Series 60 AppWizard

Installation & User's Guide

1. Introduction

This document is an installation & user's guide for Series 60 AppWizard.

2. AppWizard

AppWizard is a plug-in module for Microsoft Visual Studio 6.0. The wizard is a helper tool that can be used for rapid Series 60 UI application generation. It creates a minimal application that is based on the user's selection of features. Generated code can be compiled and run on the emulator and target device.

The wizard creates all the necessary project files (MMP files, resource files, and so on) that are needed for a full application.

2.1 Supported platforms

AppWizard has been tested on Windows NT 4. It should also work on Windows 2000. In Windows 98, opening the workspace to Developer Studio does not work.

3. Installation

The wizard consists of two files: AvkonAppwiz.awx and AvkonAppWiz.hlp. Therefore, installation is a straightforward process:

1. Locate the Developer Studio Template directory. If you have installed Microsoft Visual Studio to the default directory, the Template directory path is: "c:\Program Files\Microsoft Visual Studio\Common\MsDev98\Template".

2. Copy AvkonAppWiz.awx and AvkonAppWiz.hlp to the template directory.

3. Start Visual Studio and select File --> New. In this dialog, select the Projects tab. The Series 60 AppWizard item should be in the list:
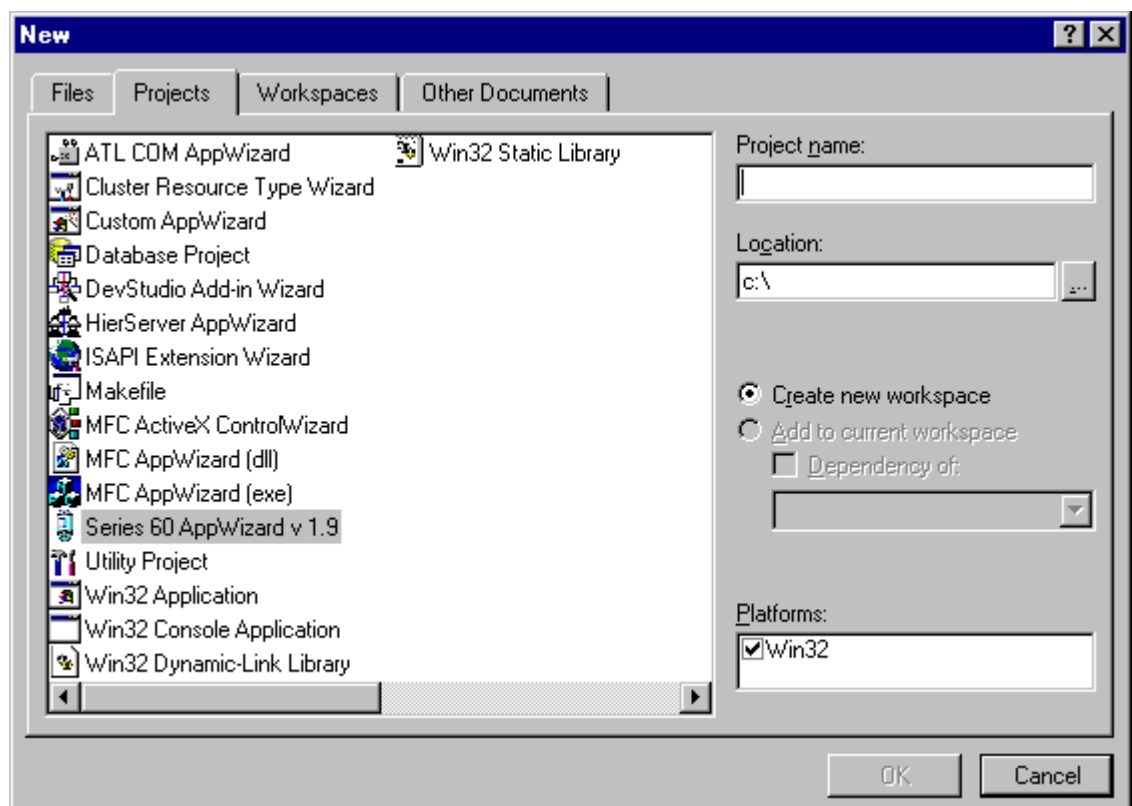

Figure 1: *New project dialog*

If you cannot find AppWizard in the list, check that you copied AvkonAppWiz.awx to the correct directory. (There should be other *.awx files already in the directory.)

Before proceeding, ensure that the Series 60 SDK and tools (for example, Perl) are installed. The wizard relies on the SDK and its scripts in order to work properly.

4. Creating an application

4.1 Starting

You can create applications with the help of a wizard, a series of dialogs where you can decide which features you want to be included.

You first need to create a new project (and a new workspace). You can do this in the dialog in Figure 1.

Specify the project name in the **Project name** field. *Note: The maximum length for the project name is **16 characters**. If this length is exceeded, the application is not recognised by Symbian OS when trying to launch it.*

In the **Location** field, specify the path where you want the new application to be created. The path can be freely chosen, but make sure that the drive is the same as that where the SDK is installed.

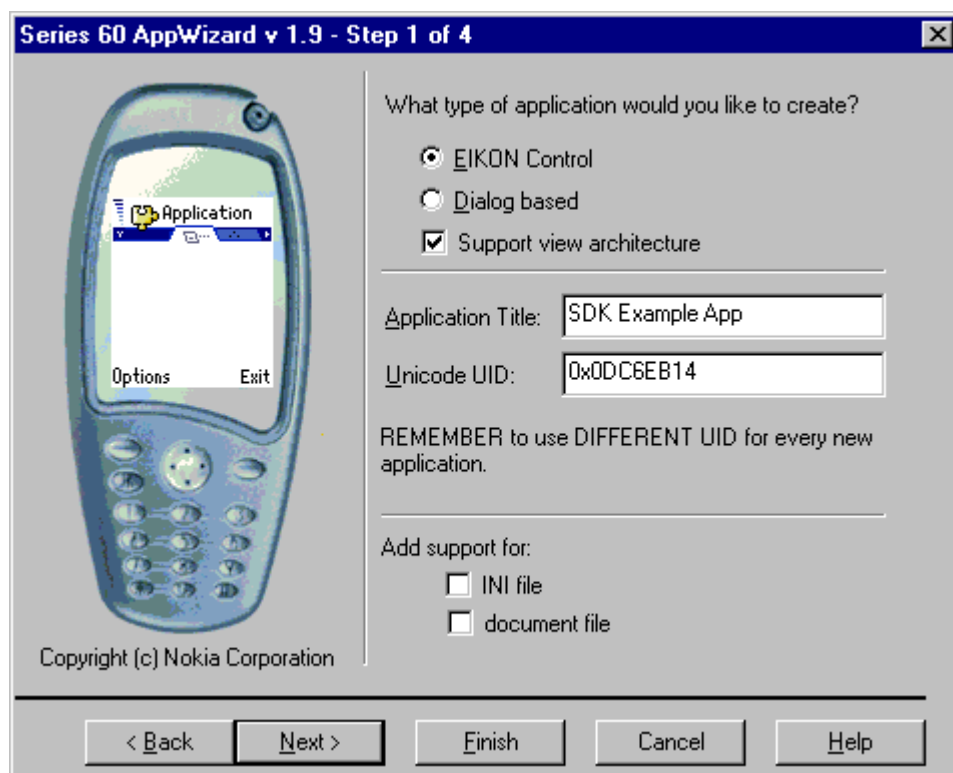Click **OK** when you are finished with this dialog.

4.2 Step 1

Figure 2: *First screen of AppWizard*

On this screen, you need to select a few basic options for the new application.

For application architecture there are the following options:

- **EIKON Control**
- **Dialog based**

If you select **EIKON Control**, the application main view is derived from the CCoeControl class. This is the most common way to create the main view (available since ER5).

If you select the **Dialog based** option, the main view is based on the CEikDialog.

View architecture is a new application architecture type. In view architecture the application is divided into views and each view takes on many of the capabilities of a traditional application UI, for example, each view has its own control stack.

If you select the **Support view architecture** option, AppWizard will create two views for the application. Each view contains either a CCoeControl or CEikDialog based container window.

In the **Application Title** field, type the title you want to give the application. The title will appear in the application shell's application list and in the application context pane. *Note: There are two forms of application title: long and short. AppWizard will use the title string defined in this step for both, but you can change the strings later (defined in the application's LOC file).*

Each application must have a unique identification ID (UID). AppWizard fills the Unicode UID: field with a random experimental UID. Before shipping the product, you should request a proper UID from Symbian Ltd.

The followingt two options are related to file system usage:

- **INI file** (creates an INI file for application settings)

- **Document file** (creates a document file for application data)

If you do not need support for either an INI or document file, leave these options unselected. This will reduce unnecessary file system access.

After selecting all necessary options, click **Next** to go to the next screen where more options can be chosen. You can also click **Finish** to jump to the last step of application creation.

4.3　　Step 2

Figure 3: *Second screen of AppWizard*

On this screen, you can enter copyright and author information. This information will appear in the beginning of generated source files.
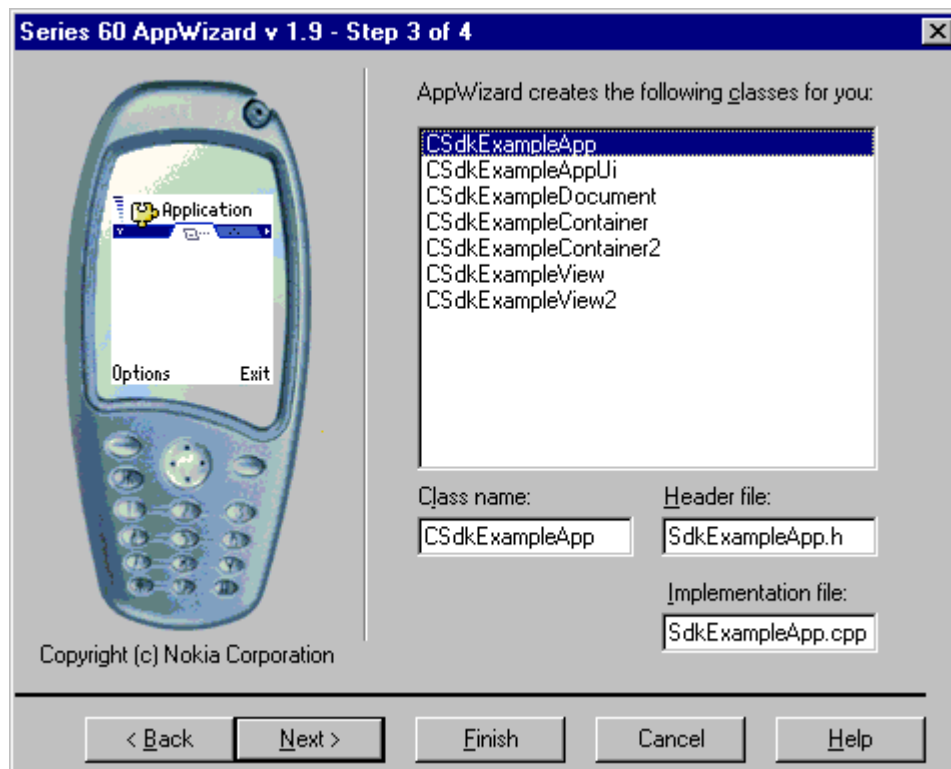
4.4   Step 3

Figure 4: *Third screen of AppWizard*

On this screen, you can change default names for classes and header and source files. Select the class name from the upper list. You can then modify the names in the fields below.

4.5   Step 4

Figure 5: *Fourth screen of AppWizard*

On this screen, you can change generated file locations. The wizard will create separate directories for include, source, group, data, and test sources. The settings are stored to the registry and are effective the next time you use the wizard. You can restore default settings by clicking the **Restore default values** button.
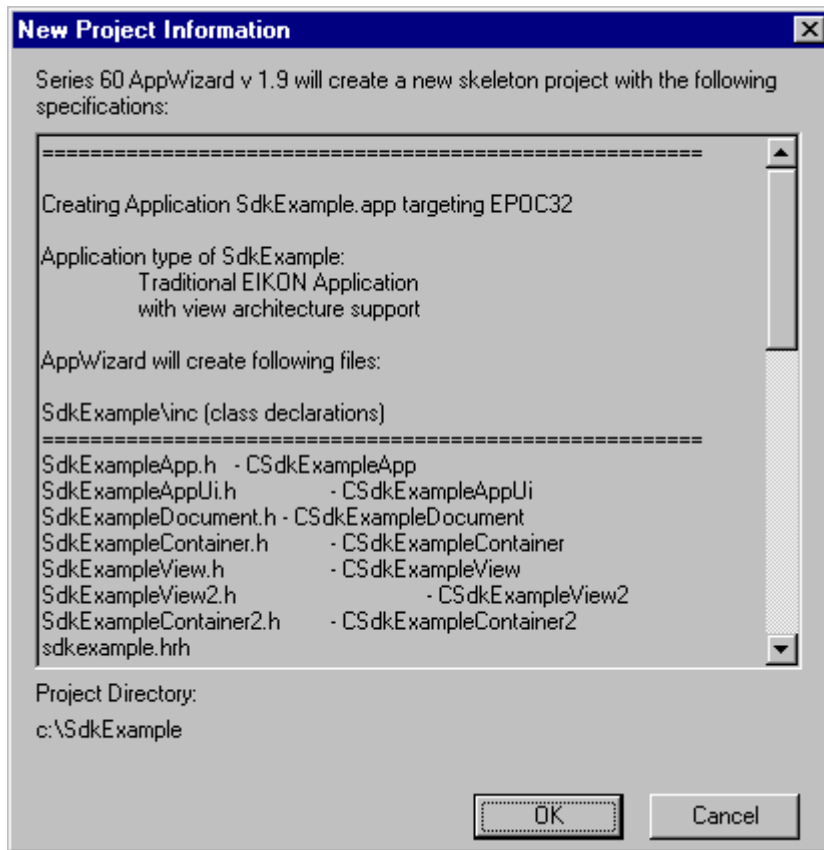
4.6   New Project Information

Figure 6: *New Project Information screen*

This is the final confirmation screen for your application. All the options selected in previous steps are listed here. Click **OK** if you want to create a new project with the listed options.

When you click **OK**, AppWizard creates a new project and opens it in Visual Studio. *Note that this will take some time!*

4.7   Building

A newly created project can be compiled directly in Developer Studio. To test with the emulator, select the Execute command from the menu. For the executable field, give the path to EPOC.EXE (for example, "c:\symbian\6.1\series60\epoc32\release\wins\udeb\epoc.exe").

Your application is located by default in the Other folder in the application shell.