

Overview

Symbol mobile computers were designed using a Flash File System architecture that supports application persistence. To survive a hard reset (also known as a cold boot), user applications and data should be installed in the non-volatile, flash memory of the mobile computer. The information presented below, applies to Symbol mobile computers running Windows CE v4.2, Windows CE 5.0 and Windows Mobile 2003.

Important Note #1.

This document contains several important notes that are formatted exactly like the one you are reading now. Pay particular attention to these notes. They provide some valuable warnings, guidance and hints. While reading the entire paper is recommended, the key points can be obtained by reading through these important notes.

Flash File System

The Flash File System contains two user accessible folders, Platform and Application. The Platform folder is reserved for system use and contains many of the system files and software drivers. Modification of the Platform folder should be avoided. The Application folder is available for customer use and may contain all of the files required for end-user applications.

RegMerge and CopyFiles

RegMerge and CopyFiles are system components that allow files to persist across a cold boot. These components run as a part of the hard reset process and copy files and registry settings from the flash folders to the volatile memory of the device. The volatile memory (or RAM) folders include: the root directory, Windows, My Documents and Program Files.

RegMerge and CopyFiles are Built-in drivers that are launched early in the boot sequence by the Windows CE Device.exe process.

RegMerge

RegMerge is a built-in driver that allows entries found in .REG files to be stored in the RAM-based, Windows CE registry. RegMerge runs during a cold boot and looks for .REG files, first in the root of the Platform folder and then in the root of the Application folder. It merges the registry changes found in these files into the system registry. For example, what follows is an excerpt from one of the REG files found in the Application folder of an MC9000 mobile computer:

```
[HKEY_CURRENT_USER\Software\Symbol\Launcher\symsetup\Settings]
"DetailColumns"=dword:00000001
"ButtonColumns"=dword:00000002
"TitleBar"="Test Applications"
"Number"=dword:00000000
"Mode"="Large Icons"
"UseProgramIcons"=dword:00000001
"Return"="Back"
```

Besides creating keys and values in the registry, RegMerge is also capable of deleting them. To remove an existing registry key, precede its name with a minus character ('-'). For example:

```
[-HKEY_LOCAL_MACHINE\Software\Symbol\App1]
```

To remove an existing registry value, assign a minus character ('-') to it. For example:

```
[HKEY_LOCAL_MACHINE\Software\Symbol\App1]
"Option1"=-
```

CopyFiles

CopyFiles is a built-in driver that is used to copy files from non-volatile memory, such as the Platform and Application folders, to RAM-based folders such as the Windows and Program Files folders. During a cold boot, CopyFiles looks for files with a .CPY extension in the root of the Platform and Application folders. Much like the RegMerge process, this process looks for CPY files, first in the root of the Platform folder and then in the root of the Application folder. These text files contain a list of copy commands that specify a source and destination separated by a ">" character. For example:

```
\Application\ScanSamp2.exe > \Windows\ScanSamp2.exe
```

This line directs CopyFiles to copy the ScanSamp2.exe application from the \Application folder to the \Windows folder. If the destination folder does not exist, then CopyFiles creates it.

In addition to copying the program file, CopyFiles can also be used to place an application shortcut in the Start Menu. The following example illustrates this:

```
\application\MC50 Demo.lnk > \%\WSM%\MC50 Demo.lnk
```

The “%WSM%” is a string substitution variable that is replaced at run time with the string “Windows\Start Menu” on English systems or the translation of this string on localized systems. The string substitution variables supported by CopyFiles are as follows:

Variable	English String	Windows Mobile	CE. NET
%WSU%	Windows\Startup	x	x
%WSM%	Windows\Start Menu	x	
%WSMP	Windows\Start Menu\Programs	x	
%WSMS%	Windows\Start Menu\Settings	x	
%WP%	Windows\Programs		x
%WDT%	Windows\Desktop		x

The Device Configuration Package (DCP) for each mobile computer contains examples of .REG and .CPY files. DCP's can be downloaded from the Symbol Developer Zone at <http://devzone.symbol.com>.

Important Note #2.

Because the CopyFiles process runs early in the boot sequence, CPY files can not be processed from Compact Flash (CF) or Secure Digital (SD) memory.

Making the Application Persist

To make an application persist across a hard reset, a REG file must be created to set the program options in the registry and a CPY file must be created to copy all of the program and data files from the Application folder to RAM folders. Multiple REG and CPY files can be used for a single application. To test the installation, all of the program files, including the REG files and CPY files, must be transferred to the Application folder of

the device using ActiveSync, and a hard reset performed on the device. The mobile computer is now ready to run the newly installed application.

Making Application Data Persist

Since the data collected by many applications is critical to the operation of the business, care should be taken to ensure the data does not get corrupted. One important way to avoid possible data loss is by storing the data collected by an application into non-volatile memory. By using non-volatile, or flash, memory the data will persist even when power is removed from the mobile computer for extended periods of time or there is a power related failure. Storing application data into non-volatile memory is easily achieved by ensuring the data files created by the application reside on the \Application folder.

The \Application folder is accessed using a FAT file system. Because of the inherent vulnerabilities of the FAT file system, the amount of time that files are left open should be minimized. Programs should close or flush data files after each update. An alternate approach would be to keep data files in the RAM-based file system which is transaction based, and periodically copy the files to the flash file system.

Important Note #3.

The “\Platform” folder is reserved for system use. Application data files should not be stored under the “\Platform” folder.

Packaging and Deployment

Application files can be packaged into a HEX file using Terminal Configuration Manager (TCM), which is supplied with the mobile computer's DCP. Once packaged, this HEX file can be loaded onto the mobile computer using TCM.

An alternate method for application installation employs Windows CE CAB files. The application files and registry settings can be packaged as a CAB file and placed in the Application folder. Using a shortcut file placed in the “Application\Startup” folder and a CPY file placed in the Application folder, this CAB file can be automatically copied to the Windows folder and launched on each cold boot. Use the Microsoft CABWiz utility, or one of the other commercially available CAB file managers, to create the CAB package.

For mass deployments of packaged applications, Symbol AirBeam or Symbol Mobility Services Platform (MSP) should be considered.

Running Applications from RAM

As discussed earlier, Symbol recommends that applications be stored on non-volatile media, such as Flash File System or Compact Flash. This increases the integrity of the system by ensuring that applications are never lost due to a critical power failure. However, Symbol does not recommend running applications directly from non-volatile media due to delays imposed by Windows CE in accessing files following a resume from suspend. In particular, applications that are launched immediately following resume should not be run directly from non-volatile media.

Automatic Startup of Applications

Most mobile enterprise applications require a single program to take over the display and hide the distractions of the remainder of the device. To achieve this focus on productivity, the customer application needs to automatically launch when the device is rebooted. All Symbol mobile computers provide for the automatic launch of programs at boot time. This includes both a hard reset (cold boot) and a soft reset (warm boot). The methods are presented in the order in which they are processed by the system.

OS Launch Keys

The first method of auto-launch is achieved through the use of built-in OS launch keys, which are processed by the Windows CE operating system. These keys define the system processes to load and the order in which they are to be launched. The processes are defined in the registry at the following location:

`HKEY_LOCAL_MACHINE\Init`

Here are some examples of OS launch keys:

Launch20	Device.exe	(starts RegMerge.dll and CopyFiles.dll)
Launch30	GWES.exe	(starts touch and display drivers)
Launch50	Explorer.exe	(launches programs in Windows startup folder)
Launch70	Startup.exe	(Symbol Startup process)

Important Note #4.

The OS launch keys are for system use only and must not be modified.

Windows Startup Folder

One of the OS launch keys defined above invokes Microsoft Windows Explorer. Part of the Explorer.exe startup process, is the launching of files located in the \Windows\Startup folder.

Important Note #5.

The order of execution is not guaranteed for items launched from the \Windows\Startup folder. Programs are launched in the order that they are found in the file system.

Programs launched from this folder may run simultaneous with the Windows Welcome.exe process, sometimes referred to as the “Dentist Appointment” screen. To insure that your programs launch after the Welcome.exe process is complete, use the \Application\Startup folder described below.

Important Note #6.

Use the Symbol \Application\Startup folder for launching applications that must start after the Welcome.exe process completes.

Symbol Startup Program Keys

The second method of auto-launch is achieved through the use of the Symbol Startup Program keys. One of the OS launch keys defined above (Launch70) invokes a Symbol created process called Startup.exe. The Symbol Startup process controls the launching of predefined programs when a device is re-booted. The Startup registry keys can be used to launch applications as well as install add-on components such as a printer driver package or the Microsoft .NET Compact Framework. The processes launched and the order in which they are launched are defined in the registry at the following location:

HKEY_CURRENT_USER\Software\Symbol\Startup\Programs

Here are some examples of Symbol Startup Program Keys:

Prog3 Microsoft .NET Compact Framework

INSTALLING APPLICATIONS FOR PERSISTENCE v2.0

Prog4	Symbol .NET Class Libraries
Prog5	Symbol AirBeam Abboot.exe
Prog7	Symbol AirBeam Abstart.exe
Prog10	Symbol Printer Drivers

For each ProgN key, the following subkeys can be set:

Name	Application file name with full path spec (string)
Command	Application command line parameters (string)
Continue	Block or not block, 1 = do not wait for program to finish (dword)
ColdBootOnly	Run on cold boot, 1 = run on cold only (dword)

A total of 20 Startup Program Keys can be defined.

Important Note #7.

Many of the Symbol Startup Program Keys are reserved for system use and caution should be used to avoid conflicts. Although these keys are not documented, customers have used them to install and launch applications. Before using a Symbol Startup Program Key, check the registry to see which of the 20 keys are defined.

Symbol Startup Folder

In addition to processing the “ProgN” registry keys defined above, the Symbol Startup process looks in a special folder for applications to launch. The Symbol Startup folder defines a set of applications to launch as well as packages to install. ***This is the preferred method for installing and launching customer applications.*** The items to launch are found in the following folder of the flash file system:

\Application\Startup

This location is user-definable in the registry as follows:

HKEY_CURRENT_USER\Software\Symbol\Startup\Path

The default value for this key is: “\Application\Startup”

The files placed in the Symbol Startup folder can be either executable (*.exe) or special Symbol defined run files (*.run). A ".RUN" is a text file with two lines. The first line is the complete name of the executable file (with full path information), and the second line is an optional command line argument to be passed to that executable file. For example, to launch OTL with the string "default" as the command line, create the OTL.run file with the following contents:

\Application\Demo\Otl.exe
default

Startup.exe also provides the ability on a Pocket PC device to delay the starting up of user-specified application until after the Welcome process has completed. The Welcome process is often referred to as the “Dentist Appointment Screen”. Any application placed in the “\Application\Startup” folder will not be launched until the Pocket PC Welcome process has completed.

Important Note #8.

The order of execution is not guaranteed for items launched from the “\Application\Startup” folder. Programs are launched in the order that they are found in the file system.

Bypassing the Welcome Screen

A frequently requested feature from enterprise users of Pocket PC devices, is the ability to bypass the Welcome screens. The Welcome screens are a series of pages that are used to calibrate the touch screen, teach how to use a stylus and set the time zone. Beginning with the release of Windows Mobile 2003, the Welcome pages can be disabled using the following registry key:

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Welcome]
"Disable"=dword:FFFFFFFD
```

By setting “Disable” to 0xFFFFFFF, all Welcome pages are disabled, except for Touch Screen Calibration. The value represents a bit mask, where each bit represents a configurable option. If the key does not exist, then all pages are enabled. The configurable bits are as follows:

Disable Touch calibration	0x02
Disable Stylus Help	0x04
Disable Pop Up Menu Help	0x08
Disable Time Zone Set	0x10
Disable Complete Page	0x20

Important Note #9.

Touch Calibration is critical to the proper operation of the device and must not be disabled.

Automatic CAB Installation

To use the startup folder to automatically install a CAB file, the CAB file needs to be placed in the \Application\Startup folder.

If some special processing is required, such as suppressing the UI prompts to the user, a RUN file needs to be created. This run file must launch the Microsoft utility called Wceload.exe with the CAB file name and the desired options on the command line. The syntax for calling Wceload is as follows:

Wceload [Options] <CAB file>

<CAB file> Specifies the location of the cab file to install or remove.

[Options] Specifies one of the following command line options

/noaskdest	Specifies that the user is not prompted for the installation directory.
/askdest	Specifies that the user is prompted for the installation directory.
/delete 0	Does not remove the cab file
/delete 1	Removes the cab file. This is the default value.
/delete 2	Treats the file as blocks of data, and dynamically deletes them.

The following options are new for Windows CE 5.0

/noui	User not prompted for input during the install. By default, prompts answered 'Yes'.
/nouninstall	Installed application cannot be removed. If option used, unload file not generated.

By default, an unload file is generated during installation with **wceload**. The unload filename has the format *<Software Provider Name> <Program Name>*. This file will not be generated if the *nouninstall* option is used.

Prior to Windows CE 5.0, a silent install could only be achieved using the headless version of “wceload”, called Wceldcmd.exe.

Important Note #10.

Because Wceldcmd is not officially supported by Microsoft for devices that have displays, caution should be exercised when using it.

Windows Mobile 5.0

Microsoft Windows Mobile 5.0 introduces a new persistent storage model. Under this new model, the Windows CE Registry and the folders that were previously RAM-based, such as the Windows folder, are now located in non-volatile memory. Although the application deployment concepts presented in this paper will be supported on Windows Mobile 5.0 devices for backward compatibility, customers should avoid using RegMerge and CopyFiles, and move toward a more standard CAB file deployment strategy.

Important Note #11.

Support for RegMerge and CopyFiles is not guaranteed in future versions of Windows Mobile 5.0 devices from Symbol Technologies. Customers should begin using the more standard CAB file method of application deployment.

Although tools have been provided in the past, which allow customization of the Platform folder, most customers confine their changes to the Application folder. The Platform folder contains important system files, which if removed, could cause instability in the device.

Important Note #12.

The first Windows Mobile 5.0 devices from Symbol Technologies will not contain a user accessible Platform folder, so customers should avoid using or even changing this folder.