Motorola A925 SDK Users Guide

Version : 1.0

Date : 14-Mar-2004

Revision History

Revision #	Date	Description
0.1	22-Sept-2003	Initial draft.
1.0	14-Mar-2004	New Format.

Contents

1. Introduction

1.1 Purpose

The purpose of this document is to provide a users guide for the Motorola A925 SDK. Topics on setting up the development environment and issues with using the emulator are covered in this document.

1.2 Target Audience

This document is intended to be utilized by application developers of the Motorola A925.

Familiarity with Symbian OS 7.0, UIQ 2.0 or 2.1, and Java 2 Micro Edition is assumed and recommended.

2. Development Environment Configuration

2.1 Overview

The A925 SDK can be used as a standalone environment or in conjunction with the Symbian 7.0 SDK for UIQ 2.0 or 2.1. The SDK is easier to use as a standalone environment, but this section will allow developers to use the Symbian 7.0 SDK for UIQ framework to work with the A925 SDK.

2.2 A925 SDK w/o Symbian 7.0 SDK for UIQ

If you do not have the Symbian 7.0 SDK for UIQ installed, then the A925 SDK will require the following:

- ActiveState Perl
- Java Runtime Environment

Developers familiar with Symbian programming may wish to map the A925 SDK onto a separate logical drive. This can still be done by using the following command:

```
subst Q: C:\Symbian\A925SDK
```

With the above environment, the following environment variables will need to be added:

- EPOCROOT=\
- PATH=\epoc32\gcc\bin;\epoc32\tools;%PATH%

While using the A925 SDK without a separate logical drive is possible, there are potential incompatibilities on the WINS emulator with applications that do not include drive letters in their file and directory paths.

2.3 A925 SDK with Symbian 7.0 SDK for UIQ

2.3.1 device.exe Kit Management

The use of the devices.exe kit manager is optional for environments with the Symbian 7.0 SDK for UIQ installed.

To add the A925 SDK to the list of supported kits in the Symbian 7.0 SDK for UIQ, run the following from the command line:

```
devices –add C:\Symbian\A925SDK C:\Symbian\A925SDK @A925:com.motorola.A925
```

To make the A925 SDK the default device environment, run the following from the command line:

devices –setdefault @A925:com.motorola.A925

For applications which do not have drive letters in directory and file paths, the use of devices.exe will not correct WINS emulator incompatibilities. Setting C:\Symbian\A925SDK as a separate logical drive corrects those issues.

For more details on devices.exe, refer to the UIQ SDK documentation.

2.3.2 Unique A925 Tools

With the Symbian 7.0 SDK, the PC directory--C:\Program Files\Common Files\Symbian\Tools--is added to the PC's PATH environment variable.

The use of devices.exe from the UIQ SDK points the application stub files in the C:\Program Files\Common Files\Symbian\Tools directory to the <EPOCROOT>\epoc32\tools directory assigned in the devices.exe syntax.

This directory includes stub files for most GCC and generic Symbian tools. But there are two tools that are unique to the Motorola A925 SDK which need to be added to the C:\Program Files\Common Files\Symbian\Tools directory:

- signmidlet – for digital signing of J2ME MIDlets
- wsp – for Winsock operation

To ensure that these two tools are available for the A925 SDK environment with the Symbian 7.0 SDK environment settings, copy the contents of C:\Symbian\A925SDK\epoc32\stub\tools into the C:\Program Files\Common Files\Symbian\Tools directory.

2.4 Native Build Targets

Although the Motorola A925 SDK supports two Application Binary Interfaces (ABIs)—ARMI and THUMB, most application developers should build for THUMB for device releases. ARMI libraries are included due to dependencies on some Motorola THUMB components.

3. Motorola APIs

3.1 Overview

This section briefly describes the various Motorola APIs supported in the Motorola A925 handset supported by the SDK.

For Symbian 7.0 and UIQ 2.0 and 2.1 APIs and usage, please consult the Symbian 7.0 SDK for UIQ 2.0 or 2.1 documentation.

3.2 C++/EPOC APIs

Documentation of the contents and usage of the following APIs are located in the SDK directory: c:\Symbian\A925SDK\docs.

3.2.1 Audio

In addition to the Media Server audio interfaces from the UIQ SDK, the Motorola A925 handset uses a Motorola Audio API for enhanced sound generation and audio playback. Note that audio playback is not supported on the WINS emulator so testing of applications using the Audio API must be performed on the A925 device.

3.2.2 Camera Capture

The A925 handset uses the Camera Capture API for taking image snapshots of with the device’s built-in digital camera. The WINS emulator does not support the Camera Capture API, so applications using it will need to be tested on the target hardware.

3.2.3 Video Playback

The A925 handset uses the Video Playback API for playback of MPEG-4 video content. The WINS emulator does not support the Video Playback API, so applications using video will need to be tested on the handset.

3.3 J2ME Gaming API

The Motorola A925 device supports a J2ME Gaming API. The WINS emulator supports most of the Gaming API except for those dealing with audio playback.

Documentation of the contents and usage of the J2ME Gaming API is located in the directory

C:\Symbian\A925SDK\GamingAPI

The classfiles for building J2ME Gaming API MIDlets are located in the PC directory

C:\Symbian\A925SDK\GamingAPI\bin

4. Motorola A925 Key Mappings

4.1 Overview

In addition to the touchscreen, the Motorola A925 device has 15 separate hardware buttons. The A925 buttons are assigned to specific scancodes and keycodes which are detailed here.

For details on the usage of scancodes and keycodes please consult the UIQ SDK documentation.

4.2 Restrictions

Most device buttons are assigned to specific applications and are unavailable for use by developers. The following keys are “consumed” by other applications:

- Send – consumed by Phone application
- End – consumed by Phone application
- Shortcut – launches app assigned in Control Panel
- Browser – launches Browser app
- Speakerphone – toggles speakerphone functionality
- Volume Up/Down – consumed by audio drivers
- Voice Tag – consumed by Contacts application

Additionally, pressing Game A within three seconds of pressing the Shortcut key will launch the touchscreen calibration. Pressing Game B within three seconds of pressing the Shortcut key locks the touchscreen.

4.3 Available Keys

The buttons that are accessible for developer use are as follows:

- Up – Scancode: EStdKeyDevice4, Keycode: EQuartzKeyFourWayUp
- Down - Scancode: EStdKeyDevice5, Keycode: EQuartzKeyFourWayDown
- Left - Scancode: EStdKeyDevice6, Keycode: EQuartzKeyFourWayLeft
- Right - Scancode: EStdKeyDevice7, Keycode: EQuartzKeyFourWayRight
- Select - Scancode: EStdKeyDevice8, Keycode: EQuartzKeyConfirm
- Game A - Scancode: EStdKeyApplicationA
- Game B - Scancode: EStdKeyApplicationB

The scancodes and keycodes above can be referenced in the e32keys.h and QuartzKeys.h header files.

5. GDB On-Device Debugging for EPOC

5.1 Overview

The Motorola A925 device supports on-device debugging of C++/EPOC applications with the help of the GNU Debugger (GDB). GDB runs on a host computer that is connected to the device via serial interface. A “stub” program running on the device provides the debugging services to GDB.

For details on setting up your project for GDB please consult the UIQ SDK documentation.

For general usage and documentation on the GNU Debugger, please consult the GDB website at http://www.gnu.org/software/gdb/gdb.html.

5.2 Restrictions

The A925 device only supports GDB operations over a serial port (the GDBstub.exe version on the device does not support setting the CSY). Thus, it is necessary for developers who wish to use this functionality to obtain an A925-compatible serial port.

Information about Data Cable **S9141**:

- Adapter SYN0279B with charging port
- 9-pin cable SKN6315A
- Compatible with Motorola 120, 120c, 120e, 120t, 120x, 270c, 280, 60, 60c, 60g, 60i, 60p, 60s, 60t, 60x, T720, T720i, T721, T721g, T721i, T722, T722i, T725, T730, T730c, T730i, T731, Timeport 270c, Timeport 280, v120, v120c, v60, v60c, v60g, v60i, v60p, v60s, v60t, v60x, v66, v70, A830, A920, and A925 cell phones.

Since GDB requires a free serial COM port, Windows PCs with Symbian Connect/Desktop Suite installed will need to disable the serial COM port in mRouter to avoid blocking GDB from using the port.

5.3 Starting GDBstub.exe on the A925

Since the GDBstub.exe, GDBseal.dll, and GDBseng.dll files reside on the A925 device, the developer follows the below steps to run the GDB stub program.

1. Launch File Manager (qfileman) on the device.

2. Goto Z:\System\Programs.

3. Run gdbstub.exe.

4. Plug the serial cable into the device. This step must be run after Step 3. Otherwise, the communication link between GDB and the GDB stub will not be established.

The File Manager (qfileman) can be downloaded from Symbian's developer website:

http://www.symbian.com/developer/downloads/tools.html

6. Redirector On-Device Debugging for Java

6.1 Overview

The Motorola A925 device supports on-device debugging of pJava and J2ME applications with the help of the Redirector application. Redirector takes over all three of Java's standard I/O streams—System.out, System.in, and System.err—and provides you with the means to redirect these streams to a console window. In addition it also allows you to redirect output to a log file, or to a serial port.

Two versions of the Redirector are provided for developers. One is for the WINS target environment and is already bundled on the WINS emulator. The other is built for the THUMB target environment and can be installed on A925 hardware. The Redirect_thumb.SIS application is located in the C:\Symbian\A925SDK\epoc32\tools directory.

For general information on the Redirector tool please consult the UIQ SDK documentation.

6.2 Restrictions

Developers who wish to use Redirector to output standard I/O over the serial port will need to obtain an A925-compatible serial port.

Please consult information in Section 5.2 Restrictions on serial port part numbers and information.

When using Redirector over the serial port, Windows PCs with Symbian Connect/Desktop Suite installed will need to disable the serial COM port in mRouter to avoid blocking terminal software from using the port.

6.3 Starting Redirector on the A925

Running Redirector with Java applications for console or file output on the A925 is described in the UIQ SDK documentation.

However, because the A925 serial port implementation is different than UIQ, running Redirector with the serial port requires the following procedure:

1. Launch Redirector on the device in serial port mode.

2. Launch a Java application which generates standard I/O.

3. Plug the serial cable into the device. This step must be run after Step 2. Otherwise, the communication link between Redirector and the serial port will not be established at cable plugin.

7. Winsock

7.1 Overview

To aid developers in testing internet-aware applications, the Motorola A925 SDK emulator provides a mechanism for accessing the socket connection of the development PC using a Winsock connection.

7.2 Restrictions

The Winsock implementation for the WINS emulator does not support proxies used by the development PC’s network connection.

The Winsock implementation also disables Windows RAS operations when enabled.

7.3 Usage

The Winsock tool is located in the \epoc32\tools directory of the Motorola A925 SDK.

7.3.1 Status

To check whether the Winsock component is enabled or disabled for the WINS emulator, run the following command:

```
wsp s
```

7.3.2 Enable

To enable the Winsock implementation for the WINS emulator, run the following command:

```
wsp e
```

7.3.3 Disable

To disable the Winsock implementation for the WINS emulator, run the following command:

```
wsp d
```

8. Digitally Signed Applications

8.1 Overview

The Motorola A925 supports digitally signed applications, including J2ME MIDlets. This section outlines issues with signing applications—EPOC or J2ME—with digital certificates.

For details on digitally signing a SIS file, see the UIQ SDK documentation.

8.2 Restrictions

Only J2ME MIDlets signed with a RSA-based key are supported in the A925 secure installer. J2ME MIDlets signed with a key not based on RSA will be treated as an unsigned MIDlet.

8.3 SIS PKG Files

8.3.1 Product/Platform Version Compatibility

The A925 supports the product/platform version compatibility feature for SIS file packages. The UID should identify the earliest possible platform version or product to maximize the number of phones the package can be installed on.

As a UIQ 2.0 device, the Motorola A925 supports the UIQ 2.0 Platform ID and the Motorola A920 Platform ID. The Motorola A925 is not a UIQ 2.1 device.

For generic UIQ 2.0 applications, it is recommended to use the UIQ 2.0 ID by inserting the following line into the PKG file:

```
(0x101F617B), 2, 0, 0, {"UIQ20ProductID"}
```

For Motorola A920/A925 applications where unique Motorola APIs are used, it is recommended to use the Motorola A920 Platform ID by inserting the following line into the PKG file:

```
(0x101F8375), 1, 0, 0, {"MotorolaA920PlatformProductID"}
```

If your PKG file supports more than one language, the dependency string will need to be defined multiple times. For example, for a PKG file with support for three languages, the following line should be used:

```
(0x101F617B), 2, 0, 0, {"UIQ20ProductID", "UIQ20ProductID",
"UIQ20ProductID"}
```

8.3.2 Condition Block Usage

The UIQ SDK documentation on PKG file format details the use of condition blocks for controlling the installer based on device attributes in the \epoc32\include\hal_data.h file.

Refer to Appendix B - HALData Attributes for the HALData attribute settings for the Motorola A925.

8.4 J2ME Signing Procedure

The procedure for digitally signing and installing a J2ME MIDlet is as follows:

1. Identify the location of the MIDlet JAR and JAD files.
2. Identify the location of the secure key and certificate on the PC hard drive.
3. Run the “signmidlet” tool in the X:\Symbian\A920Emul\epoc32 with the following syntax:

   signmidlet.exe -k <.key file> -s <.cer file> -d <.jad file> <.jar file>

   where <.key file> is the path and filename of the private key file, <.cer file> is the path and filename of the public key certificate file, <.jad file> is the path and filename of the J2ME JAD file, and <.jar file> is the path and filename of the J2ME JAR file.

Note that digital keys and certificates are not provided by Motorola.

9. Emulator Application Install

9.1 Overview

For development testing, applications can be installed on the Motorola A925 SDK emulator in a number of different ways. This section outlines the different options available to the developer.

9.2 C++/EPOC

9.2.1 WINS UDEB Build Target

By building on a WINS UDEB build target, the developer automatically inserts compiled applications into the SDK emulator environment. The next time the emulator is run, the App Launcher will automatically find the application and display it in the main menu.

9.2.2 Copying Files Into WINS UDEB Directory

If the developer has compiled files, manually copying the compiled files into PC directory

\epoc32\release\wins\udeb\z\system\apps\<app name>\

will “install” the application into the emulator.

Like building in the WINS UDEB build target, the App Launcher will automatically find the app and display it in the main menu.

9.2.3 Using Emulator App Installer

If the developer wishes to test the integrity of a packaged SIS file, copy the SIS file into \epoc32\wins\c\ and run the App Installer from the App Launcher->Launcher->Install on the WINS emulator.

When the emulator’s App Installer displays the locations to install your application, you will see two names with the hard drive volume which the emulator resides on.

The first one listed is the “A:” drive mapped to \epoc32\wins\a\ while the second one is the “C:” drive mapped to \epoc32\wins\c\. For proper operation, you must install applications on the “C:” (2nd) drive.

Note that using the App Installer to uninstall the app is also possible with this implementation.

9.3 J2ME MIDlets

Unlike C++/EPOC applications, J2ME MIDlets cannot be copied into the emulator directories directly. Therefore, to install J2ME MIDlets, use the same steps outlined in 9.2.3 Using Emulator App Installer.

Appendix A - Remote Access Service PC Setup

Overview

Windows 2000 and NT come with internet access software which can be setup to use a PPP connection between the WINS emulator and the Remote Access Service (RAS) server.

The UIQ SDK documentation details how to configure the SDK emulator to use RAS. This section details the steps on setting up the RAS service on a Windows 2000 or NT PC.

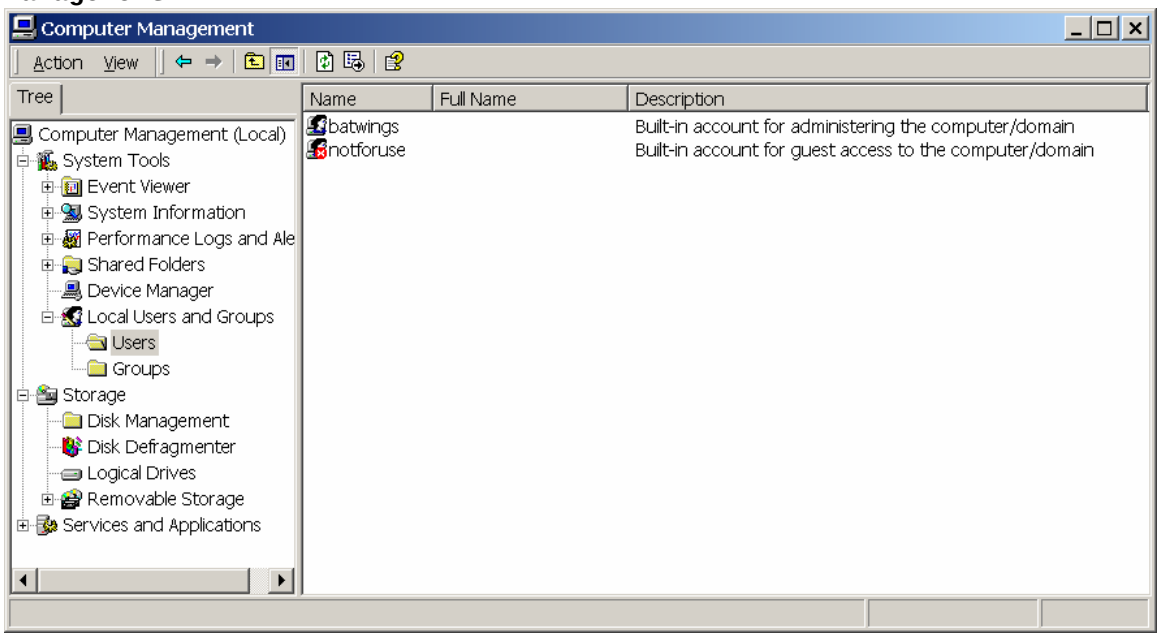
Restrictions

RAS will not work with Winsock support enabled.

Note that the Motorola A925 emulator uses the PC COM1 port. The following steps assume you are using the PC's COM2 port for the RAS server. If you are using a PC with only one COM port, you cannot use the same machine as both the emulator and the RAS server.

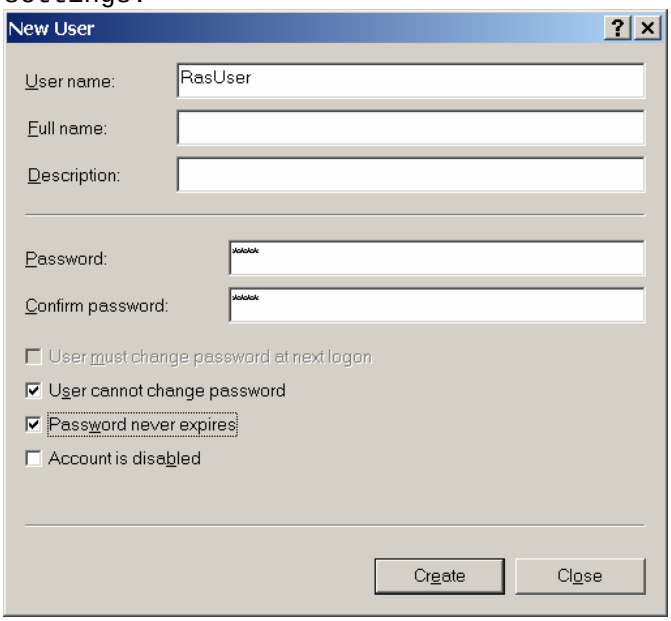
Windows 2000 Setup

1. From the **Start** menu, choose **Programs**, **Administrative Tools**, and select **Computer Management**.
   As an alternative, you can select **Settings, Control Panel**, and **Administrative Tools** from the **Start** menu and then select **Computer Management**.



2. In the left-hand panel of the **Computer Management** dialog box, expand "Local Users" and "Groups", and then select "Users". Right-click in the panel and select "Add New User" to create a new account. Create account with an account name of "RasUser" and a password of "pass". This corresponds to a preexisting internet account in the A920 emulator called "NT RAS".

3. Double-click on the “RasUser” account to check the settings. Under “General”, the boxes labeled “User cannot change password” and “Password never expires” boxes should be checked, and the box labeled “Account is disabled” should be unchecked. Under “Member Of”, the account should be a member of “Guests”. Click “OK” to save the settings.



4. From the **Start** menu, choose **Settings, Control Panel**, and select **Phone and Modem Options**. Click on the “Modems” tab. If “Communications cable between two computers” is installed on COM1, then remove that modem. If “Communications cable between two computers” is installed on a COM2, then skip to step 6.

5. Click “Add…” to open the modem wizard. In the next page, check the box labeled “Don’t detect my modem” and then click “Next”. In the following page, click on “Communications cable between two computers”, and then click “Next”. In the following page, select the serial port that you want to use for RAS connections. Now, click “Next” followed by “Finish” to close the wizard.

6. Right-click on “Communications cable between two computers”, and then select “Properties”. Select a COM2 as the serial port. Set the maximum port speed to 115200 baud, and then click “OK” twice to save the new settings.

7. From the Start menu, choose **Settings, Network and Dial-up Connections**, and select **Make New Connection** to start the Network Connection Wizard. In the first page of the wizard, click “Next”. In the second page, select “Accept incoming connections”, and then click “Next”. In the third page, select “Communications cable between two computers”, and then click “Next”. In the fourth page, De-select “Do not allow virtual private connections”, and then click “Next”. In the fifth page, make sure that the box labeled “guest” is checked, and then click “Next”.

8. In the next page, select “Internet Protocol (TCP/ IP)”. Double-click on “TCP/IP” and make sure that “specify TCP/IP address” checked. Then enter a range of IP addresses that will be used by the RAS server and the devices connecting the server. (e.g. 10.0.0.1 to 10.0.0.2 will assign 10.0.0.1 to the server and 10.0.0.2 will get assigned to one device when Ras is started, Use this server address in “hosts” file) Then, click “OK”, click “Next”, and then click “Finish” to exit the wizard.

Windows NT Setup

Dial-Up Networking

Ensure that Dial-Up Networking has been installed.

1. Double click **My Computer**
2. Double click the **Dial-Up Networking** icon
3. You will be prompted to insert the installation CD-ROM to copy the necessary files
4. When complete, a prompt will appear to add a RAS device (modem) if none are already installed
5. Tick the ‘Don’t detect my modem; I will select it from a list’
6. From the list, choose ‘Dial-up networking serial cable between 2 PCs’.
7. Choose port COM2. Then select ‘Properties’, 115200 baud.
8. Then click **Continue** on the Remote Access Setup box if no network protocols are installed choose **TCP/IP** and click **Continue** again.
9. Windows NT will then install TCP/IP Support
10. Finally you will be prompted to restart the machine

Adding RAS

If you are not prompted for this information, you can manually add the components.

1. Click on the **Network** icon in *Control Panel*
2. Click the Services tab
3. Highlight Remote Access Service
4. Click **Add...**
5. Add Remote Access Service
6. When complete a prompt will appear to add a RAS device (modem) if none are already installed. Click **Yes.**
7. If you are not prompted for a modem, go to '**Modems**' in *Control Panel*.
8. Tick the 'Don't detect my modem; I will select it from a list'
9. From the list, choose 'Dial-up networking serial cable between 2 PCs'. Choose port COM2. Then select 'Properties', 115200 baud.
10. If no network protocols are available Windows NT will prompt you to add one, select **TCP/IP** and click continue
11. Windows NT will then install TCP/IP Support
12. Finally you may be prompted to restart the machine

To manually add another modem, click **Add..** on the **Remote Access Setup** and then choose **Install Modem...** or use the **Modems** icon in the **Control Panel**

TCP/IP Properties

To set the TCP/IP Properties, follow the below instructions.

1. From the **Start** menu, select **Settings**
2. Open the **Control panel.**
3. Click on the **Network** icon
4. Choose the **Services** tab
5. Highlight '**Remote Access Service**'
6. Click **Properties…** The 'Remote access setup' dialog appears.
7. Highlight 'Dial-up Networking Serial Cable between 2 PCs'.
8. Choose '**Configure'.** The 'Port Usage' dialog appears. Select '**Dial out or receive calls**'.
9. Return to the 'Remote access setup' dialog. Click on the **Network** … button.
10. In the Network protocols:
11. Check **TCP/IP**
12. Uncheck IPX/SPX Compatible
13. Uncheck **NetBEUI**
14. Under '**Encryption Settings:**' choose 'Allow any authentication including clear text'
15. Then click **TCP/IP Configure...**
16. Check **Use static address pool**
17. Under Begin and End, enter the upper and lower limits of a range of IP addresses that are valid for your network and are not already in use.
18. Alternatively, select '**This computer only**', and use the range 10.1.0.1 to 10.1.0.254. This will suffice if you do not wish to access other machines apart from your own PC.
19. Return to the Network Services, multi-tabbed dialog. (Which was reached from the control panel by clicking 'Network'.)
20. If you wish to access other machines apart from your own PC, select the 'Protocols' tab, highlight 'TCP/IP' and click 'Properties…'. Select the 'Routing' tab and tick the 'Enable IP forwarding' option.
21. Quit all dialog boxes, and reboot.

RAS User Account

To create a RAS user account, the following steps are provided.

1. From the **Start** menu, select **Run…** and enter the program name '**MUSRMGR**'.

2. The 'User Manager' program appears

3. Look for the name 'RasUser' in the list of user names. (Exactly as shown, with no spaces, and with the same capitalization.) If it is present, double click on it.

4. If the name 'RasUser' is not already in the list of user names, choose '**New User**' from the 'User' menu. Enter 'RasUser' in the '**Username**' box, exactly as shown, with no spaces, and with the same capitalization.

5. Enter some suitable comment, such as 'EPOC32 Test RAS User' in the 'Full Name' and 'Description' boxes.

6. Enter '**pass**' in the '**Password**' and 'Confirm' boxes. *In lower case.*

7. Untick the '**User must change password at next logon**' option.

8. Tick the '**User cannot change password**' option.

9. Tick the '**Password never expires**' option.

10. Select the '**Dialin…**' button. Choose '**Grant dialin permission to user**'…

11. Return to the User Properties dialog and click the '**Groups…**' button. Add the group name 'Users' into the '**Member of**' list.

12. Quit the 'User Manager' program.

Starting RAS Service

To start the RAS service, at the command line prompt, enter the following:

```
NET START "REMOTE ACCESS CONNECTION MANAGER"
NET START "REMOTE ACCESS SERVER"
NET START "SIMPLE TCP/IP SERVICES"
NET START EVENTLOG
START RASMON
```

It would be best to save these commands in a suitable batch file.

RAS should now be running and should be listening to COM2. Enter the command 'NET START '; this should indicate that RAS is running.

When you connect a computer to your PC's COM2 port, the RASMON program should display a 'CD' light when you make a connection. You can choose options in the RASMON program which will display data-transmitted/received activity indicators.

Troubleshooting

- Check that RAS is running by typing NET START at the command line prompt.
- Run the RASMON program to monitor activity on the port.
- Check the following points in the Modems icon in the Control Panel.
  1. Click the **Connection** tab
  2. Click Connection
     - Data bits: 8
     - Parity: None
     - Stop bits: 1
  3. Click Advanced
     In case you use PCMCIA ADAPTER:
     - Check Use flow control
     - Check Hardware (RTS/CTS)
  4. Un-Check Use flow control

Appendix B - HALData Attributes

This information is provided to the developer for condition block usage in PKG files. The \epoc32\include\hal_data.h header file may need to be consulted to identify the appropriate enum or value.

Note however, several of these values set in the Motorola A925 software do not match actual A925 hardware. For example, ECPUABI = arm4 even though the native build target for A925 applications is THUMB. Also, EDeviceFamily = crystal even though the A925 is a Quartz device. Therefore, the below information may not be useful to the developer except as condition block parameters in PKG files.

```
EManufacturer                = Motorola
EManufacturerHardwareRev     = 0x3
EManufacturerSoftwareRev     = 0x001
EManufacturerSoftwareBuild   = 0x4b32f
EModel                       = 0x42525453
EMachineUid                  = ParagonEuropean
EDeviceFamily                = crystal
EDeviceFamilyRev             = 0x001
ECPU                         = arm
ECPUArch                     = 0x400
ECPUABI                      = arm4
ECPUSpeed                    = 132710
ESystemStartupReason         = cold
ESystemException             = 0
ESystemTickPeriod            = 15625
EMemoryRAM                   = 4
EMemoryRAMFree               = 4
EMemoryROM                   = 12
EMemoryPageSize              = 0x1000
EPowerGood                   = 1
EPowerBatteryStatus          = good
EPowerBackup                 = 1
EPowerBackupStatus           = good
EPowerExternal               = 1
EKeyboard                    = full
EKeyboardDeviceKeys          = 0
EKeyboardAppKeys             = 0
EKeyboardClick               = 1
EKeyboardClickState          = 1
EKeyboardClickVolume         = 0
EKeyboardClickVolumeMax      = 1
EPen                         = 1
EPenX                        = 208
EPenY                        = 320
EPenDisplayOn                = 0
EPenClick                    = 1
EPenClickState               = 1
EPenClickVolume              = 0
EPenClickVolumeMax           = 1
```

```
EMouse                          = 0
ECaseSwitch                     = 0
ELEDs                           = 0
EIntegratedPhone                = 0
ESystemDrive                    = 0x2
EDisplayXPixels                 = 0
EDisplayYPixels                 = 0
EDisplayXTwips                  = 0
EDisplayYTwips                  = 0
EDisplayColors                  = 65536
EDisplayState                   = 1
EDisplayContrast                = 0
EDisplayContrastMax             = 0
EBacklight                      = 0
EBacklightState                 = 0
EDisplayIsMono                  = 0
EDisplayIsPalettized            = 0
EDisplayBitsPerPixel            = 0
EDisplayNumModes                = 0
EDisplayMemoryAddress           = 0
EDisplayOffsetToFirstPixel      = 0
EDisplayOffsetBetweenLines      = 0
EDisplayPaletteEntry            = 0
EDisplayIsPixelOrderRGB         = 0
EDisplayIsPixelOrderLandscape = 0
EDisplayMode                    = 0
EDisplayBrightness              = 0
EDisplayBrightnessMax           = 0
EDebugPort                      = 0
ELocaleLoaded                   = 0
ELanguageIndex                  = 1
EEnableTouchScreen              = 0
EDisableTouchScreen             = 0
EDigitiserSwitchOn              = 0
EDigitiserSwitchOff             = 0
```