

Key event handling in S60 Browser 7.1 and S60 WRT 1.1

Overview

The S60 Browser and WRT (for widgets) have offered developers the ability to capture select and navigation key events using either `onKeyPress` or `onClick`, `onKeyDown`, and `onKeyUp` events.

Some early versions of S60 Browser 7.1, however, do not include an implementation for capturing `onKeyPress` events in either the browser or widgets.

Description

S60 Browser 7.1, for S60 5th Edition and the latest S60 3rd Edition, Feature Pack 2 devices, has undergone a major update. This update involves adopting the WebKit core version 525. This new core introduces changes in the way key events are triggered and delegated to the DOM event handlers.

During the implementation of the new version support for identifying key events with `onKeyPress` was accidentally omitted. This omission has been addressed in the browser version 7.1.15679 and `onKeyPress` support will be retained in future versions. However, some devices have already shipped with earlier versions of the browser. This article shows how to create widgets that will work across all browser versions.

In summary, for 7.1 browsers earlier than version 7.1.15679:

- **Centre (Selection key) triggers mouse (`onClick`) events only, key press events (`onKeyPress`) are not triggered.**
- **Navigation keys (left/right/up/down) trigger key (`onKeyDown` and `onKeyUp`) events only, key press events (`onKeyPress`) are not triggered.**

Solution

Developers wanting to ensure that widget or website JavaScript code works on all versions of the S60 Browser should **replace key press events monitoring with monitoring of `onKeyDown` and `onKeyUp` events for the left, right, up, and down keys and `onClick` events for the selection key.**

Code example - handling left/right/up/down key events

Old code

```
document.onkeypress = function(event){
    // add left/right/up/down keypress event handling code here
}
```

New code - migrate to this

```
document.onkeydown = function(event){
    // add left/right/up/down keydown event handling code here
}
```

Additionally, developers relying on key events from the selection key must **migrate to monitoring the click event.**

Code example - handling selection key events

Replace `DOMElement` with a reference to the DOM element you are monitoring.

Old code

```
DOMElement.onkeypress = function(event){
    // code for handling selection key events
}
```

```
}
```

New code - migrate to this

```
DOMelement.onclick = function(event){  
    // add code for handling selection events  
}
```