How do I suppress the “Did you mean to switch apps?” warning message from my XAML WebView control?

devblogs.microsoft.com/oldnewthing/20181119-00

November 19, 2018

Raymond Chen

A customer had an application that used a UWP XAML WebView control. There are some points at which the user can click an element in the hosted HTML which triggers the main app to navigate to another part of the app entirely. The customer didn’t give any details, but I can imagine this being, say, an app that contains multiple mini-web-apps (say, some web games), and on the main page of each game, there’s a button called *Play a different game*.

Or maybe it’s something else entirely. Whatever.

The customer managed to find a way to do this: The app registered for a protocol, let’s call it `contoso-game:`, and the *Exit* button navigates the WebView control to `contoso-game:mainmenu`. This works fine, but when the user clicks the *Exit* button, there is a warning dialog that appears first:

Did you mean to switch apps?
Did you mean to switch apps?
“Contoso” is trying to open “Contoso”.
Yes No
Is there a way to suppress this dialog box? It looks silly to have the program ask permission to switch to itself.

Your program can handle the `WebView.UnsupportedUriSchemeIdentified` event.

```
<WebView
    ...
    UnsupportedUriSchemeIdentified="OnUnsupportedUriSchemeIdentified" />

void OnUnsupportedUriSchemeIdentified(
    WebView sender,
    WebViewUnsupportedUriSchemeIdentifiedEventArgs e)
{
    // Silently allow all navigations back to the app itself.
    if (e.Uri.Scheme == "contoso-game")
    {
        e.Handled = true;

        // Use the existing URI parser on the main page.
        MainPage.InternalNavigateToUri(e.Uri);
    }
}
```

When the user clicks a link that uses a `contoso-game` protocol, we mark the event as handled (so that the WebView control takes no further action), and then forward the URI to the existing code in our main page that deals with URI activations. Basically, we short-circuit the activation and handle it internally. This is particularly useful if there is more than once running instance of the Contoso program: Short-circuiting the activation means that the navigation is handled by instance the user the clicked on.

Note that intercepting the event from the WebView means that you don't need to register the protocol in their application manifest because the protocol is never activated. Instead, the program intercepts it and performs an internal navigation.

Of course, if the customer were using the `contoso-game` for other reasons, then they should leave it in their manifest.

Raymond Chen

Follow