

Windows Mobile/CE advanced Host installation

1.0 Table of contents

1.0	Table of content	1
2.0	Introduction.....	2
2.1	Prerequisites.....	2
2.2	WiseMo Host support overview, CPU types and WM/CE versions.....	2
3.0	Standard installation	3
4.0	myCloud customized installation	5
5.0	Custom installation.....	8
5.1	Installation files	8
5.2	General installation information	9
5.3	Add Host app to a device image.....	10
5.4	Creating a CAB installer	11
6.0	WiseMo Host information and control API	13
6.1	API references and configuration.....	13
6.2	Demo program	16
7.0	Appendix 1 – Windows CE host.inf file.....	19
8.0	Appendix 2 – Windows Mobile (Smartphone and PocketPC) host.inf file	24
9.0	Appendix 3 - Host.exe command line arguments	29

2.0 Introduction

Although Windows Mobile and Windows CE are legacy operating systems they are still in use by various devices. While Windows Mobile has limited use today, Windows CE remains a critical OS for embedded systems, known for its modularity and real-time capabilities, and continues to be used by specialized devices. This document describes features for advanced configuration and installation of the Windows Mobile and CE Host. Specifically, the document describes how to include Host configuration and license in the installation and how to build your own CAB installer. Also, the document describes the WiseMo Host message based API to send and receive commands and information. Appendix 3 describes command line parameters for the main Host app.

2.1 Prerequisites

The document assumes knowledge of Windows Mobile and Windows CE – hereafter referred to as WM/CE. Although WM/CE have different user interfaces they are quite similar in many ways and will be described together in this document.

Building a CAB installer assumes general administrator knowledge of how to install and run command line tools in Windows.

The document also assumes general knowledge of the WiseMo Guest and Host modules and the terminology used herein – for example what a Guest and a Host is.

The description of the message based API assumes knowledge of Windows Mobile C++ programming.

2.2 WiseMo Host support overview, CPU types and WM/CE versions

A number of installation packages are available for installing the WiseMo Host on WM/CE devices. These installation packages cover different Windows Mobile and Windows CE versions and different CPU types (x86 and ARM based architectures). The packages don't have any restrictions for subversions of the x86 and ARM CPUs because it causes more problems than it solves – for example, newer CPUs aren't covered by the fixed CPU flags and an installation package with CPU restriction can only target one particular CPU type leading high number of (almost) identical installation packages. The missing CPU restriction means that an incorrect installation package can be installed on an incompatible device and hence trying to run the WiseMo Host app will fail. Nothing will happen to the device but please beware that you install the right CPU version.

The installation packages are available as .cab files for installation directly on the Mobile device, see the table below.

OS Version	CPU	Installation File
Windows CE 5, CE 6 and Embedded Compact 7	ARM, PXA, Cortex	WiseMoHost_CE500_ARMV4I.cab
Windows CE 5, CE 6 and Embedded Compact 7	x86	WiseMoHost_CE500_x86.cab
Windows Embedded Compact 2013, CE 8	ARM, PXA, Cortex	WiseMoHost_CE800_ARMV7.cab
Windows Embedded Compact 2013, CE 8	x86	WiseMoHost_CE800_x86.cab
Windows Mobile 6.5 Handheld Windows Mobile 6.0, 6.1, 6.5 Classic and Professional Windows Mobile 5.0 for Pocket PC (Devices with touch screens)	ARM, PXA, Cortex	WiseMoHost_WM5PPC.cab
Windows Mobile 6.0, 6.1, 6.5 Standard Windows Mobile 5 for Smartphone (Devices <u>without</u> touch screens)	ARM, PXA, Cortex	WiseMoHost_WM5SP.cab

The installation is also often provided as an .MSI package for installation via a Windows PC – but with the important limitation that it requires that Windows Mobile Device Center is installed on the Windows PC which is only possible for Windows 7 and older (on Windows 10 with various tricks).

To configure advanced settings for the Host you must download and install the Windows Host Manager. A stand-alone Host Manager for Android and Windows Mobile/CE can be downloaded [here](#).

The Windows Host Manager is included in the MSI Host installation packages.

3.0 Standard installation

Download or copy the appropriate WiseMo Host installation package to the device. On the device simply double click the .cab file and the installation will start. The installation program will ask a few questions and subsequently install the WiseMo Host on the device.

When choosing the installation folder, please be careful because some devices have folders and entire disks that will be restored to a device image during a hard or soft reset (cold or warm boot).

If the Host app is installed to a folder that is restored the Host app will simply disappear after a hard or soft reset. Which disks or folders are restored is device specific so one will have to consult the device manual to

learn more about that. For large scale deployment it can make sense to add the Host app, associated files and configuration to the device image. Please consult the device manual for information about that.

The .cab file is installed on the device by an WM/CE executable called Wceload.exe.

You can use this tool to install or uninstall the .cab file from a location that you specify. If you do not specify a location, the cab file will be installed or uninstalled in the default location, which is the Program Files directory on your target device.

```
wceload [/noaskdest | /askdest | /noui | /silent | /delete <number>] <cab file location>
```

Parameters

- /noaskdest
Specifies that the user is not prompted for the installation directory.
- /askdest
Specifies that the user is prompted for the installation directory.
- /noui
Specifies that the user will not be prompted for any input during the installation. By default, prompts are answered with 'Yes'.

However, if the .cab file is unsigned, any security-related prompts will default to 'No' for security reasons, and the installation might fail.

This is the same as /silent for legacy compatibility reasons.
- /silent
Suppresses dialog boxes during the installation, and all Yes/No prompts default to 'Yes', unless the .cab file is not signed.
However, if the .cab file is unsigned, any security-related prompts will default to 'No' for security reasons, and the installation might fail.
This is the same as /noui for legacy compatibility reasons.
- /delete <number>
Specifies how Wceload removes the cab file after installation.

Value	Description
-------	-------------

0	Does not remove the cab file.
1	Removes the cab file. This is the default value.
2	Treats the file as chunk blocks of data, and dynamically deletes them.

- cab file location

Specifies the location of the cab file to install or uninstall.

Please note that Wceload.exe is an optional component that might not be included in the image on your device. In such case you must follow the instruction in paragraph *5.0 Custom installation*.

4.0 myCloud customized installation

Before starting to build your own CAB installer because you need to include a license and custom configuration file, you can probably just use a myCloud customized installation. This of course presupposes that you want to use the Host installation in a myCloud setup – but if you do, it’s quite simple.

Login on to myCloud with an Administrator account. Go to **Manage Device > Deployment**.

Host



Screen

A Host app must be installed on computers and devices you want to remote control.

Configurations	Actions
Android Host For Android 4+ devices. Available on the Google Play™ store.	 
macOS Host	  
Mobile Host	   
Windows Host	  
Windows Host Custom	  
Add new +	

In the Host column find the **Add new +** link and click it to create a new configuration. Give the configuration a name, chose **Mobile Host** for product and select the target platform.

Create new configuration

Create or modify an installation package. The installation package will be preconfigured to connect to your myCloud domain. The installation package can also include a specific license key and a customized configuration file.

Configuration

Configuration name:*

Platform

Choose product:*

- Windows Host
- Mac Host
- Mobile Host

Target platform(s):

If you are going to use the installation with myCloud, you do not have to specify a License key. If you have a license key, you can specify it in **License key** otherwise leave the field empty.

To use a custom Host configuration file, create it with the Windows Host Manager or copy it from an existing device. Click **Browse** and choose the applicable Host configuration file.

Finally, click **Save**.

This will send you back to the deployment files. Find your new configuration and click the name. From the window that pops up, click **Download Mobile Installer**.

Download the WiseMo Host for Windows CE 5, 6 and Embedded Compact 7 (ARM/PXA/Cortex)

The WiseMo Host allows your mobile device to be remotely controlled by computers running a WiseMo Guest program.

Note: Your mobile device must be connected to the internet during the installation.

Recommended download:

[Download Mobile Installer \(CAB\), 3.36Mb](#)

Distribute or copy the CAB file to the device and install it as described in paragraph 3.0 *Standard installation*

Please note the installation on the device requires internet access during the installation.

5.0 Custom installation

Sometimes, the provided CAB installation files from WiseMo may not suit your needs, you want to include customized Host UI strings, CAB installation is not supported by your device () or you might want to embed the Host app configuration (host.xml) and license (host.lic) files directly into the CAB installer (see also paragraph 4.0 *myCloud customized installation*). In such cases, you might want to create a custom CAB installation package.

Alternatively, you might want to copy the files to the device manually or by custom scripts. In both cases you will need the information provided in this paragraph.

To proceed with a custom installation, you can request the raw installation files from WiseMo by contacting support@wisemo.com

5.1 Installation files

The raw installation files will be delivered as a ZIP file containing:

- host.exe: The Host application itself.
- hostd.exe: A daemon application used to (re)start the Host.
- inventory.dll: Used by the Host app to generate an inventory of the device (optional).
- instca.dll: Custom actions for the installation program.
- host.inf: The recipe for the installation file (further details below).
- sthostce.xml: Modifications to built-in text strings if no file for the current locale is found (optional).
- Keeylook.exe: A utility to inspect device keystrokes, used when creating skins for the device (optional).
- dtlspy.exe: A support log generator program (optional).

None of these files are digitally signed by WiseMo; you will need to sign them with your own certificate if required. If you prefer using WiseMo signed files, you may copy them from an installation using the provided installation packages. Note that unsigned files may trigger warning messages on some devices or prevent you from making silent installations. For more information on signing mobile products, search for "VeriSign Mobile2Market" online.

You can include the following files in your installation:

- host.xml: Host configuration file (optional). Include this file for a predefined configuration to avoid the Host app prompting for configuration at the first start. This file can be copied from an existing configuration or generated by the WiseMo Host Manager on a Windows computer.
- host.lic: Host license file (optional). Include this file to avoid the Host prompting for licensing at the first start. This file can be copied from an existing configuration.
- sthostce_???.xml: Locale-specific modifications (e.g., ENU for US English, ENG for UK English). Include this file to support multiple locales or use sthostce.xml for general localization.

These additional files can be created or generated as needed.

5.2 General installation information

All the above mentioned files must reside in one directory. The chosen directory may be anywhere on the mobile device but choose a directory that is not deleted during a soft or hard reset.

Create a registry key named "InstDir" or "InstallDir" under [HKEY_LOCAL_MACHINE\Software\Apps\WiseMo Host], and set its string value to the chosen directory name. If no registry key is found, the default directory is "\Program Files\WiseMo Host." If creating a custom CAB file, Microsoft Windows CE will handle this entry creation.

If you want to have a shortcut for the Host app in the 'Programs' menu, place a shortcut to Hostd.exe (not to Host.exe) there. On an English mobile device that directory would be called \Windows\Start Menu\Programs.

If you are copying the files manually to a device (i.e. not using the cab installer) and if you want the Host app to start automatically on boot, you can, depending on device type, either:

- Windows CE devices: Create a registry value in [HKEY_LOCAL_MACHINE\init].
- Windows CE devices from Symbol: Create a registry key in [HKEY_CURRENT_USER\Software\Symbol\Startup\Programs]
- All device types: Create a shortcut

Windows CE devices

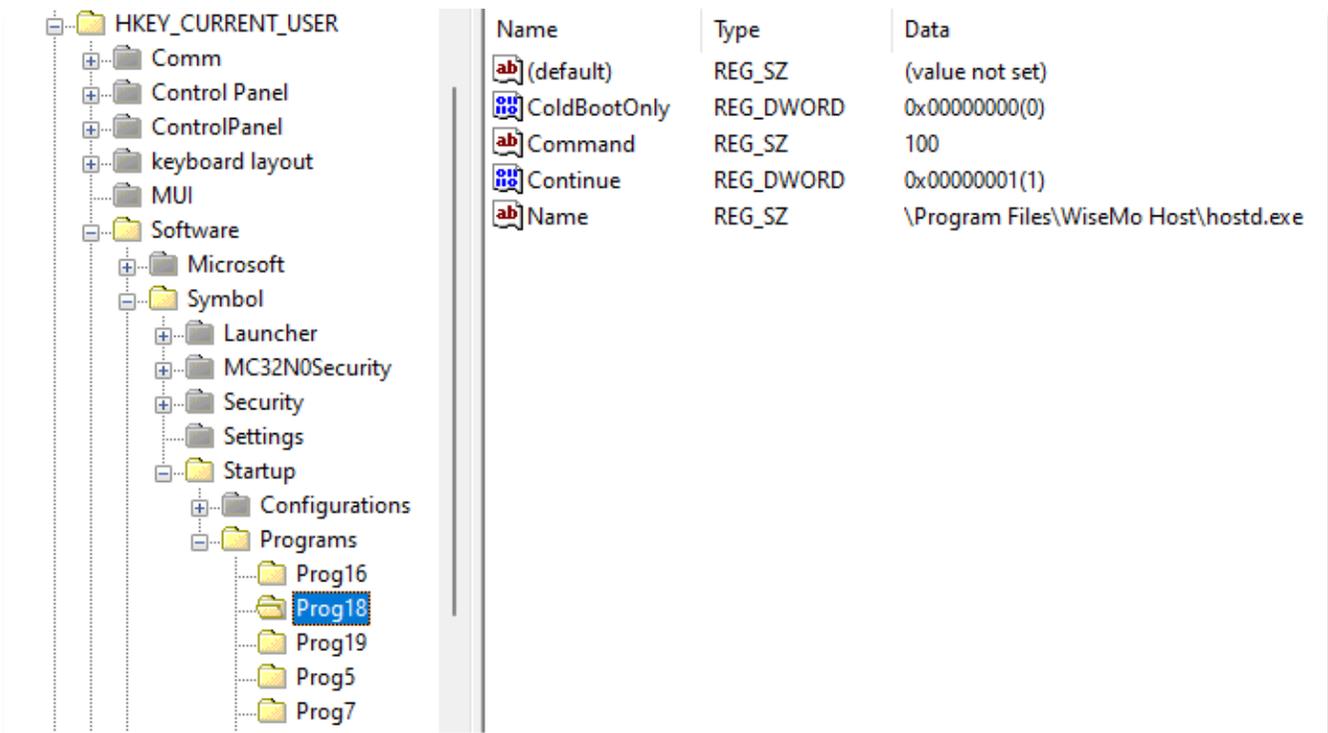
Create a registry value in [HKEY_LOCAL_MACHINE\init]. The registry value should be a String type and named "LaunchNN" (where NN is an available number in the range from 80 to 99) with the string value "\<YOUR_INSTALL_DIR>\hostd.exe".

If the Host should depend on another launch process (or processes) to be launched first, add a "DependNN" REG_BINARY type entry where NN is the number from "LaunchNN" and specify the process it should depend on in WORD Hex format, e.g. if the Host has Launch entry "Launch88" and depend on "Launch50", you would add "Depend88" specify "32 00" as the binary entry (0x32 is 50 decimal):

 Launch50	REG_SZ	explorer.exe
 Launch88	REG_SZ	\Program Files\WiseMo Host\hostd.exe
 Depend88	REG_BINARY	32 00

Windows CE devices from Symbol

If you have a device from the manufacturer Symbol (Now Zebra) you should do it a bit differently. Add a registry key under [HKEY_CURRENT_USER\Software\Symbol\Startup\Programs]. The registry key should be named "ProgNN" (where NN is an available number in the range from 1 to 99). Under this key specify the values as illustrated in the image below.



Shortcut method

Create a shortcut to hostd.exe in the device Start Up folder (normally \Windows\StartUp).

On Smartphones [HKEY_LOCAL_MACHINE\init] is protected and you should use the Start Up folder method by placing a link to hostd.exe in the Startup folder (normally \Windows\Startup).

load_at_boot

In the default Host configuration, hostd.exe will be set to start automatically at boot time. You can deactivate that by creating a host.xml file with "load_at_boot" set to "0". Setting "load_at_boot" to "0" will delete the above registry keys.

If "load_at_boot" is enabled ("1"), the above registry configuration will be adjusted accordingly when the Host app starts the first time.

5.3 Add Host app to a device image

If you are deploying the WiseMo Host to many devices you might want to include the Host installation files and configuration into a device image. To do so, you follow one of the other methods for getting the Host installed and then follow the device manufacturer's procedure for generating an image.

5.4 Creating a CAB installer

You create a CAB file by first creating an .inf file, and then using the CAB Wizard command-line tool (Cabwiz.exe) at a command prompt. The CAB Wizard is delivered with the Microsoft Windows Mobile SDK and can be found in <Platform Builder installation path>\CEPB\Bin.

CAB Wizard

If you do not have the Windows Mobile SDK, you can follow this procedure that also avoids a large installation and only extracts the necessary files:

1. Download [Windows Mobile 6 Standard SDK Refresh.msi](#)
2. Extract the contents of the .msi file using 7-Zip:
7z.exe x "Windows Mobile 6 Standard SDK Refresh.msi"
3. Extract the contents of sdkfiles.cab:
7z.exe x sdkfiles.cab
4. Find the following files and rename them to:
 - a. CGen_Common_CabWiz_Cabwiz.exe -> Cabwiz.exe
 - b. CGen_Common_CabWiz_makecab.exe -> makecab.exe
 - c. CGen_Common_CabWiz_cabwiz.ddf -> cabwiz.ddf

If you check the Details property of Cabwiz.exe, it should be version 4.5.5102.

.inf file

An .inf file consists of a number of sections, each of which starts with a section name in [brackets]. A section describes for example the target location of the files, shortcuts, and registry settings that will be contained within the cab file.

Please refer to [Creating an .inf File](#) on Microsoft.

It is beyond the scope of this guide to describe all aspects of the .inf file but if you start with the host.inf file from WiseMo, you are quite far. To get the sample host.inf file from WiseMo, follow the procedure described in paragraph *5.0 Custom installation*.

The CAB files created using an .inf file are generally processor specific. As described WiseMo has generally avoided using processor specific CAB files but if you want to create processor-specific CAB files, you must append the processor type extension to some of the section names. For example:

- [DefaultInstall] - shared by all platforms
- [DefaultInstall.sh3] - specific to the SH3 processor
- [DefaultInstall.mips] - specific to the MIPS processor

The WiseMo host.inf is not processor specific. The file contains a number of comments that you should follow. Please note that the WiseMo host.inf files are not exactly the same for different platforms – most notably, the

“StartupEntry” specifications should be included in smartphones and PocketPC installations and excluded on CE installation.

You may want to distribute additional files generated by yourself, for example the host.xml (Host configuration) and the host.lic file (contains the Host license). Specify these files in the [SourceDisksFiles] and [Files.App] sections.

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

CAB Wizard syntax:

```
Cabwiz inf_file [/dest dest_dir] [/err err_file] [/nouninstall] [/cpu cpu_type [cpu_type]] [/platform
platform_name] [/compress]
```

Parameters

- inf_file
Full path and filename to the CAB Wizard Setup .inf file.
- /dest
Destination directory for the .cab files. The default destination is the inf_file directory.
- /err
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 the window.
- /nouninstall
Do not allow this application to be uninstalled
- /cpu
The platform_label creates a .cab file for each hardware platform label that you specify. The /cpu parameter, followed by multiple platform_label values, must be the last qualifier in the command line. Specify for example “/cpu sh3 mips” to build cab files for the sh3 and mips platforms.
- /compress
Enables compression of CAB contents

The simplest example is just to execute the following command line that assumes that all files are in the same folder:

```
cabwiz host.inf
```

This will generate a cab file in the same folder with the name host.cab.

Test it on a device to see if it works as expected.

6.0 WiseMo Host information and control API

The WiseMo Host offers a message based API to send and receive commands and information that you can interface to from a customer program. WiseMo also offers a sample program to illustrate its use.

6.1 API references and configuration

The API is based on the standard Windows WM_COPYDATA messages and the accompanying COPYDATASTRUCT. You can send the commands listed as WP_xxx below and receive various events also listed below.

For reference, the COPYDATASTRUCT looks like this:

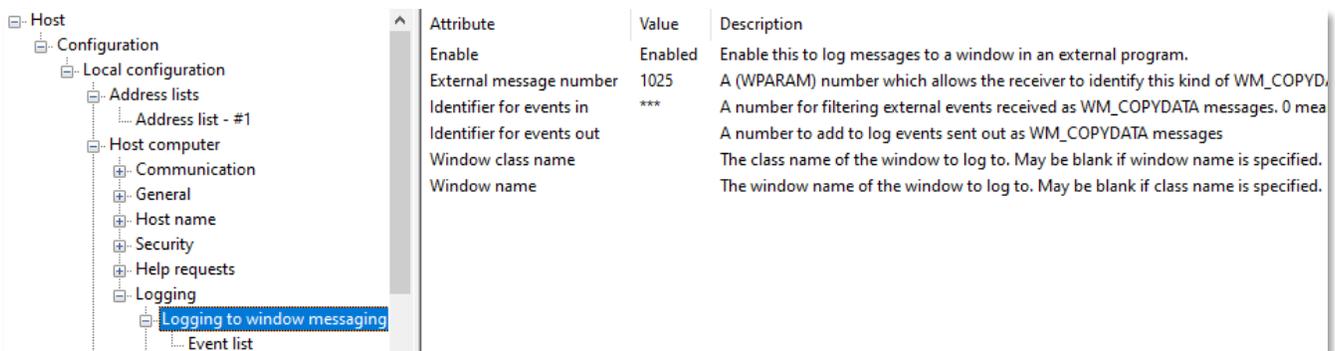
```
typedef struct tagCOPYDATASTRUCT {
    ULONG_PTR dwData;

    DWORD      cbData;

    PVOID      lpData;
} COPYDATASTRUCT, *PCOPYDATASTRUCT;
```

To receive events, you must configure it in the host.xml file. Use the WiseMo Mobile Host Manager (C:\Program Files (x86)\WiseMo\WiseMo RSM\Remote Desktop Host Manager\hostmanager.exe) installed with Mobile Host MSI installer on Windows (PS: this is not the normal Windows Host Manager). Otherwise please request it from WiseMo at support@wisemo.com

Open the host.xml file and navigate to "Logging > Log to Window Messages":



The screenshot shows the configuration tool interface. On the left, a tree view shows the configuration hierarchy: Host > Configuration > Local configuration > Address lists > Host computer > Communication > General > Host name > Security > Help requests > Logging > Logging to window messaging. The 'Logging to window messaging' item is selected and highlighted in blue. On the right, a table lists the attributes for this configuration item:

Attribute	Value	Description
Enable	Enabled	Enable this to log messages to a window in an external program.
External message number	1025	A (WPARAM) number which allows the receiver to identify this kind of WM_COPYDATA.
Identifier for events in	***	A number for filtering external events received as WM_COPYDATA messages. 0 means all.
Identifier for events out		A number to add to log events sent out as WM_COPYDATA messages
Window class name		The class name of the window to log to. May be blank if window name is specified.
Window name		The window name of the window to log to. May be blank if class name is specified.

Enter the window class name and/or the window name in the applicable attributes. The Host program will retrieve the Windows handle from the window class name and/or the window name (using FindWindow(...)) and send Windows messages to it (by using SendMessage(...)). Specify the "External message number" you want to receive in the dwData member of COPYDATASTRUCT (this is not the wParam in SendMessage(...)).

The messages you send and receive use the following message structure.

```
typedef struct _ExtMess {
    char    pass[32]; // Null terminated - max 31 characters - UTF8 - To filter mistakes away
    int     cmd;      // 4 bytes - command number - only for "outgoing"
    char    text[1];  // Null terminated - unlimited - User's choice of encoding
} ExtMess;
```

It consists of a 32 bytes long null-terminated UTF-8 identifier string to give some degree of safety, filtering away possible stray messages. Next, a four bytes long little endian integer number represents the command or event number. Lastly, a null terminated text string of in principle any length is given.

The encoding of that string may vary with individual commands and events.

If you send a message to the WiseMo Host, the WiseMo Host will only execute the command if the identifier string matches the string set as "Identifier for events in" under "Logging" - "Log to Window Messages" in the Host Manager. If that string is left blank (default), all messages are executed.

The WiseMo Host will send your program a message which includes the string set as "Identifier for events out" under "Logging" - "Log to Window Messages" in the Host Manager. How you take advantage of that is up to you.

Commands accepted by the WiseMo Host

```
#define WP_TEST      0
#define WP_EXIT      1
#define WP_EXIT_NOW  2
#define WP_EXIT_ON_IDLE 3
#define WP_PAUSE     4
#define WP_START     5
#define WP_RESTART   6
#define WP_SEND_CHATMSG 7
#define WP_HIDE      8
#define WP_SHOW      9
#define WP_REQUEST_HELP 10
#define WP_CANCEL_HELP 11
```

Events sent from the WiseMo Host

```
/*  
enum {  
    WiseMo HostAtt_Logevent_Starthost = 145  
    WiseMo HostAtt_Logevent_Stophost,      // 146  
    WiseMo HostAtt_Logevent_RequestHelp,   // 147  
    WiseMo HostAtt_Logevent_CancelHelp,   // 148  
    WiseMo HostAtt_Logevent_StartSession,  // 149  
    WiseMo HostAtt_Logevent_StopSession,   // 150  
    WiseMo HostAtt_Logevent_PasswordRejected, // 151  
    WiseMo HostAtt_Logevent_LoginRejected, // 152  
    WiseMo HostAtt_Logevent_ConfirmAccessDenied, // 153  
    WiseMo HostAtt_Logevent_HostTimeout,   // 154  
    WiseMo HostAtt_Logevent_StartRc,       // 155  
    WiseMo HostAtt_Logevent_StopRc,       // 156  
    WiseMo HostAtt_Logevent_StartChat,     // 157  
    WiseMo HostAtt_Logevent_StopChat,     // 158  
    WiseMo HostAtt_Logevent_StartNfm,     // 159  
    WiseMo HostAtt_Logevent_StopNfm,      // 160  
    WiseMo HostAtt_Logevent_SendFile,     // 161  
    WiseMo HostAtt_Logevent_RecvFile,     // 162  
  
    WiseMo HostAtt_Logevent_RequestHelpPending = 185  
    WiseMo HostAtt_Logevent_RequestHelpNoConnect, // 186  
    WiseMo HostAtt_Logevent_RequestHelpNoAnswer, // 187  
    WiseMo HostAtt_Logevent_ChatMessage,   // 188  
    WiseMo HostAtt_Logevent_Message_Displayed, // 189  
    WiseMo HostAtt_Logevent_Inventory_Delivered, // 190
```

```
}

```

6.2 Demo program

You can request a demo C++ program called ExEvent2 from WiseMo, which works on Mobile 5 ARM Pocket PC.

If you use our demo program, use "ExEvent2" as the name and leave the class blank. Set "Enable Logging" to "Enabled". In the subitem named "Eventlist", you can enable and disable individual messages as needed.

Sample program excerpts

```
// This is a demonstration how one might send WM_COPYDATA events to WiseMo Host
// Feel free to modify or rewrite this code
// The code itself is unsupported, it only serves as a demonstration what you might do
// in your own program

void CDialogSubclass::OnBnClickedButtonSend()
{
    HWND hWnd = ::FindWindow(NULL, L"WiseMo Host");
    COPYDATASTRUCT copydata;
    ExtMess *extmess;
    BOOL used;
    int n;

    if (!hWnd)
    {
        AfxMessageBox(L"No WiseMo Host running");
        return;
    }

    UpdateData(TRUE);

    n = 256; // 32 bytes pass, 4 bytes command, 220 bytes left for text
    extmess = (ExtMess*)malloc(n);
    if (!extmess)
    {
        AfxMessageBox(L"Out of memory");
        return;
    }
    memset(extmess, 0, n);

```

```

// Fill in the CMD field with the selection from the combo box
DWORD wparam = m_combo_cmd_ctrl.GetCurSel();
extmess->cmd = wparam;

// Convert the text from the upper edit field to ANSI and send it as "pass"
WideCharToMultiByte(CP_ACP, 0, m_edit_pass.GetBuffer(1), -1,
    extmess->pass, 32, "?", &used);

// Convert the text from the lower edit field to ANSI and send it as "text"
WideCharToMultiByte(CP_ACP, 0, m_edit_txt.GetBuffer(1), -1,
    extmess->text, 220, "?", &used);

// Fill in the Windows COPYDATASTRUCT and send the event to WiseMo Host
// For some reason the value of wparam in the .dwData member does not get through
// The same value is also stored in the ((ExtMess*)copydata.lpData)->cmd member
copydata.dwData = wparam;
copydata.cbData = n;
copydata.lpData = extmess;
::SendMessage(hWnd, WM_COPYDATA, (WPARAM)this->m_hWnd, (LPARAM)&copydata);

free(extmess);
}

// This is a demonstration how one might interpret WM_COPYDATA events from WiseMo Host
// Feel free to modify or rewrite this code
// The code itself is unsupported, it only serves as a demonstration what you might do
// in your own program

LRESULT CDialogSubclass::WindowProc(UINT message, WPARAM wParam, LPARAM lParam)
{
    switch(message)
    {
        case WM_COPYDATA:
        {
            COPYDATASTRUCT *pcds = (COPYDATASTRUCT*)lParam;
            ExtMess* extmess = pcds ? (ExtMess*)pcds->lpData : NULL;
            bool passok = false;
            char *text = NULL;
        }
    }
}

```

```

if (!extmess)
{
    m_list_in_ctrl.InsertString(0, L"NULL message");
}
else
{
    char ctxt[256];
    WCHAR wtxt[256];

    // This is NOT a strong password check, it is only a filter for mistakes
    if (!strcmp(extmess->pass, "mypass"))
    {
        passok = true;
    }

    // Very (too) simple buffer overflow guard
    // Assume the "rest" takes max 64 bytes of ctxt
    text = extmess->text;
    if (strlen(text) > 256 - 64) text = "overflow";

    sprintf(ctxt, "%s %d %s %s", passok ? "Ok": "!!", extmess->cmd,
        extmess->pass, extmess->text);
    MultiByteToWideChar(CP_ACP, 0, ctxt, -1, wtxt, 256);
    m_list_in_ctrl.InsertString(0, wtxt);
}
}
return TRUE;

default:
    break;
}

return CDialog::WindowProc(message, wParam, lParam);
}

```

7.0 Appendix 1 – Windows CE host.inf file

Below is a sample host.inf file for Windows CE that can be used as a starting point for your own. Copy the content to a plain text file and save it as host.inf extension.

```

;=====
; INF file for building a CAB installer for the WiseMo Host for Windows CE
;=====
;
; Feel free to modify this file and make your own CAB file for your own distribution
; Please note that your modifications are not supported by WiseMo
; and the WiseMo support will not be able to assist you in working with this file

; Do not change anything in the next three sections

; -----
; Describes the creator and version of the application. Don't change
; -----
[Version]
Signature="$Chicago$"
CESignature="$Windows CE$"
Provider=%CompanyName%

; -----
; Contains the name and path of the disk on which the application resides. Don't change
; -----
[SourceDisksNames]
1,"Application Files",,

; -----
; Describes the default installation of the application. Don't change
; -----
[DefaultInstall]
CopyFiles=Files.App
;CEShortcuts=Shortcuts,StartupEntry ; For Smartphone/PocketPC devices
CEShortcuts=Shortcuts ; For CE devices
AddReg=CEaddReg
CESetupDLL="instca.dll"

```

```

; As for Files.App, Shortcuts, StartupEntry and CEaddReg, please refer to the [Files.App],
; [Shortcuts], [StartupEntry] and [CEaddReg] sections below
; PS: CEShortcuts line depends device type

; -----
; Describes the device platform for which the application is targeted
; -----
[CEDevice]
BuildMax=0xE0000000
ProcessorType=0;   Don't check processor type

; BuildMin, BuildMax - The build information returned by OSVERSIONINFO.dwBuildNumber.
; ProcessorType - The numeric processor type returned by SYSTEMINFO.dwProcessorType. SH3 processors
normally have a value of 10003, and MIPS processors have a value of 4000.

; -----
; Contains keys and values that the .cab file will add to the registry on the device. CE Only
; -----
; Never change this registry key. It is the one and only fixpoint in the installation
; If CompanyName is "WiseMo" and AppName is "Host", this line may be excluded, though
; Windows CE will then generate that same key when the CAB file is installed.
[CEaddReg]
HKLM,\SOFTWARE\Apps\WiseMo Host,"InstallDir",,%InstallDir%

; -----
; Contains shortcuts that the installation application creates on the device
; -----
; This is the shortcut for the 'Programs' menu
; The icon inside hostd.exe is then the one visible in that menu
; Different 'brands' of the product use different hostd.exe files, but
; they all use the same host.exe
[Shortcuts]
"WiseMo Host",0,"hostd.exe"

; -----

```

```

; This is the shortcut for the 'Start Up' directory.
; -----
;[StartupEntry]
;%AppName%,0,"hostd.exe"

; On Smartphones and PocketPC
; Use this line for setting 'start at boot'. Remove it to disable 'start at boot'
; because setting the 'start at boot' in the host.xml will normally fail
; as the registry HKLM\init is not writable unless signed with a privileged certificate.
;
; On CE devices, these lines should be commented out

; -----
; Contains string definitions for one or more strings
; -----
; The factory name for the application is 'WiseMo Host'
; Your name for the application will be %CompanyName% %AppName%
; Feel free to change the AppName to anything you have agreed with WiseMo sales
; Observe that installing your application will not automatically uninstall 'WiseMo Host'
; Do not use the characters / and \ (slash and backslash) in AppName
[CEStrings]
InstallDir=%CE1%\WiseMo Host
AppName="Host"

; -----
; Contains string substitutions for application and directory names
; -----
; The factory name for the application is 'WiseMo Host'
; Your name for the application will be %CompanyName% %AppName%
; Feel free to change the AppName to anything you have agreed with WiseMo sales
; Observe that installing your application will not automatically uninstall 'WiseMo Host'
; Do not use the characters / and \ (slash and backslash) in AppName
;
;[Strings]
CompanyName="WiseMo"

; -----

```

```

; Contains the names and the paths of the destination directories for the application on the target device
; -----
[DestinationDirs]
Shortcuts=,%CE11%
;StartupEntry=,%CE4%
Files.App=,%InstallDir%

; On CE devices, StartupEntry should be commented out

; -----
; Contains the name and path of the files containing the application
; -----
; host.exe is the host module. It does the physical remote control operations
;
; hostd.exe is a small console daemon program which starts and restarts host.exe
; It is possible for you to omit the hostd.exe and start up host.exe from your
; own app for example using CreateProcess() when you feel like it. The host.exe app can take different
command line arguments, see Appendix 3 in WM-CE_Host_Advanced_Installation.pdf
;
; dtlspy.exe is a debug trace program for troubleshooting situations. You may omit it from your distribution.
;
; inventory.dll contains the inventory feature. You may omit it from your distribution.
;
; The Host is automatically started after the CAB installation. To prevent it, create an empty file
; named 'dont_start.opt' and uncomment the lines below in [SourceDisksFiles] and [Files.App] to include
; this file into the CAB.
[SourceDisksFiles]
host.exe=1
hostd.exe=1
dtlspyce.exe=1
inventory.dll=1
instca.dll=1
sthostce.xml=1
thanks.html=1
;emconfig.id=1
;dont_start.opt=1

; Additional information for [SourceDisksFiles]
; You may want to distribute additional files generated by yourself, for example:

```

```

;
; host.xml contains the host configuration. You use the WiseMo Host Manager program
; to generate this file on your desktop computer.
;
; host.lic contains the host license. First time you run the host on a mobile device
; which has no license file, you will be prompted for a license key,
; and this file will be generated. You may only distribute the file to other mobile devices
; *** IF YOU HAVE A VALID LICENSE AGREEMENT FOR THE DEVICES YOU DISTRIBUTE TO ***

; sthostce.xml contains text strings for the mobile host application.
; You may change any text here, except the help request names, which are in host.xml

; sthostce_???.xml or sthostce_????.xml contains localized versions of sthostce.xml
; ?? or ??? is ISO abbreviation for the language, as reflected in the mobile host's 'Info' pane.
; Examples
; ENG is UK-English, ENU is US-English, EN is any English variant
; SPA is Spanish as written in Spain, SP is any Spanish variant
; PTG is Portuguese, PTB is Brazilian, PT is any Portuguese variant
; The mobile host figures out what ??? currently is, then searches in the following order
; 1) sthostce_????.xml, sthostce_???.xml, sthostce.xml. If no file is found, the built in texts are used.

; WiseMo does not distribute localized versions of sthostce.xml. A skeleton file is delivered with this file.

; -----
; Files.App
; -----
[Files.App]
host.exe
hostd.exe
dtlspyce.exe
inventory.dll
instca.dll
sthostce.xml
thanks.html
;emconfig.id
;dont_start.opt

```

8.0 Appendix 2 – Windows Mobile (Smartphone and PocketPC) host.inf file

Below is a sample host.inf file for Windows Mobile (Smartphone and PocketPC) that can be used as a starting point for your own. Copy the content to a plain text file and save it as host.inf extension.

```

;=====
; INF file for building a CAB installer for the WiseMo Host for Windows Mobile (Smartphone and PocketPC)
;=====
;
; Feel free to modify this file and make your own CAB file for your own distribution
; Please note that your modifications are not supported by WiseMo
; and the WiseMo support will not be able to assist you in working with this file

; Do not change anything in the next three sections

; -----
; Describes the creator and version of the application. Don't change
; -----
[Version]
Signature="$Chicago$"
CESignature="$Windows CE$"
Provider=%CompanyName%

; -----
; Contains the name and path of the disk on which the application resides. Don't change
; -----
[SourceDisksNames]
1,"Application Files",,

; -----
; Describes the default installation of the application. Don't change
; -----
[DefaultInstall]
CopyFiles=Files.App
CEShortcuts=Shortcuts, StartupEntry ; For Smartphone/PocketPC devices
;CEShortcuts=Shortcuts ; For CE devices
AddReg=CEAddReg
CESetupDLL="instca.dll"

```

```

; As for Files.App, Shortcuts, StartupEntry and CEaddReg, please refer to the [Files.App],
; [Shortcuts], [StartupEntry] and [CEaddReg] sections below
; PS: CESHortcuts line depends device type

; -----
; Describes the device platform for which the application is targeted
; -----
[CEDevice]
BuildMax=0xE0000000
ProcessorType=0; Don't check processor type

; BuildMin, BuildMax - The build information returned by OSVERSIONINFO.dwBuildNumber.
; ProcessorType - The numeric processor type returned by SYSTEMINFO.dwProcessorType. SH3 processors
normally have a value of 10003, and MIPS processors have a value of 4000.

; -----
; Contains keys and values that the .cab file will add to the registry on the device. CE Only
; -----
; Never change this registry key. It is the one and only fixpoint in the installation
; If CompanyName is "WiseMo" and AppName is "Host", this line may be excluded, though
; Windows CE will then generate that same key when the CAB file is installed.
[CEaddReg]
HKLM,\SOFTWARE\Apps\WiseMo Host,"InstallDir",,%InstallDir%

; -----
; Contains shortcuts that the installation application creates on the device
; -----
; This is the shortcut for the 'Programs' menu
; The icon inside hostd.exe is then the one visible in that menu
; Different 'brands' of the product use different hostd.exe files, but
; they all use the same host.exe
[Shortcuts]
"WiseMo Host",0,"hostd.exe"

; -----

```

```

; This is the shortcut for the 'Start Up' directory.
; -----
[StartupEntry]
%AppName%,0,"hostd.exe"

; On Smartphones and PocketPC
; Use this line for setting 'start at boot'. Remove it to disable 'start at boot'
; because setting the 'start at boot' in the host.xml will normally fail
; as the registry HKLM\init is not writable unless signed with a privileged certificate.
;
; On CE devices, these lines should be commented out

; -----
; Contains string definitions for one or more strings
; -----
; The factory name for the application is 'WiseMo Host'
; Your name for the application will be %CompanyName% %AppName%
; Feel free to change the AppName to anything you have agreed with WiseMo sales
; Observe that installing your application will not automatically uninstall 'WiseMo Host'
; Do not use the characters / and \ (slash and backslash) in AppName
[CEStrings]
InstallDir=%CE1%\WiseMo Host
AppName="Host"

; -----
; Contains string substitutions for application and directory names
; -----
; The factory name for the application is 'WiseMo Host'
; Your name for the application will be %CompanyName% %AppName%
; Feel free to change the AppName to anything you have agreed with WiseMo sales
; Observe that installing your application will not automatically uninstall 'WiseMo Host'
; Do not use the characters / and \ (slash and backslash) in AppName
;
[Strings]
CompanyName="WiseMo"

; -----

```

```

; Contains the names and the paths of the destination directories for the application on the target device
; -----
[DestinationDirs]
Shortcuts=,%CE11%
StartupEntry=,%CE4%
Files.App=,%InstallDir%

; On CE devices, StartupEntry should be commented out

; -----
; Contains the name and path of the files containing the application
; -----
; host.exe is the host module. It does the physical remote control operations
;
; hostd.exe is a small console daemon program which starts and restarts host.exe
; It is possible for you to omit the hostd.exe and start up host.exe from your
; own app for example using CreateProcess() when you feel like it. host.exe can start
; a help request from the command line, see the file oem_install.txt
;
; dtlspy.exe is a debug trace program for troubleshooting situations. You may omit it from your distribution.
;
; inventory.dll contains the inventory feature. You may omit it from your distribution.
;
; The Host is automatically started after the CAB installation. To prevent it, create an empty file
; named 'dont_start.opt' and uncomment the lines below in [SourceDisksFiles] and [Files.App] to include
; this file into the CAB.
[SourceDisksFiles]
host.exe=1
hostd.exe=1
dtlspyce.exe=1
inventory.dll=1
instca.dll=1
sthostce.xml=1
thanks.html=1
;emconfig.id=1
;dont_start.opt=1

; Additional information for [SourceDisksFiles]
; You may want to distribute additional files generated by yourself, for example:

```

```

;
; host.xml contains the host configuration. You use the WiseMo Host Manager program
; to generate this file on your desktop computer.
;
; host.lic contains the host license. First time you run the host on a mobile device
; which has no license file, you will be prompted for a license key,
; and this file will be generated. You may only distribute the file to other mobile devices
; *** IF YOU HAVE A VALID LICENSE AGREEMENT FOR THE DEVICES YOU DISTRIBUTE TO ***

; sthostce.xml contains text strings for the mobile host application.
; You may change any text here, except the help request names, which are in host.xml

; sthostce_???.xml or sthostce_????.xml contains localized versions of sthostce.xml
; ?? or ??? is ISO abbreviation for the language, as reflected in the mobile host's 'Info' pane.
; Examples
; ENG is UK-English, ENU is US-English, EN is any English variant
; SPA is Spanish as written in Spain, SP is any Spanish variant
; PTG is Portuguese, PTB is Brazilian, PT is any Portuguese variant
; The mobile host figures out what ??? currently is, then searches in the following order
; 1) sthostce_????.xml, sthostce_???.xml, sthostce.xml. If no file is found, the built in texts are used.

; WiseMo does not distribute localized versions of sthostce.xml. A skeleton file is delivered with this file.

; -----
; Files.App
; -----
[Files.App]
host.exe
hostd.exe
dtlspyce.exe
inventory.dll
instca.dll
sthostce.xml
thanks.html
;emconfig.id
;dont_start.opt

```

9.0 Appendix 3 - Host.exe command line arguments

The host.exe app takes the command line parameters listed in the table below. If you need to specify any of these parameters you must use the host.exe app and not hostd.exe.

Argument	Description
-n	The following name will be used as Host ID, i.e. “-n MyName” will set the Host name to MyName. This overwrites what may be configured in host.xml.
-sh	Start Host app as hidden
-uu	Remove Host references in the registry and exit the Host process.
-rr	Exit Host process and start it again. This can for example be used after having updated the host.xml configuration file.
-sm	Start Host app in silent mode to suppress error messages on startup, for example error messages if the myCloud configuration fails.
-wl <0 or missing = enable (def), 0 = disable>	Activate logging via Windows Messages. “-wl 0” will disable logging even though configured via host.xml
-wn <Window Name>	Log to the window with the following name, i.e. “-wn LogWindow” will search for the window by “LogWindow” with FindWindow function. If the name parameter is omitted, the name configured in host.xml will be used (if any).
-wc <Window Class name>	Log to the window with the following window class name, i.e. “-wc LogWindowClass” will search for the window by “LogWindowClass” with FindWindow function. If the class name parameter is omitted, the name configured in host.xml will be used (if any).
-wo <Identifier>	The specified name will be passed to the receiver of the log messages, e.g. “-wo WiseMoHost”
-wi <Identifier>	The specified identifier will be passed to the receiver of the log messages, e.g. “-wi PassPhrase”

The command line arguments are case insensitive.

The normal Windows Mobile command line processing applies when using parameters with spaces apply. So, a parameter with a space must be enclosed in quotes like for example this “My Name”.

Example 1

Example, which will let WiseMo Host log all events which are enabled in the 'Log to window messages' – 'Eventlist' to another window which has the window class name "WCE_MyApp". By default, all events in the eventlist are enabled, -wl will turn on or off the boolean found in 'Log to window messages' – 'Enable logging', which by default is set to 'disabled'

```
Host.exe -wl -wo "MyApp" -wc "WCE_MyApp"
```

Example 2

Example, which will disable all logging to windows messages even if defined in configuration file

```
Host.exe -wl 0
```

Example 3

Example, which will instruct Host.exe to process only events from messages having "MyID" as identifier

```
host -wi "MyID"
```