Confirming that the user wants to leave the app

This article explains discusses why it is a "bad idea" to prompt users to confirm whether to exit an app.

Introduction

Windows Phone apps typically use a page stack for navigation - pages are added to the stack as the user navigates "forward" through the app, and removed when the user presses the back button. Going backwards eventually returns the app to the "first page", and the app closes/exits when the back button is pressed again. Pressing the Windows button will "leave" the app, but from a user's perspective will not actually exit it (unless the app is restarted from the start screen).

Whether the user closes or "leaves" the app, the Windows Phone guidelines for this kind of app is that navigation should proceed out of the app without asking the user for confirmation. If necessary the app should automatically save required data and restore it when needed (see Section 5.2.4.x of the Technical certification requirements for Windows Phone for more information on the expected back button behavior).

Note: Other types of apps may have different guidelines - for example games are expected to pause and bring up the menu when the back button is pressed during actual gameplay .

Developers from other platforms may question this design approach. Usually this is due to a lack of understanding of the application framework and UI design.

This article explains why forcing confirmation on exit is not required, and why in fact it is not possible to implement confirmation when "leaving" (via the Windows button).

Confirming app exit

Why try to save user data on exit

Typically apps try and force a confirmation on exit to avoid accidentally closing the app. There are a few motivations behind this.

1. The company behind the app isn't familiar with the OS and closing apps via the back button and so they assume the users won't be either. This is not a good justification because it produces an inconsistent experience across apps in the platform.
2. The navigation model within the app is confusing and users aren't always clear where they are in the app - as a result they don't realize that pressing the back arrow will exit the app. This should be solved with better UX design.
3. Users often close accidentally and have to relaunch the app, which is very slow. Asking if the user wants to close the app is faster than restarting the app and so seen as beneficial. A better solution is to improve startup/launch time. This would also help all users of the app and not just those who close the app accidentally.

The other main reason for wanting to confirm application exit is to store unsaved data. In Windows Phone it is normally better to automatically save data (login credentials being the usual exception) and then restore it (or ask the user if they want it restored when the user returns to the app). The nature of the data, the app and how they return to it can also have an impact on what is most appropriate.

But I really want to ask them for confirmation

This cannot be done "completely", and for the reasons in the previous section it is better to work with the platform than against it.

It's easy to ask a user to confirm that they want to leave an app (or page) by overriding the `OnBackKeyPress` method on a page:

```
protected override void OnBackKeyPress(System.ComponentModel.CancelEventArgs e)
{
    var result = MessageBox.Show("Are you sure you want to exit the app?", "Please confirm", MessageBoxButton.OKCancel);
    e.Cancel = result != MessageBoxResult.OK;
    base.OnBackKeyPress(e);
}
```

However this will not prevent the user from leaving the app in other ways, for example by selecting the "Windows Start" button. While this triggers application level events these cannot block indefinitely (and actually only have a few seconds to run). As a result you cannot ask for user confirmation and block.

This is good platform design. If an app could throw up a message box when someone left an app for any reason (and cancel the navigation dependent upon the result) this would enable the creation of an app that could prevent the user EVER leaving the app. This would be the equivalent of being able to override the other hardware buttons or stop someone launching another app from system UI (E.g. Toasts, notifications or answering incoming calls).

(C) Copyright Nokia 2013. All rights reserved.

References

- Confirming that the user wants to leave the app (Nokia Developer blogs)

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.