



# Windows<sup>®</sup> Embedded

## **New VoIP Solutions in Windows<sup>®</sup> Embedded CE 6.0 R2**

Writer: Douglas Boling

Published: November 2007

**Summary:** Windows Embedded CE 6.0 R2 contains new VoIP solutions. This paper describes those VoIP Solutions

# Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2007 Microsoft Corporation. All rights reserved.

Microsoft, Windows and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

---

# Table of Contents

<b>Introduction</b> .....	<b>1</b>
<b>The VoIP Application Suite</b> .....	<b>1</b>
The Homescreen Application .....	1
The Phone Application.....	2
The Settings Application.....	4
The Bootstrap Application.....	7
<b>Source Potential</b> .....	<b>7</b>
<b>Conclusion</b> .....	<b>9</b>



## Introduction

Windows Embedded CE has been out for a while but the developers at Microsoft have not been resting on their laurels. The embedded team has released an update to Windows Embedded CE 6.0 dubbed Windows Embedded CE 6.0 R2. While this new release improves Windows Embedded CE in a number of areas, this white paper will focus on the improvements in the Voice over IP (VoIP) area.

Windows Embedded CE has supported VoIP functionality since Windows Embedded CE 4.2 with both a VoIP “stack” that provides the basic voice over IP communication as well as a predefined operating system configuration complete with the VoIP stack and a suite of applications created to provide a solution for a Windows Embedded CE powered VoIP phone. Over the years, the VoIP support has been upgraded to improve the features of the stack as well as improve the accompanying applications. This effort has continued in Windows Embedded CE 6.0 R2.

The effort to improve the VoIP functionality extends beyond just the Voice over IP project. Microsoft has provided the complete source code to the VoIP application suite. This source is in the `\wince\public` tree and therefore OEMs are free to modify and use this source code in any Windows Embedded CE based system whether it is a VoIP device or not. This suite of applications provides a great foundation for embedded application suites regardless of the type of system.

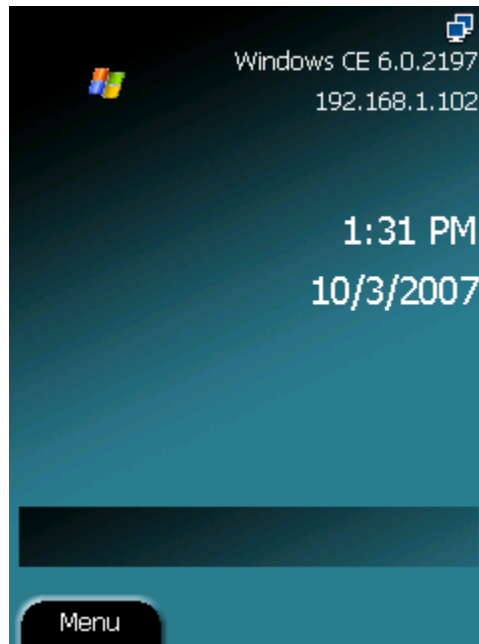
Looking first at the VoIP software, the improvements are centered on added features to the phone software. The new release provides support for video calls, multiparty conferencing, and back-end server compatibility. These new features are welcome improvements to the VoIP package and allow OEMs to build both full-featured desktop video phones as well as hand held handsets with a set of focused features.

## The VoIP Application Suite

The VoIP application suite has been upgraded as well. The VoIP suite includes five applications; the home application, the phone application, the settings application, the information application, and finally a bootstrap application. Each of the applications in the suite also use a set of common code implemented as a set of stand-alone DLLs.

## The Homescreen Application

The home screen application provides the shell functionality for the VoIP suite. It provides a central point for the phone software, launching other applications, providing the network status, voice mail and email state, and monitoring system status such as the memory state. A screen capture of the home screen application is shown below.



**Figure 1. The homescreen application in portrait mode**

The screen capture is taken from a “narrow” implementation of the application suite. There are two complete sets of resources for the VoIP suite, a narrow set and a wide set. The narrow set configures the applications for a 240 by 320 portrait mode screen while the wide set of resources is configured for a wide, 320 by 240 screen. Of course, the resources, like the entire source for the VoIP suite is available to be modified, but these two sets provide the starting point.

The home screen displays a status bar across the top of the display with various status icons. In the above screen, only the network icon is displayed but others are displayed depending on the state of the system. The time and date are displayed in the middle of the screen. This narrow layout is designed for hand held handsets and therefore this user uses only two “application buttons” across the bottom of the screen. The home screen application only uses the left button for the “Menu” button.

## The Phone Application

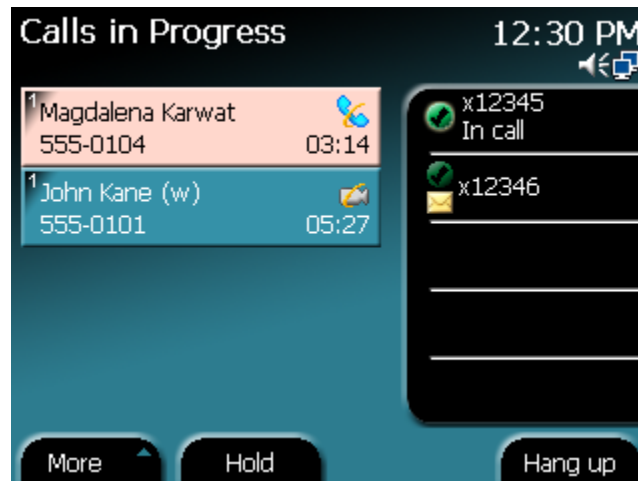
When the user starts typing a phone number on the keypad, the home screen application brings up the phone application which displays a dial window as shown below.



**Figure 2. The phone application shown while dialing a number**

The phone application is responsible for the user interface to the VoIP stack. While the phone application doesn't have much of a user interface, it does have some significant functionality. It receives the dial information from the user, interprets the number depending on the dial plan specified in the provisioning files, and initiates and manages the call through the VoIP stack.

Another view of the phone application is shown below.



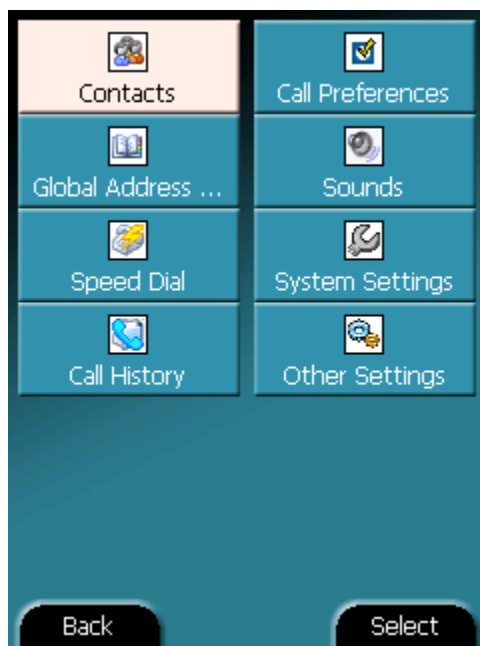
**Figure 3. The phone application while two phone calls are in progress**

This is the wide screen layout of the application that is designed for desktop phones. Notice that in this configuration, the right side of the screen is used to label the various phone lines supported by the phone. In this case, the phone supports two lines, with both active at the time of the screen capture. Also notice, that the wide screen layout supports four application buttons across the bottom of the screen. These locations would coincide with four physical buttons located below the screen.

It's important to understand that in both previous screen shots, the same application is running. The only difference is the resource files being used by the application to provide screen formatting information. Just with these two screen shots, it's easy to see that the resource files provide a high degree of flexibility in the screen layout without changing a single line of C or C++ code.

## The Settings Application

The settings application, PhSettings.exe, provides the user interface to the other applications in the suite. This application is launched when the user selects the "Menu" button on the home screen. The main screen of the settings application is shown below.



**Figure 4. The Settings application shown in portrait mode**

This screen can be configured via an XML file that can be specified in build. This, default XML file, can be further overridden by an XML specified in the registry. Each of the buttons shown above in a grid format will launch either a new application or will display a control panel-like dialog box to configure the features of the VoIP phone. An example of the XML used to configure this screen is shown below

```
<VoipTree>
  <Menu Identifier="TopLevelMenu" ScreenType="Grid">
    <Item Launch="phinfo.exe exchange" Text="Contacts"
      Bitmap="IDB_CONTACTS_ICON">
```

```

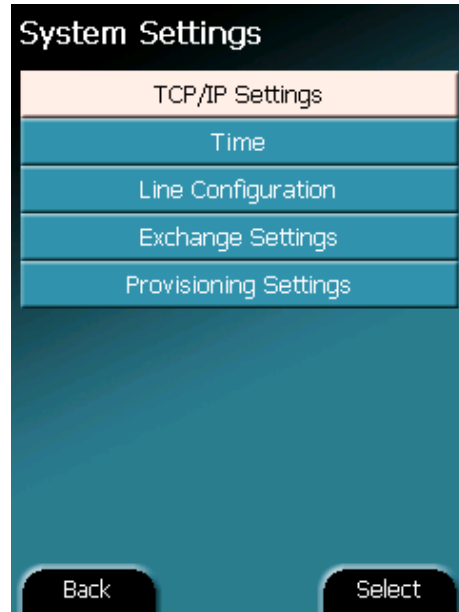
        BitmapSel="IDB_CONTACTS_ICON"/>
    <Item Launch="phinfo.exe gal"           Text="Global Address List"
        Bitmap="IDB_GAL_ICON"
        BitmapSel="IDB_GAL_ICON"/>
    <Item Launch="phinfo.exe speeddial"     Text="Speed Dial"
        Bitmap="IDB_SPEEDDIAL_ICON"
        BitmapSel="IDB_SPEEDDIAL_ICON"/>
    <Item Launch="phinfo.exe calllogs"      Text="Call History"
        Bitmap="IDB_CALLHISTORY_ICON"
        BitmapSel="IDB_CALLHISTORY_ICON"/>
    <Item Launch="phsettings.exe CallPreference"
        Text="Call Preferences"
        Bitmap="IDB_CALLPREFERENCES_ICON"
        BitmapSel="IDB_CALLPREFERENCES_ICON"/>
    <Item Launch="phsettings.exe Sounds"    Text="Sounds"
        Bitmap="IDB_SOUNDS_ICON"
        BitmapSel="IDB_SOUNDS_ICON"/>
    <Item NextMenu="SysSettings"           Text="System Settings"
        Bitmap="IDB_SYSTEMSETTINGS_ICON"
        BitmapSel="IDB_SYSTEMSETTINGS_ICON"/>
    <Item NextMenu="OtherSettings"         Text="Other Settings"
        Bitmap="IDB_OTHERSETTINGS_ICON"
        BitmapSel="IDB_OTHERSETTINGS_ICON"/>
</Menu>

<Menu Identifier="OtherSettings" Title="Other Settings"
        Icon="IDB_MENU_ICON" ScreenType="List">
    <Item Launch="phsettings.exe volume"   Text="Volume"/>
    <Item Launch="phsettings.exe setpin"   Text="Set Pin"/>
    <Item Launch="phsettings.exe calendar" Text="Calendar"/>
</Menu>
<Menu Identifier="SysSettings" Title="System Settings"
        Icon="IDM_MENU_ICON" ScreenType="List">
    <Item Launch="phsettings.exe ipconfig" Text="TCP/IP Settings"/>
    <Item Launch="phsettings.exe TimeSettings"
        Text="Time" />
    <Item Launch="phsettings.exe LineConfig"
        Text="Line Configuration" />
    <Item Launch="phsettings.exe Exchange"
        Text="Exchange Settings" />
    <Item Launch="phsettings.exe provupdate"
        Text="Provisioning Settings" />
</Menu>
</VoipTree>

```

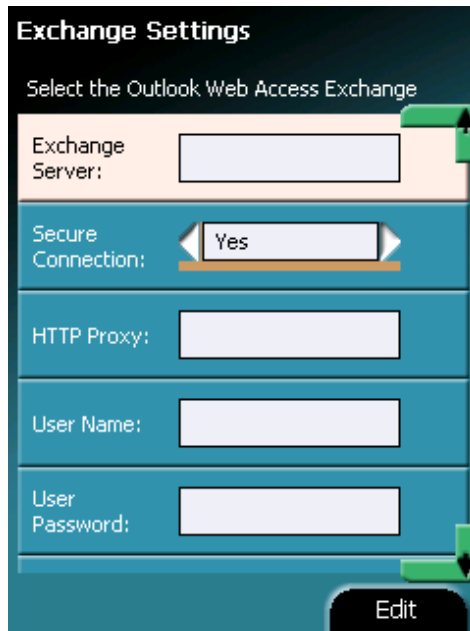
The XML above uses a simple structure to define each of the grid buttons shown on the settings main screen. Notice that each of the first six buttons launch the PhInfo application with various command lines. The PhInfo application will then display the requested information. The last two buttons don't launch an application. Instead, they display submenus on separate screens that then can launch the PhSettings application again with varying command line strings.

Selecting the "System Settings" button above displays the following screen.



**Figure 5. The System Settings control panel screen**

This screen demonstrates the “List” format of the settings application. The items in this list were specified in the XML shown previously. Selecting any of these items will launch a new copy of the PhSettings application with a command line string that will cause a specific dialog box to be displayed. For example, the Exchange Settings item will cause the following dialog to be displayed.



**Figure 6. The Exchange settings window**

This dialog is created using some of the custom controls available to the phone suite. As can be seen from these various screens, the look-and-feel of the application suite doesn't look anything like the classic Windows look-and-feel with its taskbar window and Start button.

The PhInfo application provides the data management portion of the VoIP application suite. It provides a contact list, manages the phone history, as well as interfaces with both Exchange server as well as Outlook via ActiveSync. The screen below shows the PhInfo application (in wide screen format) showing the call history log.



**Figure 7. The PhInfo application showing call history**

## The Bootstrap Application

The last application in the VoIP suite is one of the most important and interesting of the applications, it is the bootstrap application. The bootstrap application starts when the phone initially powers up. It connects with a server and downloads the provisioning information for the phone. It is the bootstrap application that provides the information the phone needs to know "who it is" with information such as its phone number, the phone's location and who's desk the phone sites on.

The user interface of the bootstrap application consists of a simple progress bar as it marks time while finding and interacting with the server but the code behind this application contains jewels that are hard to find these days. It's time to turn to the potential of the source code behind these applications and the use of this code in systems far beyond the realm of voice over IP.

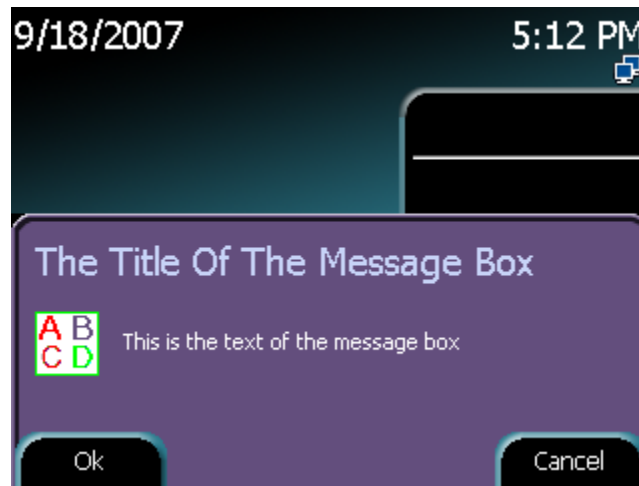
## Source Potential

The VoIP application suite provides a complete ecosystem for a VoIP client system from bootstrap to configuration to communication. While the suite does its job well, the applicability of this suite and the code behind it is much broader than VoIP.

As mentioned above, the complete source code to the VoIP application suite is provided in the Windows Embedded CE build tree. For Windows Embedded CE 6.0 R2, the new suite is provided in the public tree under the FP\_VOIP directory. This is a new directory for the VoIP code superseding the code in the VOIP directory which is still present in the build tree. The old VoIP suite is still available for backward compatibility.

The opportunity of this code is presented in three ways. First, the code can be easily modified for various configurations of VoIP hardware with little or no change to the application code. As mentioned above, layout of the screens and the items in the settings application menus are easily configurable by changing the resources and XML configuration files. This is aptly demonstrated by the narrow and wide screen formats where the two screen layouts are quite different in spite of the fact that they differ only in resource files.

In addition to modifying the look and feel via resource and XML files, the code is structured in a way that makes it fairly easy to write companion applications that have the same look and feel as the current VoIP applications. Many of the useful customized classes and handy functions are packaged in a separate DLL. The exported entry points for this DLL, dubbed PhCommon.dll, provide a simple way to create message boxes, create skinned controls, and other user interface elements that conform to the “skinned” look of the VoIP suite. The figure below shows a customized message box that was displayed with a call to the PhCommon DLL.



**Figure 8. A simple message box using one of the functions in PhCommon**

The message box looks nothing like the standard “Windows” message box that is typically displayed on a Windows Embedded CE system. In addition to PhCommon, all the applications in the VoIP suite are written in C++ with a standard class structure. This class library can be utilized by OEMs needing to write additional applications that extend the function of the VoIP platform.

The third opportunity of the code has nothing to do with VoIP. The application source code provides a great starting point for application suites for other embedded systems. The suite of applications covers most of the basic needs of an embedded suite. The home screen application provides a shell application. The PhSettings application is an example of a configurable application launcher. The PhInfo application demonstrates how to communicate with both Outlook and Exchange server, as well as being a good example of how to manage data using the Windows Embedded CE database API. Finally the bootstrap application demonstrates how to log on to a remote server using the WinINet API and download and interpret XML provisioning files.

## Conclusion

The extensive upgrading of the VoIP stack and application suite is quite useful to Windows Embedded CE OEMs in many ways. Obviously, OEMs interested in the VoIP platform gain added functionality out of the box. The VoIP application suite is also upgraded to provide a better solution for developers. But in addition to the new function, the VoIP suite is a great resource for developers needing to write applications for all types of embedded systems. Regardless of how the VoIP software is used, new example software is always welcome in Windows Embedded CE. Check out the VoIP software, it's pretty cool stuff.

### **For more information:**

Windows Embedded Web site:

<http://www.microsoft.com/windows/embedded/default.mspx>