Compiling and Debugging Basics

Compiling and Debugging Basics
2020/09/18 R17
1

QNX

NOTES:

QNX, Momentics, Neutrino, and "Build a more reliable world" are registered trademarks in certain jurisdictions, and Qnet is a trademark of QNX Software Systems.

All other trademarks and trade names belong to their respective owners.

This section is intended to give enough familiarity with the IDE to be able to do the exercises in the Realtime Programming and Drivers courses.

You will learn the basics of:

- Eclipse
- host-target development environment
- editing, compiling, running and debugging

Compiling and Debugging Basics
2020/09/18 R17
2

QNX

NOTES:

Compiling and Debugging Basics

Topics:

→ **Eclipse Basics**
   Targets
   Projects and Source
   Compiling
   Running and Debugging
   Versions
   Exercise
   Conclusion

Compiling and Debugging Basics
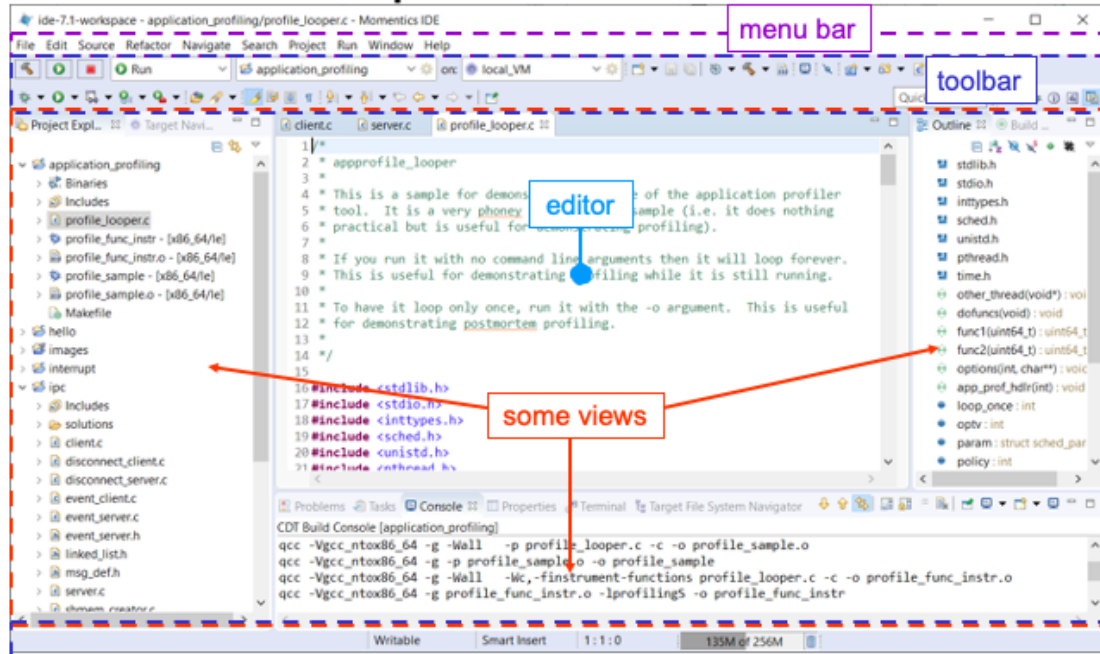2020/09/18 R17
3

QNX

NOTES:

The QNX Momentics IDE is based on Eclipse:

- an open source platform, written in Java, for building IDEs

- we inherit a lot of behavior and terminology from this:
  - an Editor is a component of the IDE where you edit (or browse) a resource (such as a C source file)
    - open editors by double-clicking on resources (files)
  - a View is an area that provides: navigation, information, control (but generally not editing)
  - a Perspective is: a collection of views, editors, menu items, and tool bar buttons that are helpful for doing a specific task

Compiling and Debugging Basics
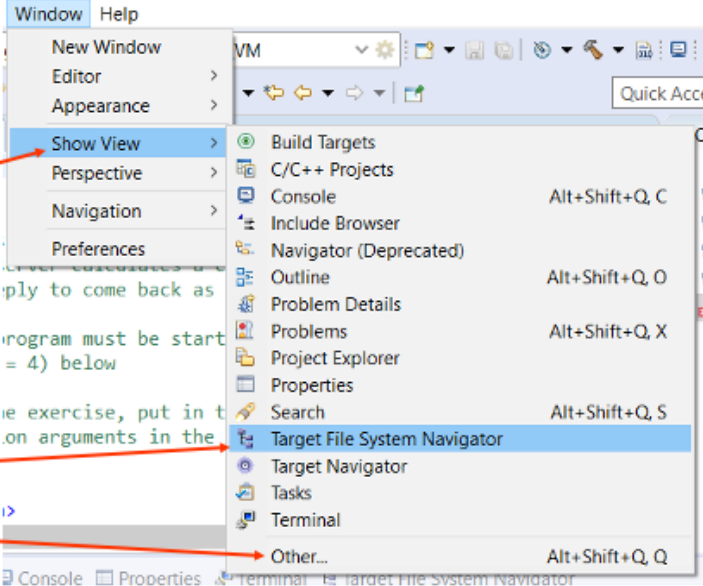2020/09/18 R17
4

‽ QNX
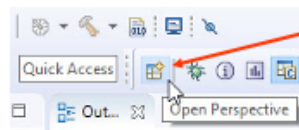
NOTES:

NOTES:

NOTES:

IDE Basics – Adding new views

To show a particular view:

① go to the Window menu…

② … and choose Show View

③ pick a view.
If the view you want is not available, choose Other… for a complete list of all the views available...

Window Help

New Window
Editor >
Appearance >
Show View >
Perspective >
Navigation >
Preferences

Build Targets
C/C++ Projects
Console   Alt+Shift+Q, C
Include Browser
Navigator (Deprecated)
Outline   Alt+Shift+Q, O
Problem Details
Problems   Alt+Shift+Q, X
Project Explorer
Properties
Search   Alt+Shift+Q, S
Target File System Navigator
Target Navigator
Tasks
Terminal

Other...   Alt+Shift+Q, Q

Quick Acce

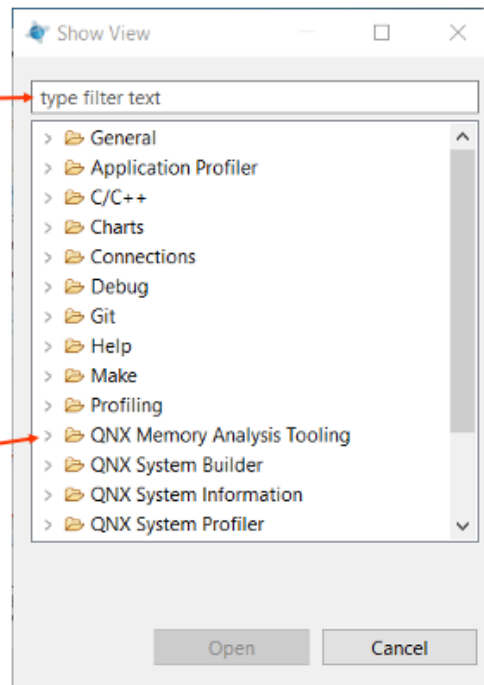**Compiling and Debugging Basics**
2020/09/18 R17
7

:: QNX

NOTES:

The list of views in the menu will vary depending on which perspective you're in. It generally contains the views associated with the perspective, many of which are displayed by default. It can also be used to find, or restore lost or accidentally closed views.

NOTES:

This filtering idea will crop up in many places, often when the IDE presents a plethora of choices the filter option will also be present to aid in navigation.
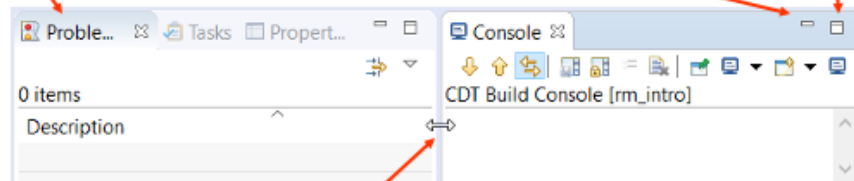
8

Working with Views:

– views are generally stacked or "tabbed" with other views:

to bring a view to the front, click on its tab

you can minimize a group of views, moving them to the closest side of the IDE

you can maximise a group of views as well

Proble... ⏹ Tasks Propert...

0 items

Description

Console ⏹

CDT Build Console [rm_intro]

any border between views can be used to resize the views that share the border

Compiling and Debugging Basics
2020/09/18 R17

QNX

9
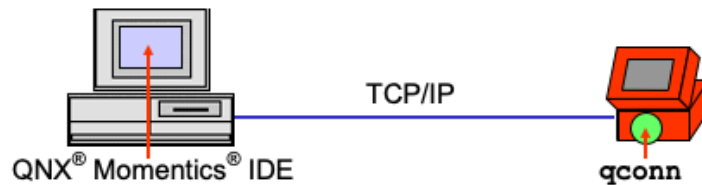
NOTES:

Compiling and Debugging Basics

Topics:

- **Eclipse Basics**
- → **Targets**
- **Projects and Source**
- **Compiling**
- **Running and Debugging**
- **Versions**
- **Exercise**
- **Conclusion**

QNX

NOTES:

QNX uses a cross-development setup:

Host running
Windows/Linux/macOS

Target running
QNX Neutrino

TCP/IP

QNX® Momentics® IDE

qconn

- **qconn** is a program on the target that must be running for the IDE to deal with the target
- in the IDE you have to tell it how to find the target:
  - generally this is an IP address or hostname
  - this information is stored in a Target Project
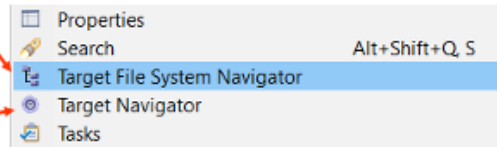 - once the Target project is created, it can be used multiple places

Compiling and Debugging Basics
2020/09/18 R17
11

QNX

NOTES:

Target Interaction – Two Main views

There are two main views for interacting with the target:

- created from Window→Show View:

- Target File System Navigator:
  - viewing target's file system
  - copying files to/from target
- Target Navigator:
  - creating/deleting Target Projects
  - seeing what processes are running
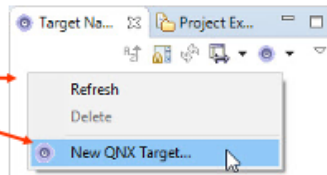 - killing processes

- The QNX System Information perspective has many views for collecting target information
  - it includes a Target Navigator view by default

Compiling and Debugging Basics
2020/09/18 R17
12

NOTES:

Creating a Target System project:

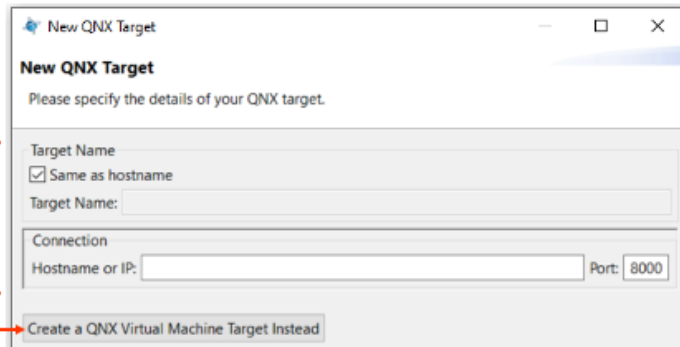from the Target Navigator view right-click in here and choose New QNX Target...

- at this point you have a choice...

create a target project for an actual target board

OR

a target project for a virtual machine (e.g. for a Virtual Box VM)

Target Na... 23 Project Ex...

Refresh
Delete
New QNX Target...

New QNX Target

New QNX Target
Please specify the details of your QNX target.

Target Name
☑ Same as hostname
Target Name:

Connection
Hostname or IP: Port: 8000

Create a QNX Virtual Machine Target Instead

Compiling and Debugging Basics
2020/09/18 R17
•13

QNX

NOTES:

NOTES:

Accessing your Target - Target System project for virtual machine

Target System project for a virtual machine:

fill in a name representing your target. This will be the Target System project's name

the virtual machine that you're using (e.g. vbox for Virtual Box)

the architecture running on your VM

the IP address can be found automatically once/if your VM is running

put extra command line options here to be added to the `mkqnximage` command line used by the IDE to create and run an image for your VM.

choose Finish and your Target System project will be created and `mkqnximage` will create and run the image

New QNX Virtual Machine Target

QNX Virtual Machine Target
Edit the properties of the QNX Virtual Machine

Target Name: local_VM
VM Platform: vbox
CPU Architecture: x86_64
IP Address: <leave blank for automatic>
Extra Options: <leave blank for default options>
Preview: QNX_TARGET=C:/Users/stdufresne/qnx710/target/qnx7 C:\Users\stdufresne\qnx710\host\win64\x86_64\usr\bin\bash C:/Users/stdufresne/qnx710/host/common/bin/mkqnximage --noprompt --

Finish   Cancel

Compiling and Debugging Basics
2020/09/18 R17
•15

QNX

NOTES:

If this is the first time using VirtualBox, make sure it has a network interface before you run `mkqnximage`. Also, you have to enable DHCP.

Often you will need a command-line on your target, some ways include:

- ssh/telnet session:
  - in the Target Navigator, right click on the Target and select "Start SSH Session…" or "Start Telnet Session"
  - or run a ssh/telnet client manually on your host
- serial connection:
  - generally requires a hardware connection, with a null-modem cable
  - use the Terminal view, or a serial terminal application on your host
- run a shell from the IDE:
  - double-click on a shell in the Target File System Navigator
- physical console, real or through a KVM switch

Compiling and Debugging Basics
2020/09/18 R17
16

QNX

NOTES:

Topics:

Eclipse Basics

Targets

→ **Projects and Source**

Compiling
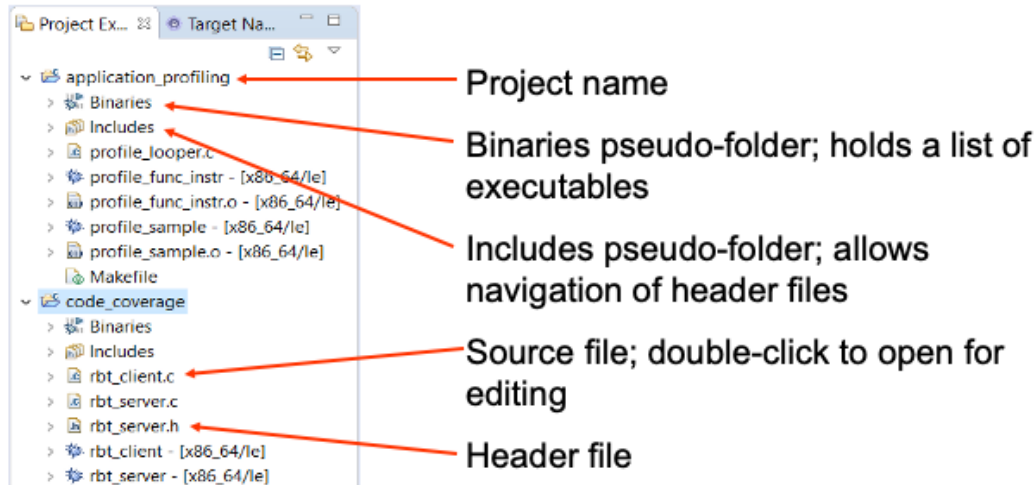
Running and Debugging

Versions

Exercise

Conclusion

QNX
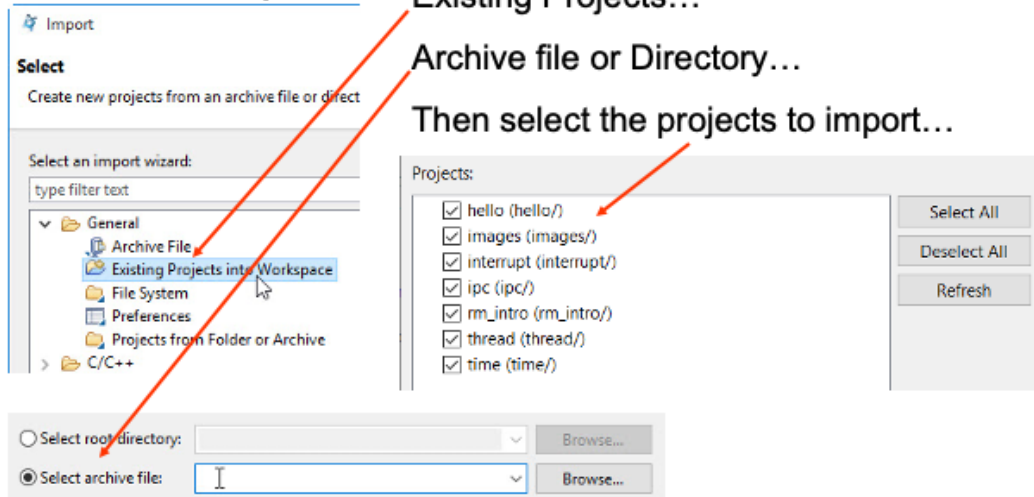
NOTES:

NOTES:

NOTES:

IDE Basics – Exporting Project(s)

To save all your projects to a zip file:

– Project -> Clean…

– File -> Export

Select Archive File then Next

Select All projects

Browse to or enter zip file name
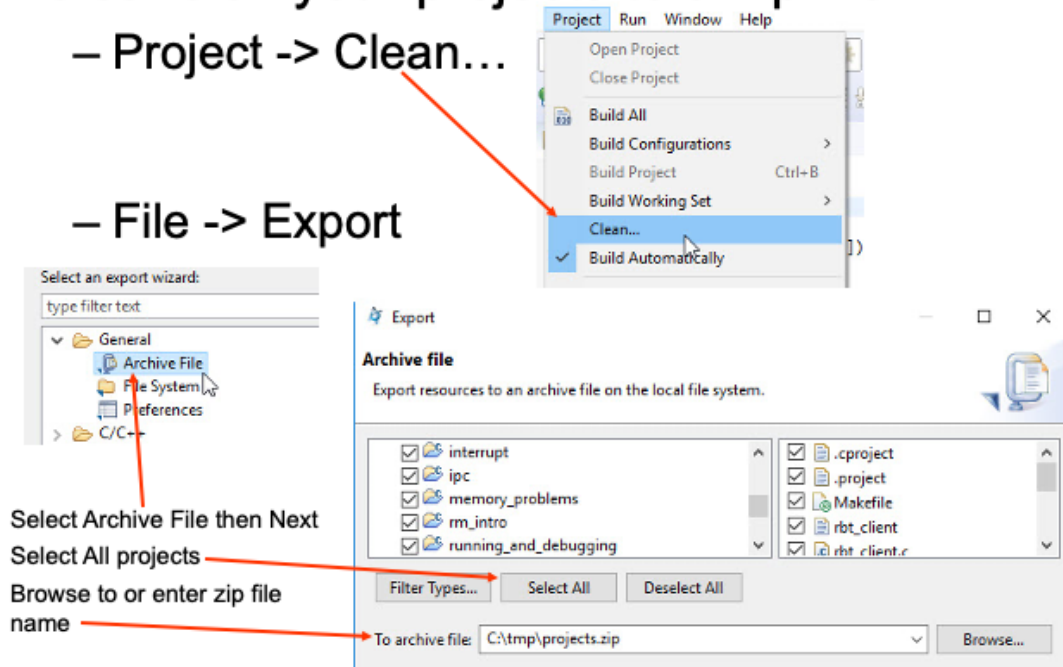
All content copyright
QNX Software Systems Limited,
a subsidiary of BlackBerry
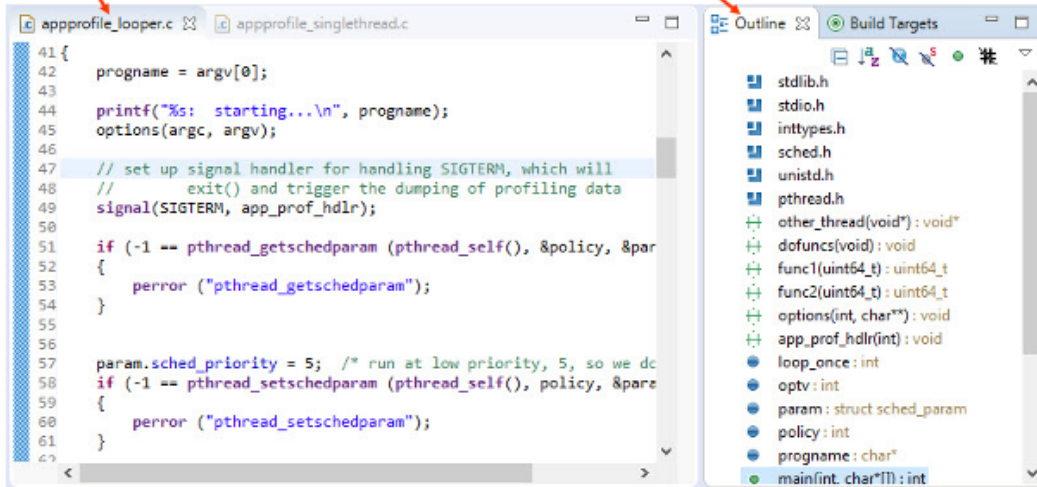
Compiling and Debugging Basics
2020/09/18 R17
20

QNX

NOTES:

NOTES:

Some editing shortcuts:

- Ctrl-Space does code completion for functions, structures/classes, and code blocks
- "standard" Windows cut & paste with Ctrl-C, Ctrl-X, Ctrl-V
- undo/redo with Ctrl-Z, Ctrl-Y
- hover-help on functions in library gives quick summary of use and headers
- select function, then
  - Ctrl-Shift-N will insert `#include` lines for needed headers
    - Menu-Click -> Source -> Add Include will also do this
  - F3 will open definition/declaration
- find or find & replace with Ctrl-F
- search multiple files with Ctrl-H

Compiling and Debugging Basics
2020/09/18 R17

22

QNX

NOTES:

Shortcuts may differ as they are host platform specific.

Topics:

Eclipse Basics

Targets

Projects and Source

→ **Compiling**
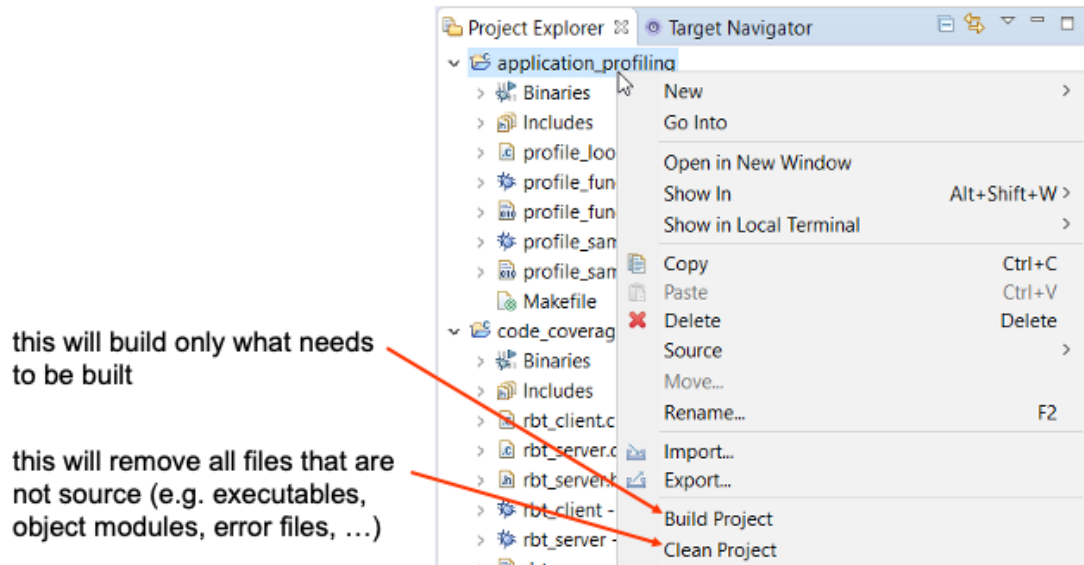
Running and Debugging

Versions

Exercise

Conclusion

QNX

NOTES:

Compiling - Building from the IDE

The IDE calls compiling "building":

– to build from the IDE, right-click on your project...

this will build only what needs to be built

this will remove all files that are not source (e.g. executables, object modules, error files, …)

All content copyright
QNX Software Systems Limited,
a subsidiary of BlackBerry

Compiling and Debugging Basics
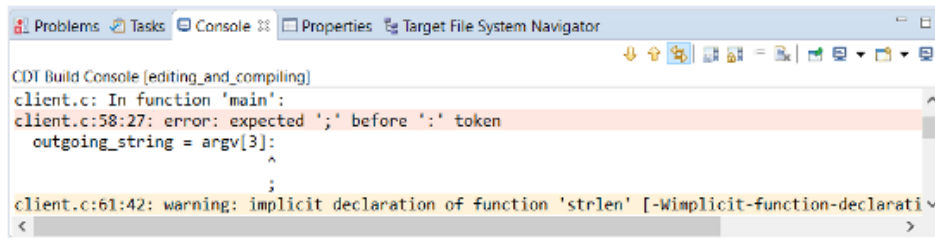2020/09/18 R17
24

QNX

NOTES:

For most projects, the IDE invokes make to actually do the builds, using the makefile in the project.

Build Project issues "make all"; Clean Project issues "make clean". Both assume that these targets exist in the makefile.
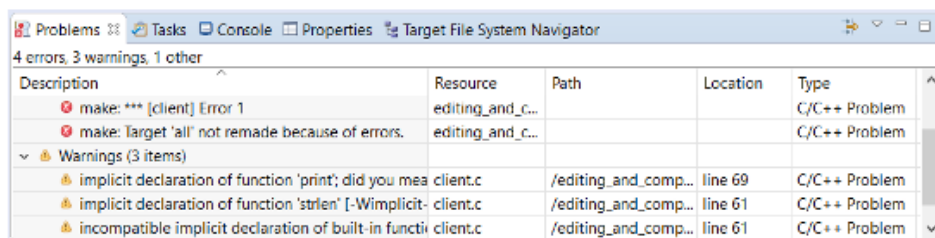
While building, the console view shows the output from the build, highlighting errors and warnings:

After the build is complete, the Problems view summarizes the errors and warnings:

Compiling and Debugging Basics
2020/09/18 R17
25

NOTES:

Double-clicking on errors or warnings in the Console or Problems view will take you to them in your editor.

Fixing Errors - Problem indicators

Many other places indicate problems:

clicking on these will go to next and previous problems (the editor must be selected for these to be enabled)
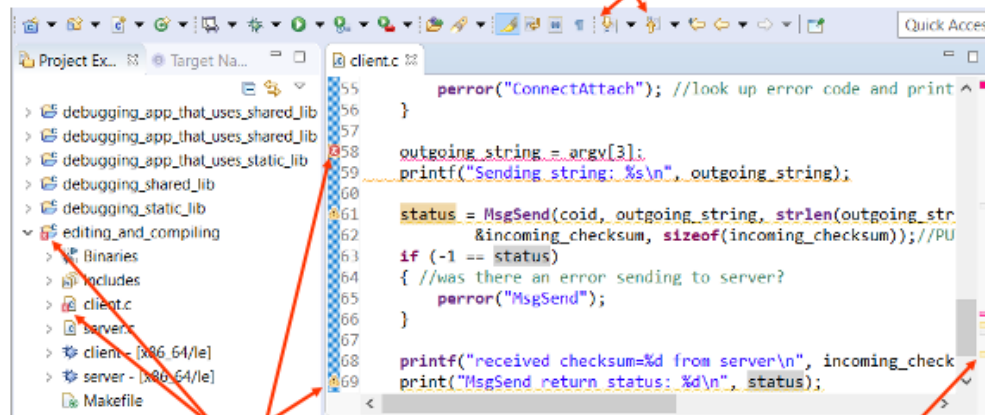
indicates that there is a problem(s)

these markers represent problems. Their relative position vertically represents their locations in the file. You can click on these.

All content copyright
QNX Software Systems Limited,
a subsidiary of BlackBerry

Compiling and Debugging Basics
2020/09/18 R17
26

NOTES:

Hover-help on most error/warning indicators will give the error/warning text.

Compiling and Debugging Basics

Topics:

Eclipse Basics

Targets

Projects and Source

Compiling

→ **Running and Debugging**

Versions

Exercise

Conclusion

:: QNX

NOTES:

Running

There are two main ways of running a program you've built in the IDE:

- copy it to the target with the Target File System Navigator then run it from the command line

- create a Launch configuration and run it from the IDE

  - if you're using the IDE for debugging, you'll need a Launch configuration

  - Launch configurations only need to be created once for a program, then can be re-used

All content copyright
QNX Software Systems Limited,
a subsidiary of BlackBerry

Compiling and Debugging Basics
2020/09/18 R17
28

QNX

NOTES:

NOTES:

Basic Launch configuration:

Name for the launch configuration: pick something descriptive

What program to run: specified by Project & Application

Where to run it: select a target system

Click Run to run the program

Run Configurations

Create, manage, and run configurations

Name: bin.client

type filter text

- C/C++ QNX Applicatic
  - bin.client
  - bin.profile_func_inst
  - bin.profile_sample
  - bin.server
- C/C++ Unit
- Launch Group
- Launch Group (Deprec
- QNX File Transfer

Main | Arguments | Environment | Upload | Debug | Source | Common

Project:

editing_and_compiling   Browse...

C/C++ Application:

client

☐ Auto-pick   Variables...   Search Project...   Browse...

Build (if required) before launching

Build Configuration: Select Automatically

○ Enable auto build   ○ Disable auto build
● Use workspace settings   Configure Workspace Settings...

Target: local_VM   Browse...

Revert   Apply

Filter matched 9 of 9 items

Run   Close

Compiling and Debugging Basics
2020/09/18 R17
30

QNX

NOTES:

Our exercises will often say something like:

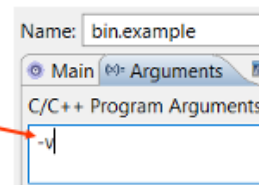run it as:
```
example -v
```

To do this from the IDE:

Put the executable name, **example**, in the Application Section

Then click on the arguments tab

And put the arguments, **-v**, in the Arguments field

Name: bin.example
◉ Main | Arguments
Project:
rm_intro
C/C++ Application:
example

Name: bin.example
◉ Main | Arguments
C/C++ Program Arguments:
-v

All content copyright
QNX Software Systems Limited,
a subsidiary of BlackBerry

Compiling and Debugging Basics
2020/09/18 R17
31

QNX

NOTES:

NOTES:

Compiling and Debugging Basics

Topics:

Eclipse Basics

Targets

Projects and Source

Compiling

Running and Debugging

→ **Versions**

Exercise

Conclusion

QNX

NOTES:

Having different versions of software on the host and target can cause problems:

- the IDE and GDB use files on the host, but the files run on the target
  - they must be the same version
  - this applies for both QNX's and your pieces
- to check versions:
  - from your target:

    `uname -a`

    `use -i` *filename*
  - from your host:

    `use -i ${QNX_TARGET}/`*path*`/`*filename*

Compiling and Debugging Basics
2020/09/18 R17
34

:: QNX

NOTES:

The `QNX_TARGET` environment variable contains the folder where you installed the SDK followed by `target/qnx7` (e.g. `C:/Users/wally/qnx710/target/qnx7`).

To set `QNX_TARGET`, from a terminal/command line window:

1. go to where you installed the SDK

   e.g. `C:\Users\wally\qnx710`

2. run the `qnxsdp-env.bat` or `qnxsdp-env.sh` that you'll find there
   - on Linux or macOS you need to source it: `source ./qnxsdp-env.sh`

Versions – Checking/changing the SDK version in the IDE

To check/change the SDK version in the IDE:
– go to Window→Preferences ② choose an SDK version from the dropdown list

① select QNX

③ Apply and Close and the IDE will ask for your permission to restart for the change to take effect
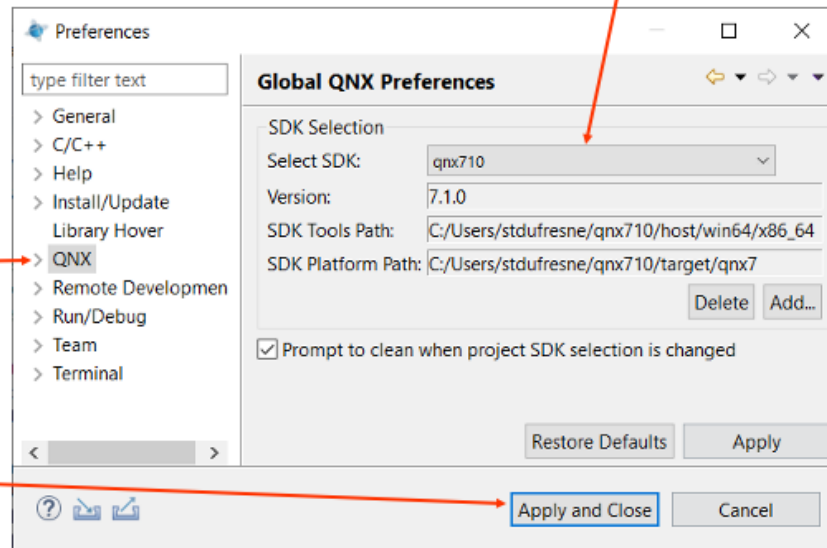
All content copyright
QNX Software Systems Limited,
a subsidiary of BlackBerry

Compiling and Debugging Basics
2020/09/18 R17
•35

NOTES:

This is also useful in the general case where you are working with different versions or different SDKs on a daily basis.

When you Apply and Close, it gives you the option of restarting or not. You must restart for the change to take effect. If you choose not to restart then you should either change your SDK selection back to what it was or restart the IDE yourself in which case the change will take effect.

Working with multiple SDKs can be done by:

- using Windows→Preferences→QNX to switch between SDKs
  - slow since it restarts the IDE
- creating a different workspace for each SDK
  - use File→Switch Workspace to switch between them (also restarts the IDE), or
  - simply run the IDE multiple times, each with a different workspace
    - when each IDE starts up, it asks which workspace you want to work with

Compiling and Debugging Basics
2020/09/18 R17
36

QNX

NOTES:

Compiling and Debugging Basics

Topics:

Eclipse Basics

Targets

Projects and Source

Compiling

Running and Debugging

Versions

→ **Exercise**

Conclusion

QNX

NOTES:

Exercise:

- in your `hello` project, compile `hello.c`
- it has errors and warnings, to demonstrate how the IDE marks build problems
- fix these
- build the project again
- run the program as (something like):

  `hello This is some text`

:: QNX

NOTES:

38

Compiling and Debugging Basics

Topics:

Eclipse Basics

Targets

Projects and Source

Compiling

Running and Debugging

Versions

Exercise

→ **Conclusion**

Compiling and Debugging Basics
2020/09/18 R17
39

QNX

NOTES:

Conclusion

In this section you learned how to:

- edit

- compile

- and run or debug your programs

Compiling and Debugging Basics
2020/09/18 R17

QNX

NOTES: