

## Re: wceload.exe and /noaskdest in WM5?

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.compactframework/2007-10/msg000>

---

- *From:* "Neville Lang" <[neville@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:neville@xxxxxxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Wed, 3 Oct 2007 01:25:37 +1000
- 

Hi Roby2222,

Sorry about the spacing in the code snippet, it seems to have double spaced all lines. I am glad this code was useful to you.

Regards,  
Neville Lang

"Roby2222" <[robby2222@xxxxxxxxxxxxxxxx](mailto:robby2222@xxxxxxxxxxxxxxxx)> wrote in message  
<news:31DF0046-51EA-4991-9C97-8033ADDC98BD@xxxxxxxxxxxxxxxx>

Neville, thanks very much! We're going to pursue this option. I appreciate the help.

Thanks,  
Roby2222

"Neville Lang" wrote:

In going over my custom installer code from a few years ago, I now remember that someone from Microsoft strongly recommended that we now install apps onto WM5 (and now WM6) devices by using ActiveSync's CeAppMgr.exe and not the WCELOAD.EXE on the device as we were used to in previous device operating systems. As you can see from the code snippet, I first detect whether the device is using WM5 or later and switch the method of CAB delivery. My code has been working fine on WM 2003, WM5 devices and also on WM6 devices. My app is a consumer app so it needs to be able to install

Re: wceload.exe and /noaskdest in WM5?

onto  
any of our customers Pocket PCs or Pocket PC phones.

Though this code may not be exactly what you are looking for, I hope it is  
of use in understanding the way we went after using the hint from the  
Microsoft person.

Regards,  
Neville Lang

~~~~~  
\*\*\* My Custom Installer class (partial code) \*\*\*

```
// Class fields
```

```
private const string CEAPPMGR_PATH =  
@"SOFTWARE\Microsoft\Windows\CurrentVersion\App  
Paths\CEAPPMGR.EXE";  
  
private const string ACTIVESYNC_INSTALL_PATH =  
@"SOFTWARE\Microsoft\Windows  
CE Services";  
  
private const string INSTALLED_DIR = "InstalledDir";  
  
private const string CEAPPMGR_EXE_FILE = @"CEAPPMGR.EXE";  
  
private const string CEAPPMGR_INI_FILE = @"MyApp.ini";  
  
private const string APP_SUBDIR = @"\MyAppFolder";  
  
private const string ACTIVESYNC_EXE_FILE = "WCESMGR.EXE";  
  
private const int BASE_AS = 371; // Base value for ActiveSync  
private  
bool bWM5 = false; // Windows Mobile 5.0 (or later)  
private  
const int WM_MIN_VERSION = 5; // WM5 or later  
  
  
private bool MyInstaller(string MsiSource, string MsiTarget)  
  
{
```

```
string strProdCab1 = "MyApp_PPC.ARM.cab";
string strProdCab2 = "MyApp_PPC.ARMV4.cab";

....

....

//---- Check the version of ActiveSync -----
if (! CheckActiveSync() )
{
return false;
}

// Initialise RAPI
if (!RAPI.IsDeviceConnected(5000))
{
HandleErrors(23);
return false;
}

//---- Get the Operating System version on the device
-----

RAPI.CEOSVERSIONINFO DeviceOS = new
RAPI.CEOSVERSIONINFO();

int iRet = RAPI.CeGetVersionEx(out DeviceOS);

if (iRet == 0)
{
```

Re: wceload.exe and /noaskdest in WM5?

```
HandleErrors(24);

return false;

}

// Check if the Device's operating system is WM5 or later
if (DeviceOS.dwMajorVersion >= WM_MIN_VERSION)

{

this.bWM5 = true;

}

....

....

if (this.bWM5)

{

// Find the location where the application will be installed

string installPath = this.GetAppInstallDirectory();

// Create target directory off the ActiveSync folder for my App

Directory.CreateDirectory(installPath);

// Copy the two CABs and AppMgr INI file to the ActiveSync
directory

System.IO.File.Copy(Path.Combine(MsiTarget, strProdCab1),
Path.Combine(installPath, strProdCab1), true);
```

Re: wceload.exe and /noaskdest in WM5?

```
System.IO.File.Copy(Path.Combine(MsiTarget, strProdCab2),
Path.Combine(installPath, strProdCab2), true);

System.IO.File.Copy(Path.Combine(MsiTarget, strProdCabIni),
Path.Combine(installPath, CEAPPMGR_INI_FILE), true);

// Get the path to CeAppMgr.exe

RegistryKey keyAppMgr =
Registry.LocalMachine.OpenSubKey(CEAPPMGR_PATH);

string appMgrPath = (string) keyAppMgr.GetValue(null);

keyAppMgr.Close();

// Run CeAppMgr.exe to install the files to the device

Process.Start(appMgrPath, "\"" + Path.Combine(installPath,
CEAPPMGR_INI_FILE) + "\"");

}

else // True if device o/s earlier than WM5

{

...

...

RAPI.PROCESS_INFORMATION info = new
CustomInstaller.RAPI.PROCESS_INFORMATION();

Cursor.Current = Cursors.WaitCursor;

RAPI.CopyFileToDevice(MsiTarget + strProdCab2, @"\" + strProdCab2,
true);

RAPI.CeCreateProcess("WCELOAD.EXE", "/noui /noaskdest " + @"\" +
```

Re: wceload.exe and /noaskdest in WM5?

```
strProdCab2, 0, 0, 0, 0, 0, null, 0, ref info);
```

```
Cursor.Current = Cursors.Default;
```

```
}
```

```
} // MyInstaller method
```

```
//=====
```

```
private bool CheckActiveSync()
```

```
{
```

```
if (Environment.OSVersion.Version.Major < 6) // True if Windows  
version is not Vista
```

```
{
```

```
// First check if ActiveSync installed on main computer by  
checking for the Windows CE Services key
```

```
RegistryKey regkey =  
Registry.LocalMachine.OpenSubKey(ACTIVESYNC_INSTALL_PATH);
```

```
if (regkey == null)
```

```
{
```

```
HandleErrors(17);
```

```
return false;
```

```
}
```

Re: wceload.exe and /noaskdest in WM5?

```
// Get the path where ActiveSync installed

string strInstallPath =
Convert.ToString(regkey.GetValue(INSTALLED_DIR));

if (strInstallPath == "") // True if this value not
found

{

HandleErrors(17);

return false;

}

regkey.Close();

// Check the version of ActiveSync on the desktop

string strFilename = strInstallPath + @"\ +
ACTIVESYNC_EXE_FILE;

if (System.IO.File.Exists(strFilename))

{

FileVersionInfo myFVI =
FileVersionInfo.GetVersionInfo(strFilename);

int currentAS = (myFVI.FileMajorPart * 100) +
(myFVI.FileMinorPart * 10) + myFVI.FileBuildPart;

if (currentAS < BASE_AS)

{

HandleErrors(15);

return false;

}

}

else
```

Re: wceload.exe and /noaskdest in WM5?