Archived:How to enable OTA (Over The Air) SIS install from your website

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

This article show how you can enable over-the-air installation of Symbian installation (SIS) files from your own websites without any settings on server.

Introduction

Enabling OTA installs usually requires you to manually set the correct MIME-Type on the server, so the mobile browser can recognize what kind of data server is returning and install the SIS correctly.

You can manually set the the MIME-Type as described below on your webserver, however this article targets when you don't have access to server settings for custom MIME-Types.

Symbian S60 1st and 2nd Edition

For enabling OTA installs for Symbian S60 1st and 2nd Edition SIS files, you have to set the following MIME-Type on the server:

```
Content-Type: application/vnd.symbian.install
```

Note: The recommended file extension for for 1st is 2nd Edition is .SIS.

Symbian S60 3rd Edition

For enabling OTA installs for Symbian S60 3rd Edition SIS files, you have to set the following MIME-Type on the server:

```
Content-Type: x-epoc/x-sisx-app
```

Note: The recommended file extension for for 3rd Edition SIS files is .SISX.

It is important that you set the correct MIME-Type for 1st/2nd Edition and 3rd Edition SIS files. Incorrect settings may lead to different error-prone behaviour.

How scripts work

Most Server side programming languages allows altering the response headers. What we can do is, rather directly making the link to .SIS file for download, we make the link to a server side script (i.e. PHP or ASP.Net). The script then adds the required MIME-Type in response header, and reads the content of the .SIS file and sends them in response body.

Below you will find code for PHP and ASP.Net scripts to do the same. I have added an extra logic to also detect .SIS and .SISX files to return correct MIME-Type depending on the file extension.

Code for webservers supporting PHP

```
<?php
$filename = $_GET['file'];

if(endsWith($filename, '.sisx'))
{
    header('Content-type: x-epoc/x-sisx-app');
 header('Content-Disposition: attachment; filename="'.$filename.'"');
 readfile($filename);
}
```


(C) Copyright Nokia 2014. All rights reserved.

```
else if(endsWith($filename, '.sis'))
{
    header('Content-type: application/vnd.symbian.install');
 header('Content-Disposition: attachment; filename="'.$filename.'"');
 readfile($filename);
}
else
{
 print('Not file to download');
}

function endsWith( $str, $sub )
{
    return ( substr( $str, -strlen( $sub ) ) == $sub );
}
?>
```

Upload this PHP file on webserver in the same folder as the .SIS file(s), and make the link as follows:

```
<a href="http://myserver.com/downloads/download_sis.php?
file=myapp_s60v3_demo.sisx">MyApp S60 v3</a>
```

Code for webservers supporting ASP.Net

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Web" %>
<%@ Import Namespace="System.Web.UI" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Request["file"] != null)
        {
            string filename = Request["file"];
            if (((filename.EndsWith(".sis") || filename.EndsWith(".sisx"))) &&
System.IO.File.Exists(Server.MapPath(filename)))
            {
                Response.Clear();
                Response.ContentType = (filename.EndsWith(".sisx") ? "x-epoc/x-sisx-app"
: "application/vnd.symbian.install");
                Response.AddHeader("Content-Disposition", "attachment; filename=" +
filename);

                /* for ASP.Net 2 */
                Response.WriteFile(Server.MapPath(filename));
                /****************/

                /* for ASP.Net 1.1 */

//Response.BinaryWrite(System.IO.File.ReadAllBytes(Server.MapPath(filename)));
```

 (C) Copyright Nokia 2014. All rights reserved.

```
                /*******************/
            }
        }
    }
</script>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Download</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
    No file to download.
    </div>
    </form>
</body>
</html>
```

Upload this ASPX file on webserver in the same folder as the .SIS file(s), and make the link as follows:

```
<a href="http://myserver.com/downloads/download_sis.aspx?
file=myapp_s60v3_demo.sisx">MyApp S60 v3</a>
```

You can also add more functionality to these scripts. For example, you can add a download counter code to save it in database. So you can track how many times SIS file is downloaded.

Download

To make things really easy for you, I have also uploaded the files on this wiki. You can download them from the link below.

Download: media:Download sis scripts.zip

About Author

Faisal Iqbal a.k.a. chall3ng3r, is mobile application developer having expertise in Adobe Flash, Adobe Flash Lite, Symbian S60, MS .Net and related technologies.

Find out more at www.orison.biz/blogs/chall3ng3r/

chall3ng3r 22:22, 29 August 2008 (EEST)

 (C) Copyright Nokia 2014. All rights reserved.

 (C) Copyright Nokia 2014. All rights reserved.