



TUTORIAL

wavecom®

wavecom  confidential ©

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement. Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.



TUTORIAL

Version: 001 / 1.0

Date: October 30th, 2001

Reference: WM_SW_OAT_UGD_001

1	Introduction	1
1.1	Purpose	1
1.2	References	1
1.3	Glossary	1
1.4	Abbreviations	1
2	Generating Applications With the Open AT Wizard	2
2.1	Creating an Application With the Wizard	2
2.1.1	<i>Step 1: Launching the Wizard</i>	
2.1.2	<i>Step 2: Project Type</i>	
2.1.3	<i>Step 3: Other Settings</i>	
2.2	Directory Architecture	4
2.3	Generating Remote Application Binary	5
2.4	Running the Remote Application	5
3	Generating an Application without the Open AT Wizard	6
3.1	Directory Architecture	6
3.2	Creating Embedded Applications	6
3.2.1	<i>Coding the Application</i>	
3.2.2	<i>Application binary generation</i>	
3.3	Download	7
3.4	Running the Embedded Application	10
3.4.1	<i>With HyperTerminal</i>	
3.4.2	<i>With the Terminal Emulator</i>	
3.4.3	<i>Target Monitoring Tool</i>	

List of figures

Figure 1: Project Type	2
Figure 2 : Directory Structure	4
Figure 3: Connection Type	7
Figure 4: Phone Call Destination	7
Figure 5: COM1Serial Port Features	7
Figure 6: Modem Connection Check	8
Figure 7: File Download with HyperTerminal	8
Figure 8: File Selection	9
Figure 9: Application Downloading process (continued)	9

WAVECOM, WISMO are trademarks or registered trademarks of Wavecom S.A. All other company and/or product names mentioned may be trademarks or registered trademarks of their respective owners.

1 Introduction

1.1 Purpose

This tutorial describes the process to develop an embedded application using the Open AT Wizard for debugging (“Debug Mode”), and MS-DOS generation tools for the “Target mode”.

1.2 References

- I. Development Guide
- II. Tools Manual
- III. AT Command Interface Guide

1.3 Glossary

- AT commands**Set of standard modem commands.
- AT function**Software that processes AT commands and AT subscriptions.
- Embedded application**User application sources to be compiled and run on a Wavecom product.
- Embedded Core software**Software that includes the Embedded application and the Wavecom library.
- Target**Open AT compatible product supporting an Embedded Application.
- Target Monitoring Tool**Set of utilities used to monitor a Wavecom product.
- Receive command**Process that intercepts AT responses.
- pre-parsing**
- Remote Application**Set of libraries enabling the User to run his application on a PC.
- Send command**Process that intercepts AT commands.
- pre-parsing**
- Wavecom library**Library delivered by Wavecom to interface Embedded application sources with Wavecom Core Software functions.
- Wavecom Core Software**Set of GSM and open functions supplied to the User.

1.4 Abbreviations

- API**Application Programming Interface
- CPU**Central Processing Unit
- IR**Infrared
- KB**Kilobyte
- OS**Operating System
- PDU**Protocol Data Unit
- RAM**Random-Access Memory
- ROM**Read-Only Memory
- RTK**Real-Time Kernel
- SMA**SMall Adapter
- SMS**Short Message Services
- SDK**Software Development Kit

2 Generating Applications With the Open AT Wizard

2.1 Creating an Application With the Wizard

This Wizard provides a user-friendly way to create an Open AT project. It is used to manage the following operations:

- Visual C++ , project creation,
- Compilation, link and library settings.

The different steps in creating an Open AT project are listed in the following paragraphs.

2.1.1 Step 1: Launching the Wizard

- ❶ From the main Visual C++®, screen, select “File” -> “New...”,
- ❷ In the “New” window, select the “Projects” tab,
- ❸ From the list box, select “Wavecom Open AT”,
- ❹ In the “Location” text field, choose the path corresponding to the location where the project is going to be created,
- ❺ In the “Project Name” text field, indicate a project name,

For this example, enter “Call Monitoring” as the project name.

NOTE: this name is added to the previously chosen path.

- ❻ End the initialization using “OK”.
- ❼ The Open AT Wizard “Welcome” first screen is displayed. Press the “Next >” button to continue.

2.1.2 Step 2: Project Type

The screen shown in Figure 1 is displayed.



Figure 1: Project Type

Different choices are proposed:

- New Project:** The Open AT Wizard will insert template files into the project (appli.c and appli.h) with the basic functions (see the minimum embedded application code in the Development Guide). An optional file (utils.c. with utils.h header), containing the string and memory manipulation functions, is available. This is the default option.
- Sample Project:** The Open AT Wizard will insert the Call Monitoring sample application of the Open AT CD-ROM.
- Existing Project:** no file will be added to the project. This option is useful when the user wants to create a new project using existing application code files. These files will then have to be manually added to the project.

For this example, select the “**Sample Project**” choice.

2.1.3 Step 3: Other Settings

The user has to define two parameters which are used in the Remote Application environment:

- Stack Size:** this parameter is not used in the Remote Application environment, but is written in appli.c. It defines the size of the target call stack memory (see Stack Size in the Development Guide).
- Update Wavecom delivery files:** this option enables the Open AT wizard to re-install headers and library files needed in the Remote environment. It creates a “**V100**” directory (if there is none) with updated files, at the same level as the project directory.

For this example, check the “**Update Wavecom delivery files**” box.

Use the “**Finish**” button to create this embedded application project.

A message box appears that summarizes the directory structure to be created.

Press “**OK**” to finish creating the project.

2.2 Directory Architecture

The typical directory tree structure of the application generated by the Open AT wizard is shown below :

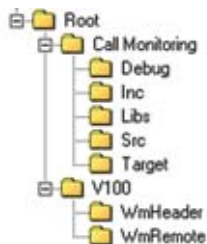


Figure 2 : Directory Structure

The “**Root**” directory is the directory entered in the “**Location**” text field in the Open AT wizard step ❶.

The “**Call Monitoring**” directory is specific to the new application (this was the text entered in the “**Project Name**” text field in the Open AT wizard step ❶).

It is made up of the following sub-directories :

- ❑ **Debug**: contains the Remote Application binary, which will be generated by a “**Build**” command,
- ❑ **Target**: contains the Embedded Core Software binary, ready to be downloaded into the Target (see §3 for target binary generation),
- ❑ **Src**: contains the User Open AT sources,

For this example, the “**src**” directory contains three files:

- appli.c
- utils.c
- at_com.c

- ❑ **Inc** : contains the User Open AT headers,

For this example, the “**inc**” directory contains three files:

- appli.h
- utils.h
- at_com.h

- ❑ **Libs**: contains the Windows libraries related to one part of the target code sources (for Remote Application Execution).

The “**V100**” directory contains the softwares supplied by Wavecom with this version of Open AT. This directory is only generated if the “**Update Wavecom delivery files**” checkbox has been selected in the wizard step ❷. It includes the following sub-directories:

- ❑ **WmHeader**: contains the Open AT API header files,
- ❑ **WmRemote**: contains the Open AT library file “**wmremoteat.lib**” for the Remote Task Environment.

2.3 Generating Remote Application Binary

See the **Development Guide** and the **AT Command Interface** to create the embedded application.

With this example, a call monitoring system has been implemented.
Refer to the source files and to the readme.txt file on the Open AT CD-ROM,
in the **"Samples\Call Monitoring"** directory.

The remote application binary is generated as described in the **Tools Manual**: use the **"Build"** -> **"Build xxx.exe"** command (or F7 key). If any compilation or link error occurs, correct them and restart the build operation.

2.4 Running the Remote Application

In the Target Monitoring Tool

Start the Target Monitoring Tool (from the Windows Start Menu).

Select **"Commands"** -> **"Auto Detect"** to detect the target speed.

Select **"Commands"** -> **"Init Target"** to initialize the target in debug mode.

Select **"Traces"** -> **"Open"** to open the trace window.

Select **"Commands"** -> **"Get Informations About Target"** to initialize the trace mode.

In the Terminal Emulator

From the Target Monitoring Tool toolbar, start the Terminal Emulator.

Type the **"AT"** command in the Terminal Emulator window.

The module should answer the **"OK"** string, in blue characters. If not, check your target connection or check if another program is using the serial port. Then, retry the operation.

In Visual C++®

Launch the application with the **"Build"** -> **"Start Debug"** -> **"Go"** command (or F5 key).

The Remote Application Controller window appears.

In the Remote Application Controller

Select the **"CUSTOMER trace levels"** you want to show in the Target Monitoring Tool.

For this example, the call monitoring application uses Levels 1 to 6.
You can check these levels in the Remote Application Controller window.

Select the **"Start"** button.

In the Target Monitoring Tool

Traces should appear in blue in the trace window, indicating that the remote application is running.

In the Terminal Emulator

The modem is ready to receive **AT commands**.

For this example, the **"NO SIM"** string will be displayed if no SIM is inserted;
otherwise the **"ENTER PIN CODE"** string is displayed.
You can now use the call monitoring application
(cf. the readme.txt file on the CD-ROM, in the **"Samples\Call Monitoring"** directory.

3 Generating an Application without the Open AT Wizard

3.1 Directory Architecture

In the directory of your choice, create a new directory using the name of the application. In this directory, create two directories named **"src"** and **"inc"**.

The **"src"** directory will contain all the user's source files.

The **"inc"** directory will contain all the user's header files.

The first call to the Wavecom **"c"** tool will create a **"target"** directory, at the same level as the **"src"** directory. This **"target"** directory will contain the files generated by Wavecom tools, including the embedded application binary.

For this example, copy the **"Call Monitoring"** directory from the CD-ROM directory **"Samples\Call Monitoring"**, and remove all **"read only"** files attribute.

3.2 Creating Embedded Applications

3.2.1 Coding the Application

See the **Development Guide** and the **AT Commands Interface** to create the embedded application. One of the source files (for example **"appli.c"**) must contain the minimum required code (See Minimum embedded application code in the Development Guide). Do not forget to copy all the user source files in the **"src"** directory, and the header files in the **"inc"** directory.

For this example, a full Call Monitoring system has been implemented. Refer to the source files, and to a readme.txt file from the CD-ROM directory **"Samples\Call Monitoring."**

3.2.2 Application binary generation

- In a MS-DOS command window, call the **"WmMake xxx"** command in the root directory, where **"xxx"** is the makefile name.

For this example, call the **"WmMake call_monitoring"** command from the **"Call Monitoring"** directory

- Check in the created **"target"** directory for the new **".obj"** compiled files. In the **"src"** directory, the **".lst"** files contain a compilation log. If compilation errors occur, correct them and restart the operation
- Check in the **"target"** directory for the new **"call_monitoring.dwl"** file. This file is the Embedded Application binary, ready to be downloaded to the target.

3.3 Download

In order to download the “.dwl” file to the target, follow these steps:

- 1 Launch HyperTerminal;
- 2 Configure it to open the COM port (see figures 3, 4 and 5);



Figure 3: Connection Type



Figure 4: Phone Call Destination



Figure 5: COM1Serial Port Features

③ Type the “AT” command in the HyperTerminal window. The response must be “OK” (see figure 6). The modem will echo characters on the serial link only if it is actually configured (In Figure 6, the “aaaaaa” string is typed and the modem echoes it: the modem is connected);

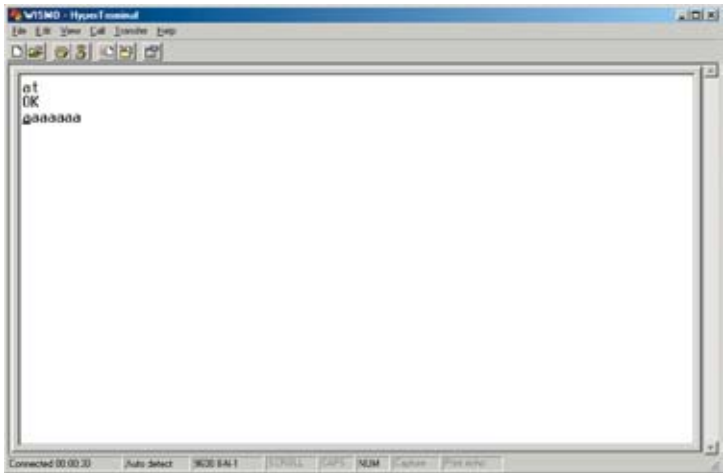


Figure 6: Modem Connection Check

④ Type the “at+wdwl” command to start the download mode. A “+WDWL: 0” response is given, and a set of characters should appear: the module has acknowledged the download mode (see Figure 7);

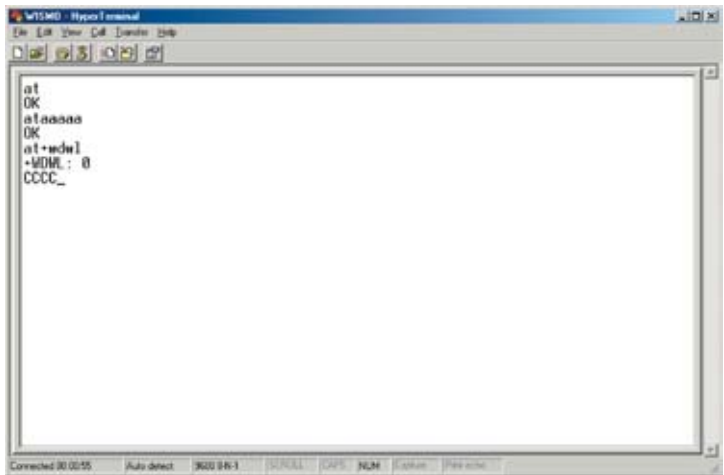


Figure 7: File Download with HyperTerminal

⑥ From the “Transfer” menu, select “Send a file...” (see Figure 8);



Figure 8: File Selection

⑦ Select the “target\xxx.dwl” file (in this sample : “call_monitoring.dwl”), and the “1K Xmodem” protocol. Press the “Send” button (see Figure 9);

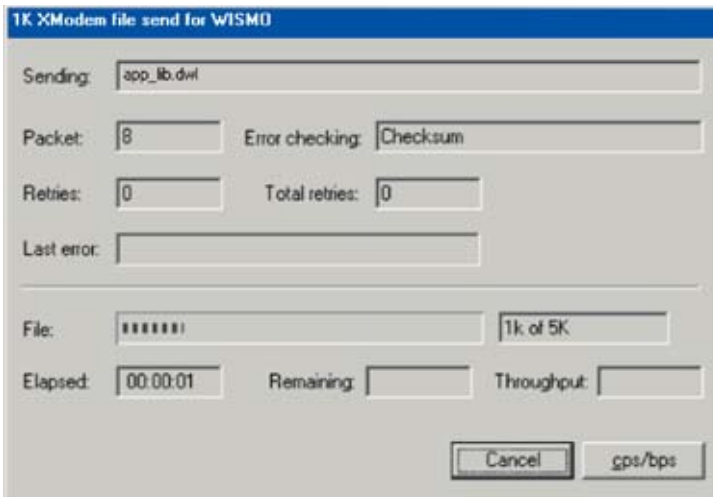


Figure 9: Application Downloading process (continued)

⑧ As soon as the download process is terminated (the download window disappears), reset the target with the “at+cfun=1” command.

3.4 Running the Embedded Application

3.4.1 With HyperTerminal

Type the **"at+wopen=1"** command, to start the embedded application (see **AT Commands Interface** for more information).

Your application is now running in nominal mode.

For this example, if no SIM card is inserted, the modem should display **"NO SIM"**, otherwise it will display **"ENTER PIN CODE"**. You can now use the call monitoring application (see the readme.txt file on the CD-ROM, in **"Samples\Call Monitoring"** directory).

3.4.2 With the Terminal Emulator

In the Target Monitoring Tool

Close the HyperTerminal program.

Start the Target Monitoring Tool (from the Windows Start Menu).

Select **"Commands"** -> **"Auto Detect"** to detect the target speed.

Select **"Commands"** -> **"Init Target"** to initialize the target in debug mode.

Select **"Traces"** -> **"Open"**, to open the trace window.

Select **"Commands"** -> **"Get Informations About Target"** to initialize the trace mode.

Select **"Traces"** -> **"Set Diagnose Levels"**. Select **"CUS"** in the **"Parameter"** list, check trace levels in the **"Bitmap"** zone, and use the **"Send Level"** button before closing the dialog box.

For this example, the call monitoring application uses Levels 1 to 6, so check these levels in the Set Diagnostic Levels window.

In the Terminal Emulator

From the Target Monitoring Tool toolbar, start the Terminal Emulator.

Type the **"at"** command in the Terminal Emulator AT window.

The module should answer the **"OK"** string in blue characters. If not, check your target connection, or if another program using the serial port is not running, then retry the operation.

Type **"at+wopen=1"** command in the Terminal Emulator AT window (see **AT Commands Interface** for more information).

In the Target Monitoring Tool

Select **"Commands"** -> **"Init Target"** to initialize the target in debug mode (the target should have reset after the **"at+wopen=1"** command).

Traces should appear in black in the trace window, indicating that the remote application is running.

In the Terminal Emulator

The modem is ready to receive AT commands.

For this example, the **"NO SIM"** string will be displayed if no SIM is inserted; otherwise the string **"ENTER PIN CODE"** is displayed.

You can now use the call monitoring application (cf. the readme.txt file on the CD-ROM, in the **"Samples\Call Monitoring"** directory).

wavecom[®]

WAVECOM S.A. - 12, boulevard Garibaldi - 92442 Issy-les-Moulineaux Cedex - France - Tel: +33 (0)1 46 29 08 00 - Fax: +33 (0)1 46 29 08 08
WAVECOM, Inc. - 4810 Eastgate Mall - Second floor - San Diego, CA 92121 - USA - Tel: +1 858 362 0101 - Fax: +1 858 558 5485
WAVECOM Asia Pacific Ltd - 5/F Shui On Centre - 6/8 Harbour Road - Hong Kong, PRC - Tel: +852 2824 0254 - Fax: +852 2824 0255

www.wavecom.com