



<b>ID</b>	CS001160	<b>Creation date</b>	October 30, 2008
<b>Platform</b>	S60 5th Edition	<b>Tested on devices</b>	Nokia 5800 XpressMusic
<b>Category</b>	Web Runtime (WRT)	<b>Subcategory</b>	S60 Platform Services

**Keywords (APIs, classes, methods, functions):** device.getServiceObject(), Service.AppManager

## Overview

This code snippet demonstrates how to use the AppManager Service API to discover the applications that are installed on the device.

## Source: widget.xhtml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <script type="text/javascript" src="script.js" />
    <title>WRT Application</title>
  </head>
  <body>
    <div id="bodyContent" class="bodyContent">
    </div>
  </body>
</html>
```

## Source: script.js

```
var serviceObj = null;

window.onload = init;

// Initializes the widget
function init() {
  // Obtain the AppManager service object
  try {
```

## CS001160\_-\_Listing\_installed\_applications

```
        serviceObj = device.getServiceObject("Service.AppManager",
            "IAppManager");
    } catch (ex) {
        alert("Service object cannot be found.");
        return;
    }

    // We are interested in applications, so let's define the criteria
    // respectively
    var criteria = new Object();
    criteria.Type = "Application";

    // Obtain the list of installed applications
    var result = serviceObj.IAppManager.GetList(criteria);
    var appList = createAppList(result.ReturnValue);
    appList.sort();
    displayList(appList);
}

// Creates the list of installed applications
function createAppList(iterator) {
    var list = new Array();
    try {
        iterator.reset();
        var item;
        while ((item = iterator.getNext()) != undefined) {
            var txt = "";
            txt += item.Caption + " (";
            txt += item.Uid + ")";
            list.push(txt);
        }
    } catch (ex) {
        alert(ex);
    }
    return list;
}

// Displays a list on the screen
function displayList(list) {
    var listElement = document.createElement("ol");
    for (var i = 0; i < list.length; i++) {
        var listItemElement = document.createElement("li");
        var textElement = document.createTextNode(list[i]);
        listItemElement.appendChild(textElement);
        listElement.appendChild(listItemElement);
    }
    var bodyContentElement = document.getElementById("bodyContent");
    bodyContentElement.appendChild(listElement);
}
```

## Postconditions

The example displays an alphabetically ordered list of installed applications and their UIDs on an HTML page.

## Supplementary material

- You can test the application listing features in action in a simple, executable application into which this code snippet has been patched. The application is available for download at:  
[Media:WRTStub\\_CS001160.zip](#)
- You can examine all the changes that are required to implement the above mentioned features in an application. The changes are provided in unified diff and color-coded diff formats:  
[Media:CS001160\\_Listing\\_installed\\_applications.diff.zip](#)
- For general information on applying the patch, see [Using Diffs](#).
- For unpatched stub applications, see [Example stub](#).