View Vs Container

This article explains the conceptual difference between S60 views and containers.

The basic reasons for confusion

SDK examples

Most of the example applications name the `CCoeControl`-derived class **View** (maybe because they mean the generic term : View means representation of data, View may be derived from `CAknView` or from `CCoeControl` and its a generic term). To avoid confusion it should in fact be named **Container** because the view class that derives directly from `CCoeControl` should be called **Container**.

Lack of proper documentation

There is no straightforward documentation which explains what view and container are, what the basic difference between them is, and when each of them should be used.

What exactly is a view

- In abstract terms, a view means **representation of data**.
- If we talk in terms of classes, then a View class in S60 is derived from **`CAknView`**.
- It is not an actual control.
- It owns a *Container* control (derived from **`CCoeControl`**) to create its view.

What exactly is a Container

- As stated above it is derived from **`CCoeControl`**.
- It actually is a control, all the controls are derived from abstract base class **`CCoeControl`**.
- It is a drawable control and may contain one or more controls.

Why do we need Views

- Most of the normal applications don't need views (derived from **`CAknView`**).
- You need views when your application has multiple screens that form complex navigational paths.
- When you want to save data on every view switching, to update the model with the newly entered or updated data.
- When you want to send data among screens or to external applications.
- It handles switching of views, handling menu-commands, sending keyboard/pointer events to respective container class.

Why do we need a Container

- Well certainly you may want to display some information on the screen, this is the class which will actually have access to screen. For example, you may want to display a list box or a query dialog for user input.
- Most of the normal applications don't require view classes, container will satisfy the basic requirements.

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.