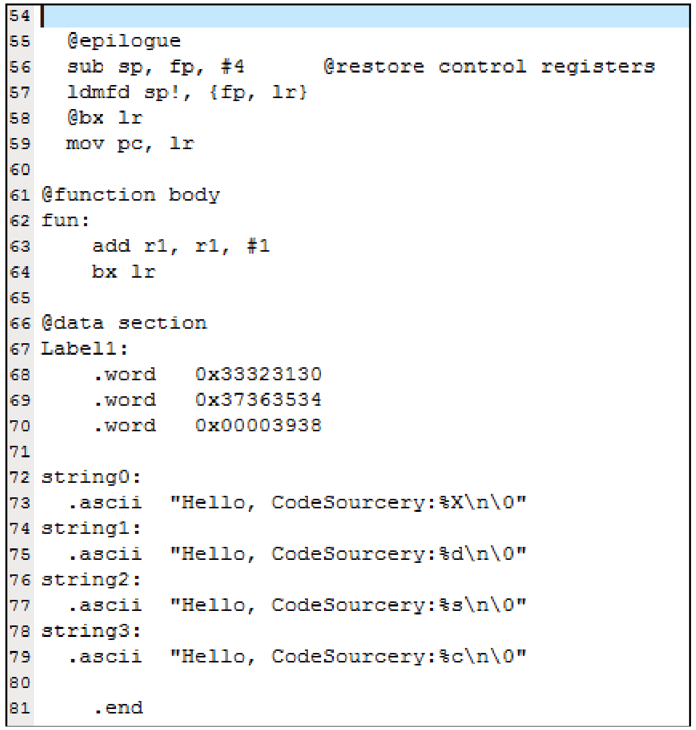
- http://www.codesourcery.com/sgpp
- The as manual of CodeSourcery G++



```
@this is comment

@the information is that tells arm-none-eabi-as what arch. to assemble to
  .cpu arm926ej-s
  .fpu softvfp

@this is code section
@note, we must have the main function for the simulator's linker script
  .text
  .align  2   @align 4 byte
  .global main
main:

    @prologue
  stmfd sp!, {fp, lr}    @store content of control registers
  add fp, sp, #4

  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  @ printf prototype:
  @      int printf ( const char * format, ... );
  @
  @ To use printf correctly,the first argument "format" must be stored at the address pointed by the content of register "r0".
  @ The second argument is the first value to be print,which is stored in r1.
  @ The third argument is the second value to be print,which is stored in r2. And so on.
  @ If the value to be print is a string,the address of the string should be pass to printf.
  @
  @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

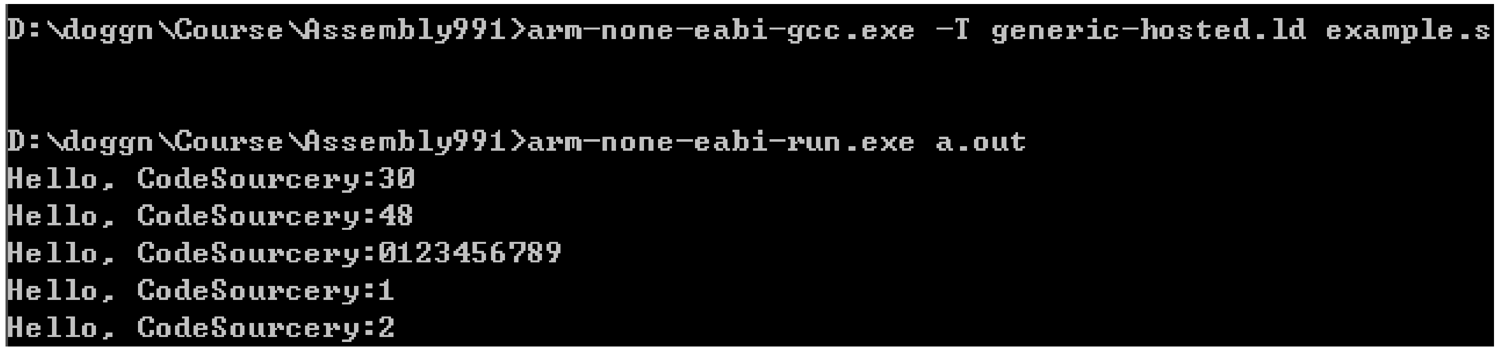


```

  @print hex value
  ldr r0, =string0
  mov r1, #48
  bl printf

  @print decimal value
  ldr r0, =string1
  mov r1, #48
  bl printf

  @print string
  ldr r0, =string2
  ldr r1, =Label1
  bl printf

  @print character
  ldr r0, =string3
  ldr r1, =0x00000031
  bl printf

  @an example of using fuction
  mov r1, #1
  bl fun
  ldr r0, =string0
  bl printf
```



```

  @epilogue
  sub sp, fp, #4      @restore control registers
  ldmfd sp!, {fp, lr}
  @bx lr
  mov pc, lr

@function body
fun:
    add r1, r1, #1
    bx lr

@data section
Label1:
    .word   0x33323130
    .word   0x37363534
    .word   0x00003938

string0:
  .ascii  "Hello, CodeSourcery:%X\n\0"
string1:
  .ascii  "Hello, CodeSourcery:%d\n\0"
string2:
  .ascii  "Hello, CodeSourcery:%s\n\0"
string3:
  .ascii  "Hello, CodeSourcery:%c\n\0"

    .end
```

Execution result



```
D:\doggn\Course\Assembly991>arm-none-eabi-gcc.exe -T generic-hosted.ld example.s


D:\doggn\Course\Assembly991>arm-none-eabi-run.exe a.out
Hello, CodeSourcery:30
Hello, CodeSourcery:48
Hello, CodeSourcery:0123456789
Hello, CodeSourcery:1
Hello, CodeSourcery:2
```