10 Steps to Developing a QNX Program

Quickstart Guide

QNX SOFTWARE SYSTEMS

© 2008, QNX Software Systems GmbH & Co. KG.
A Harman International Company. All rights reserved.
QNX, Aviage, Momentics, Neutrino, Photon and Photon microGUI are trademarks of QNX Software Systems GmbH & Co. KG, which are registered trademarks in certain jurisdictions and used under license by QNX Software Systems International Corporation. All other trademarks and trade names belong to their respective owners.
Printed in Canada. 002497 MC502.03

Quickstart Guide

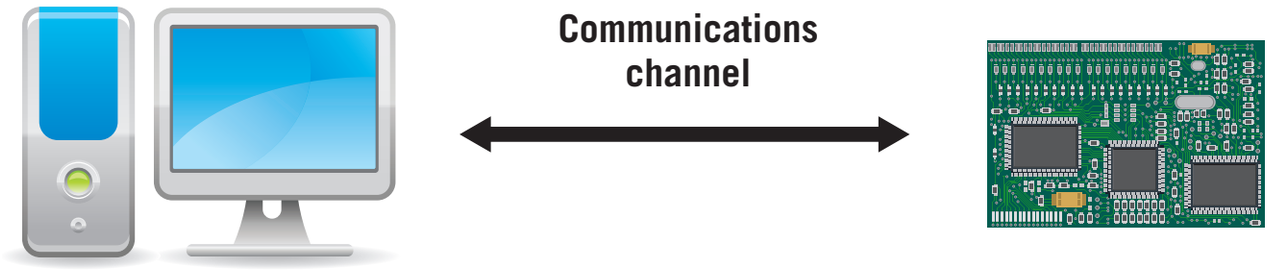
Install and configure the QNX Software Development Platform, which includes the QNX Neutrino® Realtime Operating System (RTOS) and the QNX Momentics® Tool Suite, so you can start developing right away!

1. Requirements
2. Installing the QNX Software Development Platform on the development host
3. Installing the QNX Neutrino RTOS on the target system
4. Networking with the QNX Neutrino RTOS
5. Creating a program project
6. Communicating with the QNX Neutrino RTOS
7. Compiling and linking
8. Preparing to launch the program
9. Starting and debugging
10. Making the program your own

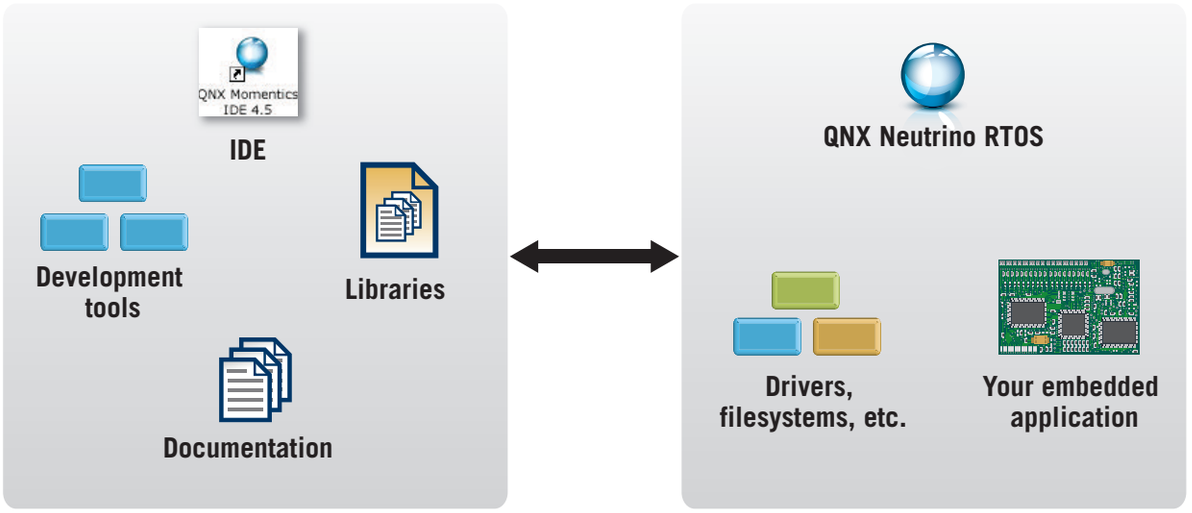
Requirements

To write programs that run under the QNX Neutrino Realtime Operating System (RTOS), the first thing you need is the QNX Software Development Platform (SDP). This includes the QNX Momentics Tool Suite, which contains everything you need to develop programs that run under the QNX Neutrino RTOS: compiler, linker, libraries and other operating system (OS) components, precompiled for all CPU architectures that the OS supports. On Windows and Linux, the tool suite features an extensive Integrated Development Environment (IDE).

You can install QNX SDP on a QNX Neutrino RTOS system for *self-hosted development*, or you can install it on a Windows Vista, Windows 2000, Windows XP, or Linux *development host* and install the OS on a *target system*:



The development host runs the QNX Momentics Tool Suite; the target system runs the QNX Neutrino RTOS itself plus all the programs you're going to develop:



QNX Momentics Tool Suite **Target system**

If you don't have the QNX Software Development Platform DVD, you can download an evaluation version from **`www.qnx.com/products/evaluation/`**. If you want to evaluate the QNX Neutrino RTOS on x86 targets only, you can download the (much smaller) QNX SDP for x86 targets.

The DVD includes an installer for each host OS. There's also a CD that contains just the QNX Neutrino RTOS-hosted version, for systems that don't have a DVD drive.

To become familiar with the QNX Neutrino RTOS, you have several choices:

- You can install the self-hosted version of the development platform on a normal PC that has a free partition of about 1.5 GB. (The OS itself, as you'd install it on a runtime system, requires only about 300 MB.) Installing the OS won't damage any existing partitions. You can also boot the QNX Neutrino RTOS directly from the DVD or CD, in case you don't have enough room on your hard disk.
- You can run the QNX Neutrino RTOS on a *reference platform*, a reference design made by a CPU vendor (e.g. with a PPC, XScale, or SH CPU). You'll need a QNX board support package (BSP) for your platform. The documentation that comes with each BSP explains how to install the operating system on that target system. For more information, see our Foundry27 website, http://community.qnx.com.
- You can install and run the QNX Neutrino RTOS as a virtual machine in a VMware session. Although VMware is a handy way to try the OS, you should note that virtual machines don't necessarily support hard realtime.

Since the QNX Neutrino RTOS is designed the same way for all platforms and is used the same way, for this Quickstart Guide we'll use Windows as a development host, and a PC as the target.

2 Installing the QNX Software Development Platform on the development host

Boot your Windows Vista, 2000, or XP system and insert the **QNX Software Development Platform DVD**. If the installation doesn't start automatically, simply run the program **`qnxsdp-6.4.0-`***`nnnnnnnnnnnn`***`-win32.exe`** (where *nnnnnnnnnnnn* is a build number), which you'll find in the root directory of the DVD. You'll be guided through the installation process. For more information, see the *Installation Guide*.

The installation program will ask you for a license key. If you downloaded an evaluation version of QNX SDP from our website, you should have received an email containing the key. Otherwise, you'll find your key on the box that contains the DVD and CD.

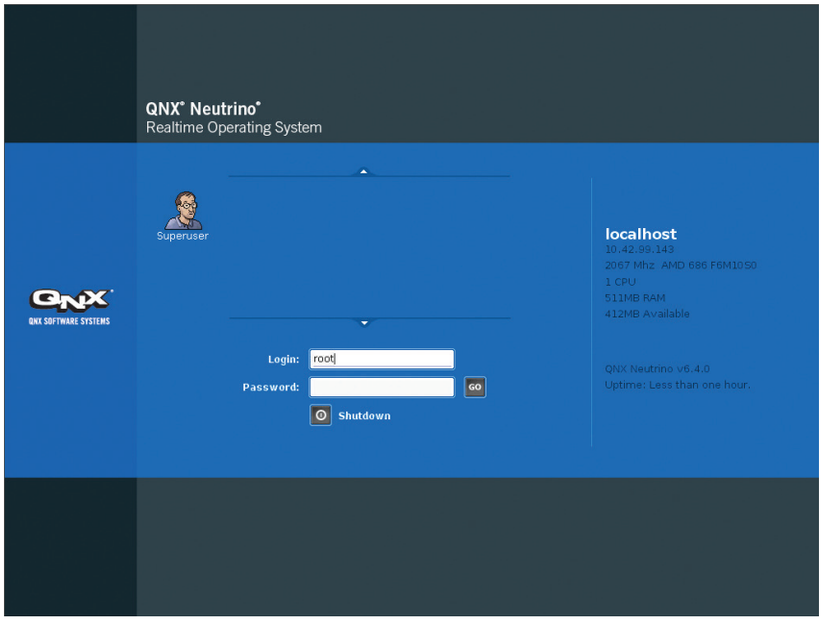
After the installation, you'll find an icon for the QNX Momentics IDE on your Windows desktop:

3 Installing the QNX Neutrino RTOS on the target system

Now insert the **QNX Software Development Platform QNX Neutrino RTOS Host DVD or CD** in the drive of your second machine and boot from it. If you don't have this disk, you can download an evaluation version from **`www.qnx.com/products/evaluation/`**.

At this point, you can choose to start the OS directly from the disk (ideal for initial testing) or install the QNX Neutrino RTOS onto your hard disk. Please choose the installation to hard disk and follow the onscreen instructions. For more details, see the *Installation Guide*.

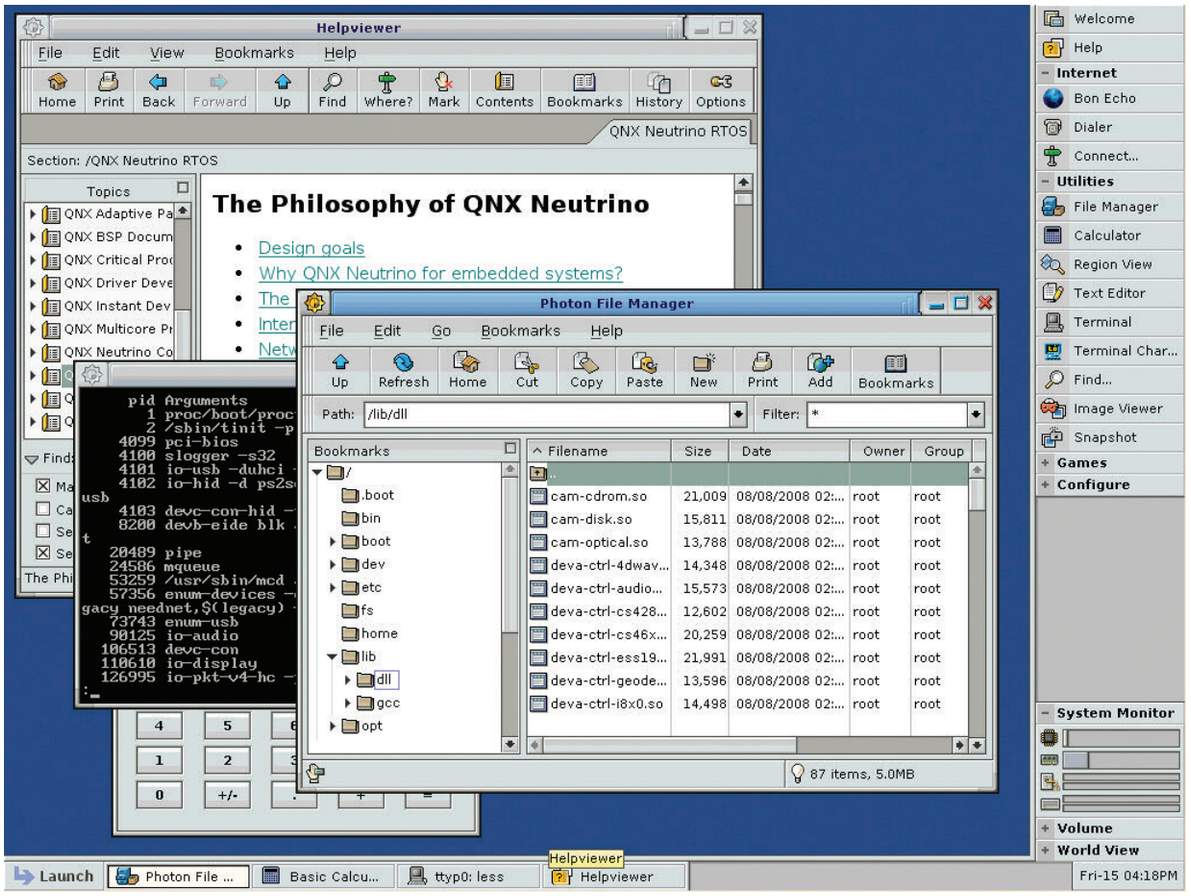
After rebooting, your hardware will automatically be detected. Once you select the graphics mode, you can log in as **`root`** without a password. Note your machine's IP address on the right side of the screen; you'll need it later.



You now are sitting in front of a preconfigured, fully featured QNX Neutrino Realtime Operating System including the QNX Photon® microGUI® windowing system. (Of course you can run the QNX Neutrino RTOS without graphics, too.) On the right side of the screen, you'll find an icon bar. From there, open a shell by opening the **Utilities** group and then clicking on **Terminal**. To see a list of the processes that currently exist in your system, type: **`pidin | less`**

Each process is optional, which means that later in your design, you can remove processes to save resources — or you can add other processes to increase the system's functionality. This also applies for graphics, networking, or audio; each QNX Neutrino RTOS component is a single process that you can load dynamically. Type **`q`** to exit the **`less`** command.

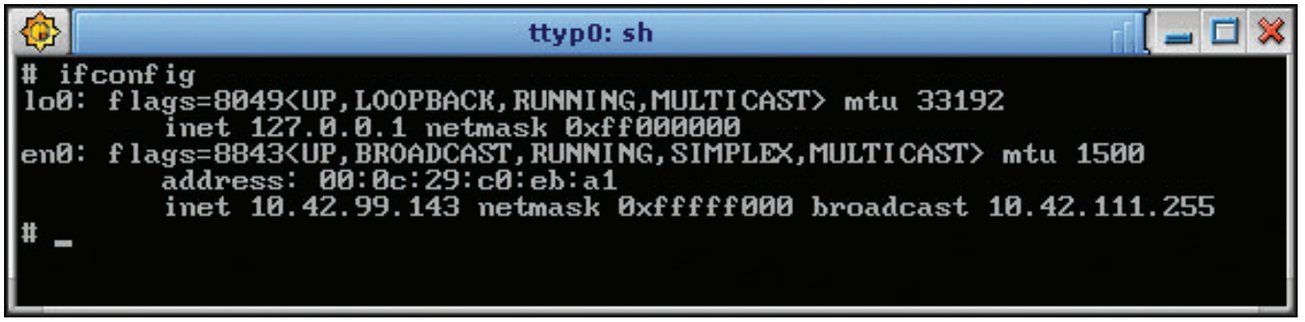
With the QNX Neutrino RTOS installation that you just created, you can easily familiarize yourself with many QNX capabilities and features. And all this without the need to create and configure a boot image yourself!



Lots of features and still capable of hard realtime: a self-hosted QNX Neutrino system.

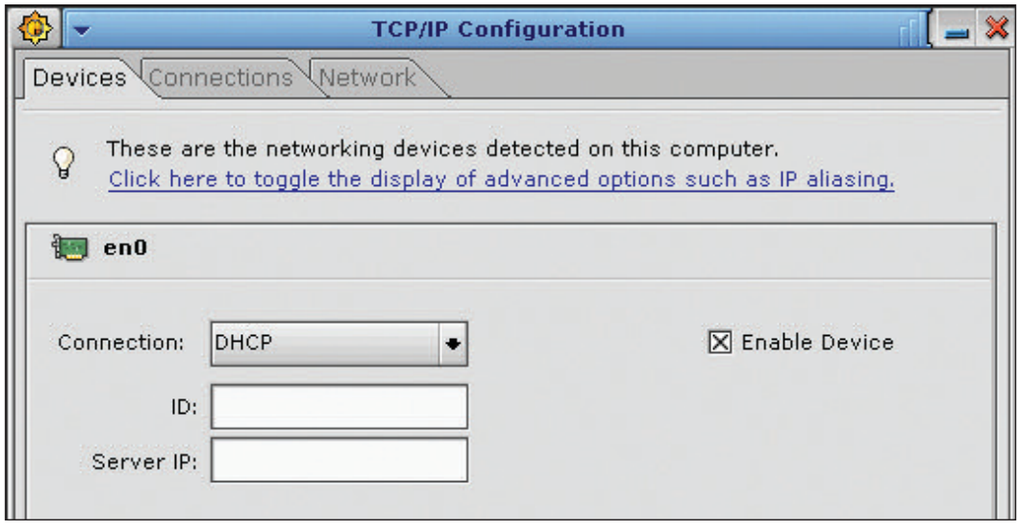
4 Networking with the QNX Neutrino RTOS

Now connect your QNX Neutrino RTOS machine (your target) to the network. Your development machine should be on the same network. With a DHCP server available, your QNX Neutrino RTOS machine will receive an IP address automatically. You can view it or change it using the **`ifconfig`** command, as described in the *Utilities Reference*.

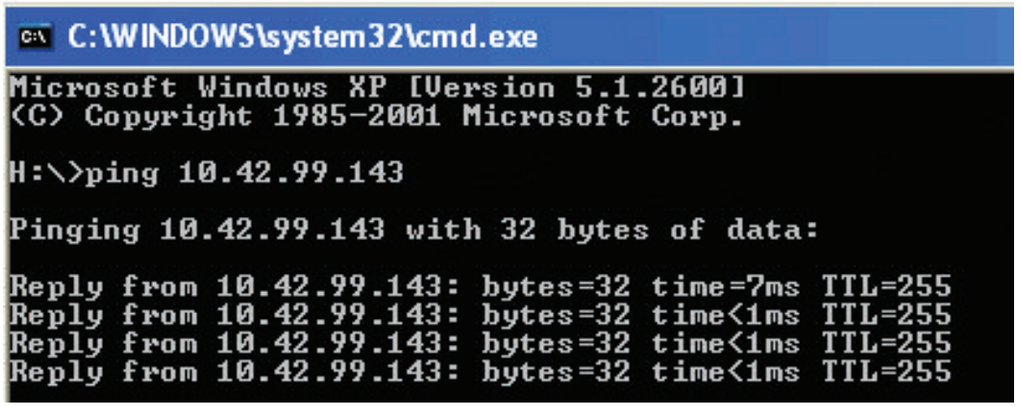


Alternatively, you can use the network configuration tool that comes with the QNX Neutrino RTOS installation. You'll find it under the **Configure** item on the icon bar on the right side of the screen, or in the **Launch** menu.

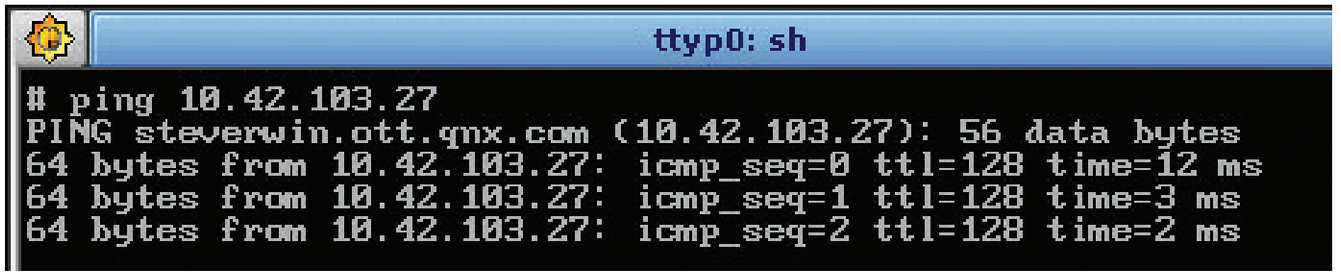
Under **Devices**, you can select whether you would like to use DHCP or a manually assigned IP address. Under **Network**, please enter the IP addresses of the Gateway and DNS.



On your Windows development host, open a **cmd** window and use **ping *IP_address*** to check that your Windows development host can reach your QNX Neutrino RTOS system (target) on the network:



In the same **cmd** window, use **ipconfig** to determine your host's IP address. On the target system, use this IP address to make sure that your target can reach your host:



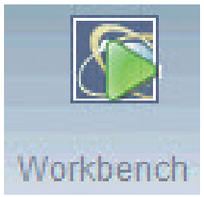
Note: If your host machine uses a firewall, you might not be able to **`ping`** it from the target. On Windows XP you might have to enable **Allow incoming echo request** in the ICMP settings; choose **Control Panel** from the **Start** menu, then choose **Security Center,** and then choose the **Advanced** tab.

If the network doesn't work properly on your target machine, you may be using an unsupported network card. For a full list of supported hardware, visit **`http://www.qnx.com/developers/hardware_support`**. If you have further questions regarding hardware support, please call your local sales representative.

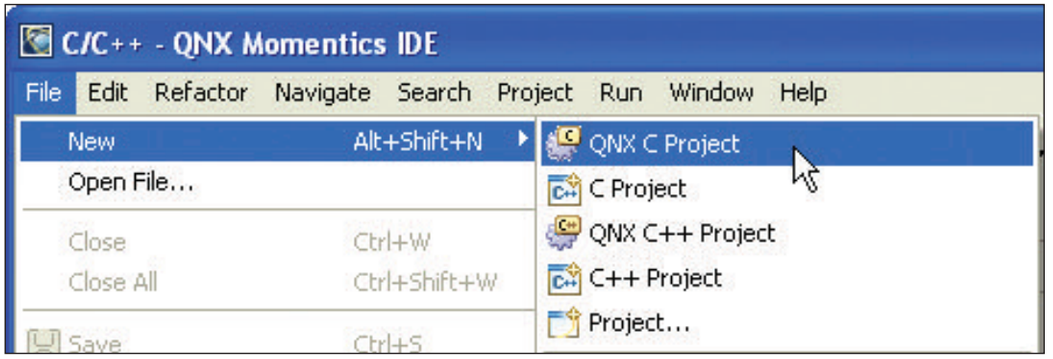
Hint: The full QNX Neutrino RTOS installation uses automatic hardware detection to start the corresponding device drivers. With the **`enum-devices -n`** command, you can see what hardware was detected by the enumerators and which drivers have been started accordingly during booting.

5 Creating a program project

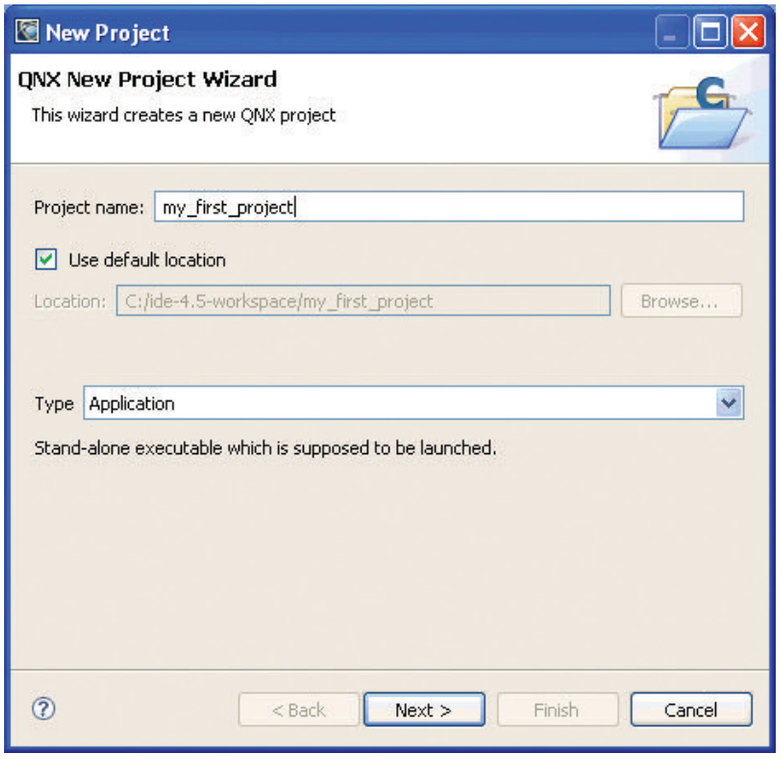
Start the QNX Momentics IDE on your development host. The first time you start the IDE, it asks you to choose a *workspace*, a folder where it can store your projects and other files. The IDE then displays its Welcome page. When you're ready to start, click the Workbench icon:



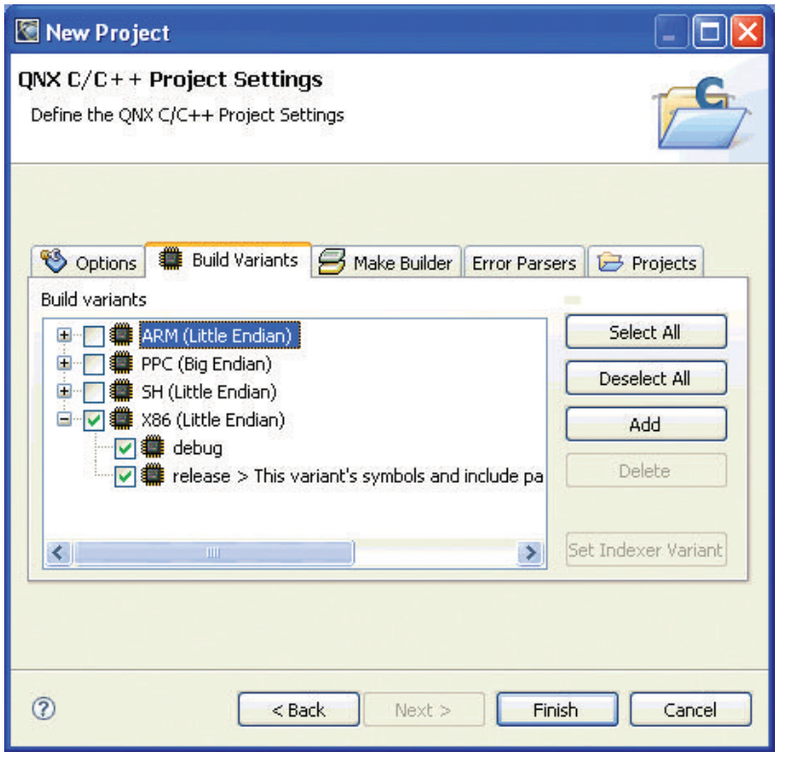
Now create a QNX C Project: from the **File** menu, select **New > QNX C Project:**



In the resulting dialog, give your project a name and then click **Next**. You now need to select a CPU architecture for the binary you're creating. To do this, go to the **Build Variants** tab. For a PC as target, choose x86. For projects on other processors, select the corresponding CPU type: PPC, SH, or ARM. You can also select compilation with or without debug information; we'll be using both later, so make sure the debug and release variants are both checked.



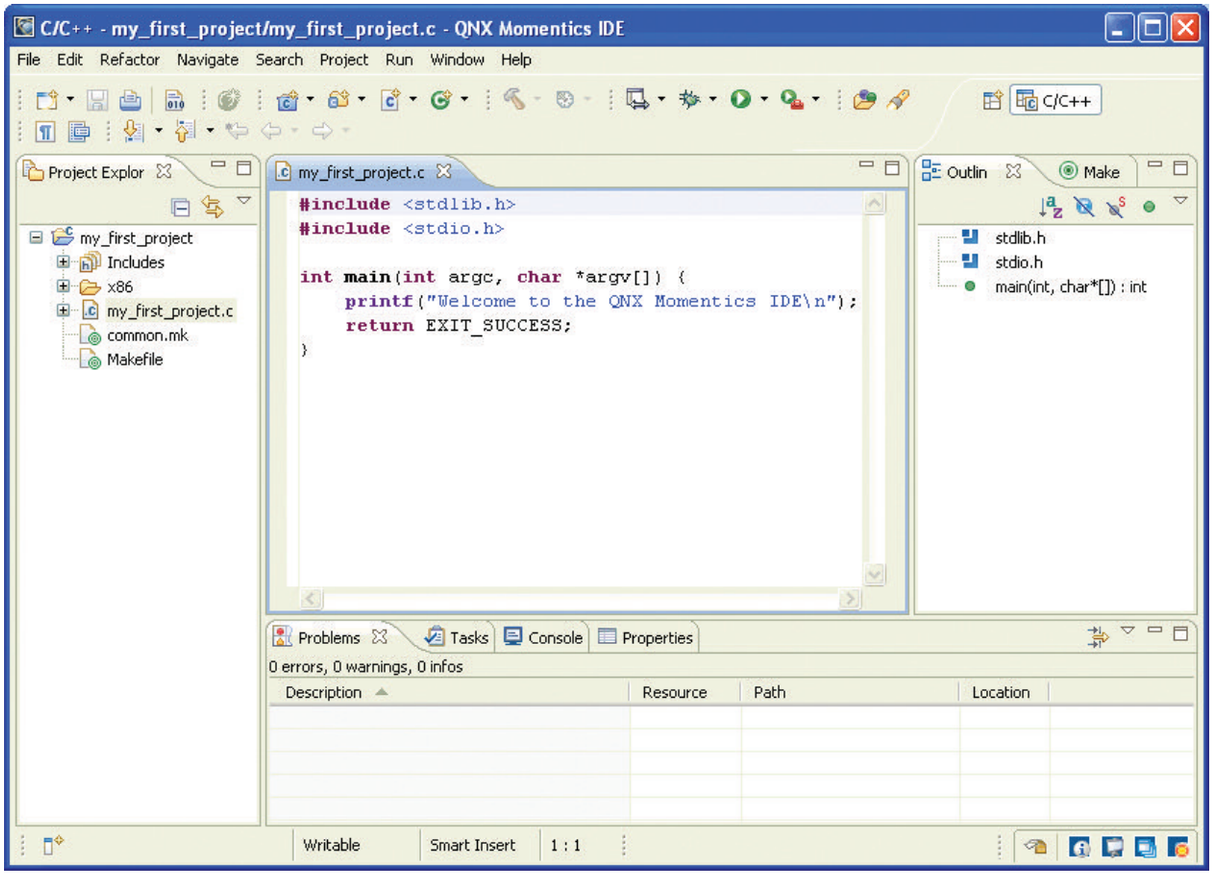
Naming your project.



Selecting build variants.

Click **Finish**. A ready-to-use project structure with a Makefile is created for you, including a small program ("Welcome to the QNX Momentics IDE"), which you will find in an automatically generated source code file.

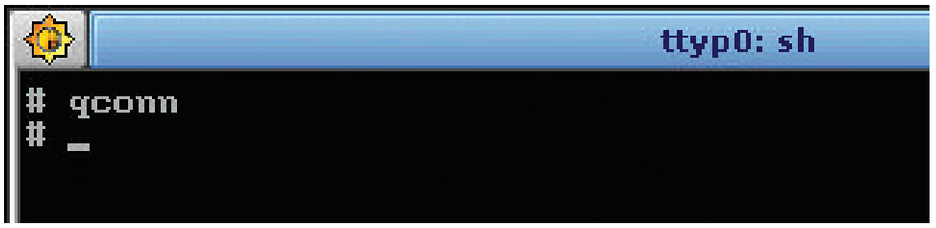
The IDE now switches to the C/C++ perspective, which features the navigator, the editor, and other useful *views*, areas that display information that's relevant to the task at hand:



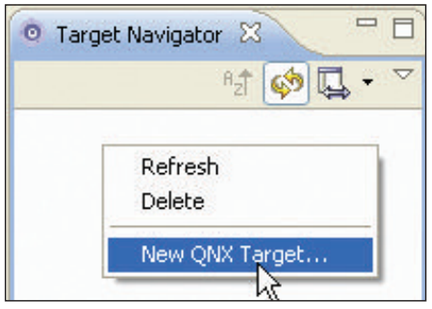
A QNX C project comes with a predefined Makefile structure.

6 Communicating with the QNX Neutrino RTOS

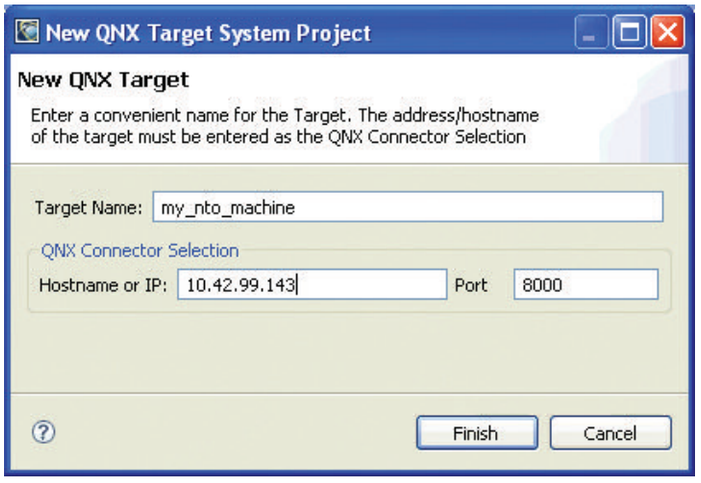
Your target system must be able to respond to requests from the development environment. To make this possible, start the program **qconn**. On a PC running the QNX Neutrino RTOS, you can do this from a terminal window.



To access your target system from the IDE, you have to create a *target project.* Open the System Information perspective: In the **Window** menu, select **Open Perspective > QNX System Information.** In the empty **Target Navigator** view, press the right mouse button and select **New QNX Target** from the context menu.

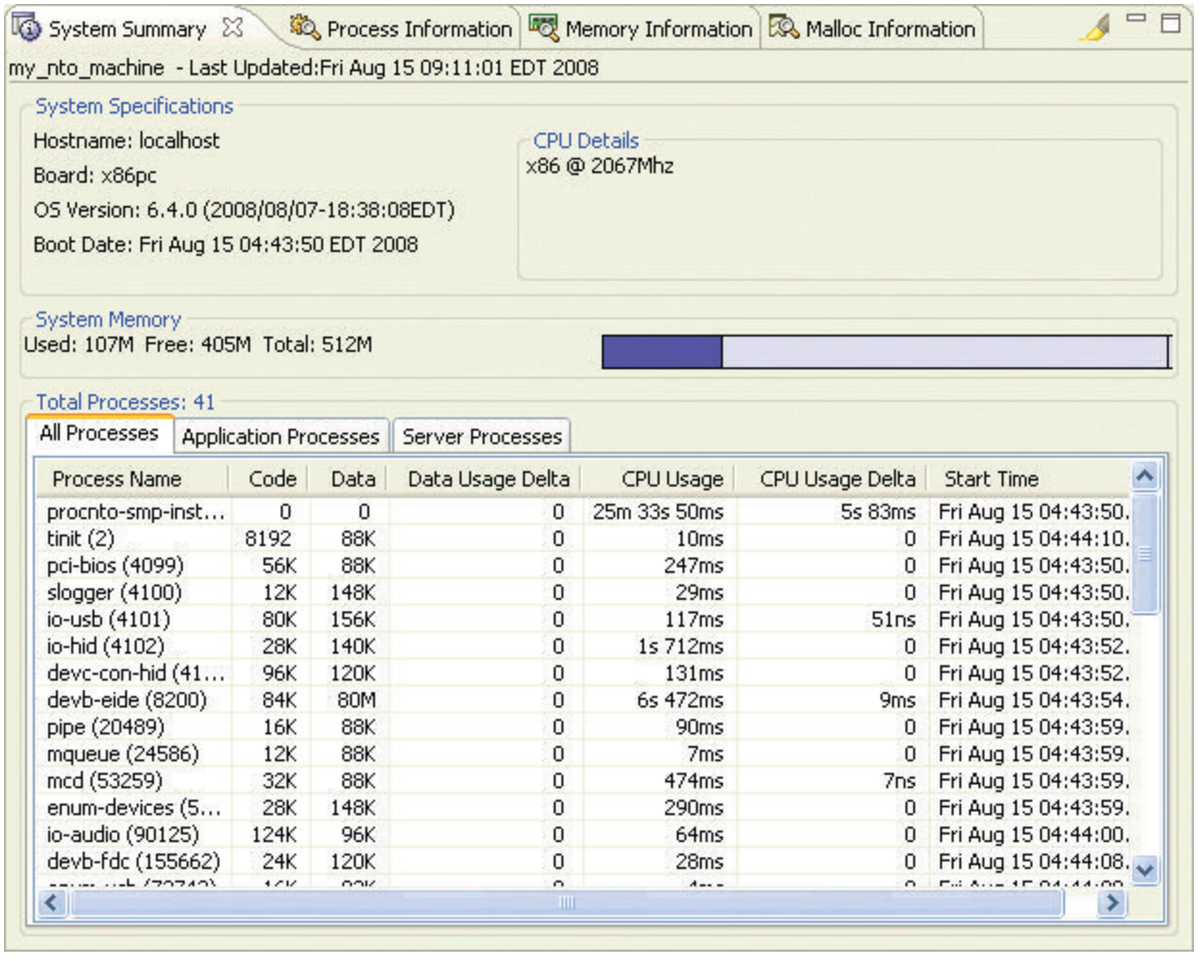


Now provide a name for your target system and enter its IP address in the corresponding field.



The Target Configuration dialog.

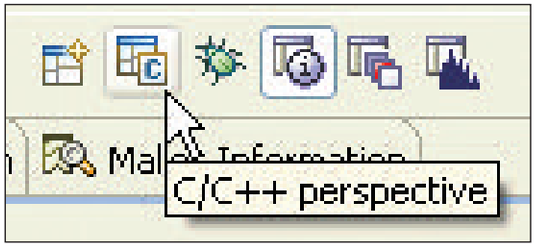
Click **Finish,** and then select your new target in the Target Navigator. You will now see a list of all the processes in your QNX Neutrino RTOS system. The views (the tabs at the top) provide other information to you. You can find even more useful views in the **Window** menu under **Show View**.



Here you see what's going on inside your QNX Neutrino RTOS system.

7 Compiling and linking

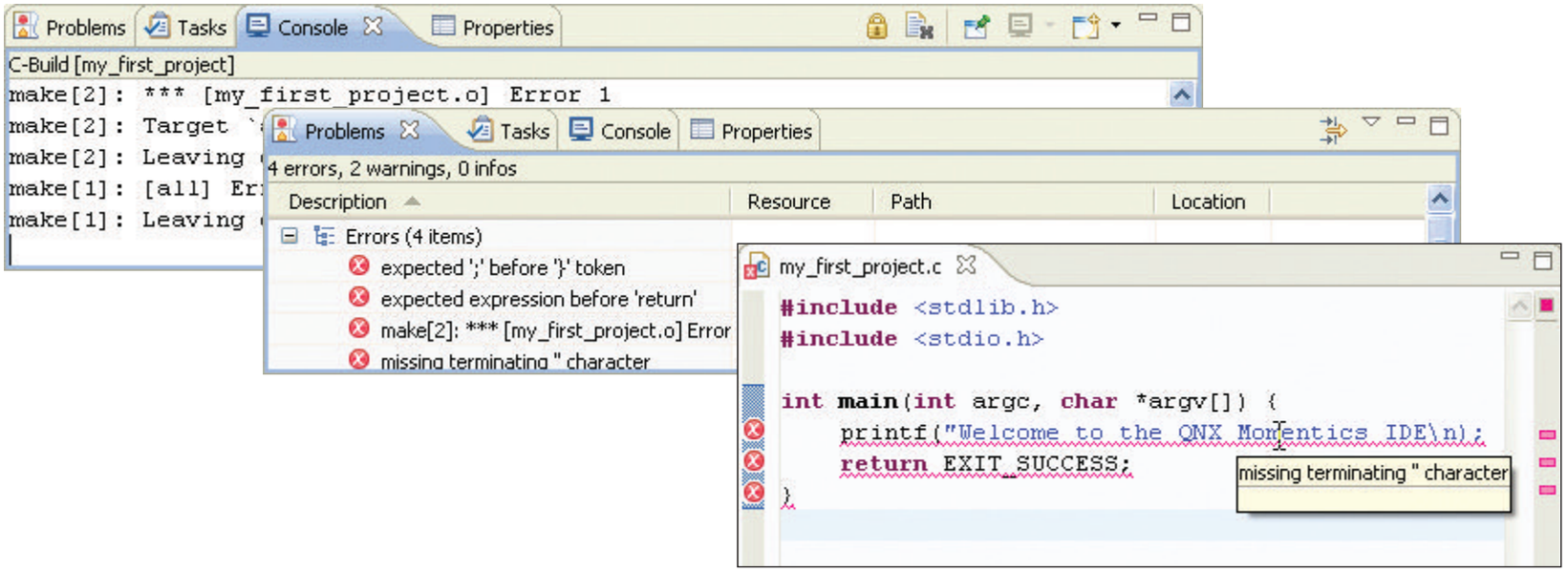
Now switch back to the C/C++ perspective by choosing its icon in the right side of the toolbar.



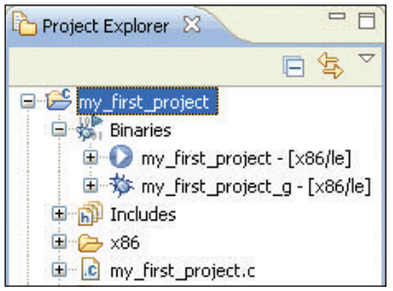
Before you compile, you may want to select compilation with or without debug information. To do so, right-click the project name in the C/C++ Projects view, and then choose **Properties**. Click **QNX C/C++ Project**, click **Build Variants**, and then click on the small + symbol next to x86. Make sure that both the debug and release variants are checked. Click **OK**; the IDE offers to rebuild the project.

During the creation of the QNX C Project, a QNX-made directory structure with **Makefiles** was generated. Now to create a binary, please right-click the project name, and then select **Build Project**. The compiler and linker will now do their work.

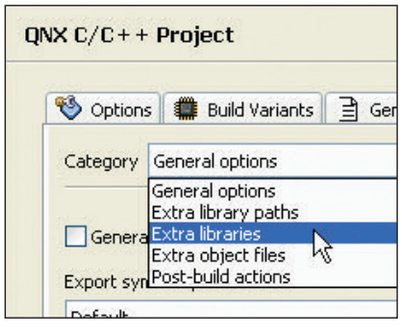
You will find the compiler output in the C-Build output in the Console view, including any errors (you shouldn't see any errors, but we've added one in the examples below). However, if errors occur during compiling, you will find the Problems view more useful, because it displays the output of the compiler in an interpreted and more readable fashion than the Console view. The Editor view also gives you information about an error if you leave the pointer over it.



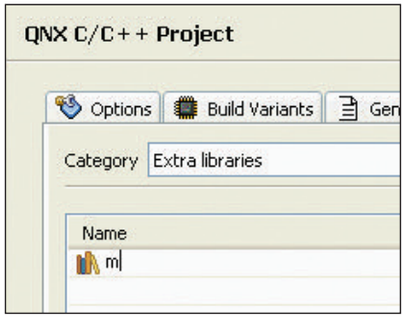
After the build operation, your binaries will be displayed in the **Binaries** folder. Physically, they're located in the CPU directory under "**o**" (for object) and "**o-g**" (**-g** for the debug option passed to the compiler). The IDE automatically created the corresponding **Makefiles**.



The QNX library **`libc.so`**, which contains many basic functions, is linked dynamically to your binary by default. If you want to add other libraries later, you can do so under the **Project** > **Properties** section. From there, click on **QNX C/C++ Project**, then **Linker**, and then choose **Extra Libraries** in the **Category** field:



Click **Add,** and type the name of the library, without the **`lib`** prefix or the extension. For example, to add the math library, **`libm.so,`** you just have to type **`m`** in the Name field:

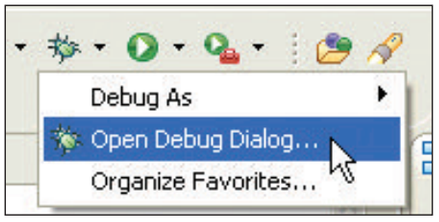


Click **OK.** The linker will now link the library when you build the project.

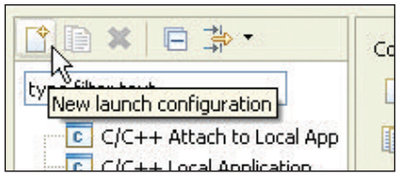
8 Preparing to launch the program

To run and debug the newly built program on your target system, you need to create a *launch configuration*. It consists of various settings that affect how the program starts (e.g. command-line parameters, environment variables). You enter these once, and then you can use this collection of settings again and again.

Now create your own launch configuration: from the dropdown menu beside the "bug" icon on the toolbar, select **Open Debug Dialog. . .** :

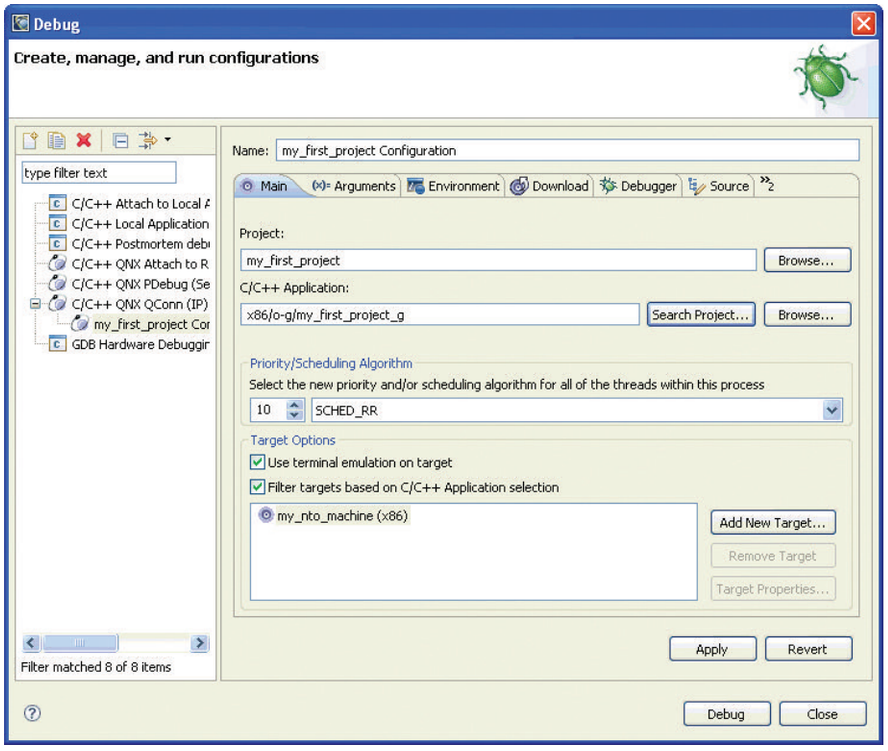


A dialog window opens, where you can start existing launch configurations, change them, or create new ones. On the left, select **C/C++ QNX QConn (IP)**. This type of launch configuration is meant for network-based (cross) development with the QNX Neutrino RTOS running on the target system, using the **`qconn`** program. Now click on the **New launch configuration** icon:



You will now be presented with many configuration possibilities that all deal with starting your executable program. Right now, only the **Main** tab needs your input. Later, however, you should also take a look at what the other tabs have to offer. Next to the **C/C++ Application** field, press the **Search Project** button and choose your binary. If you compiled it with debug information, its name includes a suffix of **_g**. If you compiled it without debug information, its name doesn't include this suffix. Since we would like to start the Debugger in the next step, please choose the binary with the debug information. Click **OK**.

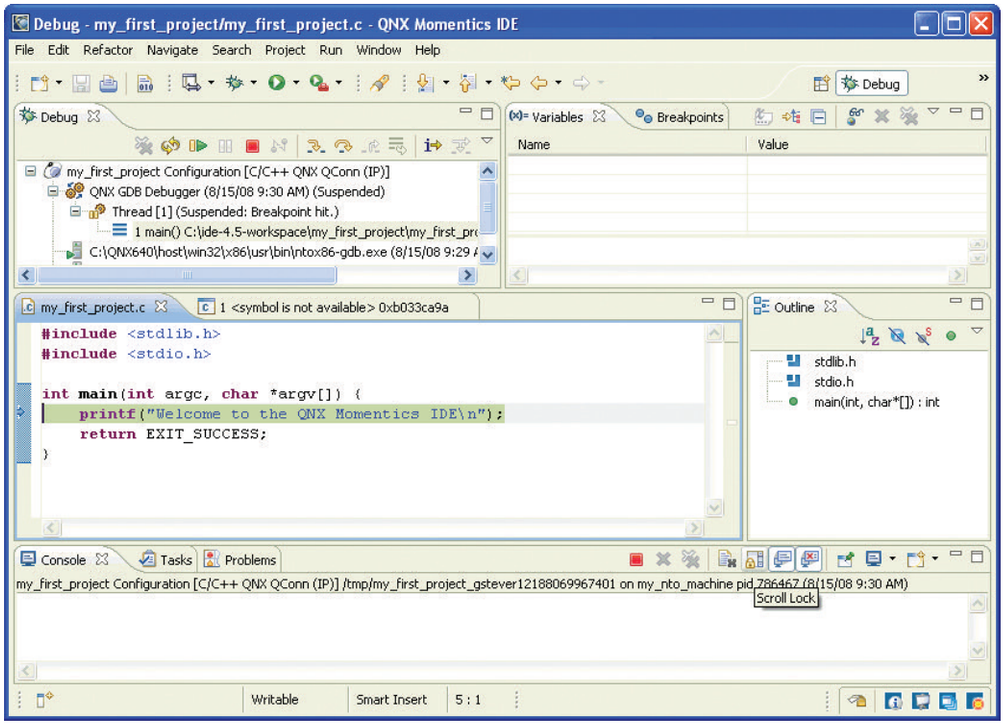
Make sure your target system is listed under **Target Options**, and then click **Apply** — the launch configuration is now ready:



9 Starting and debugging

You should still be in the Debug launch configuration dialog. You just created a configuration for launching your program, which you now can start in the debugger. To do this, please click **Debug**.

The IDE now switches to the Debug perspective and transfers your program from your development machine across the network to your target QNX Neutrino RTOS system, and then starts it in the debugger. You will see that the debugger stops in the first line of your program. In the Debug view, you'll see an overview of your process, including the call stack. Using the buttons in the main bar of the Debug view, you can control the debugger.

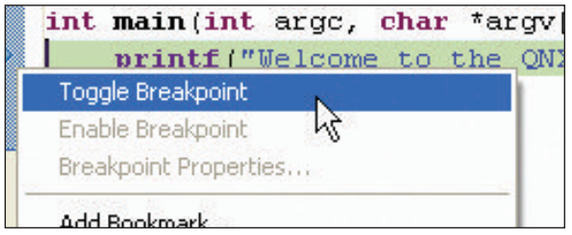


When you run or debug your application from the IDE, any input is read from the IDE's console, and any output goes to it. Once execution has passed the line that calls *printf()*, you should see the "Welcome to the QNX Momentics IDE" message in the Console window.

Using the **Step Over** button, you can jump to the next line of code.

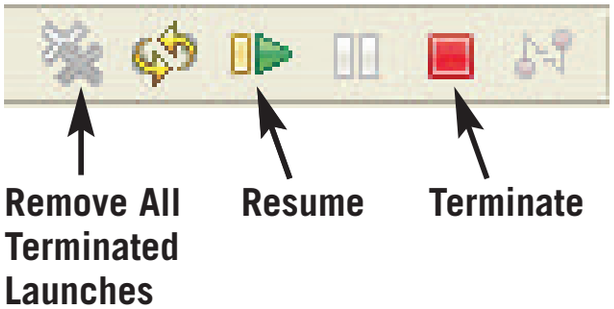
During debugging, you can watch the Variables view on the right, which displays how your variables change. You can use the **Step Into** button to let the debugger go into the code of a function (which, of course, is useful only if you have the source code for this function).

To set a breakpoint, place the mouse pointer over the left border of the source display, press the right mouse button and choose **Toggle Breakpoint** from the context menu. The breakpoint is shown as a little circle with a check mark, which you can also set or remove while you write your code.



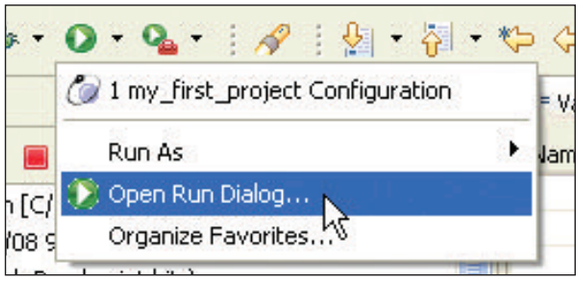
Setting breakpoints.

When the running program hits a breakpoint, it stops in the debugger, and you can, for example, examine your variables. If you click the **Resume** button, your program continues until it hits the next breakpoint. To abort program execution, use the **Terminate** button. After the program has finished running, use the **Remove All Terminated Launches** button to end the debugger and have your binary (which was previously transferred) deleted from your target system.



Note: The debugger keeps the project's files open while the program is running. Be sure to terminate the debug session before you try to rebuild your project, or else the build will fail.

To run your program as a standalone binary (without the debugger), open the dropdown menu beside the Run icon and choose **Open Run Dialog. . .** :

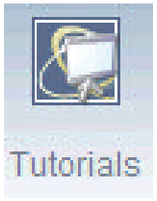


Then you can use the launch configuration (described in the previous step) to start your program. Or create a new launch configuration and select the binary without debug information. You can also transfer your binary manually using the System Information Perspective's File System Navigator (**Window** > **Show View**) and start it from the terminal. Mounting a shared network drive is also possible (see the entry for **`fs-cifs`** in the *QNX Neutrino RTOS Utilities Reference*).

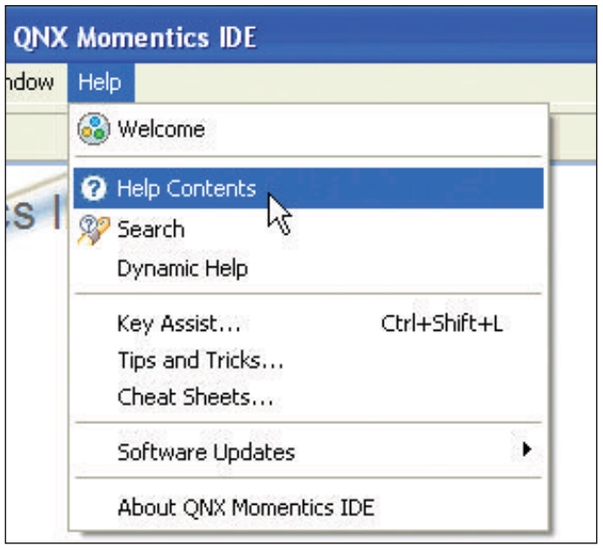
10 Making the program your own

To turn the whole thing into your own QNX program, you can modify and extend the source code we just created. Try some of our sample programs and copy code from them into your project. And, now that you've started, you'll probably want a lot more information, e.g. how to create your own threads, how the QNX Neutrino message-passing works, which process-synchronization methods are available, how to get access to I/O areas, or how to build a QNX Neutrino resource manager. But don't worry: all this is (almost) as simple as the quick start you just experienced!

The IDE includes a number of tutorials to help you get started. Choose **Help** > **Welcome** from the IDE's toolbar, and then click the Tutorials icon:



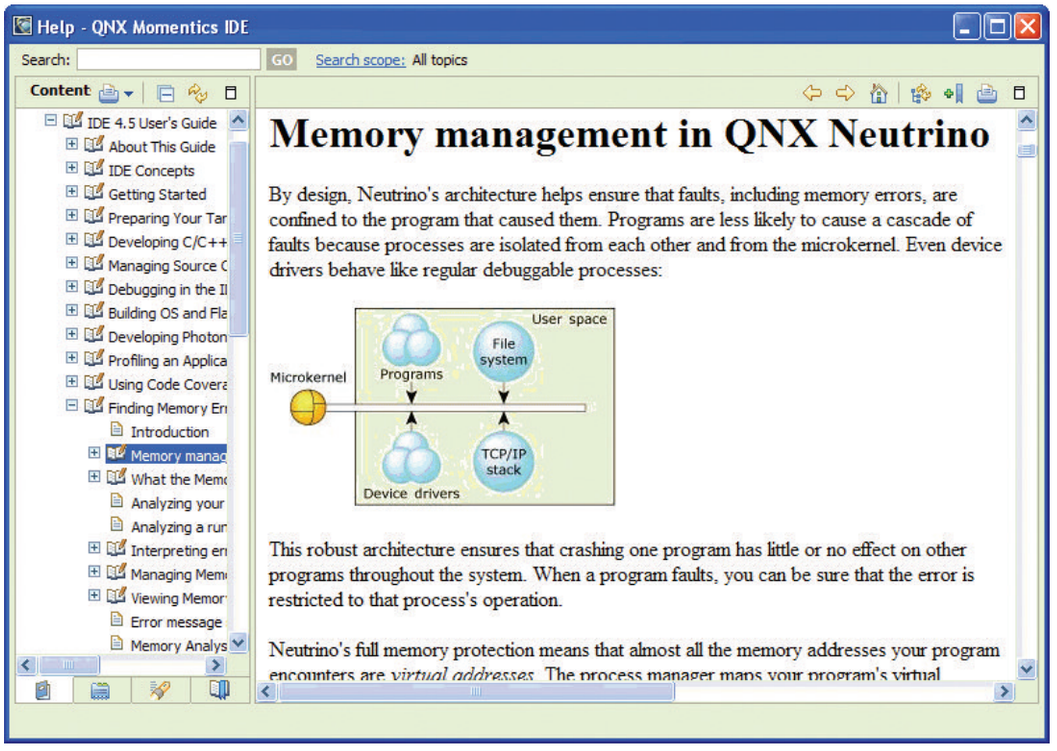
The IDE's Help system includes the QNX documentation, along with information about the Eclipse platform. In the **Help** menu, click **Help Contents**:



The *Welcome to the QNX Software Development Platform* guide helps you find out where to look for the information you need. We recommend browsing the QNX Neutrino RTOS *System Architecture* guide, the IDE *User's Guide*, and the QNX Neutrino RTOS *Programmer's Guide*.

In the Help window, please choose:

- **A Roadmap to the QNX Software Development Platform** for information about the QNX Momentics Tool Suite and the QNX Neutrino RTOS.
- **IDE User's Guide** for information about using the IDE



To view the documentation on self-hosted systems, just click the **Help** button on the icon bar on the right of the screen. Printed documentation is also available.

Source code examples covering thread creation, usage of mutexes, message-passing and other methods of interprocess communication as well as a QNX resource-manager template can be found on our FTP server **`ftp.qnx.de`**; just use an FTP client to log in as user **`qnx`** with the password **`qnx!neutrino`** and download the source package. It contains a **readme.txt** that explains how to import the source examples into the QNX Momentics IDE. The source features extensive comments, explaining what is done there. For every function you are interested in, you also should consult the *QNX Neutrino RTOS Library Reference*.

For more assistance

While you explore the QNX Momentics Tool Suite and the QNX Neutrino RTOS, you will probably have further questions. Please contact your QNX Account Manager, Field Application Engineer, or our support department. We want to be with you from the start, because we are successful only if you are!

QNX Software Systems
www.qnx.com
info@qnx.com
+1 613 591 0931

Corporate Headquarters
175 Terence Matthews Crescent
Ottawa, Ontario
Canada, K2M 1W8

North America
t: +1 800 676-0566
f: +1 613 591-3579

International
t: +1 613 591-0931
f: +1 613 591-3579

Online
info@qnx.com
www.qnx.com

QNX SOFTWARE SYSTEMS

002497 MC502.03
Printed in Canada