Component Development Environment Getting Started Guide

Document Number: CDEGS
Rev 02/2014

freescale™

Contents

Section number **Title** **Page**

Chapter 1
Introduction

Chapter 2
Inherited components

Chapter 3
Exporting and importing components

Chapter 1
Introduction

This document introduces the usage of Component Development Environment (CDE) by creating a software component that is extended using inheritance to make use of a peripheral driver.

The CDE is an Eclipse plug-in included in CodeWarrior Development Studio for MCU V10.x, and supported in any Eclipse-hosted platform (Windows and Linux) version 3.6 or later.

The CDE provides the environment to create, edit, and package embedded components using a graphical interface.

1.1 Overview

An embedded component is a software entity that exposes a specific set of methods, properties, and events providing an abstraction for peripheral I/O and CPUs. An embedded component may also encapsulate/package a software stack or RTOS adapter.

1.2 Terminology

- **Properties**: The initialization state and features supported at runtime by the component; the properties can only be defined at design time.
- **Methods**: These are the functions exposed by the component, available at runtime and are used to set and read the component state or implement behavior.
- **Events**: These are callback functions exposed by the component to attend to asynchronous events such as interrupts.
- **Driver**: A driver contains the source code of all methods and events generated by a component. Every component, except the CPU, has a driver associated with it.

- **Inheritance**: The process that allows using and/or redefining methods and events of another component.
- **Interfaces**: Defines the methods and events needed by a new component that uses the interface.

1.3 Creating embedded component

In this section, you can create an example software component. The software component computes the parity in a data byte, either even or odd, depending on the component's configuration. The properties supported by the component are odd parity or even parity and will add a 1 or a 0 to the data byte.

1. Select **File > New > Embedded Component** from the IDE menu bar. The **Target Project** page appears. Enter the name of the project or select the existing project and click **Next**.
2. The **New Component** page appears (as shown below). Enter the name of the component and select the **Component type** and click **Finish**.

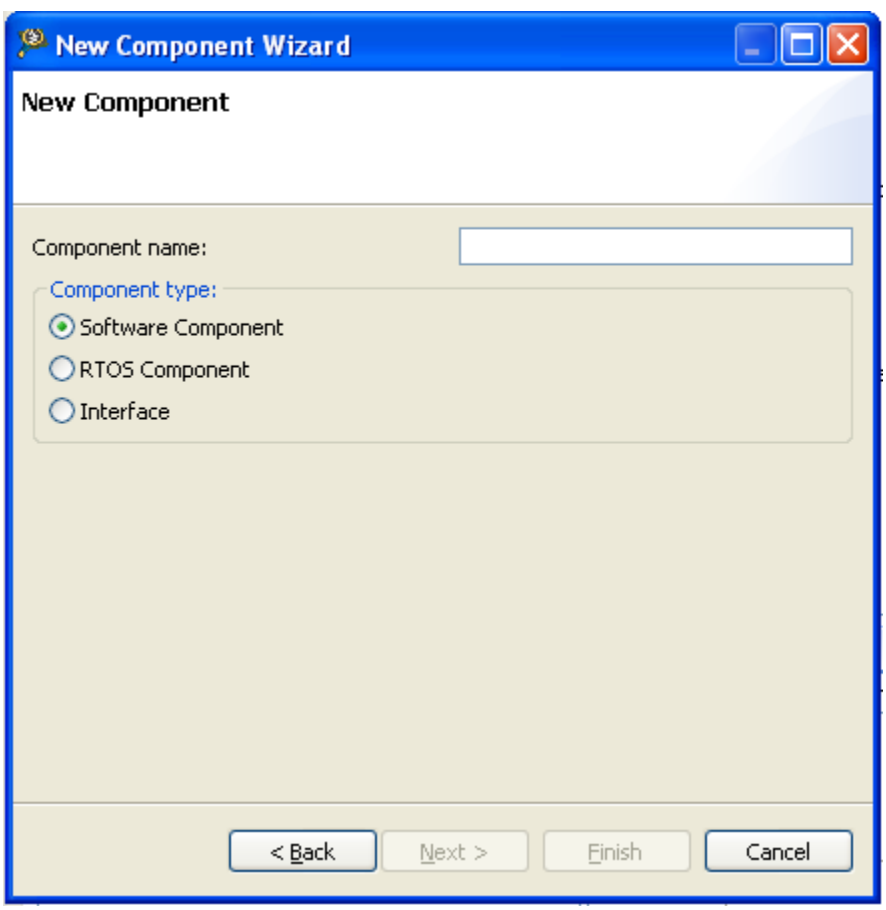

Figure 1-1. New component - create new component screen

The projects are displayed in the Project Explorer and components created in the project are displayed in the **Components** view in the CDE folder. You can create or open as many projects and create or open as many components in the project. You can also create and import new components/interfaces from **Components** view.

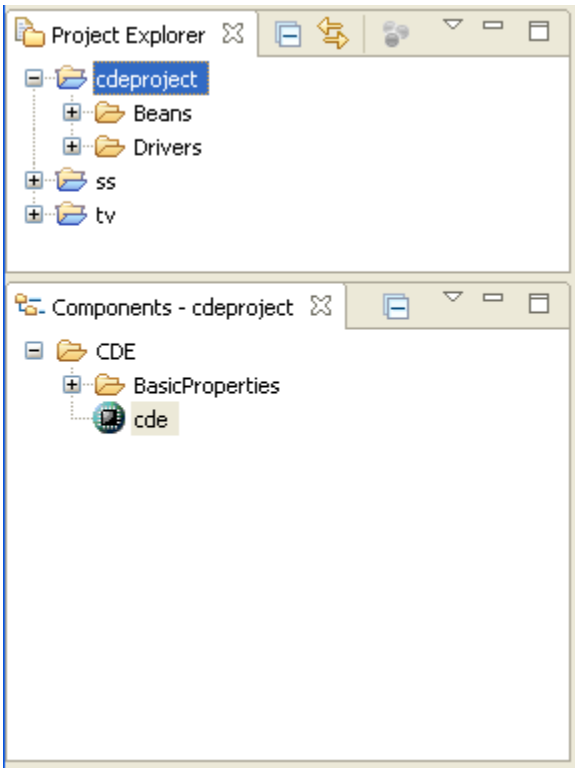

Figure 1-2. Project explorer and component view

The Component editor holds the high level information related to the component, such as author version, the icon to show the copyright information when loaded in Processor Expert. In the component editor, the project name is different from the component name.The Component editor is divided into two panels.

- Left panel: displays internal component structure and all related files.
- Right panel: displays an editor for a selected item from component. Editor is also divided into tabs. The number of tabs depend on the item that is being edited.

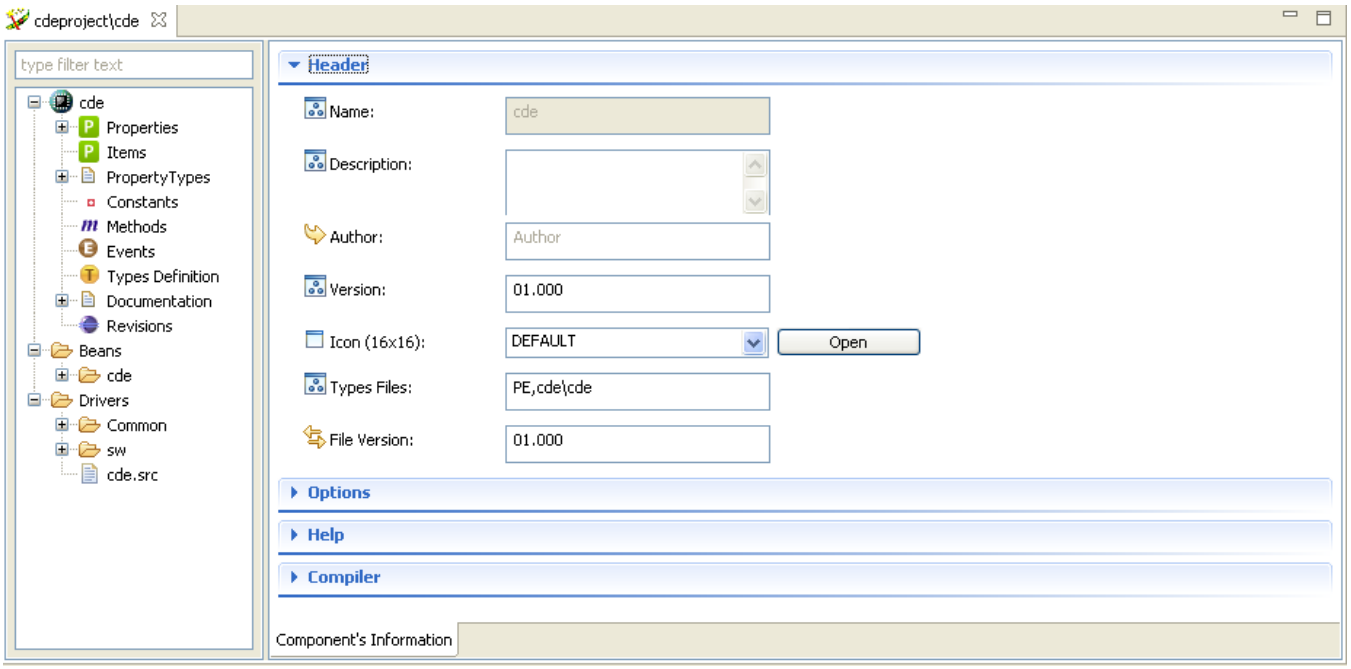

Figure 1-3. Component editor

The Component Development Environment provides an independent and specific editor for each one of the supported features of the component.

- Properties editor
- Property Types editor
- Constants editor
- Methods editor
- Events editor
- Types Definition editor

For more information about each editor and its features, see *Component Development Environment User Guide*.

You will understand how to use the editors to add properties, methods, and events.

1.3.1 Adding properties

You can set one of the properties of the component: even or odd parity.

Right-click on the **Properties** in the Component editor and select **Add Property** option from the context menu to open the **Properties** editor.

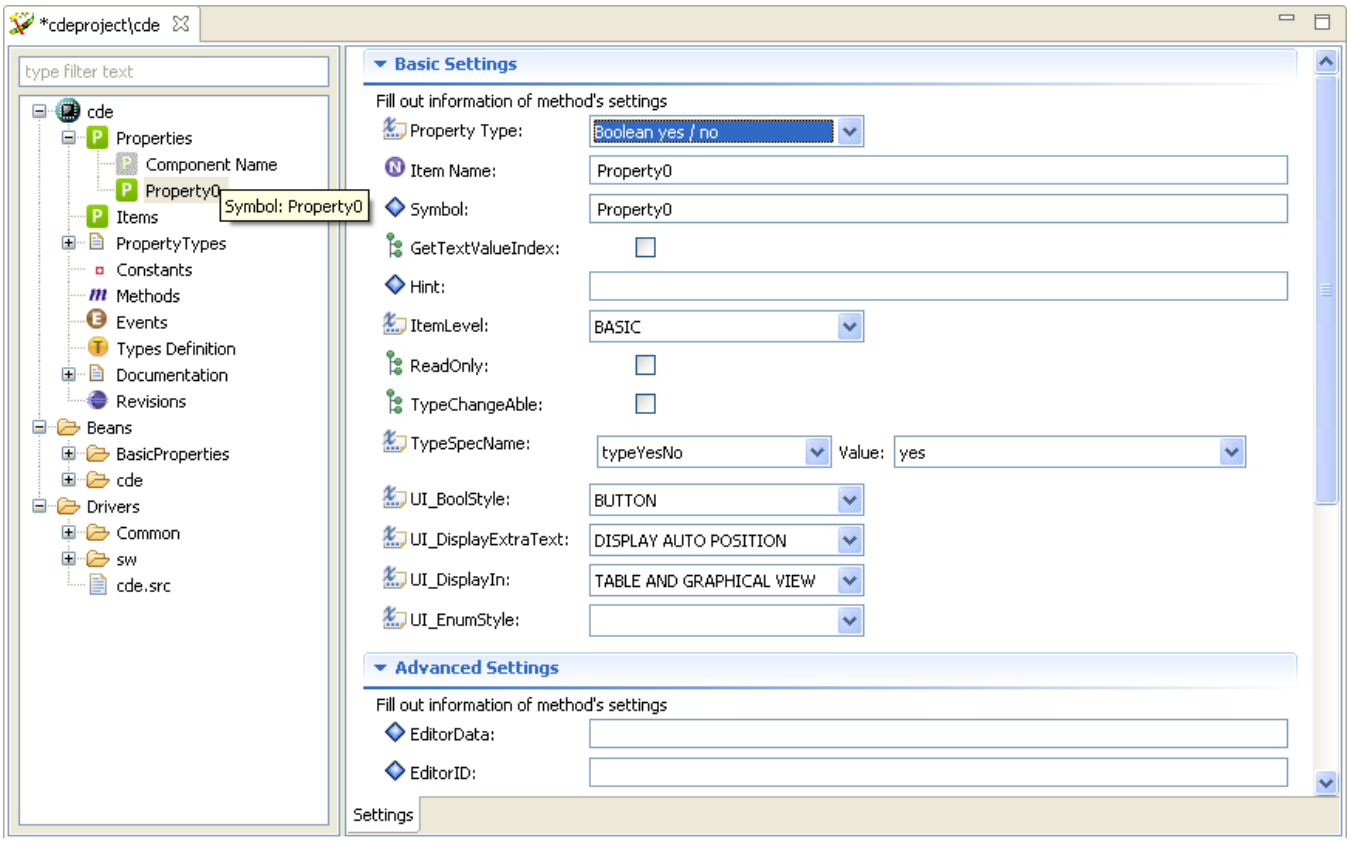

Figure 1-4. Properties editor

The **ItemLevel** item represents the user level at which the item is accessible in the Processor Expert interface. All users can see it. Some properties are for advanced or expert users only.

You can provide text in **Hint** if you wish. This is a context help that pops up over the property when you hover the cursor.

1.3.2 Adding method

The component exposes only one method that calculates the parity bit based on the property you set at design time (even or odd).

Right-click on the **Methods** in the Component editor and select **Add Method** option from the context menu to open the **Methods** editor.

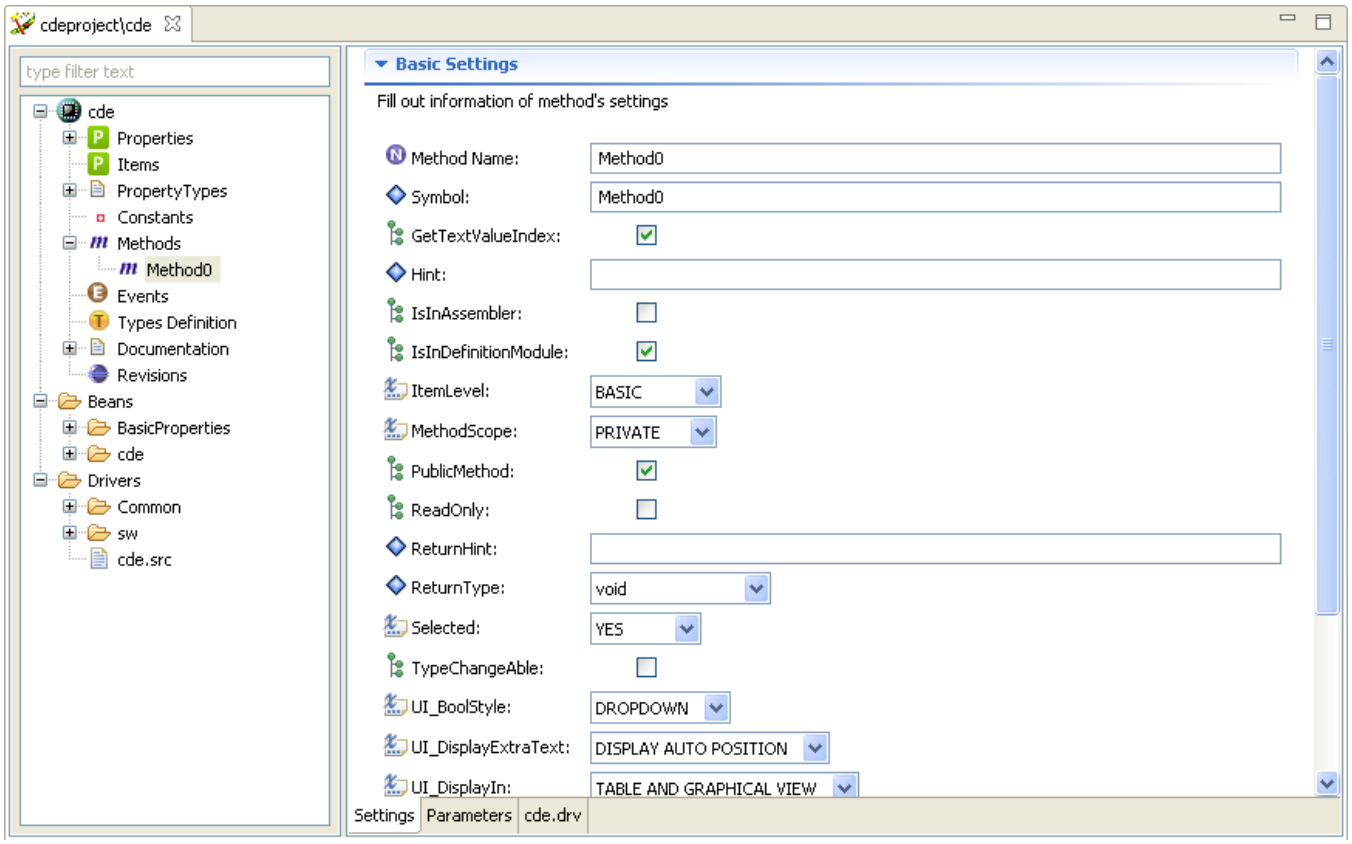

Figure 1-5. Method editor

You can configure the methods of the component. The created methods are listed on the left side, and properties of the selected method are displayed on the right side. To view parameters of a Method; click **Parameters** tab of the Component editor.

1.3.3 Component code generation script

The component ultimately generates code. The CDE provides two ways to specify the generated code, essentially one method at a time, or all at once. The first way is to edit each method independently. The second option is to edit the whole driver using the file editor. Note that the entire body of code to be generated is known as a "driver." And the file that creates that code is a script. You specify the code to be created using the scripting language: either method by method (independent method editor) or in the driver file.

1.3.3.1 File editor

To edit the driver, go to the left panel of the **Component** editor. Navigate to **Drivers -> sw**, and double-click the name of the .drv file you want to edit.

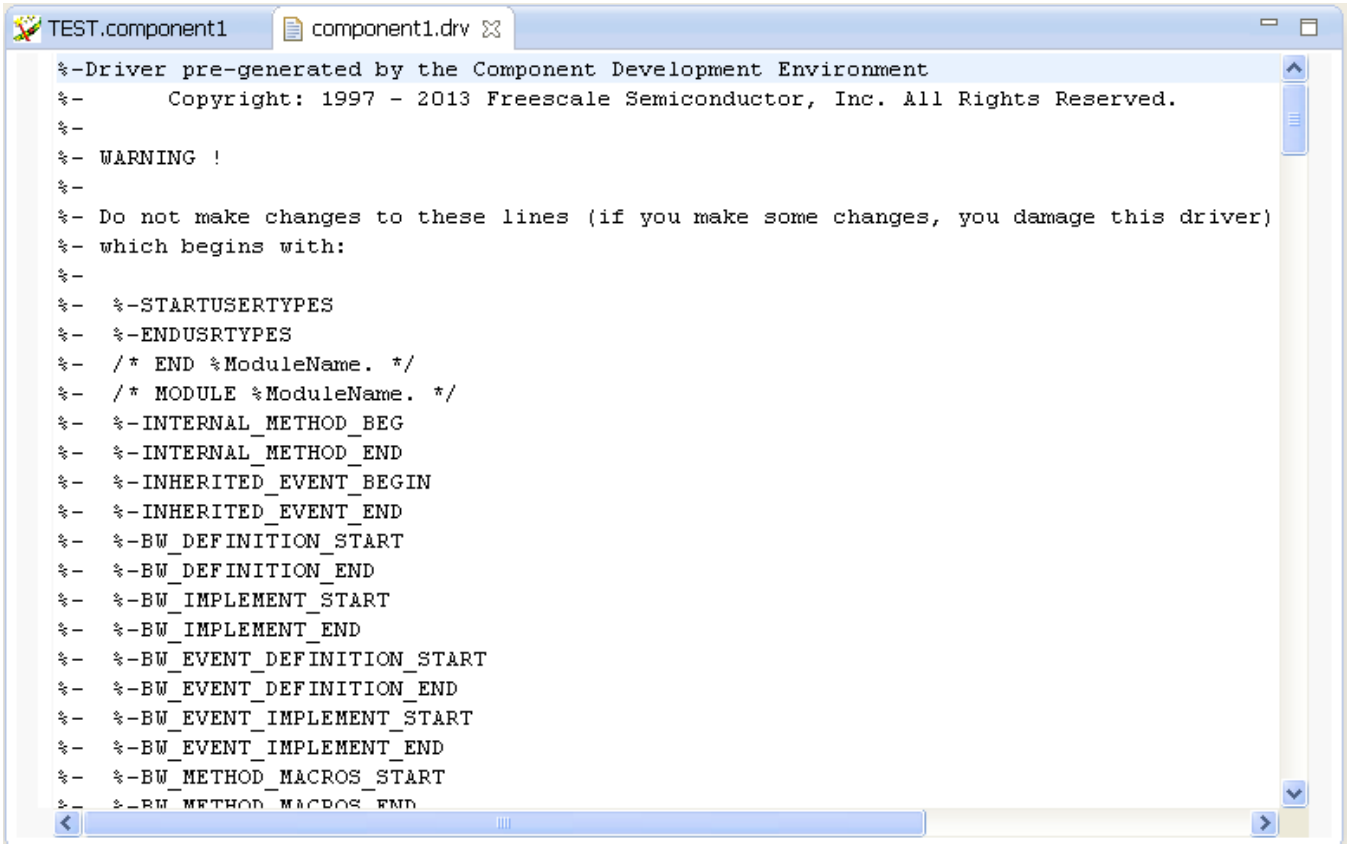

Figure 1-6. Driver source code

1.3.3.2 Implement component generation script

Now that you have seen both techniques, use the method editor to implement the driver.

1. In the **Explorer** view, double-click the **getParityBit** method name. Then click **Source Code** in the lower tab of the Method editor.
2. Type the code below into the method body.
3. Save your work.

   Note that this is C code with some %-delimited commands. Those commands are part of the Processor Expert scripting language. When Processor Expert generates code, it reads the state of the property and modifies the generated code based on the if/else condition. This minimizes the generated code.

```
int i, res = 0;

// Compute parity using shifting and XOR

for (i = 0; i < 8; i++) {
```

```
res = res ^ (Data >> i);

}

%if %EvenParity = 'yes'

return (byte)(res & 1); // mask result

%else

return (byte)(~res & 1); // mask result

%endif
```

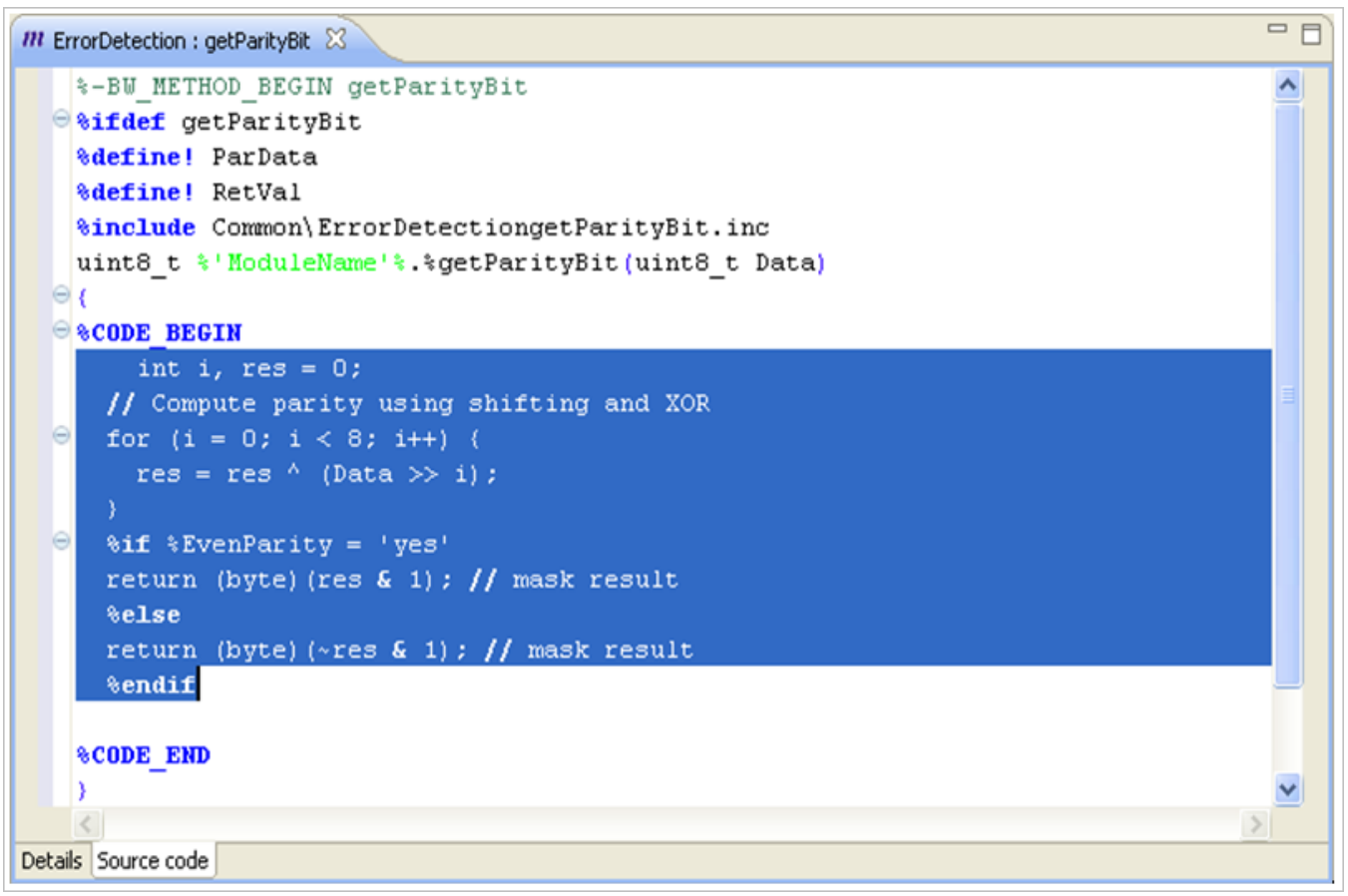

Figure 1-7. Entering code using Processor Expert scripting language

Chapter 2
Inherited components

Inheritance allows you to include (and redefine if you wish) methods and events from another component into your component. This simplifies and speeds up creating new embedded components. You can even inherit from the platform-independent components included in the Processor Expert.

The CDE provides two ways to inherit a component.

Select **File > New > Other** ... **New** wizard appears. Expand **Component Development Environment** and select **Inherited Component** option, then click **Next**.

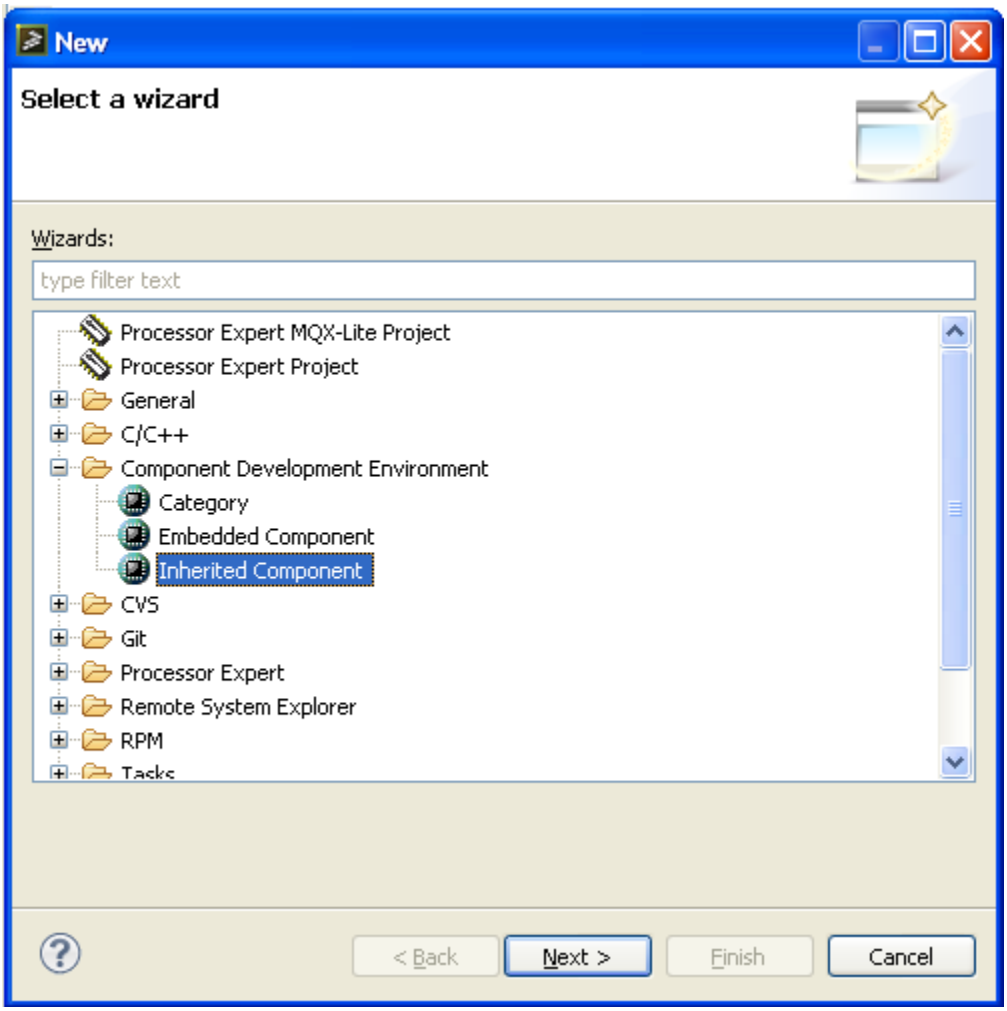

Figure 2-1. Inheritance wizard

Alternative method is, right-click on the component name or on the **Properties** in the Component editor and select **Inherit Component...**.

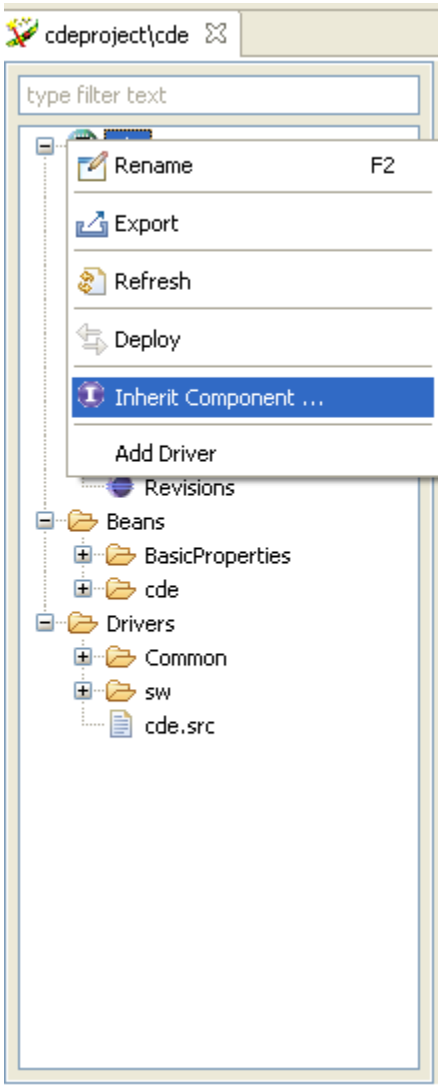

Figure 2-2. Component inheritance

2.1 Inherit component

1. Right-click on the component name or on the **Properties** in the Component Editor and select **Inherit Component...**. A message is shown indicating the current component must be closed. Click **OK**.

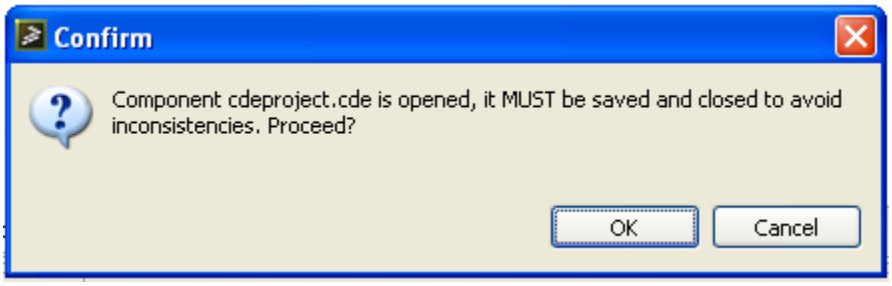

Figure 2-3. Continuing with the inheritance process

2. Choose the component to be inherited from. Select **Init_UART_VAR0**.

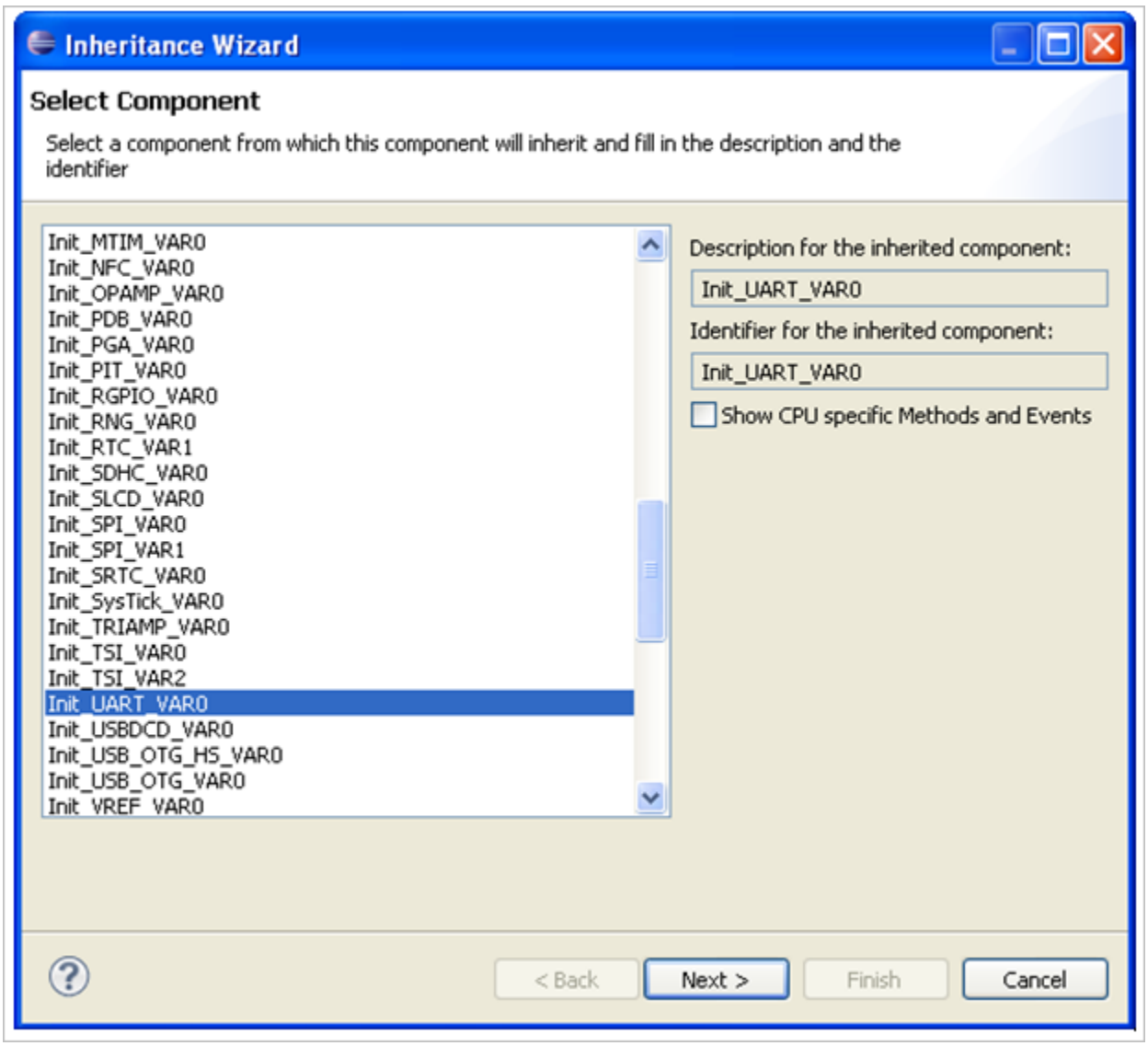

Figure 2-4. Select component page

3. Select **Exclusive usage of components methods and events**, and click **Next**.

   For more explanation about each type of inheritance, refer to the *CDE User Guide*.

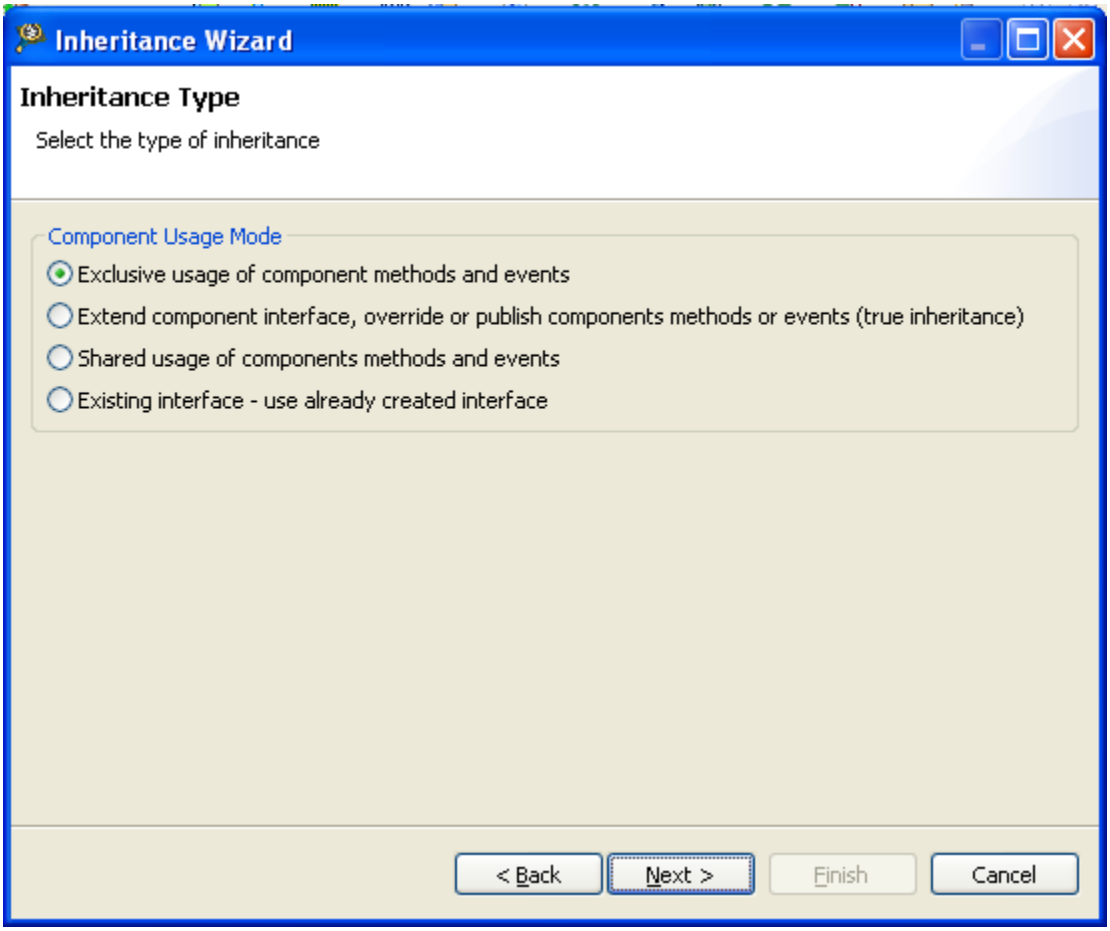

Figure 2-5. Inheritance type page

Select **Init** method to be inherited to the target component and set the **Inherit** settings to **Yes**, and then click **Next**.

For more information about each option, see *CDE User Guide*.

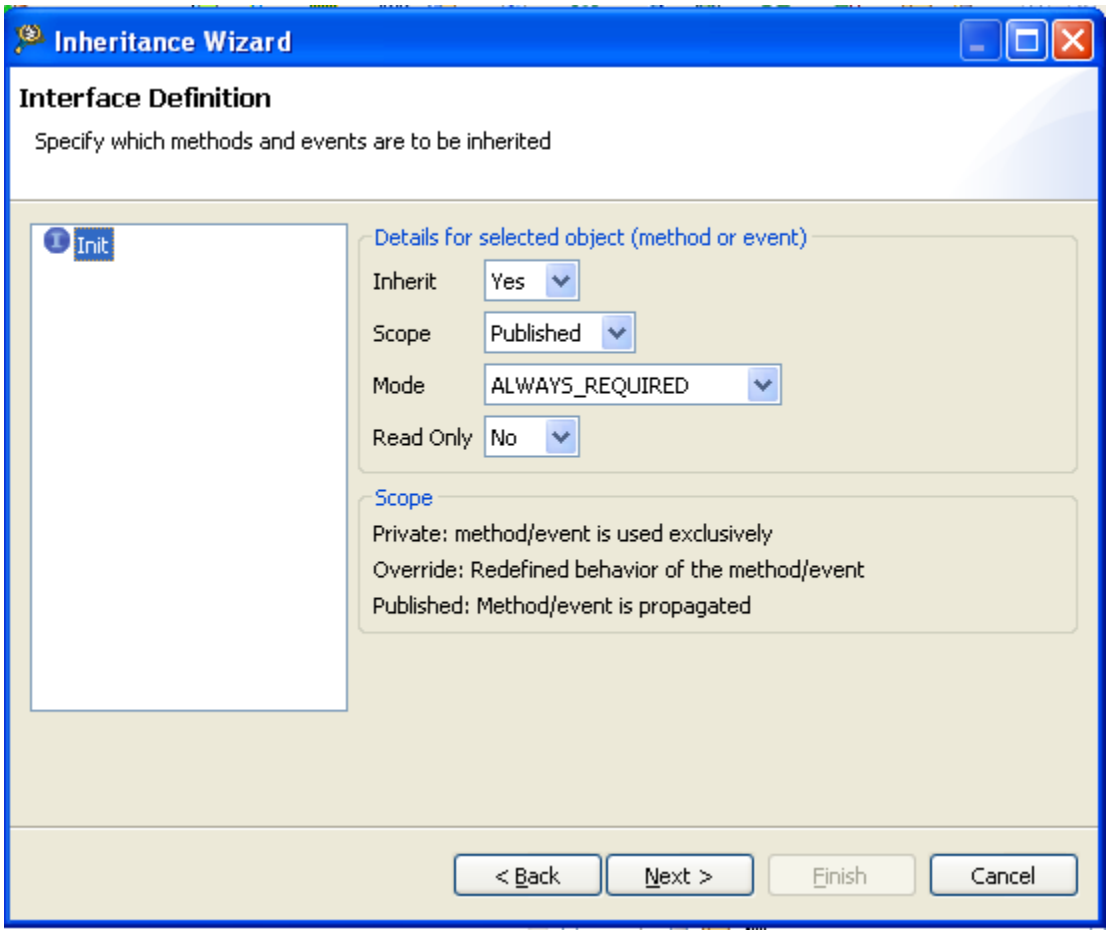

Figure 2-6. Interface definition page

4. Select **Continue without inheriting another component**, and click **Next**.

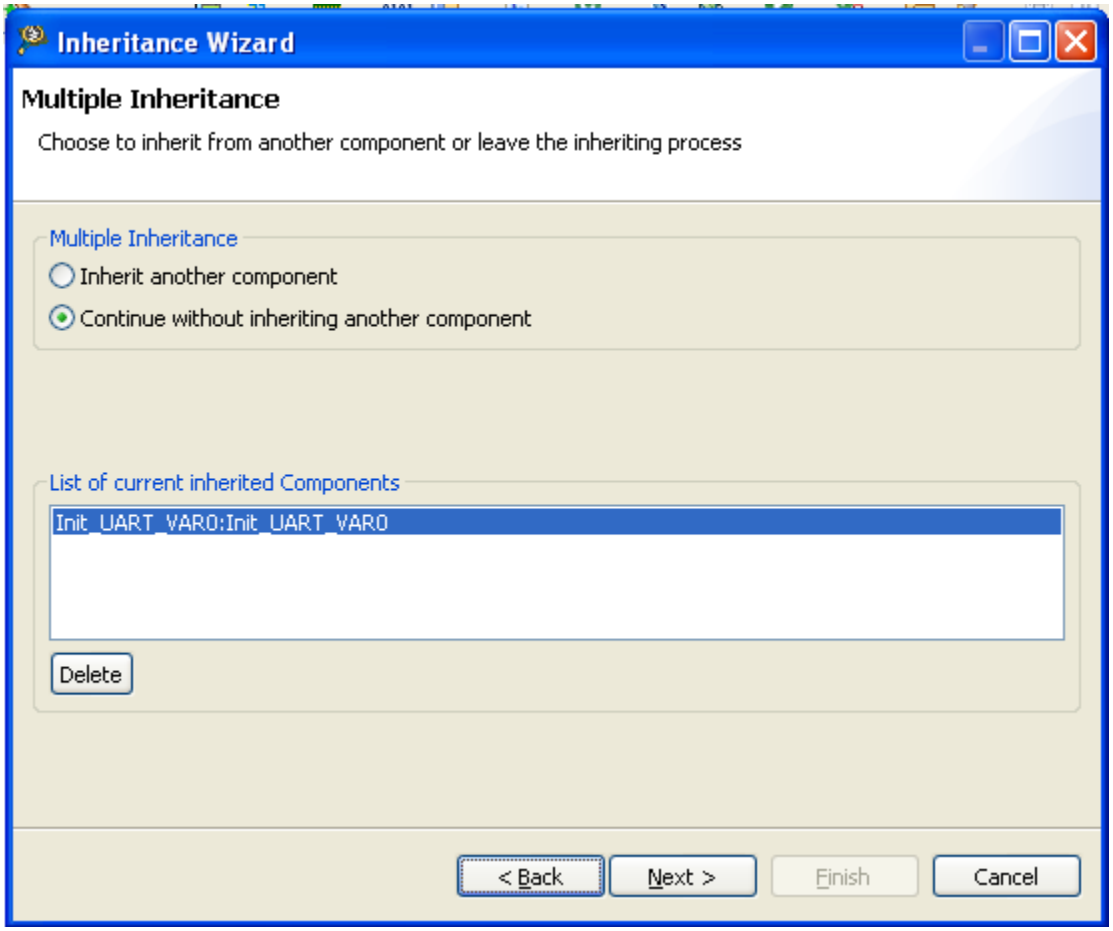

Figure 2-7. Multiple inheritance page

5. The default programming language **ANSI C** is selected; click **Finish**.

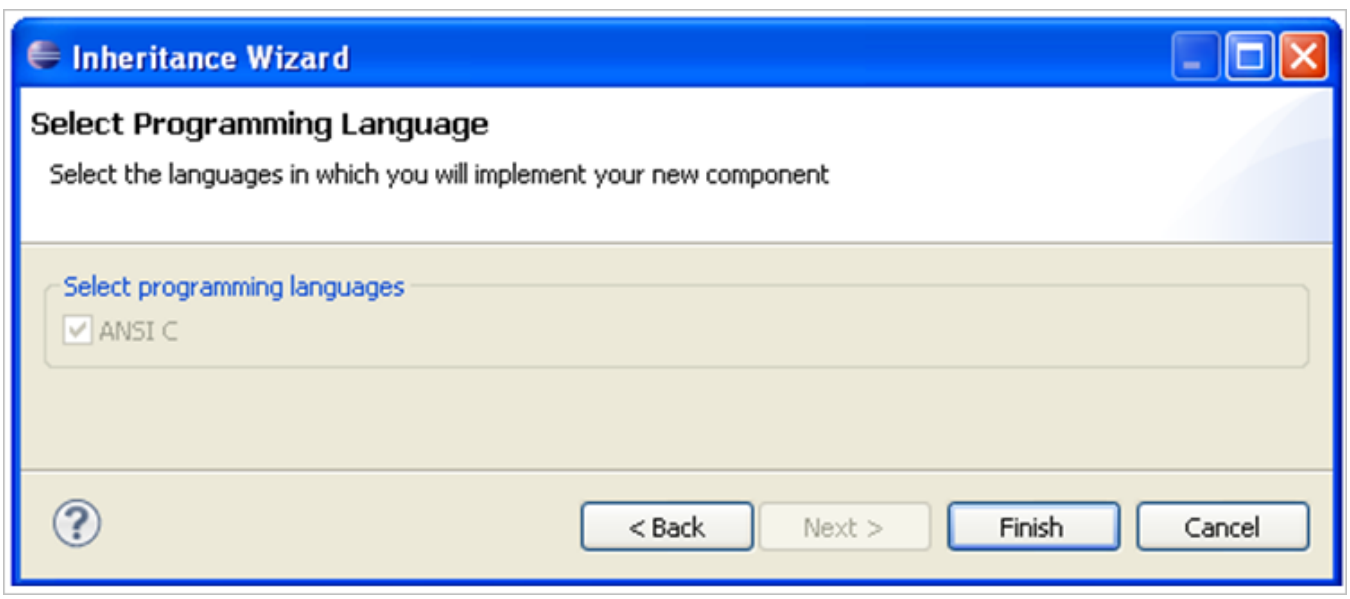

Figure 2-8. Select programming language page

Figure below shows the **ErrDet1** component with the method **Init** inherited in the target component.

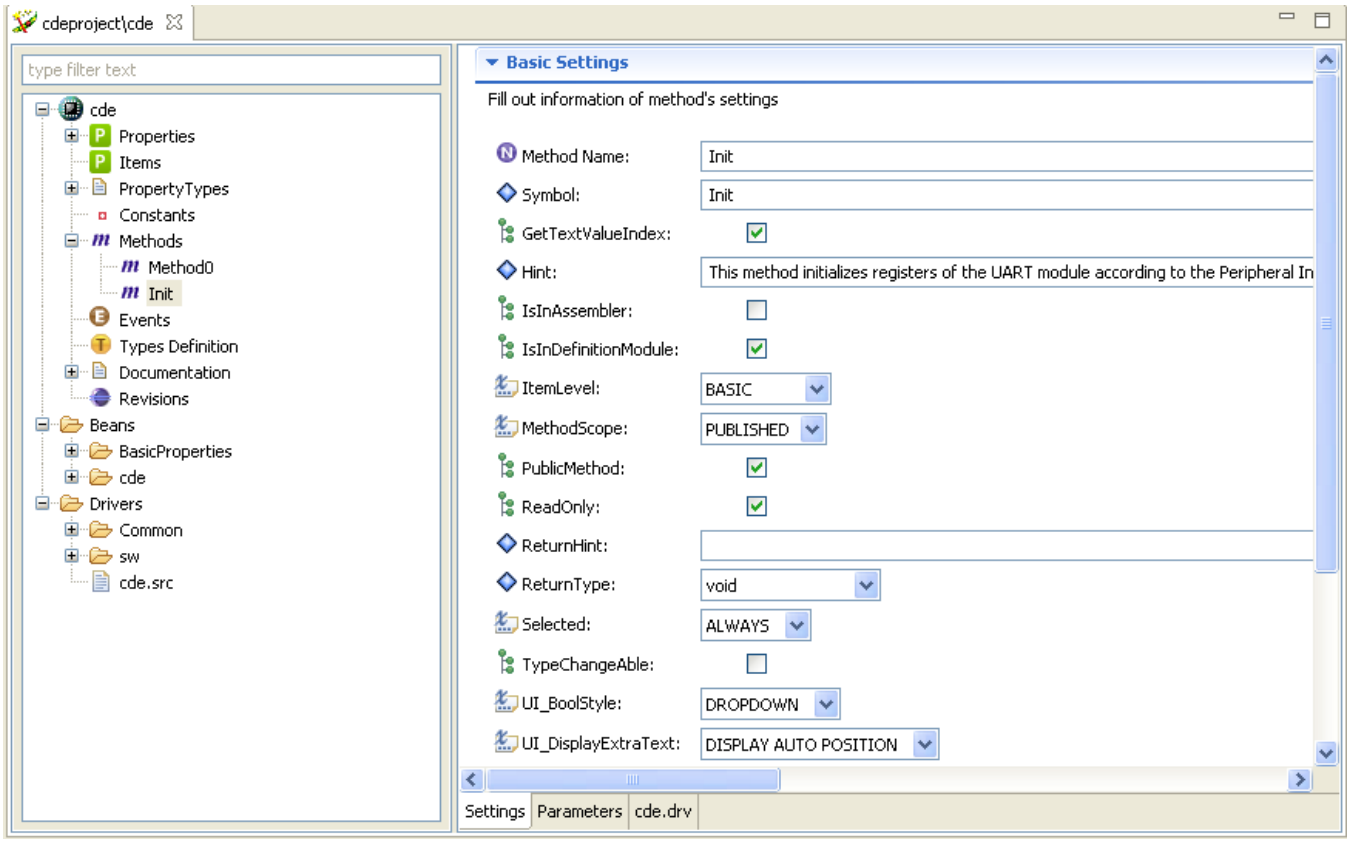

Figure 2-9. Inherited components

2.2 Deploying components

The Deploy operation copies the related component files to the Processor Expert user path allowing a Processor Expert instance to use the newly created component. It automatically appears in the Component Library for the selected Processor Expert instance to which the CDE is connected.

Right-click on the component name and select **Deploy** option.

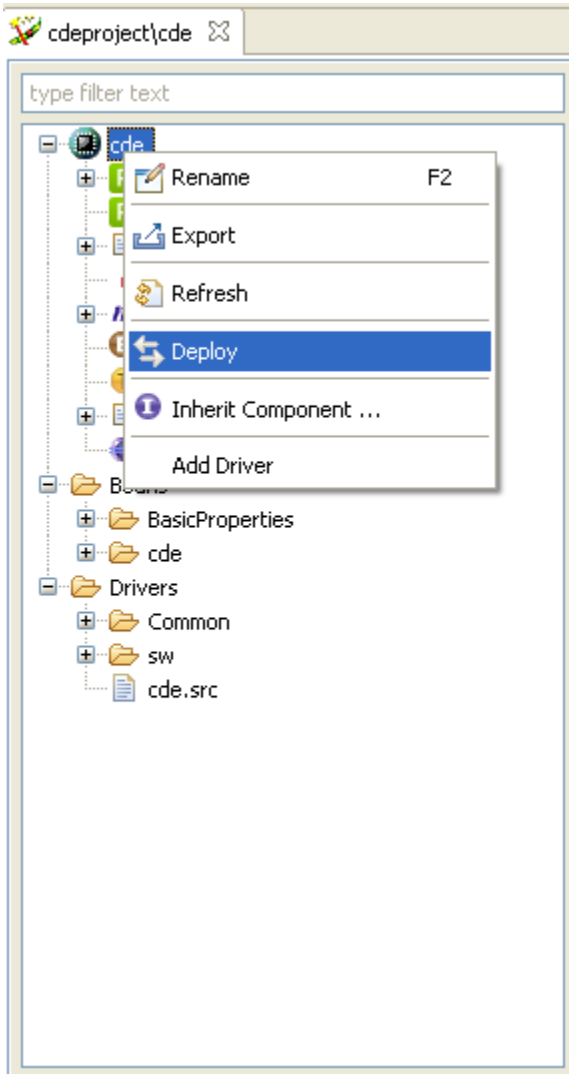

Figure 2-10. Deploy component screen

Verify the path is correct and click **Finish**.

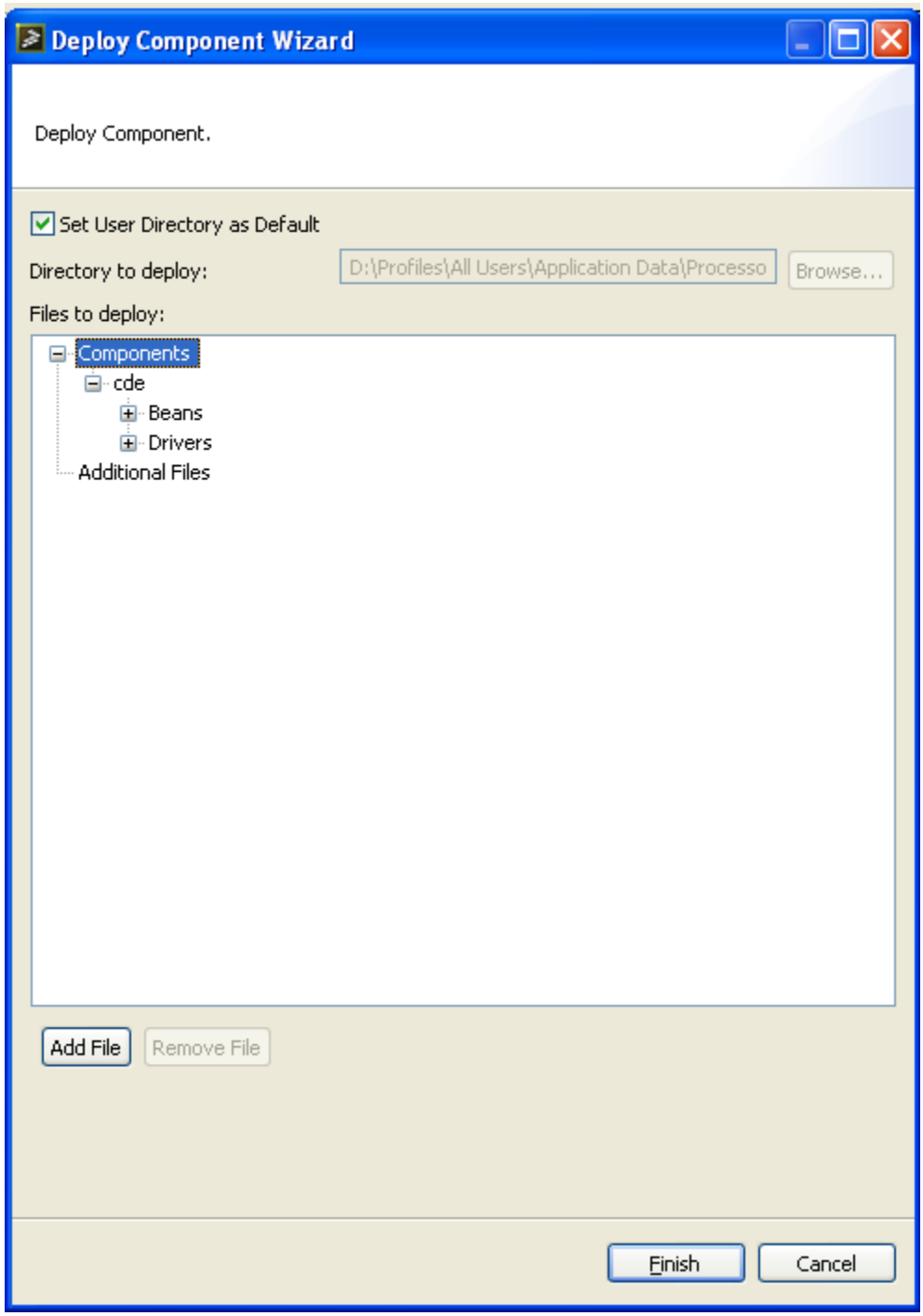

Figure 2-11. Deploy component wizard

NOTE

Under **Windows > Preferences > Processor Expert > CDE Preferences**, select the checkbox labeled **Auto Deploy Component when saved**. When selected, this option auto deploys the component every time it is saved.

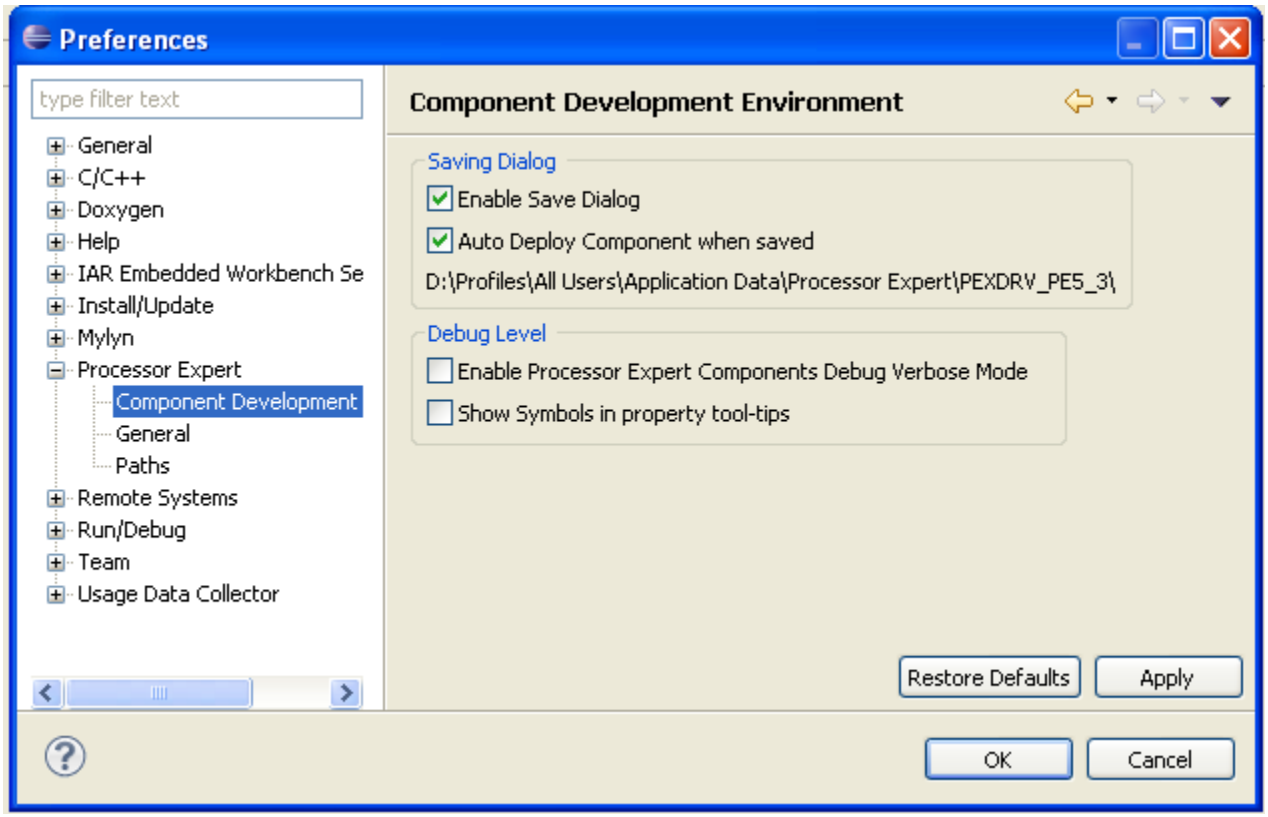

Figure 2-12. Auto deploy preferences dialog

Chapter 3
Exporting and importing components

CDE allows you to import or export component or interfaces of the selected components. This section explains:

- Exporting components
- Importing components
- HTML Help

3.1 Exporting components

The Export operation generates an installable package for distribution. This package can be imported into other Processor Expert instances, for example, on a separate computer.

1. Right-click on the component name and select **Export** option.

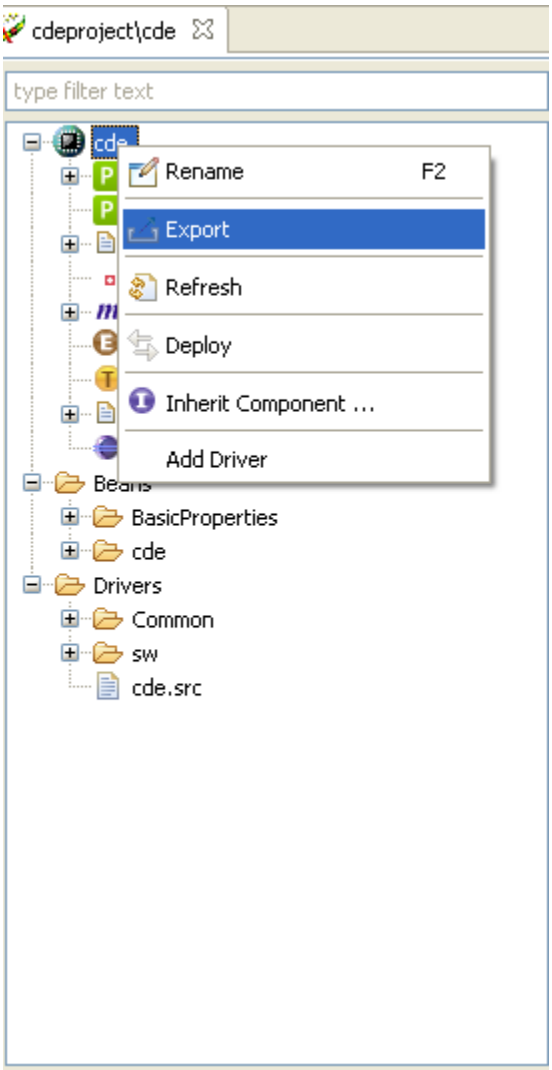

Figure 3-1. Export component screen

2. Click **Browse** ... and select a directory destination.

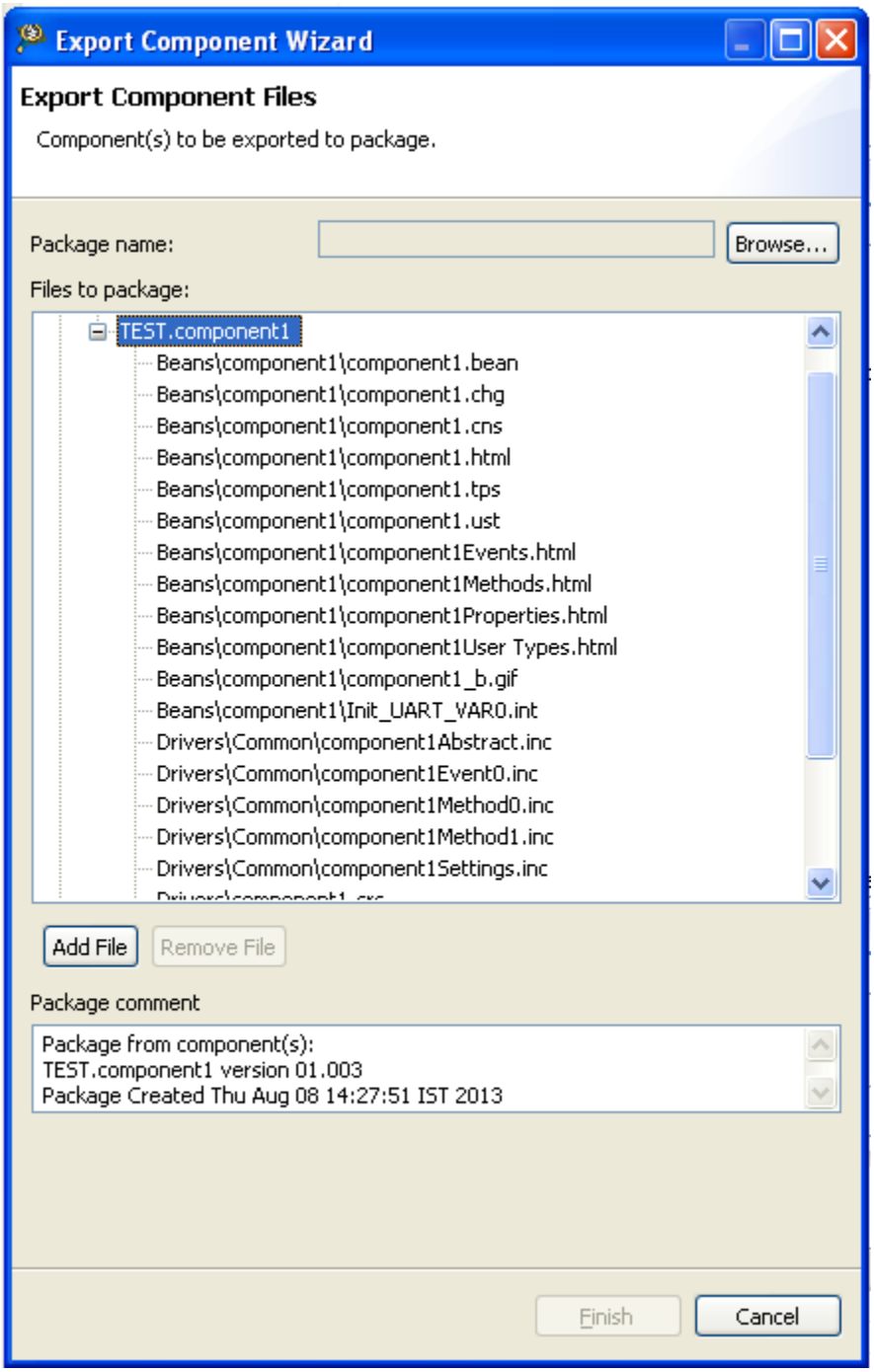

Figure 3-2. Export component files screen

3. Name the package and click **Finish**.

   You can also import the .upd file the component will appear in the **Component Library** view.

3.2 Importing components

You can import the component or interface into your current project and also can import the component from package. This package can be imported into other Processor Expert instances, for example, on a separate computer.

This topic explains:

- Import component or interface
- Import from package

3.2.1 Import component or interface

You can import the component or interface into your current project.

To import component or interface:

1. Select the project in which you want to import the component or interface. Right-click on the project name and select **Import** option. The **Import** screen appears. Select **Component Development Environment** - > **Import Component/Interface** option.

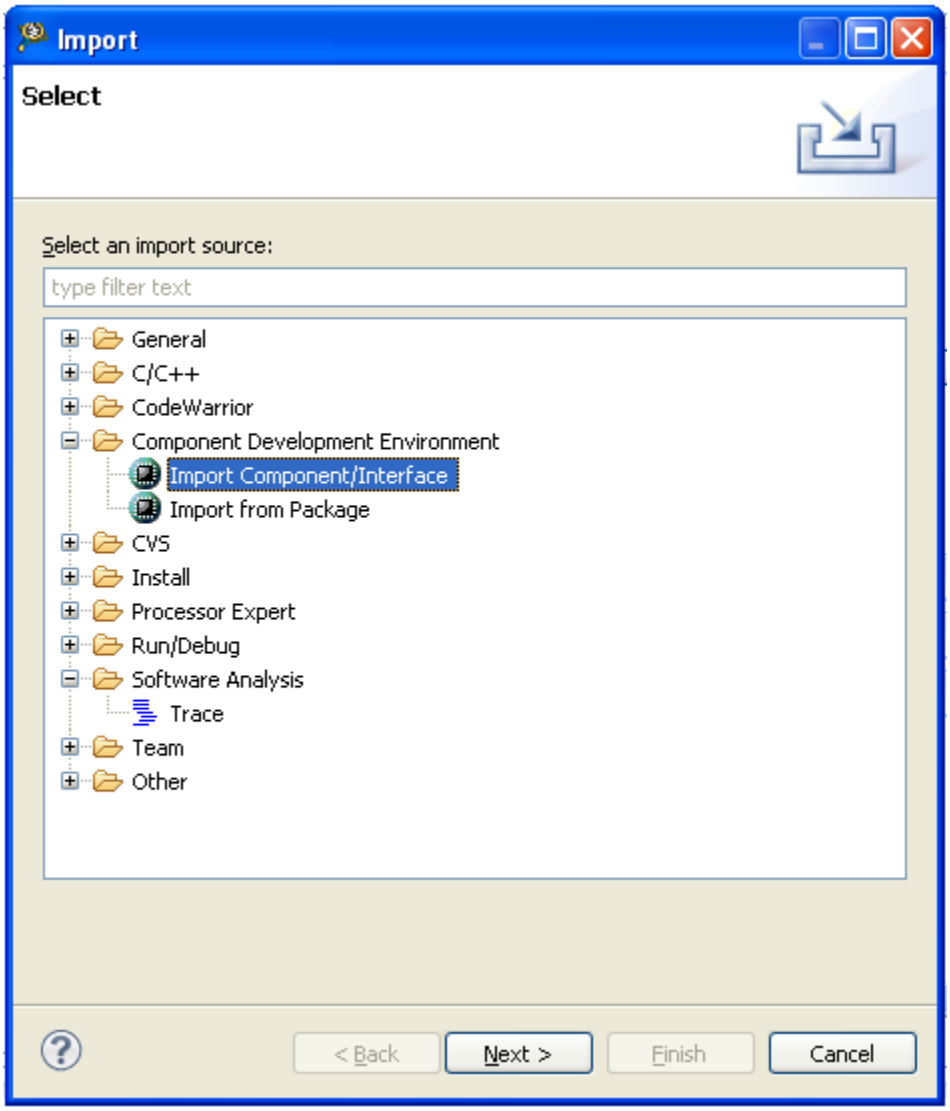

Figure 3-3. Import screen

2. Select **Next**. The **Target project** screen appears. Select the project where you want to import the component.

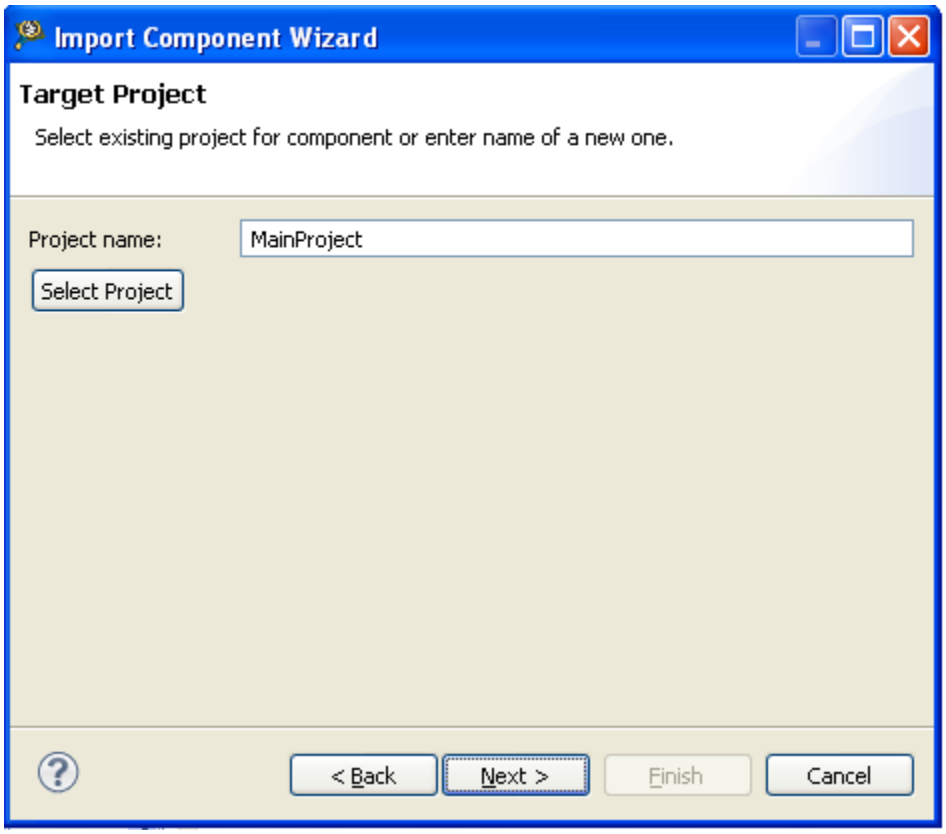

Figure 3-4. Target project screen

3. Click **Next**. The **Select component for import** screen appears.

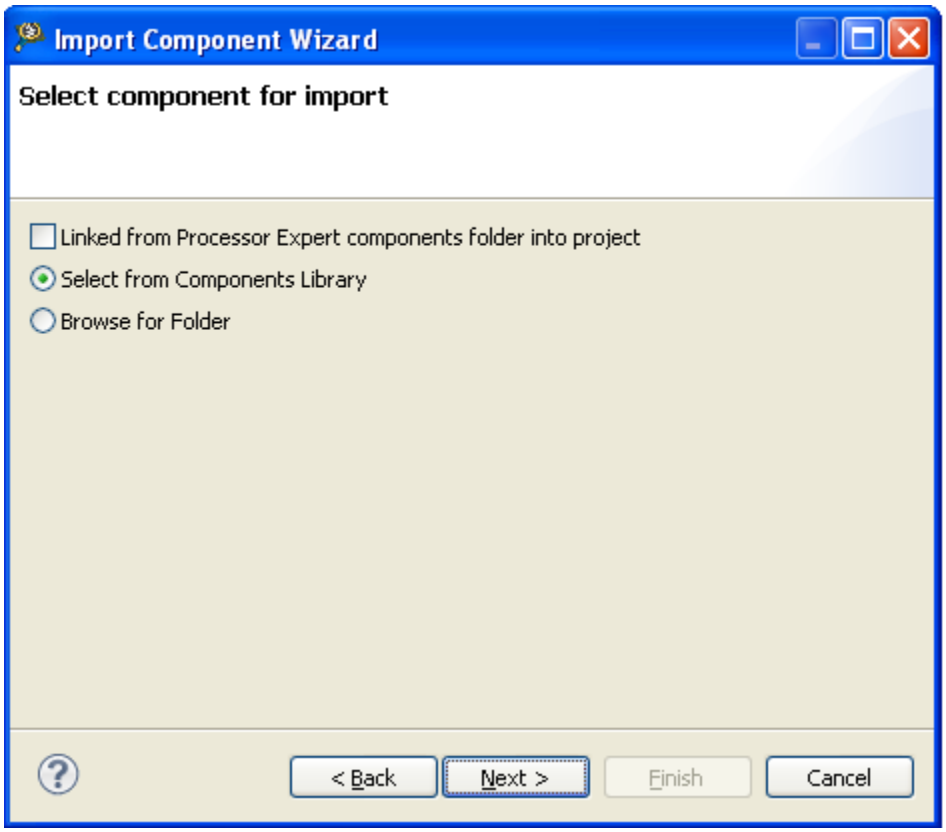

Figure 3-5. Select component for import screen

Select the required option.

- **Linked from Processor Expert components folder into project** - if this checkbox is selected, the imported component or interface in the project will be linked from PEx directory
- **Select from Components Library** - if you want to select the component or interface from the Component Library
- **Browse for Folder** - if you want to select the component or interface from your folder

4. Click **Next**. The **Select component/interface from Component Library** screen appears.

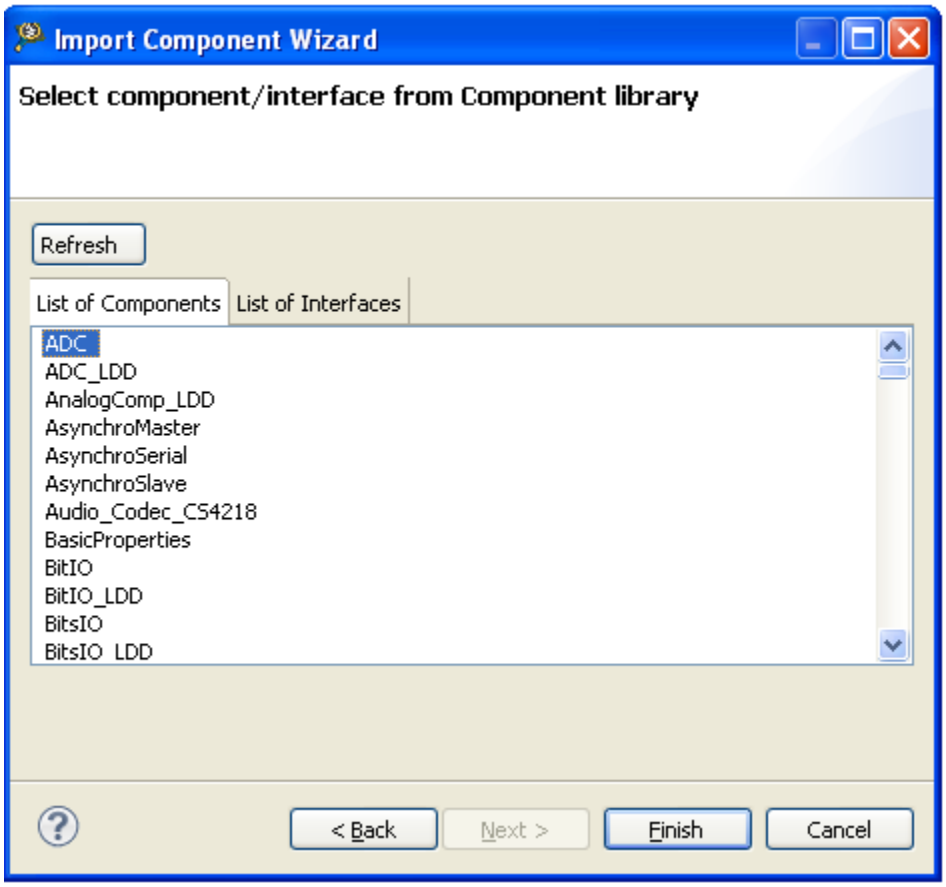

Figure 3-6. Select component/interface from Component Library screen

5. Select the required component or interface from the **List of Components** or **List of Interfaces**. Click **Finish**. You will see that the selected component is imported into the current project.

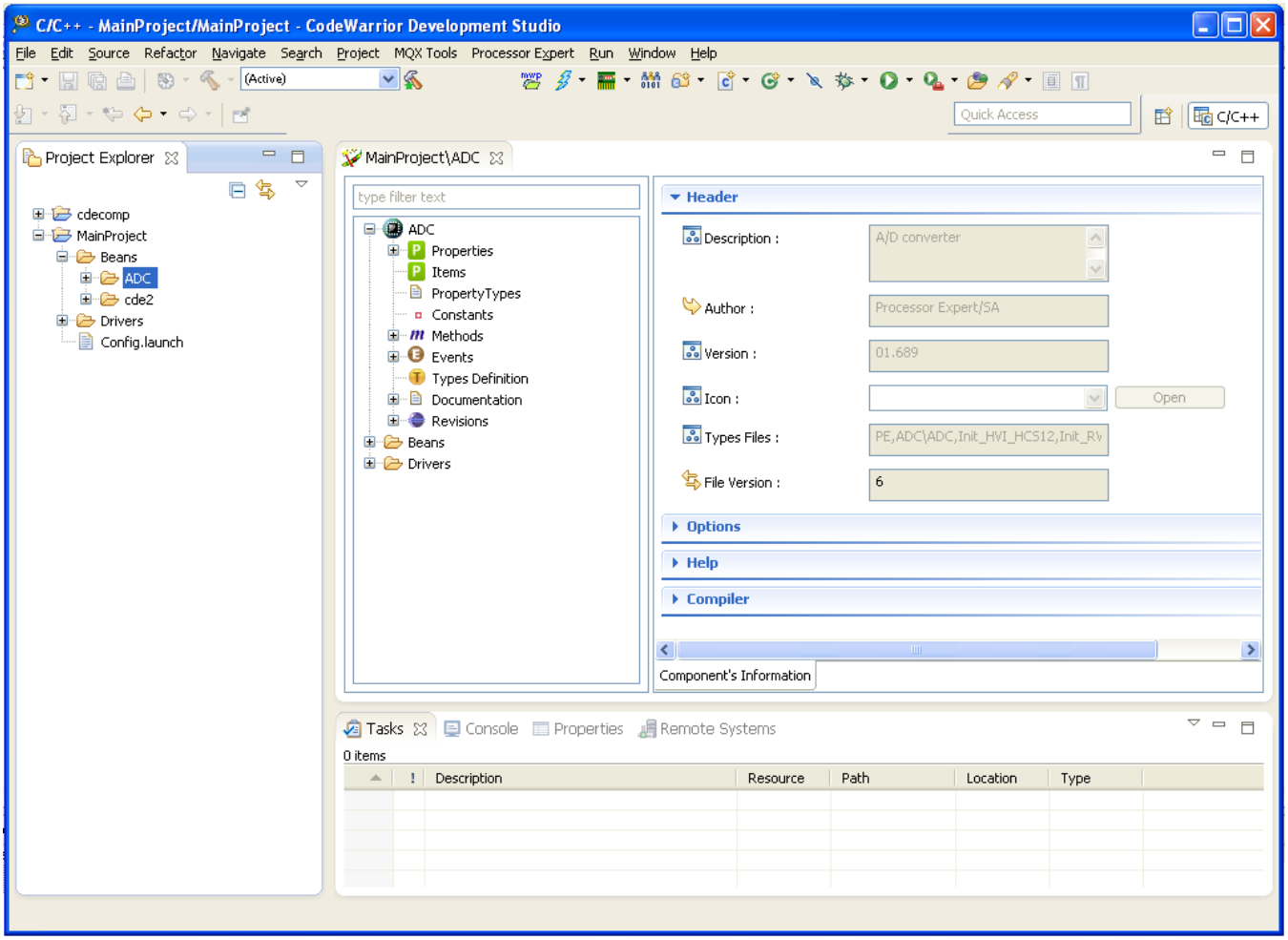

Figure 3-7. Component imported into the project screen

3.2.2 Import from package

You can import the component from package. This package can be imported into other Processor Expert instances, for example, on a separate computer.

To import component from package:

1. Select the project in which you want to import the component or interface. Right-click on the project name and select **Import** option. The **Import** screen appears. Select **Component Development Environment** -> **Import from Package** option.

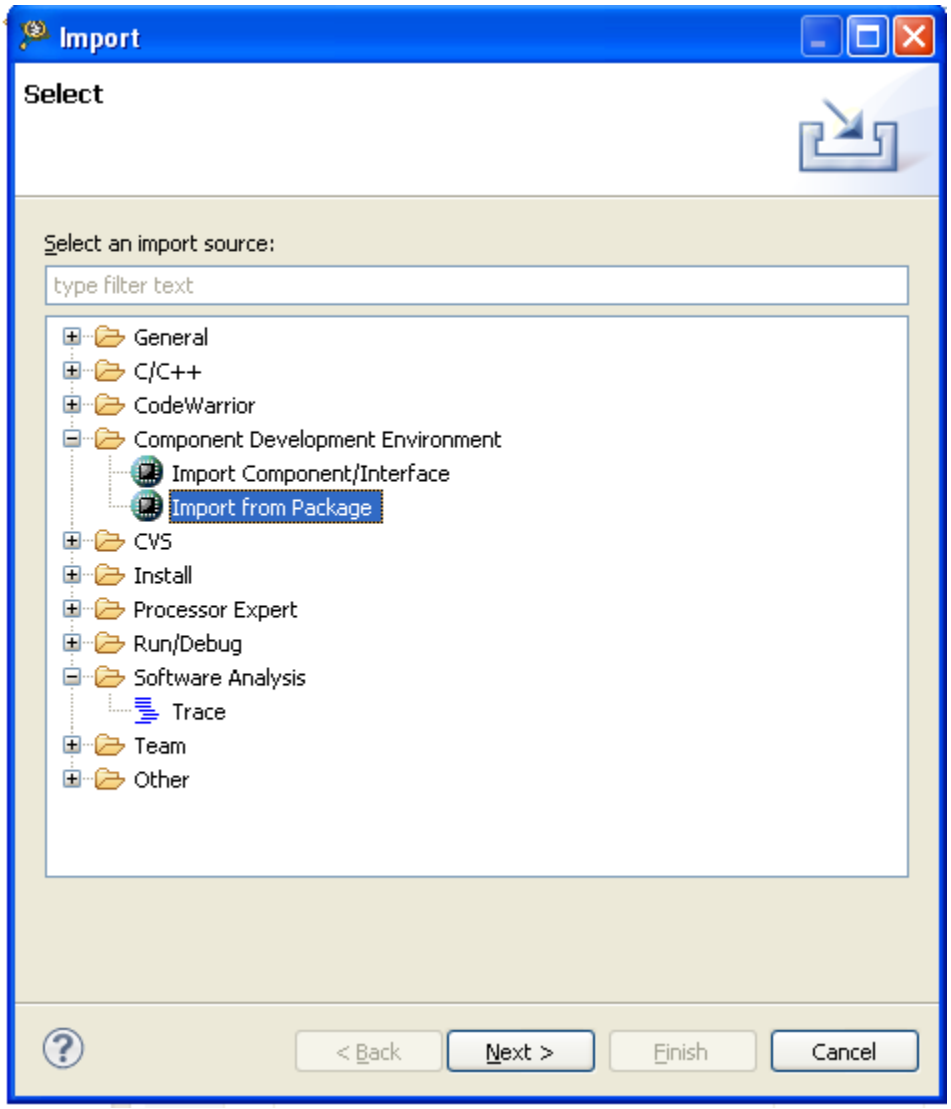

Figure 3-8. Import from package screen

2. Select **Next**. The **Import component** screen appears.

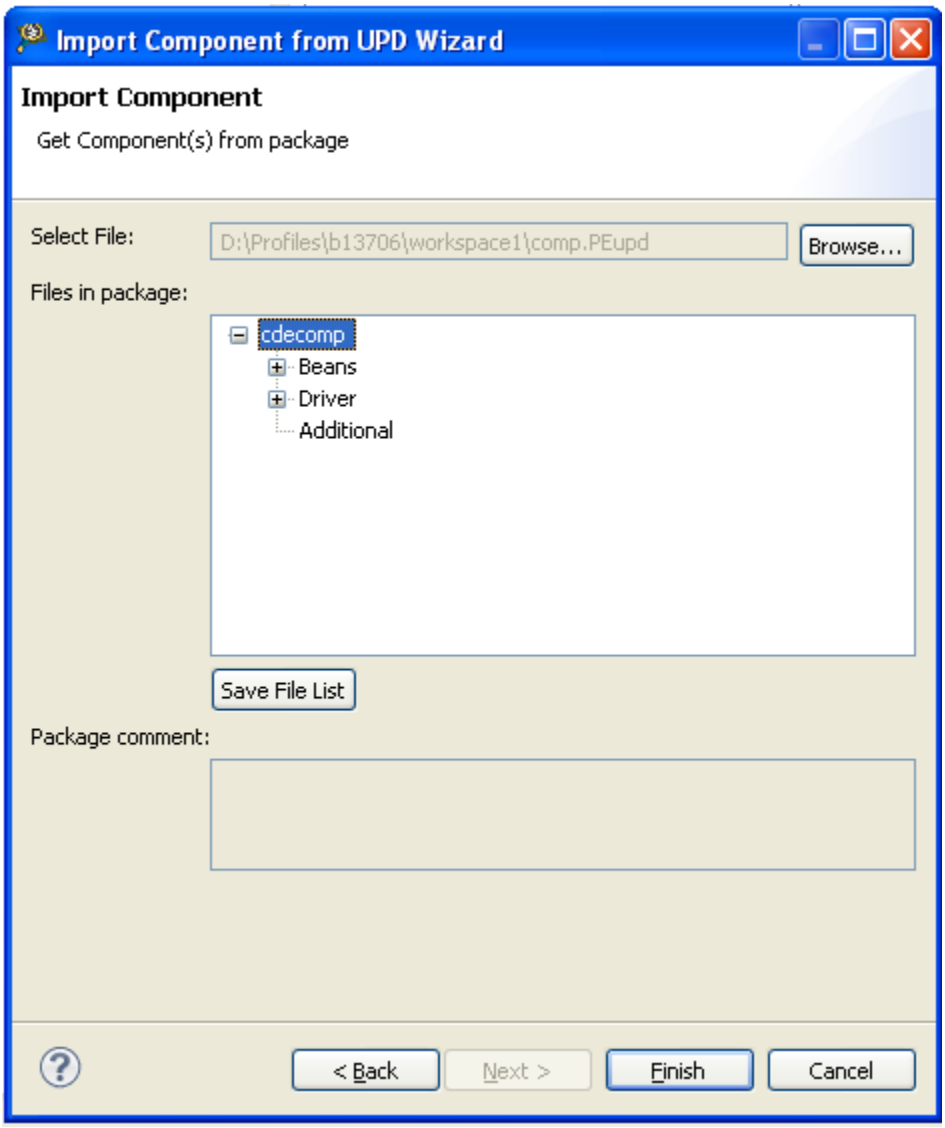

Figure 3-9. Import component screen

3. Click **Browse** to search for the package from which component can be imported.
4. Click **Finish**. You can see the **Components** view and **Project Explorer** that the required component is added.

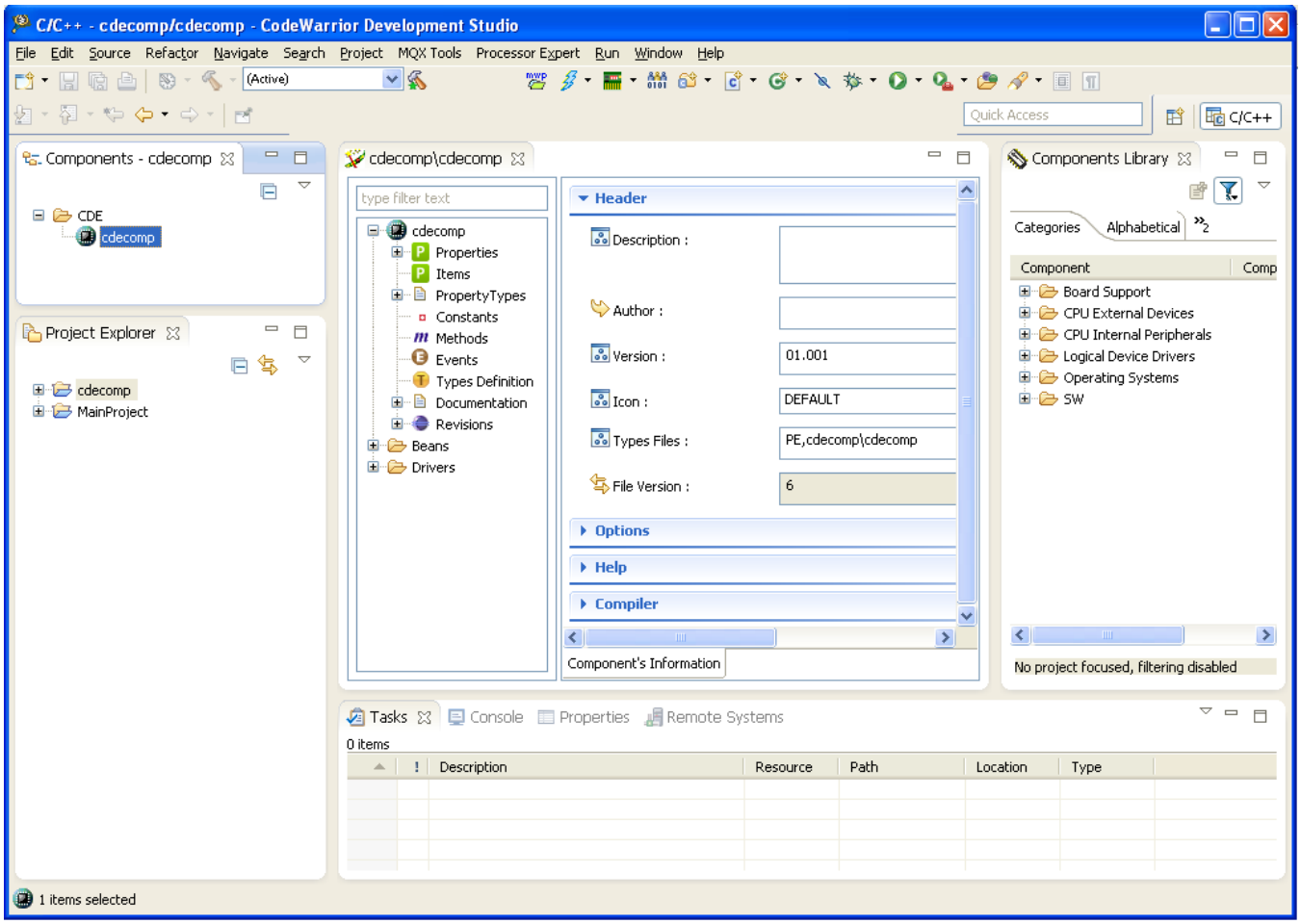

Figure 3-10. Component is added from package screen

3.3 HTML help

The CDE generates several HTML help files based on the methods and properties hints entered during design time. You will find these files in the **Beans** folder of the **Component** editor.

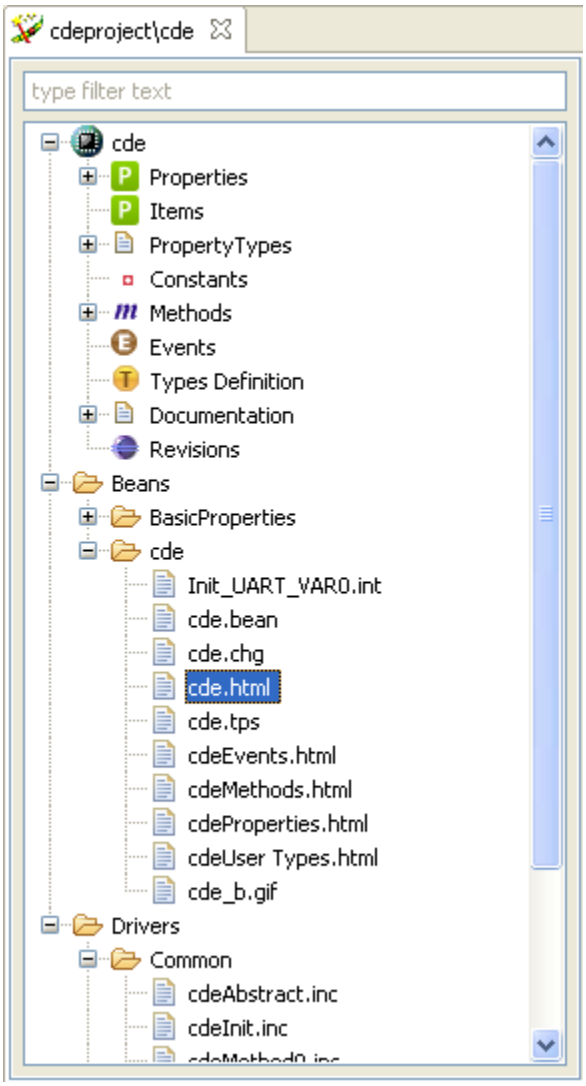

Figure 3-11. HTML help files screen

To view HTML editor, right-click on the component HTML file and select **Open With -> Default Editor** option from the context menu.

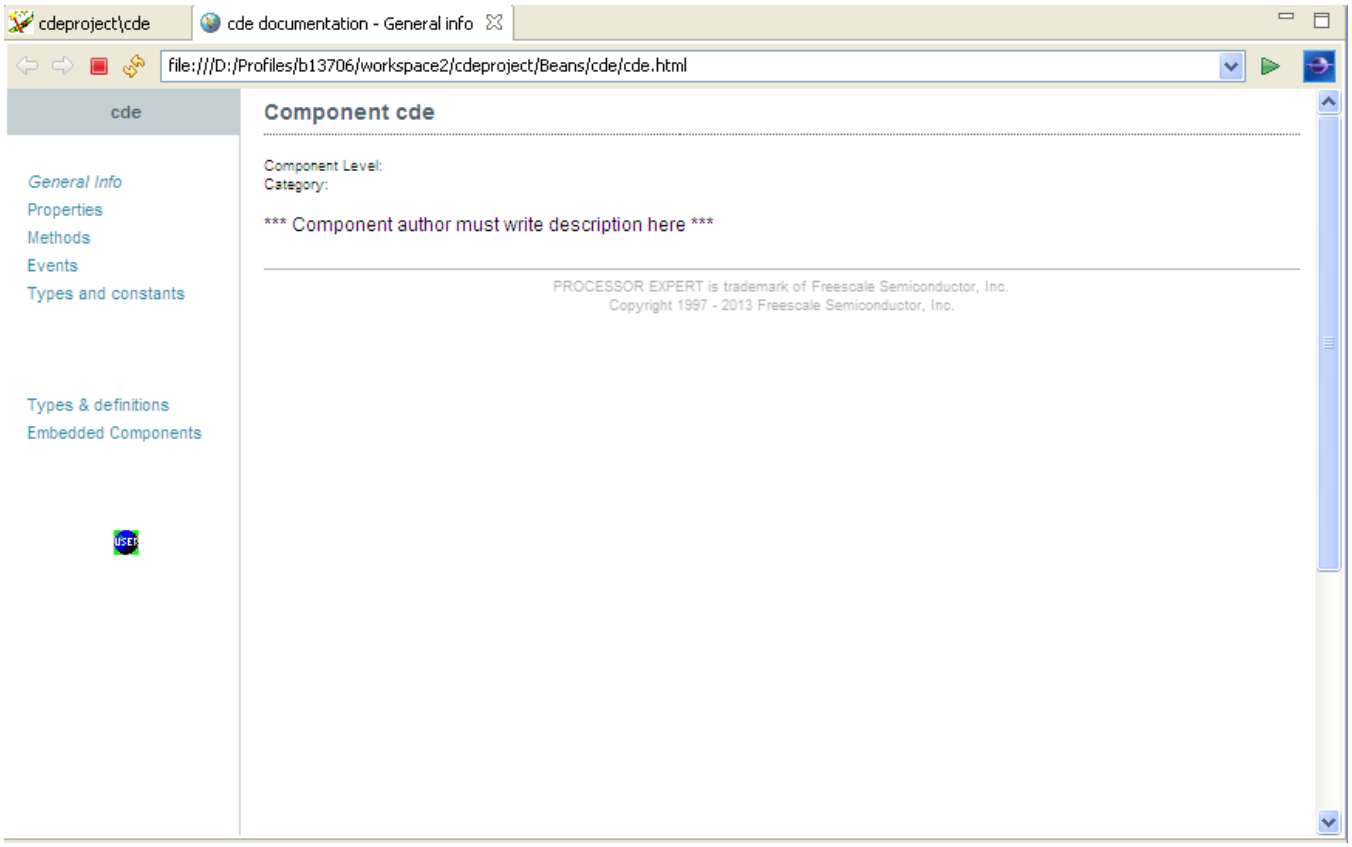

Figure 3-12. HTML editor screen

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number CDEGS
Revision 02/2014