CAB Files

How to create cab files

From MSDN: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcepbguide5/html/wce50concabwizardoverview.asp

Creating an .inf File

An .inf file consists of a number of sections that describe the target location of the files, shortcuts, and registry settings that are to be in the .cab file.

Macro Strings

For the installed application to execute, you must define a shortcut to the executable file that resides in the .inf file. The .inf file is then placed in the Start menu.

For debugging purposes, you can place applications into the Windows\Start Menu directory so that they can be executed.

You must place dynamic-link library (.dll) files in the Windows directory.

When creating an .inf file for a target device installation package you use macro strings to represent destination directories in the .inf file. This is important because directories can have different names depending on the language the target device is localized into.

The following table shows the available macro strings and their corresponding directory for a target device intended for an English-speaking audience.

Macro string	Directory
%CE1%	Program Files
%CE2%	Windows
%CE4%	Windows\StartUp
%CE5%	My Documents
%CE8%	Program Files\Games
%CE11%	Windows\Start Menu\Programs
%CE14%	Windows\Start Menu\Programs\Games
%CE15%	Windows\Fonts
%CE17%	Windows\Start Menu

Sections

The following table shows the sections that make up an .inf file.

Section	Description
Version	Describes the creator and version of the application.
CEStrings	Contains string substitutions for application and directory names.
Strings	Contains string definitions for one or more strings.
CEDevice	Describes the hardware platform the application is targeted for.
DefaultInstall	Describes the default installation of the application.
CopyFiles	Describes the default files to copy to the target device.
AddReg	Contains keys and values that the .cab file adds to the registry on the target device.
CEShortcuts	Contains shortcuts that the installation application creates on the target device.
SourceDisksNames	Contains the name and path of the disk where the application resides.
SourceDisksFiles	Contains the name and path of the files containing the application.
DestinationDirs	Contains the names and paths of the destination directories for the application on the target device.

The CAB Wizard can use a single .inf file and multiple application binaries to create multiple .cab files.

You can create separate .cab files for each specific processor or hardware platform type.

To indicate information for a specific processor or hardware platform, append a unique label that indicates the hardware platform to the following section names:

- CEDevice
- DefaultInstall
- SourceDisksNames
- SourceDisksFiles

For example, when installing the .cab file application on a hardware platform that uses the SH4 processor, use the SourceDisksNames.SH4 section name to specify the name and path of the disk where the application resides.

Version

The [Version] section is required and specifies the creator of the file and other relevant information.

```
[Version]
Signature = "signature_name"
Provider   = "INF_creator"
CESignature = "$Windows CE$"
```

Parameters

signature_name
: The name of the operating system (OS), which must have a value of either "$Windows NT$" or "$Windows 95$".

INF_creator
: The name of the company that created the application.

The following code example uses this format:

```
[Version]
Signature = "$Windows NT$"
Provider = "Microsoft"
CESignature = "$Windows CE$"
```

CEStrings

The [CEStrings] section is required and contains string substitutions for the application name and the default installation directory.

```
[CEStrings]
AppName   = app_name
InstallDir   = default_install_dir
```

Parameters

app_name
: The name of the application.

Other instances of %AppName% in the .inf file are replaced with this string value.

default_install_dir
: The name of the default installation directory.

Other instances of %InstallDir% in the .inf file are replaced with this string value.

The following code example uses this format:

```
[CEStrings]
AppName="Game Pack"
InstallDir=%CE1%\%AppName%
```

Strings

The [Strings] section is optional and defines one or more string keys, which represent strings of printable characters.

```
[Strings]
string_key = value
[string_key = value]
```

Parameters

value

String consisting of letters, digits, or other printable characters.

If the corresponding string key is used in an item that requires double quotation marks, enclose the value in double quotation marks ("" "").

The following code example uses this format:

```
[Strings]
reg_path = Software\Microsoft\My Test App
```

CEDevice

The [CEDevice] section is optional and describes the hardware platform that your application is targeted for.

All keys in this section are optional.

If a key is nonexistent, Microsoft® Windows® CE does not perform the checks specified for that key.

If a key has no data, Windows CE does not perform checking.

The exception is UnsupportedPlatforms. If this key exists but there is no data, the previous value is not overridden.

```
[CEDevice]
[ProcessorType =[processor_type]]
[UnsupportedPlatforms = platform_family_name[,platform_family_name]]
[VersionMin = [major_version.minor_version]]
[VersionMax = [major_version.minor_version]]
```

Parameters

processor_type

Value returned by SYSTEM_INFO.dwProcessorType.

platform_family_name

A list of hardware platform family names that are known to be unsupported.

If the platform_family_name specified in the [CEDevice.xxx] section is different from that in the [CEDevice] section, both platform_family_name values are unsupported for the microprocessor specified by xxx.

The list of specific unsupported hardware platform family names is appended to the previous list of unsupported hardware platform family names.

Application Manager does not display the application for an unsupported hardware platform.

If the .cab file is copied to an unsupported hardware platform, the user is warned during the setup process.

In the following code example, the [CEDevice] section is independent of the hardware platform and sets UnsupportedPlatforms to pltfrm1, but [CEDevice.SH4] appends its own list of unsupported hardware platforms:

```
[CEDevice]
UnsupportedPlatforms = pltfrm1
[CEDevice.SH4]
UnsupportedPlatforms =
```

major_version and minor_version

Numeric values that are returned by OSVERSIONINFO.dwVersionMinor and OSVERSIONINFO.dwVersionMajor.

The .cab file is valid for the currently connected target device if the version of the target device is less than or equal to VersionMax and also greater than or equal to VersionMin.

For Windows CE 2.0-based Japanese-language target devices, set VersionMin and VersionMax to 2.01.

If you use Microsoft® ActiveSync® 3.0 or later, use VersionMin and VersionMax to distinguish between black and white and color displays on a Palm-size PC.

The following table describes how to declare VersionMin and VersionMax to distinguish between different displays.

VersionMin value	VersionMax value	Palm-size PC version	Display
2.1	2.1	1.0	Black and white
2.11	2.11	2.0	Color
3.0	>3.0	3.0	Version 3.0 color

The following code example shows how to use one .inf file to distinguish between different Palm-size PC hardware capabilities:

```
[CEDevice.PPC_1]; black&white Palm-size PC 1.0
VersionMin = 2.1
VersionMax = 2.1
UnsupportedPlatforms = "HPC","HPC Pro"

[CEDevice.PPC_2]; color Palm-size PC 2.0
VersionMin = 2.11
VersionMax = 2.11
UnsupportedPlatforms = "HPC","HPC Pro"

[CEDevice.PPC_3]; color Palm-size PC 3.0 and above
VersionMin = 3.0
VersionMax = 1000.0; an arbitrary high value
UnsupportedPlatforms = "HPC","HPC Pro"
```

The following code example shows three [CEDevice] sections. One gives basic information for any CPU. The next two are specific to the SH4 and the MIPS microprocessors, and create .cab files that are valid only for the given microprocessor.

Note To create the two CPU-specific .cab files for the setup .inf file in this code example, you must run the CAB Wizard with the /cpu sh4 mips parameter.

The following section specifies version 2.0 target devices only, using VersionMin and VersionMax.

```
[CEDevice]
UnsupportedPlatforms = pltfrm1
VersionMin = 2.0
VersionMax = 2.0
```

The following section inherits all [CEDevice] settings, but overrides the version settings so that no version checking is performed:

```
[CEDevice.SH4]
ProcessorType = 10003
UnsupportedPlatforms =
VersionMin =
VersionMax =
```

The following section inherits all [CEDevice] settings, and appends pltfrm2 to UnsupportedPlatforms so that pltfrm1 and pltfrm2 are not supported for the MIPS .cab file:

```
[CEDevice.MIPS]
ProcessorType = 4000
UnsupportedPlatforms =pltfrm2
```

See Also

SYSTEM_INFO | Creating an .inf File | CAB Wizard Overview | CAB Wizard Syntax

DefaultInstall

The [DefaultInstall] section is required and describes the default installation of your application.

```
[DefaultInstall]
Copyfiles=copyfile_list_section[,copyfile_list_section]
AddReg=add_registry_section[,add_registry_section]
[CEShortcuts=shortcut_list_section[,shortcut_list_section]]
[CESetupDLL=setup_DLL]
[CESelfRegister=self_reg_DLL_filename[,self_reg_DLL_filename]
```

Parameters

shortcut_list_section

String that identifies one more sections which define shortcuts to a file, as defined in the [CEShortcuts] section.

setup_DLL

Optional string that specifies a Setup.dll file.

It contains customized functions for operations during installation and removal of the application.

The file must be specified in the [SourceDisksFiles] section.

self_reg_DLL_filename

String that identifies files which self-register by exporting the DllRegisterServer and DllUnregisterServer Component Object Model (COM) functions.

You must specify the files in the [SourceDisksFiles] section.

During installation, if installation on the target device fails to call the exported DllRegisterServer function of the file, the exported DllUnregisterServer function of the file is not called during removal.

The following code example uses this format:

```
[DefaultInstall]
CopyFiles = CopyToInstallDir,CopyToWindows
AddReg = RegSettings
CEShortcuts = Shortcuts
```

CopyFiles

The Copyfiles key, under the [DefaultInstall] section, is required and describes the default files to copy to the target device.

```
[copyfile_list_section]
destination_filename,[source_filename],[,flags]
[destination_filename,[source_filename],[,flags]]
```

Parameters

destination_filename

Specifies the destination filename.

source_filename

Optional if it is identical to destination_filename.

flags

Contains a numeric value that specifies an action to be done while copying files.

The following table shows the values that are supported by Windows CE.

Flag	Value	Description
COPYFLG_WARN_IF_SKIP	0x00000001	Warn a user if an attempt is made to skip a file after an error occurs.
COPYFLG_NOSKIP	0x00000002	Do not allow a user to skip copying a file.
COPYFLG_NO_OVERWRITE	0x00000010	Do not overwrite a file in the destination directory.
COPYFLG_REPLACEONLY	0x00000400	Copy the source file to the destination directory only if the file is in the destination directory.
CE_COPYFLG_NO_DATE_DIALOG	0x20000000	Do not copy files if the target file is newer.
CE_COPYFLG_NODATECHECK	0x40000000	Ignore date while overwriting the target file.
CE_COPYFLG_SHARED	0x80000000	Create a reference when a shared DLL is counted.

The following [CopyFiles] code example renames Help.htm and warns if it is skipped; WinGame.wav is renamed and marked as shared:

```
[DefaultInstall]
CopyFiles = CopyFilesSection

[CopyFilesSection]
"Sample Help.htm",Help.htm,,0x00000001
"Win Game.wav",WinGame.wav,,0x80000000
```

AddReg

The AddReg key, under the [DefaultInstall] section, is required and describes the keys and values that the .cab file adds to the target device registry.

```
[add_registry_section]
registry_root_string, subkey,[value_name], flags, value[,value]
[registry_root_string, subkey,[value_name], flags, value[,value]]
```

Parameters

registry_root_string

String that specifies the registry root location.

The following table shows the values that are supported by Windows CE:

Root string	Description
HKCR	The same as HKEY_CLASSES_ROOT
HKCU	The same as HKEY_CURRENT_USER
HKLM	The same as HKEY_LOCAL_MACHINE

value_name

Registry value name. If empty, the "(default)" registry value name is used.

flags

Contains a numeric value that specifies information about the registry key.

The following table shows the values that are supported by Windows CE.

Flag	Value	Description
FLG_ADDREG_NOCLOBBER	0x00000002	If the registry key exists, do not overwrite it. This flag can be used with all other flags in this table.
FLG_ADDREG_TYPE_SZ	0x00000000	The REG_SZ registry data type.
FLG_ADDREG_TYPE_MULTI_SZ	0x00010000	The REG_MULTI_SZ registry data type. The value field that follows can be a list of strings separated by commas.
FLG_ADDREG_TYPE_BINARY	0x00000001	The REG_BINARY registry data type. The value field that follows must be a list of numeric values separated by commas, one byte per field, and must not use the 0x hexadecimal prefix.
FLG_ADDREG_TYPE_DWORD	0x00010001	The REG_DWORD data type. Only the noncompatible format in the Win32 Setup .inf documentation is supported.

The following code example sets a sample application version number, using "alpha" as the "(default)" registry value. It also sets test equal to 3, and new and another equal to 6.

```
[DefaultInstall]
AddReg = RegSettings
[Strings]
reg_path = Software\Company\AppName

[RegSettings]
HKLM,%reg_path%,,0x00000000,alpha
HKLM,%reg_path%,test,0x00010001,3
HKLM,%reg_path%\new,another,0x00010001,6
```

CEShortcuts

The CEShortcuts key, a Windows CE-specific key under the [DefaultInstall] section, is optional and describes the shortcuts that the installation application creates on the target device.

```
[shortcut_list_section]
shortcut_filename,shortcut_type_flag,target_file/path[,standard_destination_path]
```

```
[shortcut_filename,shortcut_type_flag,target_file/path[,standard_destination_path]]
```

Parameters

shortcut_filename

String that identifies the shortcut name.

It does not require the .lnk extension.

shortcut_type_flag

Numeric value.

Zero or empty represents a shortcut to a file.

Any nonzero numeric value represents a shortcut to a folder.

target_file/path

String value that specifies the destination file or folder.

For a file, use the target file name — for example, MyApp.exe — that must be defined in a file copy list.

For a path, use a file_list_section name defined in the [DestinationDirs] section — for example, DefaultDestDir — or the %InstallDir% string.

standard_destination_path

Optional string value; a standard %CEx% path or %InstallDir%.

If no value is specified, one of the following is used:

- The shortcut_list_section name of the current section
- The DefaultDestDir value from the [DestinationDirs] section

The following code example uses the [DestinationDirs] path; the second line directly specifies a path:

```
[DefaultInstall]
CEShortcuts = Shortcuts

[Shortcuts]
Sample App,0,sample.exe
Sample App,0,sample.exe,%InstallDir%
```

In the following code example, which creates a shortcut to a Help file named AppHelp.htm, [Shortcuts] uses the path that [DestinationDirs] specifies:

```
[DefaultInstall]
CEShortcuts = Shortcuts

[DestinationDirs]
Shortcuts = 0,%CE2%\Help

[Shortcuts]
App Help Link,0,AppHelp.htm
```

SourceDisksNames

The [SourceDisksNames] section is required and describes the name and path of the disk where your application resides.

```
[SourceDisksNames]
disk_id= ,comment,,path
[disk_id= ,comment,,path]
```

Parameters

disk_id

Source identifier used to specify the source directory.

comment

Friendly description of the source directory.

path

Path of the disk where your application resides.

You must have a [SourceDisksNames] section that is independent of the hardware platform. Additional hardware platform-specific sections are optional.

You can use absolute or relative paths to specify the location of your application.

The following code example uses this format:

```
[SourceDisksNames]
1 = ,"Common files",,C:\app\common

[SourceDisksNames.SH4]
2 = ,"SH4 files",,sh4

[SourceDisksNames.MIPS]
2 = ,"MIPS files",,mips
```

SourceDisksFiles

The [SourceDisksFiles] section is required and describes the name and path of the files where your application resides.

```
[SourceDisksFiles]
filename=disk_id[,subdir]
[filename=disk_id[,subdir]]
```

Parameters

filename

Source filename. Enclose long filenames in double quotation marks.

disk_id

Source identifier used in [SourceDiskNames] to specify the source directory.

You must have a [SourceDisksFiles] section that is not hardware platform-specific; hardware platform-specific sections are optional.

The following [SourceDisksFiles] code example uses source identifiers corresponding to those found in the [SourceDisksName] code example:

```
[SourceDisksFiles]
begin.wav = 1
end.wav = 1
sample.hlp = 1

[SourceDisksFiles.SH4]
sample.exe = 2

[SourceDisksFiles.MIPS]
sample.exe = 2\
```

DestinationDirs

The [DestinationDirs] section is required and describes the names and paths of the destination directories for your application on the target device.

```
[DestinationDirs]
file_list_section = 0,subdir
[file_list_section = 0,subdir]
[DefaultDestDir=0,subdir]
```

Parameters

file_list_section

Section name used in "[DefaultInstall] CopyFiles".

subdir

Destination directory, which uses the format of an absolute target device path, a directory macro, or the install directory %InstallDir%.

The following [DestinationDirs] code example uses the string %CE2%, which corresponds to the Windows directory, and the string %CE1%, which corresponds to the Program Files directory:

```
[DestinationDirs]
Files.Common   = 0,%CE1%\My Subdir
Files.Shared   = 0,%CE2%
```

CAB Wizard Syntax

After you create the .inf file, use the CAB Wizard to create the .cab file.

The following example shows the command-line syntax for the CAB Wizard:

```
cabwiz.exe "inf_file" [/dest dest_dir] [/err err_file]
[/cpu platform_label [platform_label]]
```

Parameters

inf_file

Full path and filename to the CAB Wizard Setup .inf file.

dest_dir

Destination directory for the .cab files.

The default destination is the inf_file directory.

err_file

File name for a log file containing all warnings and errors that are encountered when the .cab files are compiled.

If no file name is specified, errors are displayed in message boxes.

If a file name is used, the CAB Wizard runs without the user interface (UI); this is useful for automated builds.

platform_label

Creates a .cab file for each hardware platform label that you specify.

A hardware platform label is a label that is used in the Win32 setup .inf file to differentiate between different hardware platform and processor types.

The /cpu parameter, followed by multiple platform_label values, must be the last qualifier in the command line.

The following code example creates .cab files for the SH4 and MIPS microprocessors, assuming that the setup .inf file contains the SH4 and MIPS hardware platform labels:

```
cabwiz.exe  "c:\myfile.inf"  /err myfile.err  /cpu  sh4  mips
```