

# NetBeans Mobility Pack for CDC 5.5.1 Quick Start Guide

## Introduction

---

A CDC application is an application that is meant to run on a hand-held or wireless device that has at least 2MB of memory. This configuration supports a more feature-rich JVM than cellular phones, which typically have 128 to 512KB of memory and use the Connected Limited Device Configuration (CLDC).

This document takes you through the basics of using NetBeans Mobility Pack for CDC 5.5 to create a Java Platform, Micro Edition (Java ME platform), Connected Device Configuration (CDC) application. The guide takes you through some of the basic steps of creating a new CDC Project. We will show you four ways to create a Java ME CDC project that displays a simple form in a device emulator; one way for each profile, and one way for Mobile Windows devices. This document is designed to get you going as quickly as possible.

The sections in this guide are:

- [Software Requirements](#)
- [Adding Emulator Platforms](#)
- [Creating a CDC Application](#)
  - [Foundation Profile](#)
  - [Personal Profile](#)
  - [AGUI Profile](#)
  - [Windows CE Application](#) (using the NSIcom CrEme VM)
- [Next Steps](#)

## Software Requirements

---

The software and hardware requirements for the CDC Mobility Pack are listed in the [NetBeans Mobility Pack for CDC 5.5.1 Release Notes](#).

## Adding Emulator Platforms

---

The NetBeans Mobility Pack supports the following emulator platforms:

- Sun Java Toolkit for CDC 1.0
- Ricoh Embedded Software Architecture Emulator 1.14c
- UIQ SDK 3
- Sony Ericsson M600 and P990 devices
- Nokia Series 80 Platform SDK for Symbian OS, for Java, Personal Profile
- NSI.com CrEME emulator
- SavaJe Operating System

Instructions for adding emulator platforms are listed in the [NetBeans Mobility Pack for CDC 5.5.1 Installation Guide](#).

## Creating a CDC Application

---

In this section we will create a CDC application project for each available profile and, for the Personal and AGUI platforms, create a GUI for the application using the IDE's GUI Builder. Finally we will create an application for Windows CE using the NSIcom CrEme VM.

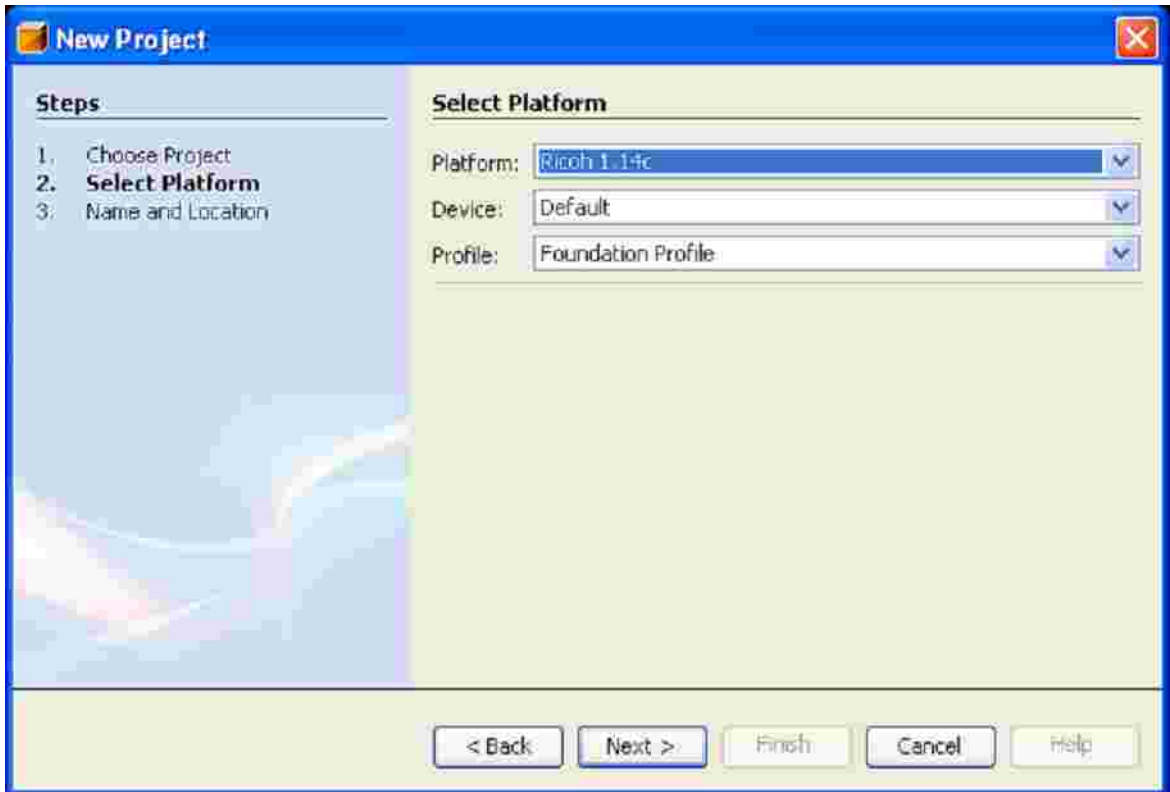
The Profiles described are:

- [Foundation Profile \(Ricoh\)](#)
- [Personal Profile \(Nokia, Sony Ericsson\)](#)
- [AGUI \(Sun Java Toolkit, SavaJe\)](#)
- [Windows CE \(NSIcom CrEme\)](#)

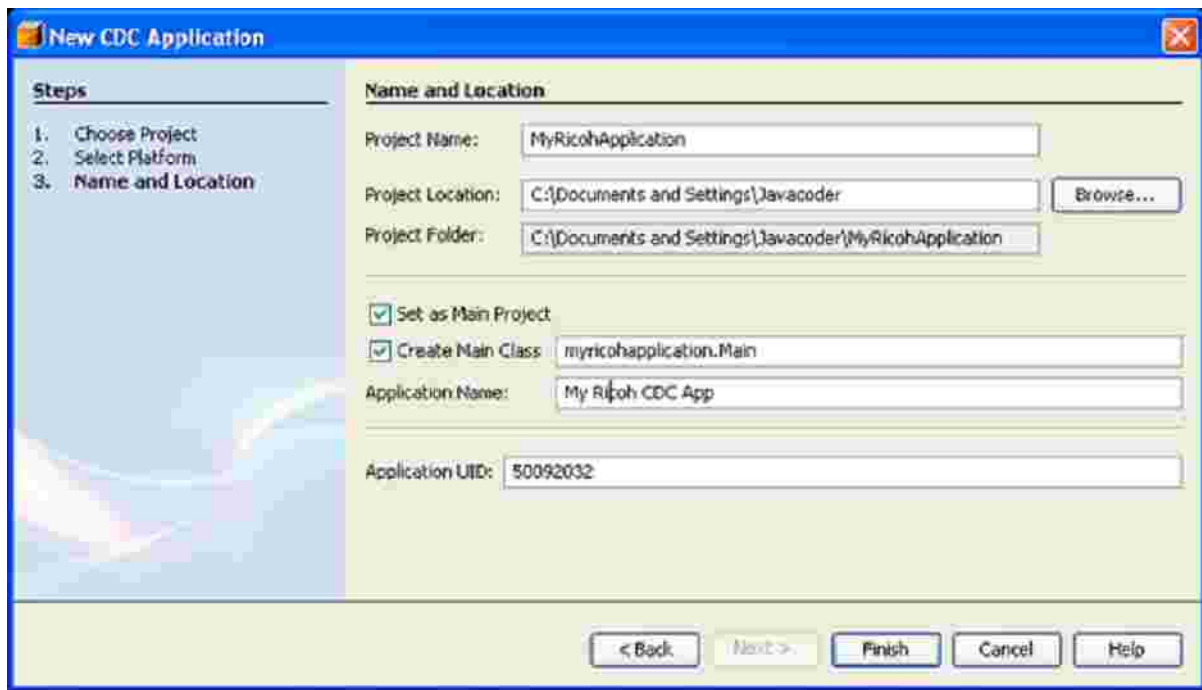
## Creating a New CDC Application - Foundation Profile

Here we will create a CDC Application, or Xlet, for the Foundation Profile using the Ricoh Embedded Software Architecture Emulator 1.14c.

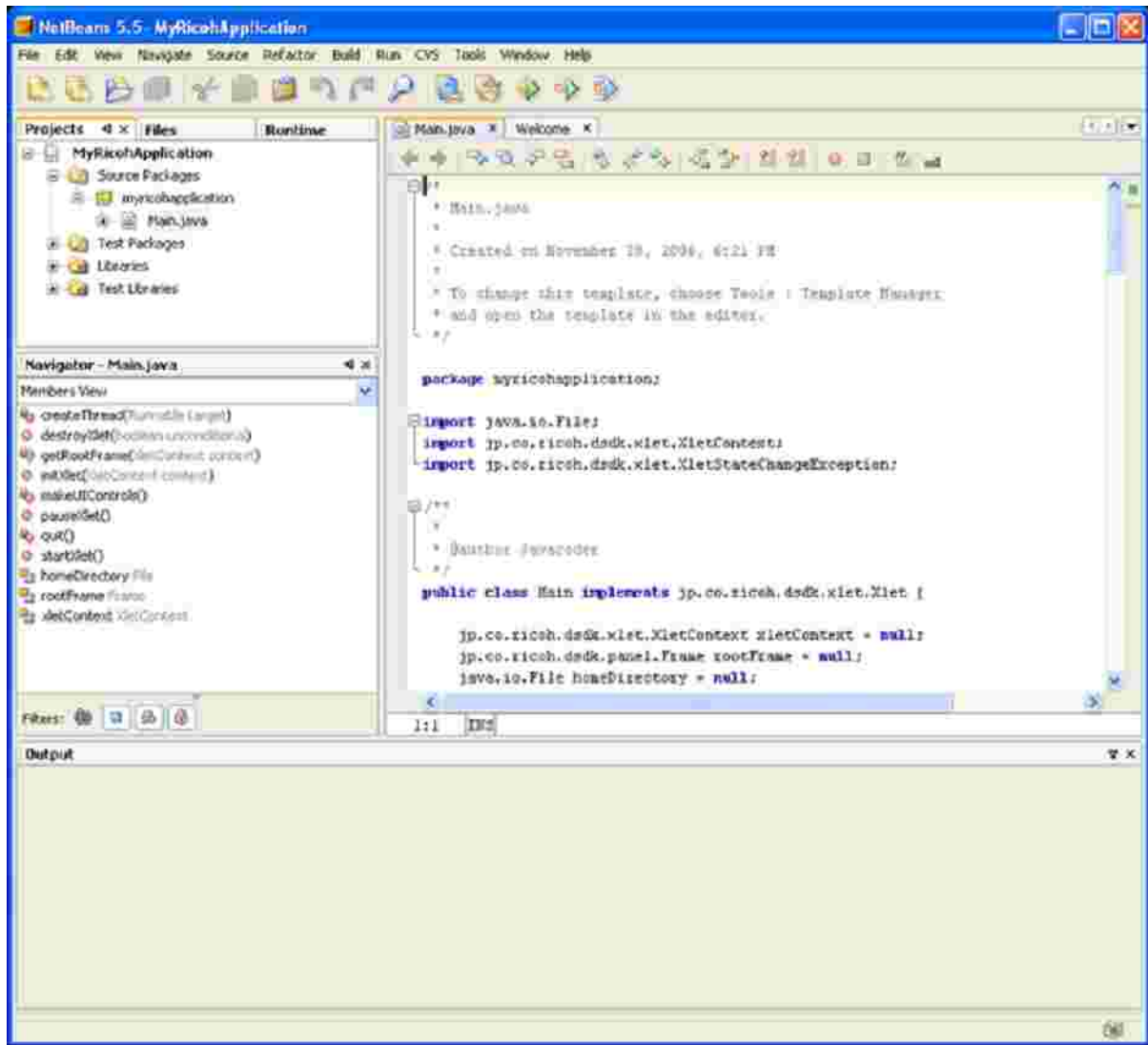
1. Choose File > New Project (Ctrl-Shift-N) from the main window.
2. Under Categories, select CDC. Under Projects, select the CDC Application template. Click Next.
3. In the Select Platform Page, choose Ricoh 1.14c.



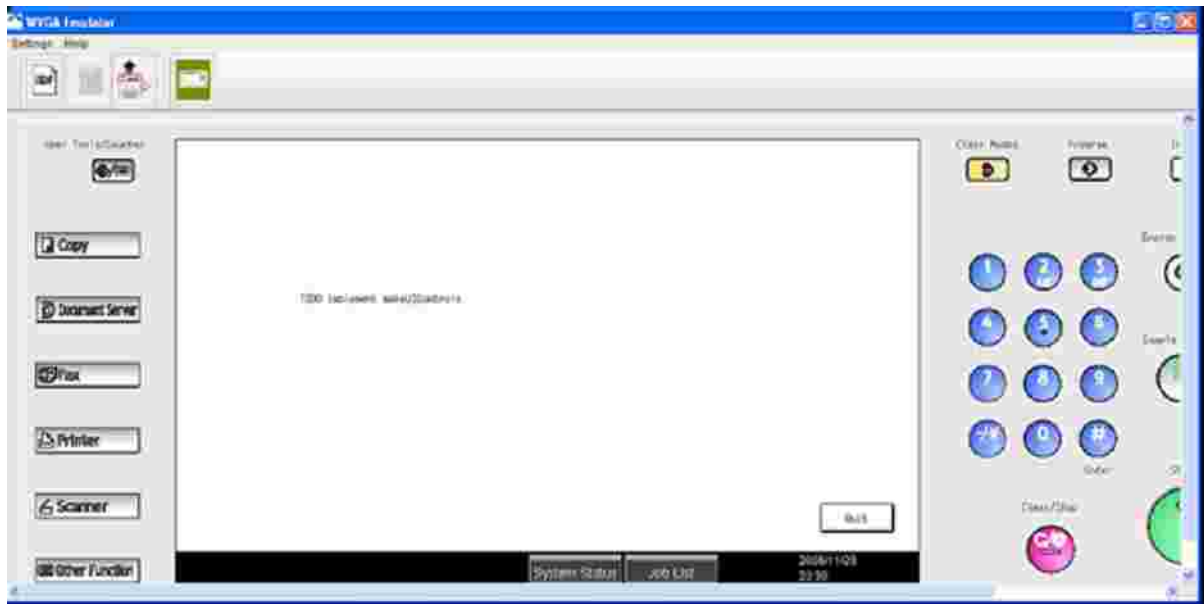
4. In the Name and Location panel, name the project `MyRicoApplication` and specify a location for the project on your computer. Leave the Create Main Class checkbox selected.
5. Leave the UID as generated.
6. Specify the Application Name, which is the human-readable name of the application. In our example, we will enter `My Ricoh CDC App` for this field.



7. Click Finish. The IDE creates the new application and opens the main class in the Source Editor. The project structure is shown in the following figure. Note that the generated code is based on the profile. In this case, the main class is derived from `jp.co.ricoh.dsdk.xlet.Xlet`. This profile does not support the Netbeans GUI Designer.



8. To run the project, choose Run > Run Main Project. The emulator displays the text, "TODO implement makeUIControls."

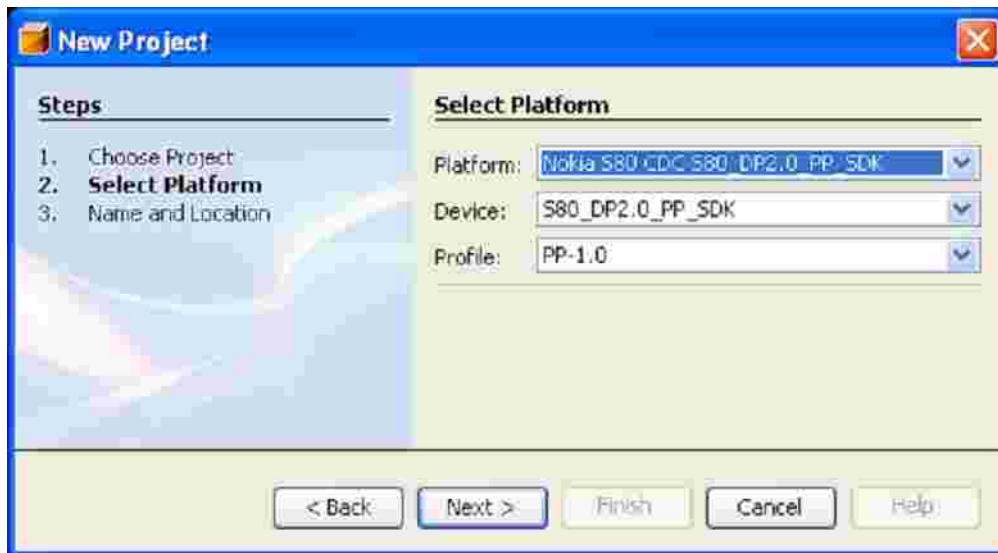


9. You can also debug the project, and create and debug tests using the JUnit framework.

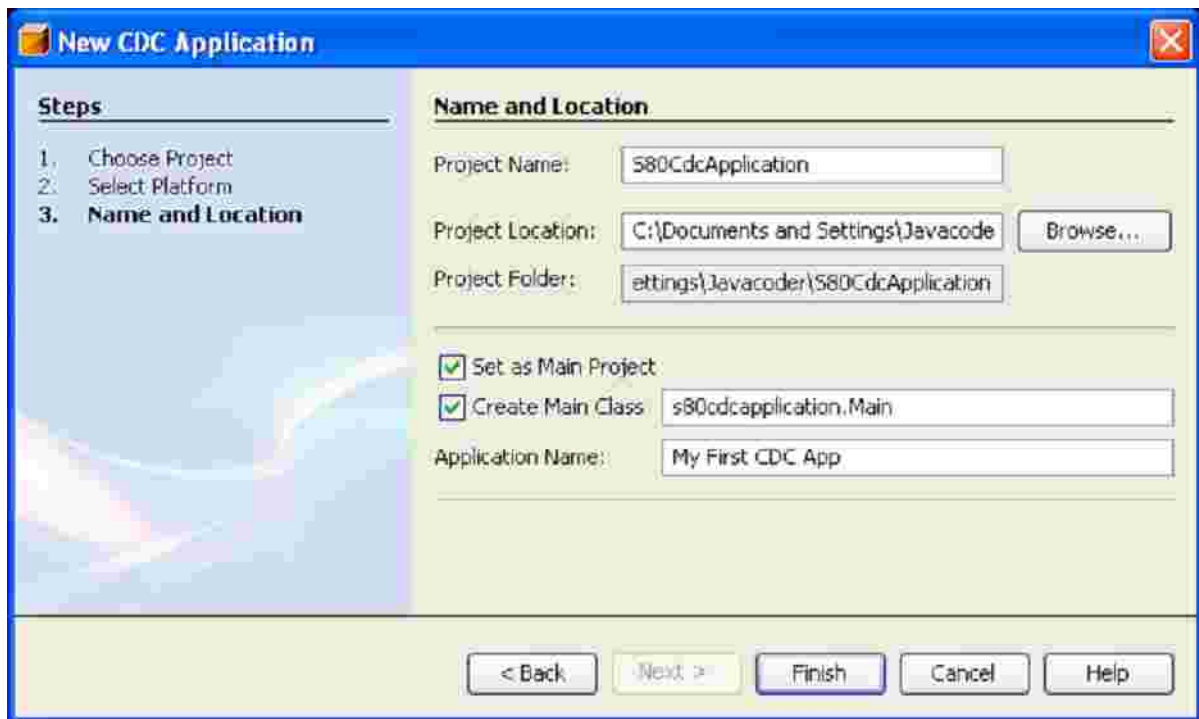
### Creating a New CDC Application - Personal Profile

Here we will create an applet for the Personal Profile using the Nokia Series 80 Platform SDK for Symbian OS, for Java, Personal Profile. The Sony Ericsson platform also supports the Personal Profile.

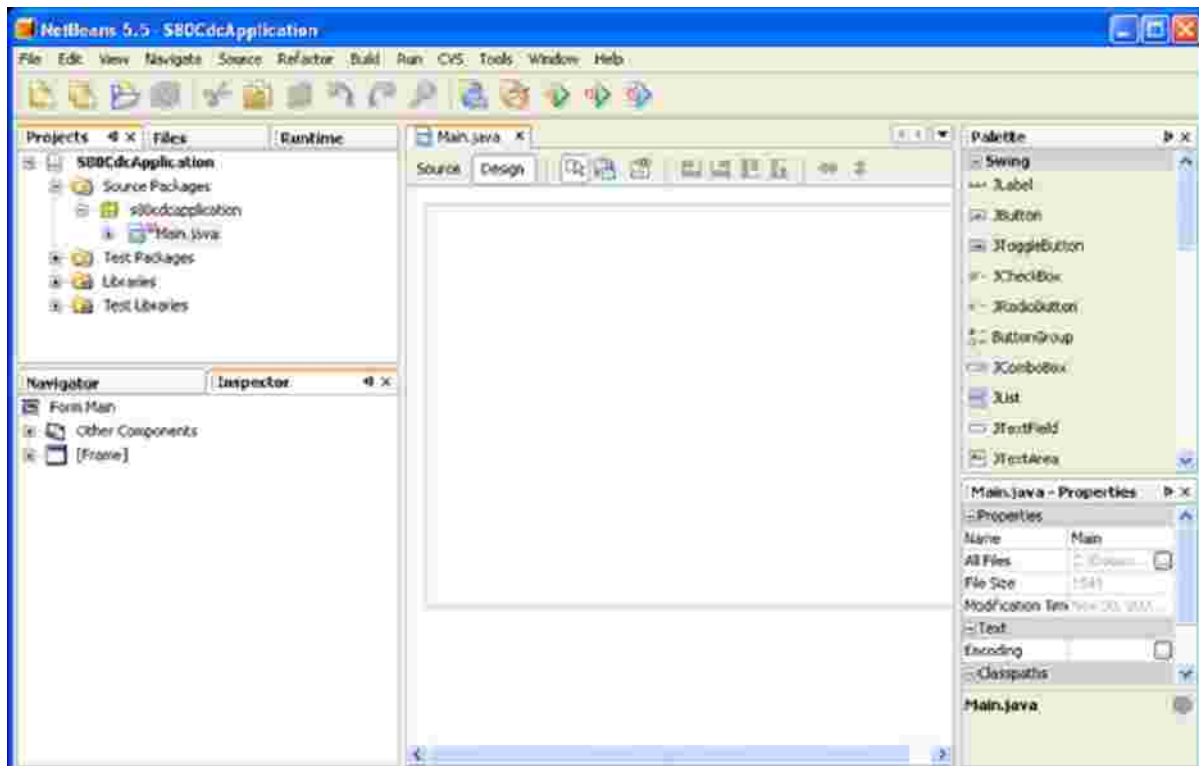
1. Choose File > New Project (Ctrl-Shift-N) from the main window.
2. Select the CDC Application template from the CDC Platform category. Click Next.
3. In the Select Platform Page, choose the Nokia S80 Platform. Click Next.



4. Name the project S80CdcApplication and specify a location for the project on your computer. Leave the Create Main Class check box selected.
5. Specify the Application Name, which is the human-readable name of the application. In our example, we will enter My First CDC App for this field.



6. Click Finish. The IDE creates the new application and opens the main class form in the GUI Builder, as shown in the following figure. Note that the generated code is based on the profile. In this case, the main class is derived from `java.awt.Frame`.



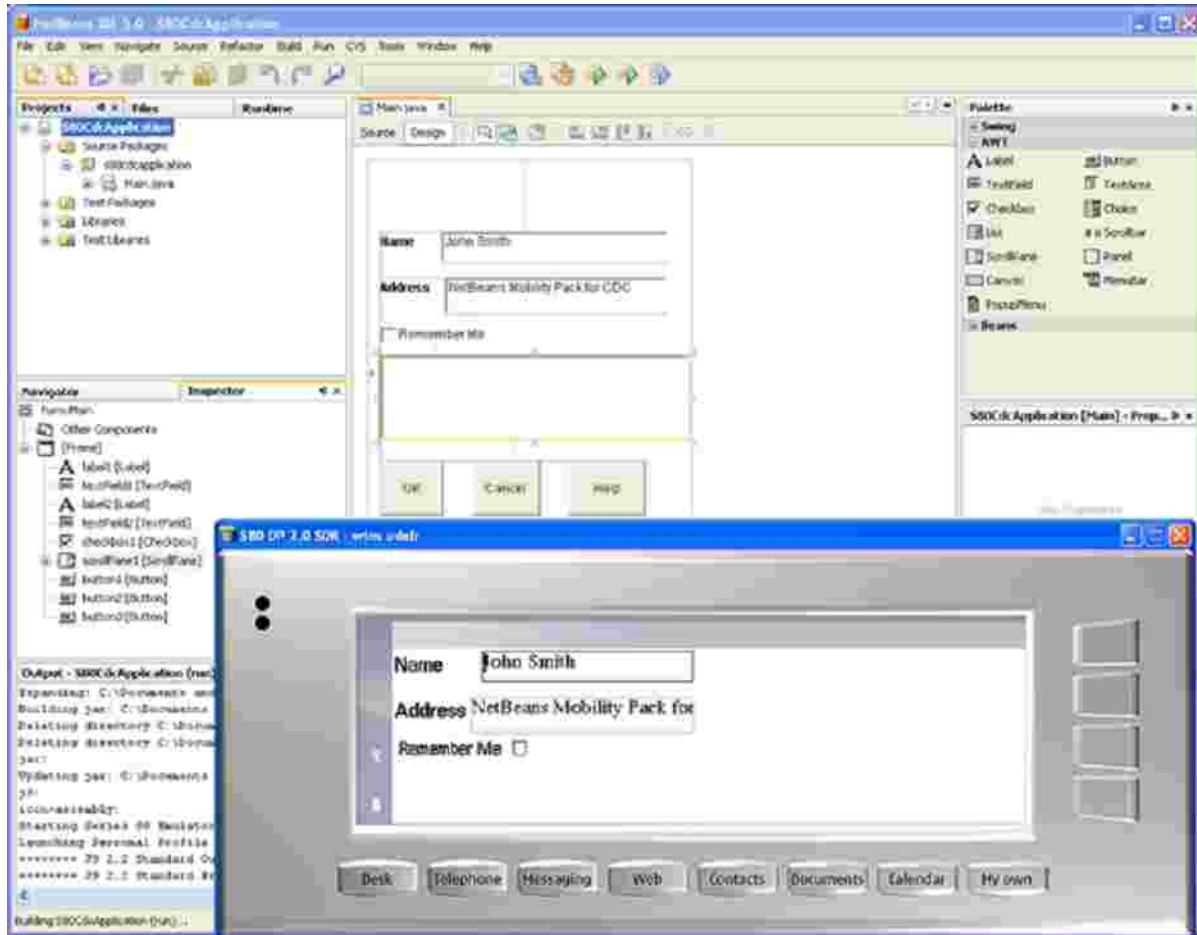
### Creating the Application Interface Using the Project Matisse GUI Builder

You can use the Project Matisse GUI Builder in the same way you use it for regular J2SE development. In the GUI Builder, right-click the `Main.java` form and choose `Set Layout > Free Layout`. Then drag and drop components from the Palette window into the Design Area of the GUI Builder. Make sure you only use AWT components in your

form. Because all the Nokia Series 80 devices support the Personal Profile, only AWT widgets are available.

For more information on using the IDE's GUI Builder, see the [Java GUIs and Project Matisse Learning Trail](#).

When you are done, right-click the project in the Projects window and choose Run Project. Your application should be displayed in the Applications menu in the device emulator. You can now run the device in the emulator.



You can also debug the project, and create and debug tests using the JUnit framework

### Building Distribution Files

You're now ready to finish your application. Right-click the project and choose Build Target Bundle. The IDE creates the SIS file in the `PROJECT_HOME/target` folder. You can see this file in the Files window. If you specified additional parameters, like certificate, private key, and password, the IDE also signs the SIS file for you.

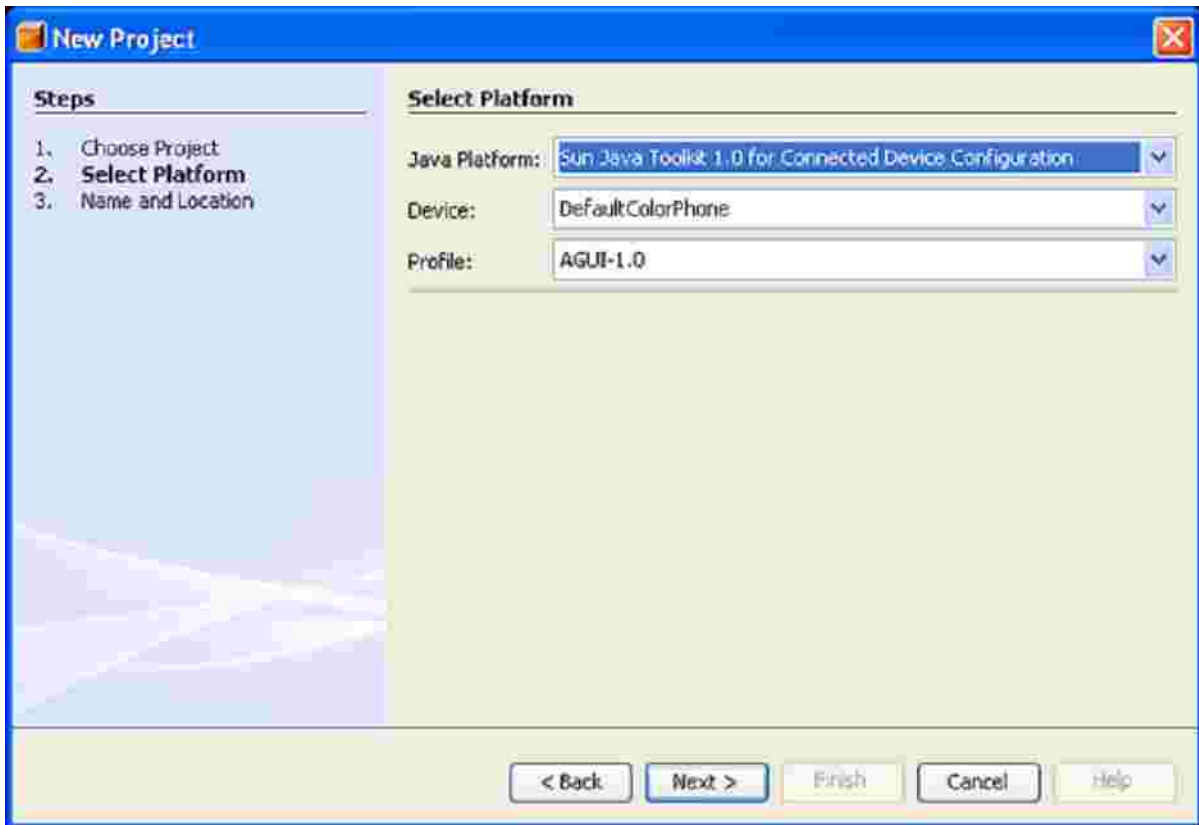
To build a Nokia project, use the Build SIS command as well. While doing an SIS build for Nokia is not supported yet, you can deploy the application using the JAR file created under `PROJECT_HOME/dist`.

### Creating a New CDC Application - AGUI Profile

Here we will create a CDC Application, or Xlet, for the AGUI Profile using the Sun Java Toolkit for CDC 1.0. The AGUI Profile is also supported by the SavaJe OS platform.

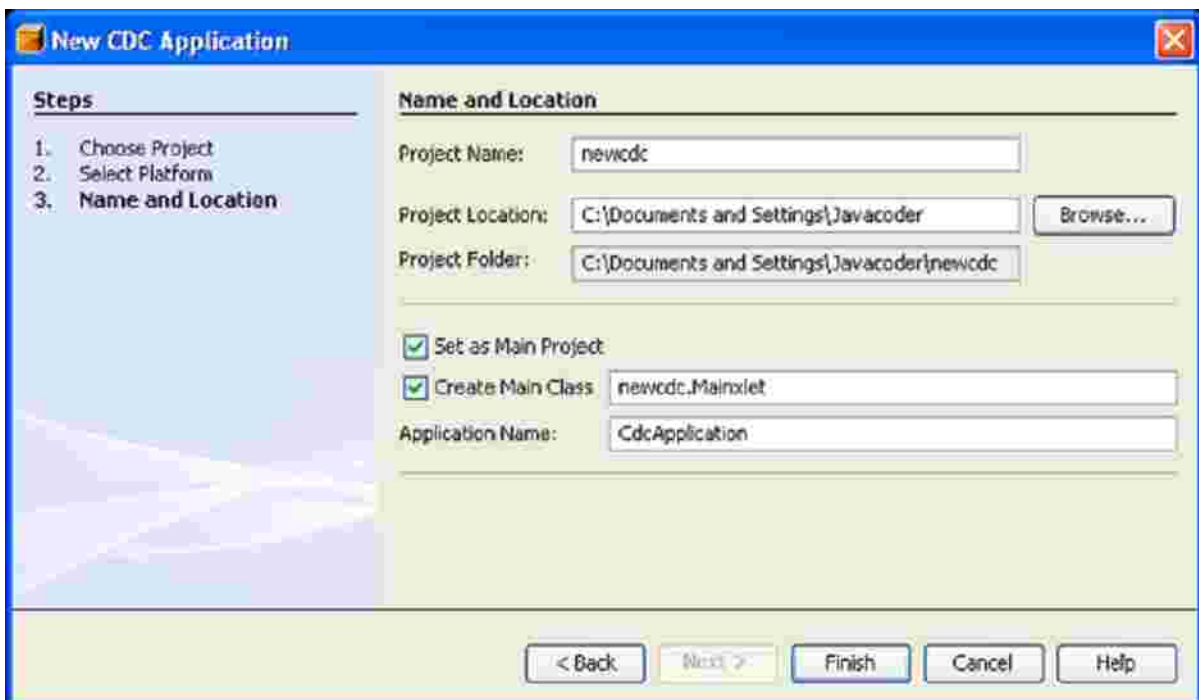
Note: For more information about developing for the SavaJe platform, see "[Developing for the SavaJe Platform](#)."

1. Choose File > New Project (Ctrl-Shift-N) from the main window.
2. From Categories, select CDC. From Projects, select CDC Application. Click Next.
3. In the Select Platform page, use the drop-down menu to choose `DefaultColorPhone` as the Device. Leave the Platform and Profile settings as they are.



4. In the New CDC Application panel:

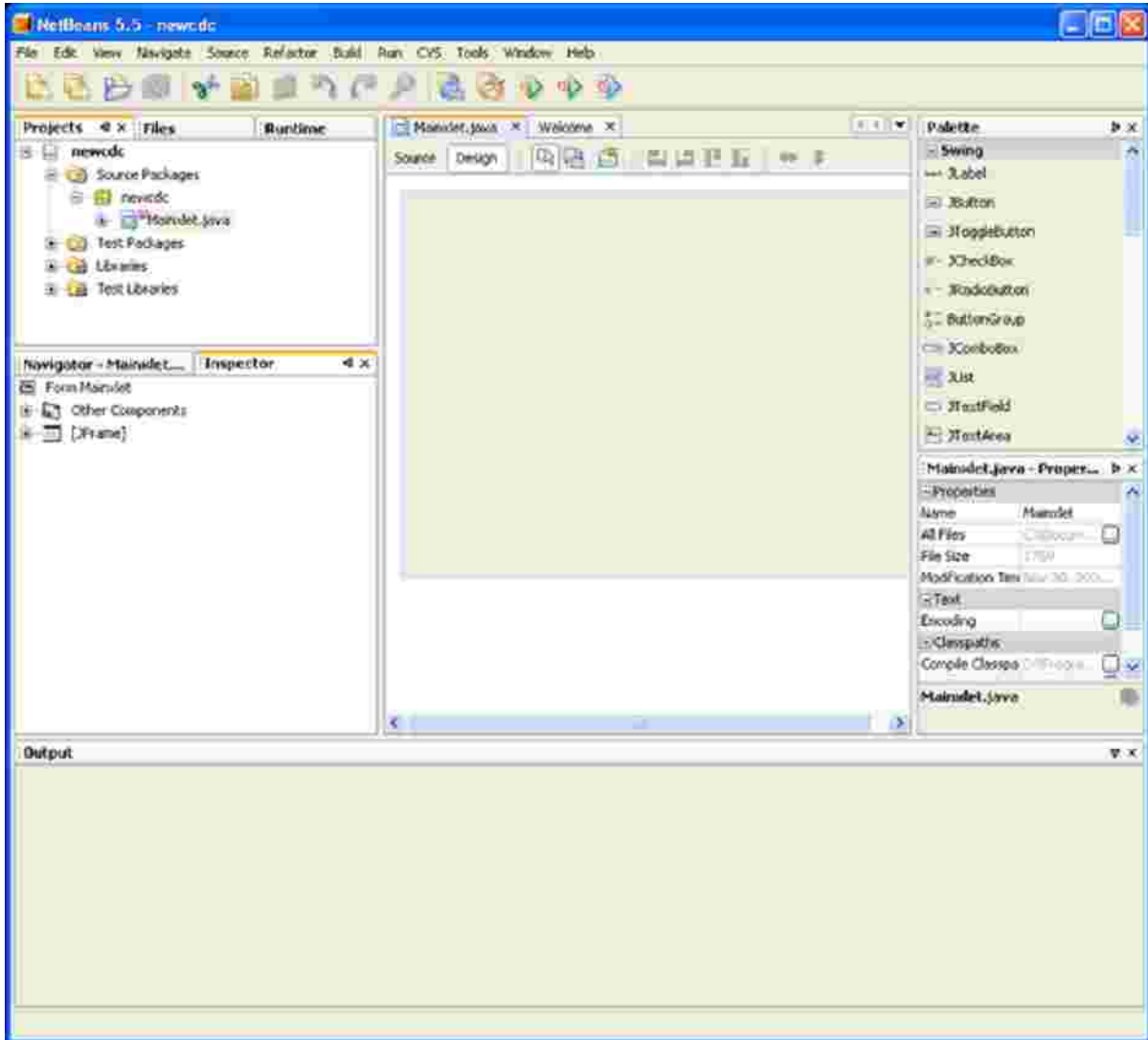
- Name the project newcdc .
- You do want to create a Main class, even though it's really an Xlet class and not a Java SE class, with `main()`.
- Change the package/classname to `Mainxlet`.
- The Application Name is the name that will show up as the bundle title (on the My Application Menu).



5. Click Finish. The IDE creates the new application and opens the main class form in the GUI Builder, as shown in



the following figure.

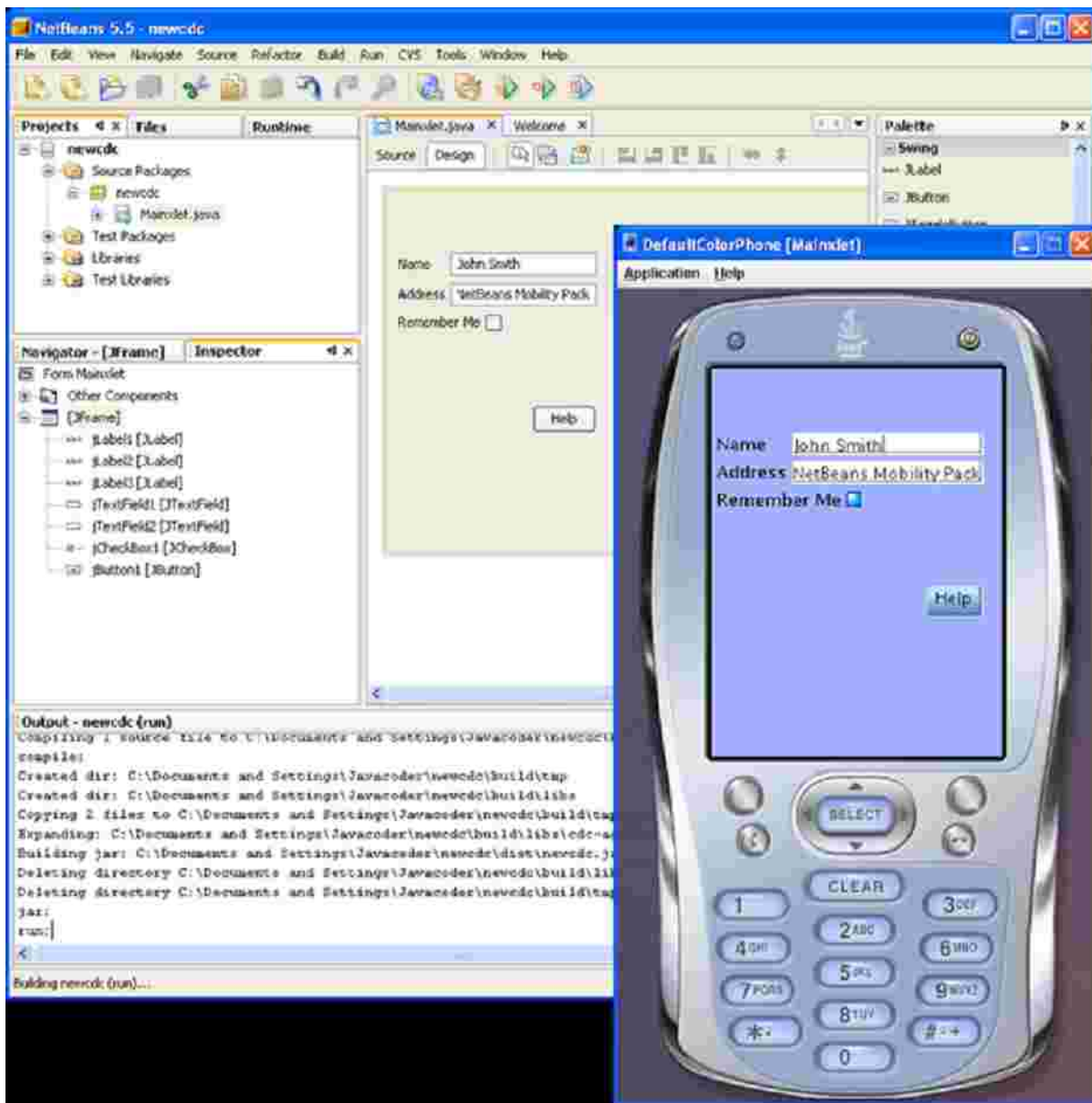


### Creating the Application Interface Using the Project Matisse GUI Builder

You can use the Project Matisse GUI Builder in the same way you use it for regular J2SE development. In the GUI Builder, right-click the Main.java form and choose Set Layout > Free Layout. Then drag and drop components from the Palette window into the Design Area of the GUI Builder. Make sure you only use Swing components in your form. Because the AGUI platform supports Swing, only Swing widgets are available.

For more information on using the IDE's GUI Builder, see the [Java GUIs and Project Matisse Learning Trail](#).

When you are done, right-click the project in the Projects window and choose Run Project. Your application should be displayed in the device emulator.



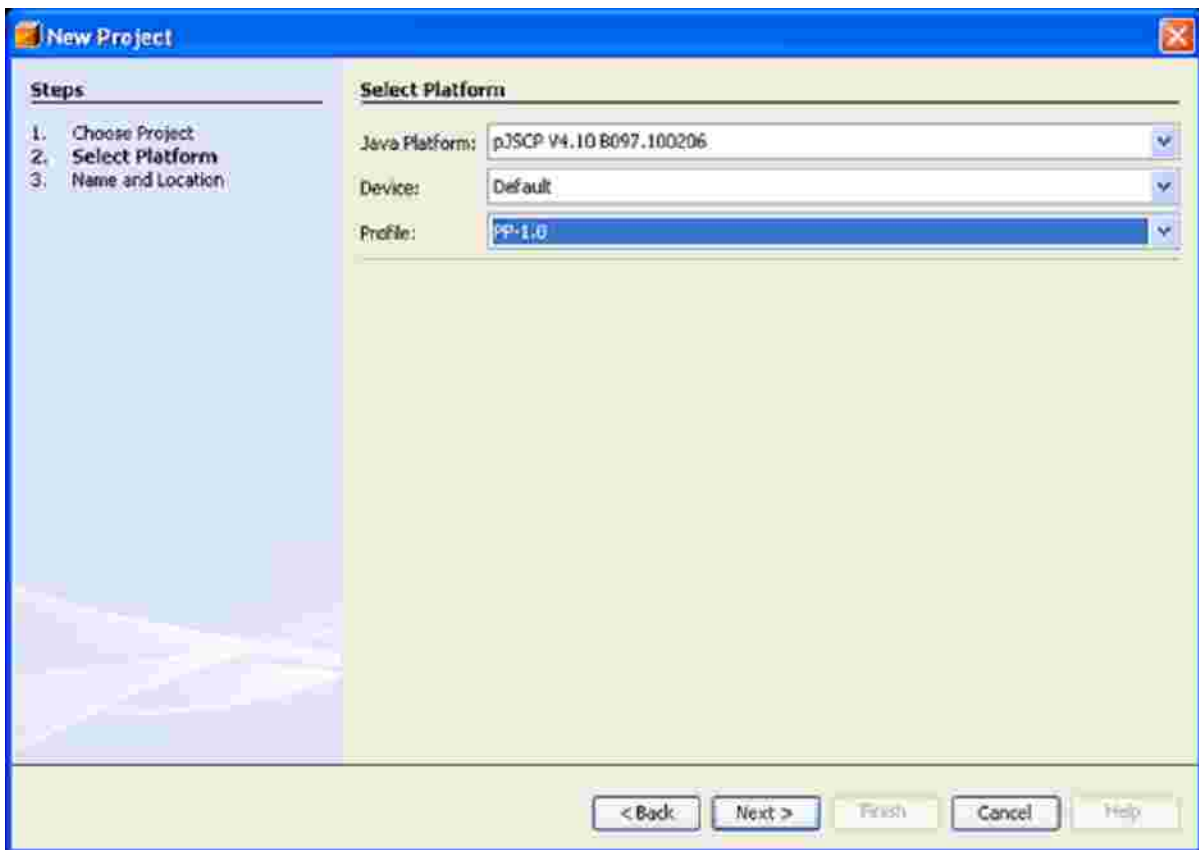
You can also debug the project, and create and debug tests using the JUnit framework.

## Creating a New CDC Application for Windows CE (Using the CrEme VM)

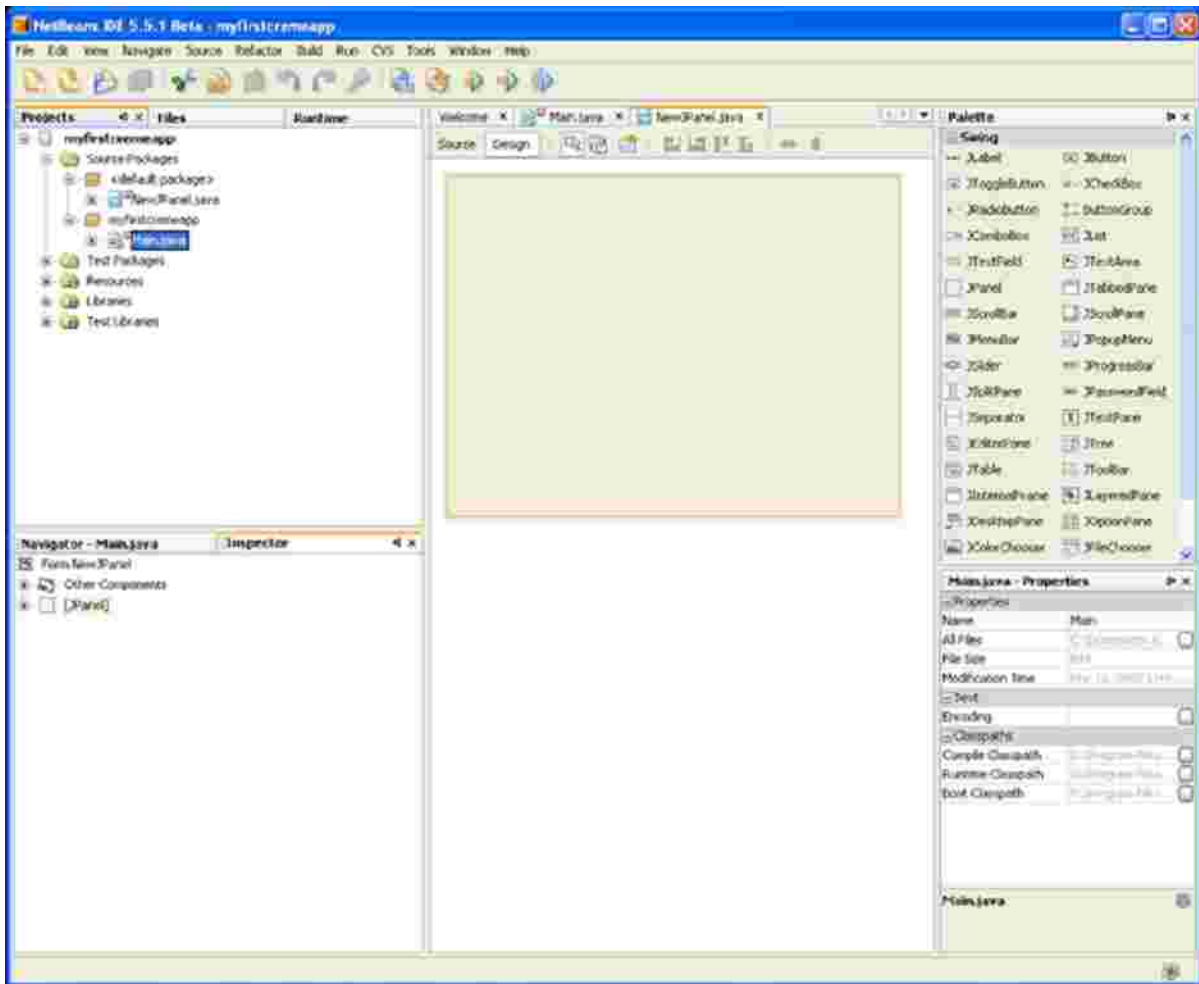
The NSIcom CrEme VM enables you to create and run Java Swing applications for devices running Windows CE.

### Creating an Application

1. Choose File > New Project (Ctrl-Shift-N) from the main window.
2. Under Categories, select CDC. Under Projects, select the CDC Application template. Click Next.
3. In the Select Platform Page, choose pJSCP V4.10. Click Next.



4. Name your project (for example, myfirstcremeapp). Click Finish.
5. In the Explorer window, right-click on the Source Packages node of your project and choose New > JFrame form. Click Finish. The NetBeans GUI Builder (Matisse) displays the new form.



Note: Make sure that the first fragment of code in the `main()` method is as is shown below when you are going to run the application in CrEme default emulator. This will make sure that Swing is loaded.

```
try {
    UIManager.setLookAndFeel(UIManager.getCrossPlatformLookAndFeelClassName());
} catch (Exception exception) {
    System.out.println("Error loading L&F: " + exception);
}
```

6. Use the Project Matisse GUI builder to create your application. You can use the Project Matisse GUI Builder in the same way you use it for regular J2SE development.

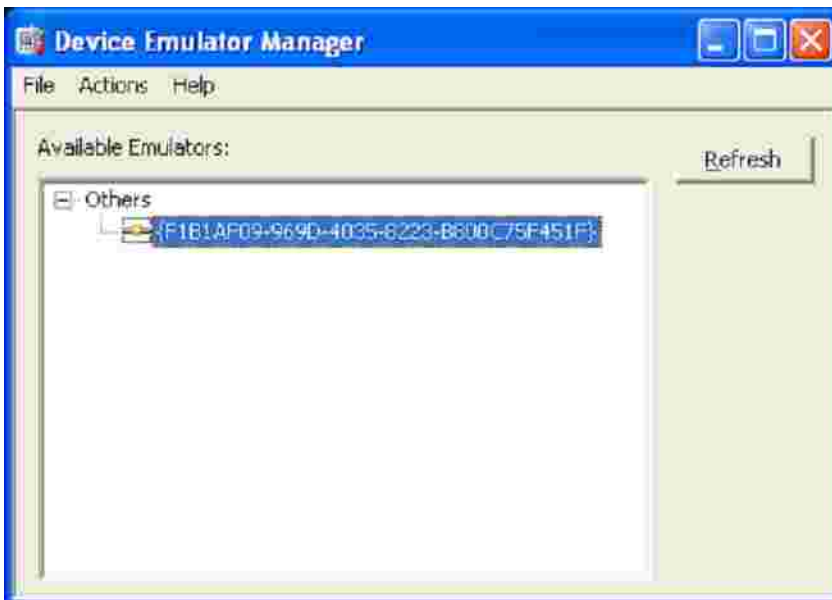
Note that if there is an exception that crashes the emulator, the emulator prints it to the standard output window and waits for a key press. You cannot, however, send the keypress from the IDE. You must instead kill the VM from the Process Explorer.

## Compiling and Running the Application on a Device or Emulator

1. Connect your mobile device using cable or Bluetooth.  
If you have a device, skip to Step 12.  
If you do not have a device, install the Microsoft Device Emulator and the Virtual Machine Network Driver for the Microsoft Device Emulator as described in the [NetBeans Mobility Pack for CDC 5.5.1 Installation Guide](#). Then follow steps 2 through 4.  
If you do have a device, skip to Step 5.
2. Start the Microsoft Device Emulator by choosing Start > Programs > Microsoft Windows Mobile 5.0 MSFP Emulator Images > Pocket PC Coldboot. This opens the Pocket PC emulator.



3. Choose Start > Programs > Microsoft Windows Mobile 5.0 MSFP Emulator Images > Device Emulator Manager. This opens the Device Emulator Manager.



4. In the Device Emulator Manager, choose Actions > Cradle.
5. Open ActiveSync by clicking on the icon in the Application tray.  
If you do not have ActiveSync installed, download and install it as described in the [NetBeans Mobility Pack for CDC 5.5.1 Installation Guide](#).

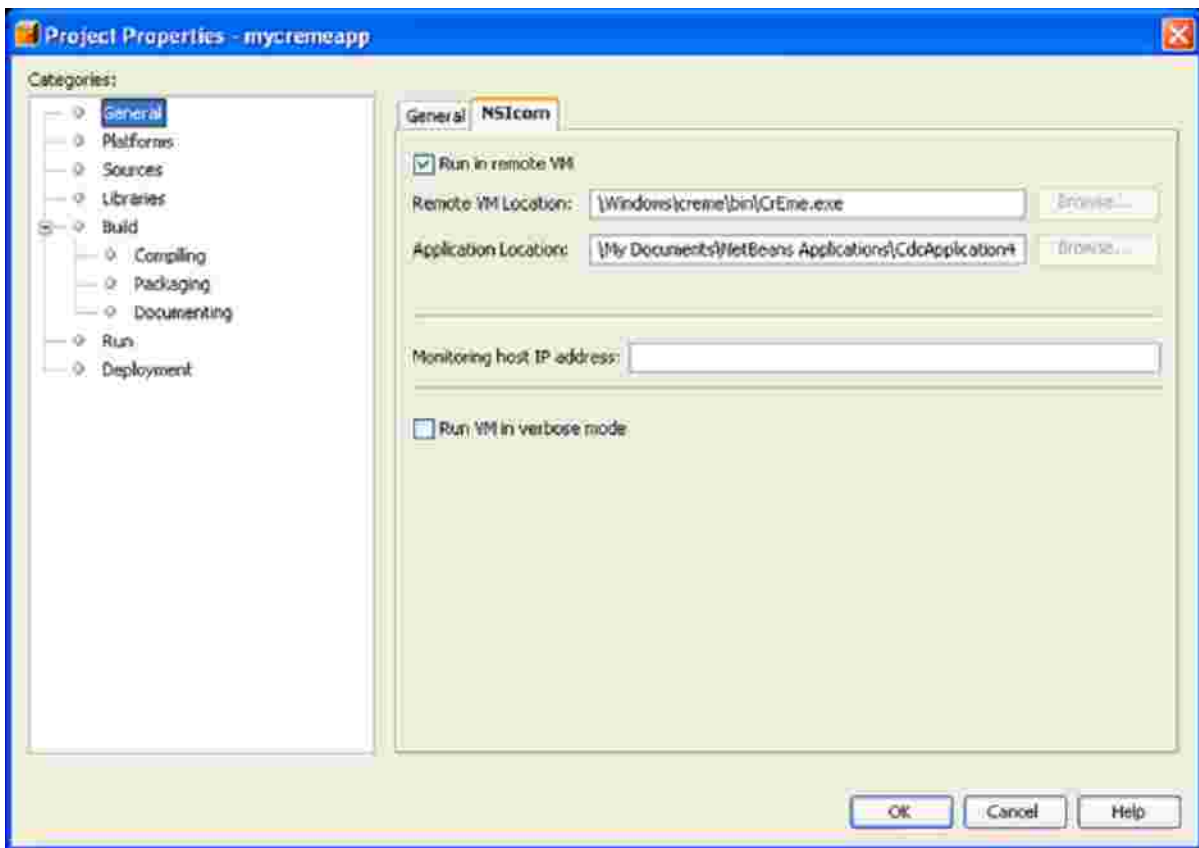
When you open ActiveSync, it should, ActiveSync should detect and connect to the device or the device emulator.

If it does not:

1. Click on the ActiveSync icon in the Application Tray.
2. Choose File > Connection settings.
3. Do one or both of the following:
  - ⊙ Check the Allow connections to one of the following check box and select DMA for the emulator, or USB or port number for a real device.
  - ⊙ Choose Run Connect which should detect the device automatically.
6. Download and install the CrEme VM on your device ([download](#)). Double-click on the downloaded file. It will install both on your PC and your device.
7. Download and install Swing extensions (CrE-ME410\_swing.CAB) for your device from NSIcom ([download](#)).  
To install the Swing extensions on your device or device emulator:
  1. In the ActiveSync dialog, choose Tools > Explore Device.  
The Mobile Device opens in an Explorer window.
  2. Copy CrE-ME410\_swing.CAB into the Explorer window.
  3. In the device or device emulator, choose Start > Programs > File Explorer.
  4. Start CrE-ME410\_swing.CAB.

## Run the Application

1. Right-click the project and choose Properties.  
The Properties page opens.
2. Under Categories, choose General.
3. Choose the NSIcom tab and check Run in remote VM. Click OK to close the Properties page.



4. Choose Run > Run Main Application.

## Next Steps

[Send Us Your Feedback](#)

In addition to the built-in help system, the NetBeans Mobility Pack for CDC is documented through a series of tutorials and articles. For information about the Java ME, CDC development features supported by the Mobility Pack for CDC, see the following documents:

- [NetBeans Mobility Pack Learning Trail](#)
- [NetBeans Mobility Pack for CDC 5.5 Docs and Support Page](#)