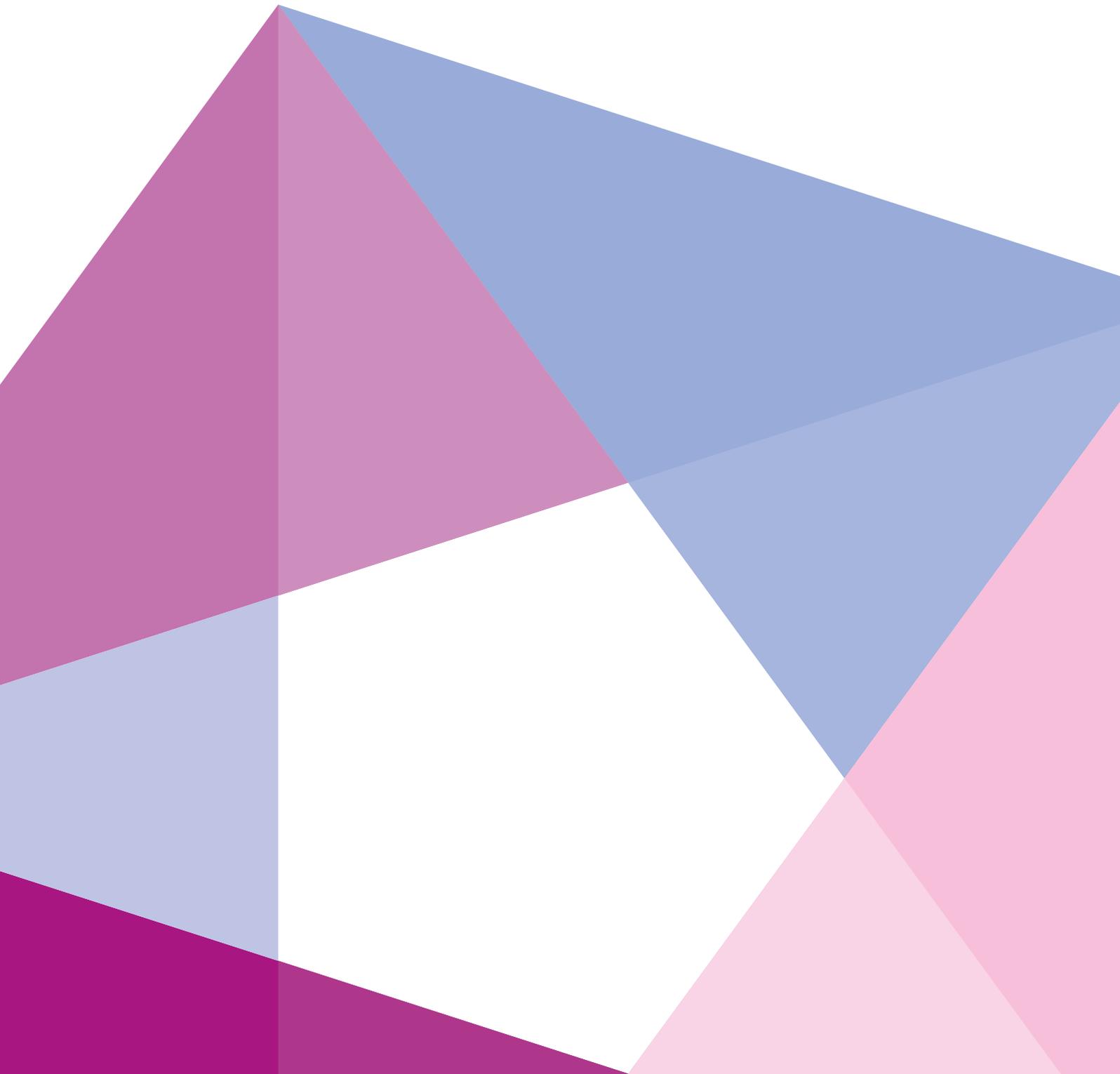




Using Eclipse with Cavium Octeon SDK



Using Eclipse with Cavium Octeon SDK

So, you finally persuaded your boss to buy this SDK from Cavium to program one of their magnificent Octeon processors. But after installing the SDK, all you get is a blinking cursor in a terminal session and you're back in the old days of using crappy editors to generate C-code, typing 'make' on the command line and parsing the output for errors. Don't even try to think of programming in C++ and using the advantages of object oriented programming; all you get from Cavium is flat, hard to maintain C. We are living in the twenty-first century, not the early seventies of the past millennium! Do I have to pay all those dollars for this? No way!

In this application note, we will first install Eclipse on your Linux system followed by building the Cavium passthrough example using Eclipse.

Installing Eclipse on your Linux system

We assume you have some knowledge about Linux and have the graphical X Window System up and running. You have installed the Octeon SDK and you have setup the environment as instructed. Verify that your able to build the passthrough example in the old fashion way (by typing **make** on the command line).

This tutorial has been verified with 64 bit Linux Centos 7.0 and Octeon SDK3.1 but will probably work with other versions of Linux and SDK, maybe with some small modifications. We assume you are a normal user (not root) when installing Eclipse. And you probably know that if you run into problems regarding installing things on Linux, Google is your best friend. Remember that most problems with Linux are caused by read and write permission issues.

At the time of writing (February 2015) the latest version of Eclipse is 4.4.1 (Luna). Because Eclipse is Java based, you should have Java installed on your system. Java 1.7 is required to run Eclipse Luna. You can check your current version by typing in a terminal window

java -version

This should reveal your current java version. To update it to the latest version, type

yum install jre

Next step is to download the latest Eclipse version from www.eclipse.org in the download section. Choose the 64 bit Linux *Eclipse IDE for C/C++ Developers* version and download it. Choose to open it using the Archive Manager after the file was downloaded (see figure 1).

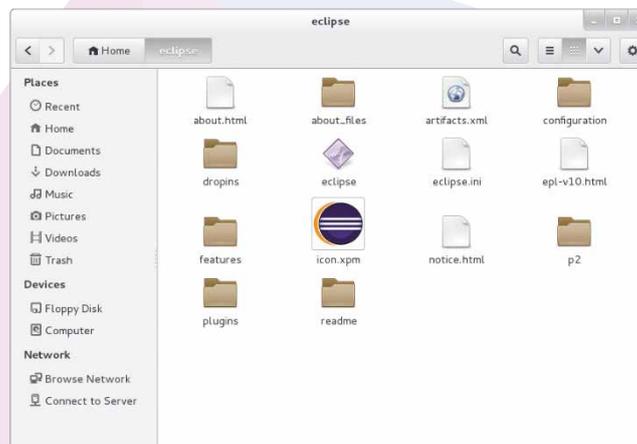


Figure 1: Directory containing Eclipse



Using the Archive Manager, extract the content to the user home directory (*/home/<user>*). After extracting there should be a directory named *eclipse* in the user home directory.

To make a shortcut from your desktop, right-click the eclipse icon and select *Make Link*. Now right-click the newly created "Link to eclipse" icon and select *Move To....* Select the desktop directory to place the shortcut on the desktop. Start Eclipse using the shortcut.

You should see the Eclipse intro screen and a little later you are asked where Eclipse should place its workspace directory. The default choice (*/home/<user>/workspace*) is not a bad one so you can keep this. Press **OK**.

Click the Workbench icon on the top-right. You are now ready to use eclipse.

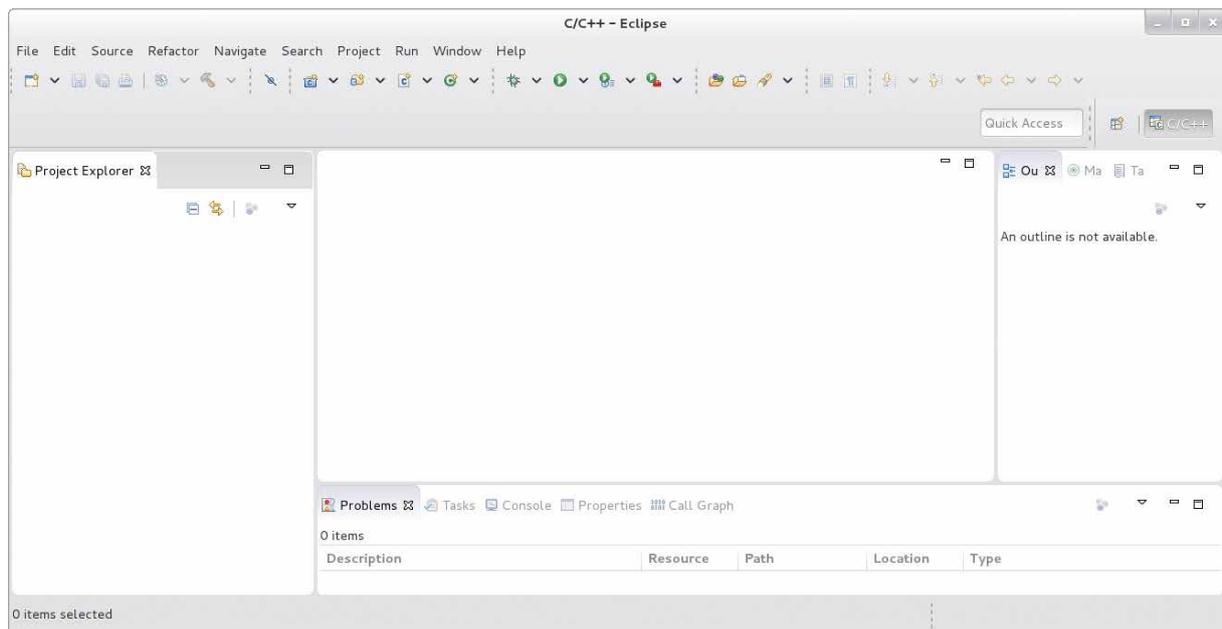


Figure 2: First start of Eclipse

Build passthrough example using Eclipse

We should now configure the Eclipse environment so it can handle Cavium C-projects. We will use the passthrough example that comes with the SDK. First, using a terminal session or a file browser, make a copy of the passthrough example into your workspace. Use the following command

```
cp -Rf $OCTEON_ROOT/examples/passthrough ~/workspace/passthrough
```

Open Eclipse and create a new C-project using *File->New->C-project*. Click *OK* and choose an empty Makefile project and select *--Other Toolchain* – under *Toolchains*. Choose *passthrough* as the project name (see figure 3). Press *Finish*.



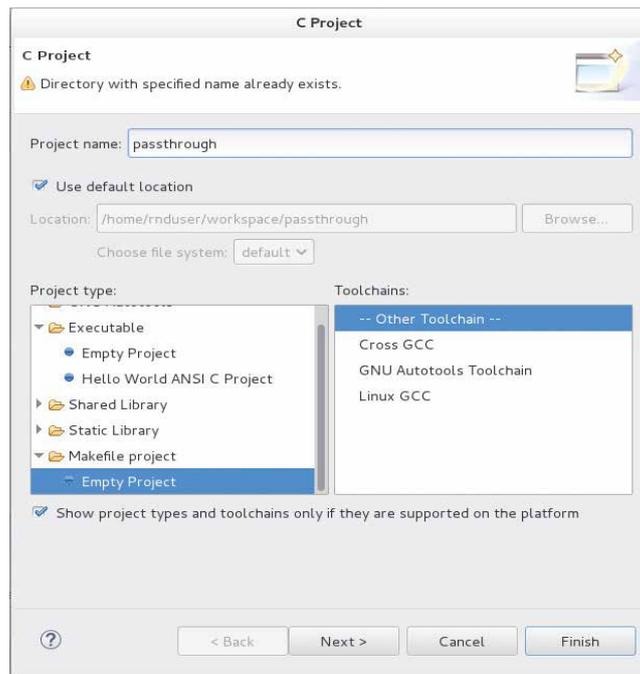


Figure 3: Definition of the passthrough project under Eclipse

Your project definition should now be visible at the left top side of Eclipse in the Project Explorer. Right click your project and select *Build Project*. The passthrough example should build without any problems. We're almost there.

Expand the project by double clicking it in the Project Explorer and double click on the *passthrough.c* file. The C-file should become visible in the source window. Double click the filename *passthrough.c* in the source window to expand the source file. It will now almost completely fill the Eclipse workspace. Notice the question marks in a yellow box in front of the inclusion of the header files. When you hover the mouse over these question marks, the cause of this warning is shown: *unresolved inclusion*. Let's do something about that.

Open up the projects' properties (*Project->Properties*) and navigate to the *Paths and Symbols* of the *C/C++ General* item. Select the *Includes* tab and select *GNU C*. To add two include paths here, press *Add* and select *File System* in the box that pops up. Select the directory */usr/local/Cavium_Networks/OCTEON_SDK/executive* and press *OK*. In the same way, add the path */usr/local/Cavium_Networks/OCTEON_SDK/tools/mipsisa64-octeon-elf/include* (See figure 4).

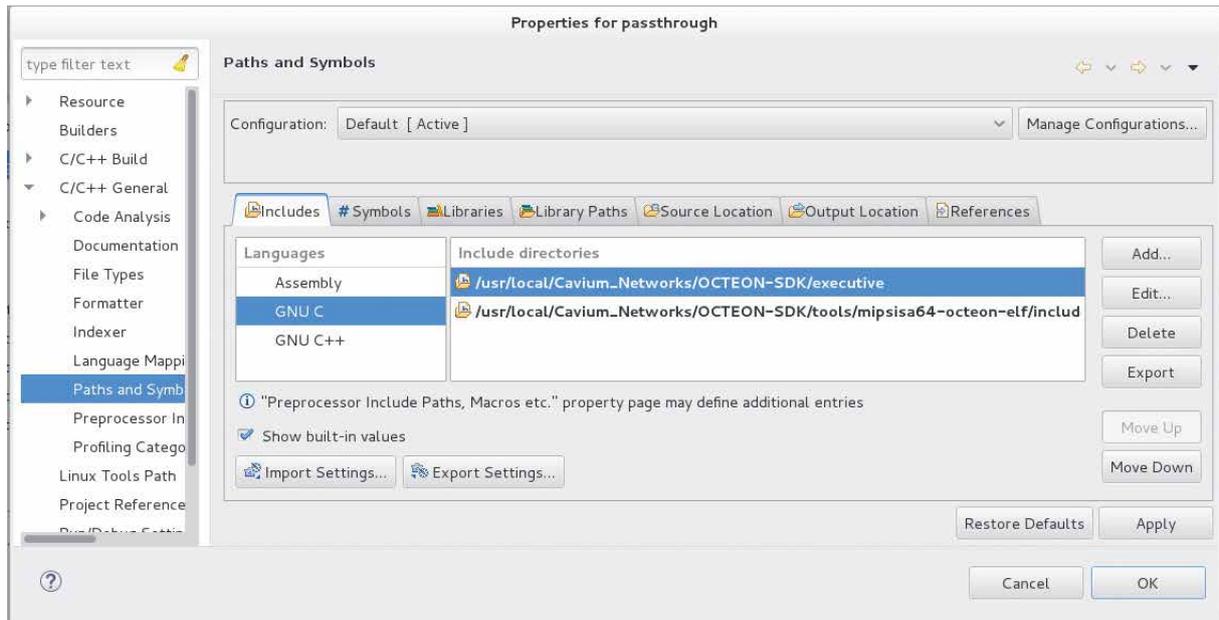


Figure 4: Add some include path to tackle warnings

Now select the *Symbols* tab and select *GNU C*. Now click *Add* and use *OCTEON_MODEL* for the name and *OCTEON_CN58XX* for the value. When using another device change it accordingly. Press *OK*.

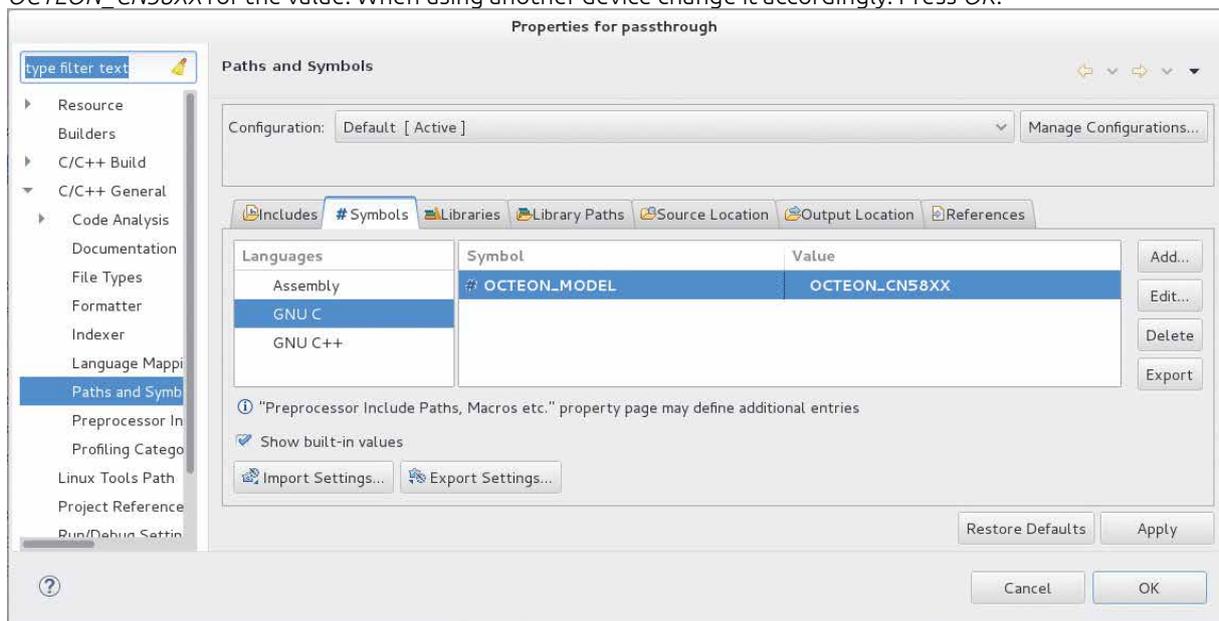


Figure 5: Add symbol to tackle warnings

Press *OK* and enter *Yes* when you are asked to rebuild the project. The warnings in the main window of Eclipse should be gone now.

Powered by Phact

Sirius is a TCP/UDP/IP protocol stack and is Phact's solution of daily handling millions of TCP and SSL connections on OCTEON hardware. Sirius is available as source code or as a library build for your OCTEON platform. It runs as Simple Exec or on OCTEON-Linux.

Phact is a one stop shop for software and hardware solutions, support and consultancy and was founded in 2015 with 25 years of experience in the business.

How? Phact originated in VSN Systemen. VSN was founded in 1991. The last couple of years, the company grew and evolved tremendously. It was time to adjust the look and feel of the company to what it had become. That's why Phact was born.

We started developing telecom hardware and software for VoIP- and SS7-carriers in 1991. For them, our engineers developed stable and professional hardware and software solutions. Additionally, we provided our customers with support and management of their solutions.

Knowledge is a key part of our business. Therefore, we are engaged in close partnerships with universities and our professional suppliers.

Phact might still be very young, but we carry years of experience in software and hardware development, support, management and consulting.



Phact BV

Keizersveld 83, 5803AP, Venray, The Netherlands

T +31 (0) 478555000 F +31 (0) 478589563

Email: info@phact.nl

www.phact.nl

www.sirius-networks.com

