



# Tools Manual for Open AT<sup>®</sup> v3.03

Revision: 001  
Date: November 2005



**wavecom**<sup>®</sup>  
*Make it wireless*


# **Tools Manual for Open AT<sup>®</sup> V3.03**

Revision: 001  
Date: 29<sup>th</sup> November 2005  
Reference: WM\_ASW\_OAT\_UGD\_00051

## Document History

Index	Date	Versions	
001	November 29 <sup>th</sup> 2005	Creation	

## Trademarks

®, WAVECOM®, WISMO®, Open AT®, and certain other trademarks and logos appearing on this document, are filed or registered trademarks of Wavecom S.A. in France or in other countries. All other company and/or product names mentioned may be filed or registered trademarks of their respective owners.

## Copyright

This manual is copyrighted by WAVECOM with all rights reserved. No part of this manual may be reproduced in any form without the prior written permission of WAVECOM.

No patent liability is assumed with respect to the use of the information contained herein.

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>8</b>
1.1	Purpose.....	8
1.2	References.....	8
1.3	Glossary.....	8
1.4	Abbreviations.....	9
1.5	Notation conventions.....	9
<b>2</b>	<b>OPEN AT® SDK ELEMENTS.....</b>	<b>10</b>
2.1	Open AT® SDK directory structure.....	10
2.2	Compatibility.....	10
2.3	SDK elements.....	11
2.3.1	Documentation.....	11
2.3.2	Samples.....	11
2.3.3	TgtGen.....	11
2.3.4	Wavecom Core Software.....	11
2.4	GCC compiler.....	11
2.5	Software Generation Toolkit (SGT).....	12
2.6	Add-on libraries.....	12
<b>3</b>	<b>OPEN AT® SETTINGS.....</b>	<b>13</b>
3.1	Tool description.....	13
3.2	Available settings.....	14
3.2.1	Open AT® libraries path.....	14
3.2.2	Open AT® Tools path.....	14
3.2.3	Add-on libraries path.....	14
3.2.4	SGT path.....	15
3.2.5	GCC compiler path.....	15
3.2.6	Synchronized Workspaces.....	15
3.2.7	Default Compiler.....	16
3.2.8	Default Memory Size.....	16
3.3	Applying changes.....	16
<b>4</b>	<b>OPEN AT® PROJECT WIZARD.....</b>	<b>17</b>
4.1	Tool description.....	17

29th November 2005

4.2	Basic settings.....	18
4.2.1	Project Name.....	18
4.2.2	Project Path.....	18
4.2.3	Used API.....	19
4.2.4	Project Type.....	19
4.2.5	Wmnew script additional options.....	19
4.2.6	Development Environment launch.....	20
4.3	Wizard modes.....	20
4.3.1	New Project.....	20
4.3.2	Sample Project.....	21
4.3.3	Existing Project.....	21
4.4	Updating existing projects.....	22
4.5	Error messages.....	22
<b>5</b>	<b>BUILDING OPEN AT® APPLICATIONS.....</b>	<b>24</b>
5.1	Open-AT® application directories architecture.....	24
5.2	Generated applications for target mode.....	25
5.2.1	Generated binary files.....	25
5.2.2	Open AT® application download.....	25
5.3	Microsoft Visual C++ 6.0 development environment.....	26
5.3.1	Creating an Open AT® application workspace for Visual C++ 6.0...26	
5.3.2	Visual C++ 6.0 Projects.....	28
5.3.3	Displaying HTML help file.....	29
5.3.4	Building application for Target Mode.....	30
5.3.5	Building application for remote mode.....	31
5.3.6	Setting breakpoints.....	32
5.4	Microsoft Visual C++ .NET development environment (2002 & 2003 versions).....	34
5.4.1	Creating an Open AT® application workspace for Visual C++ .NET 34	
5.4.2	Visual C++ .NET Projects.....	36
5.4.3	Displaying HTML help file.....	37
5.4.4	Building application for Target Mode.....	38
5.4.5	Building application for remote mode.....	39
5.4.6	Starting the Remote Task Environment.....	40
<b>6</b>	<b>DEVELOPMENT TOOLKIT.....</b>	<b>41</b>
6.1.1	Overview.....	41
6.1.2	Serial Link Manager.....	42
6.1.2.1	Description.....	42
6.1.2.2	Getting Started.....	42
6.1.3	Serial Link Manager.....	42
6.1.4	Target Monitoring Tool.....	43
6.1.4.1	Description.....	43
6.1.4.2	Getting Started.....	43
6.1.4.3	Get Traces Sent by the Wavecom Target.....	44
6.1.4.4	Select the Current Embedded Application Workspace.....	44
6.1.5	Terminal Emulator.....	45

29th November 2005

6.1.5.1	Description .....	45
6.1.5.2	Getting Started .....	45
6.1.6	Remote Application Execution .....	45
<b>7</b>	<b>REMOTE TASK ENVIRONMENT .....</b>	<b>46</b>
7.1	Basic Concepts.....	46
7.2	Project Creation .....	47
7.3	Remote Application Execution .....	47
7.3.1	Serial Port Configuration.....	47
7.3.2	Remote Application Launch .....	48
7.3.3	Traces Configuration .....	50
7.3.4	Data Flash Objects synchronization .....	51
7.3.5	Application & Data Storage Cells synchronization .....	51
<b>8</b>	<b>COMMAND-LINE SCRIPTS .....</b>	<b>52</b>
8.1	Create Open AT <sup>®</sup> projects : wmnew script.....	52
8.1.1	Script control .....	52
8.1.2	Script modes .....	53
8.1.3	Project interface options .....	53
8.1.4	Project type options.....	53
8.1.5	Project global options .....	53
8.1.6	Project source paths options .....	54
8.1.7	Project object paths & names options .....	54
8.1.8	Project library paths & names options .....	55
8.1.9	Provided libraries options .....	55
8.2	Building Open AT <sup>®</sup> projects : wmmake script .....	56
8.2.1	Script control .....	56
8.2.2	Clean options.....	56
8.2.3	Compiler options .....	56
8.2.4	Memory options .....	57
8.2.5	Debug Options .....	57
8.3	Check current Open AT <sup>®</sup> environment settings : wmcheck script .....	58
8.3.1	Script control .....	58
8.3.2	Configuration dump sample.....	58
<b>9</b>	<b>TCP/IP DEBUG TOOL .....</b>	<b>60</b>
9.1	Enable the traces.....	60
9.2	Write traces to disk .....	60
9.3	TCPDUMP format conversion .....	61
9.4	IP packets analysis .....	61

## LIST OF FIGURES

Figure 1 : Open-AT <sup>®</sup> SDK Directory Architecture.....	10
Figure 2 : Open-AT <sup>®</sup> Settings application.....	13
Figure 3 : Available SDK installed versions sample .....	14
Figure 4 : Available SGT installed versions sample .....	15
Figure 5 : Synchronized workspaces options.....	15
Figure 6 : Default compiler option .....	16
Figure 7 : Default memory size option .....	16
Figure 8 : Open AT <sup>®</sup> Project Wizard .....	17
Figure 9 : Project Name .....	18
Figure 10 : Project Path .....	18
Figure 11 : Project API.....	19
Figure 12 : Project type .....	19
Figure 13 : wmnnew additional options.....	19
Figure 14 : Available Development Environments.....	20
Figure 15 : Wizard modes .....	20
Figure 16 : Samples list.....	21
Figure 17 : Open-AT <sup>®</sup> Project Directory Architecture.....	24
Figure 18 : New Open AT <sup>®</sup> project from Visual C++ 6.0 .....	26
Figure 19 : Open AT <sup>®</sup> Project Wizard is going to be started .....	27
Figure 20 : Open AT <sup>®</sup> projects in Visual C++ 6.0 .....	28
Figure 21 : Preview the HTML help file from editor screen .....	29
Figure 22 : Application sample HTML help file display .....	29
Figure 23 : Select the active project – way 1 .....	30
Figure 24 : Select the active project – way 2.....	30
Figure 25 : Select the active project – way 3.....	31
Figure 26 : Select the configuration – way 1 .....	31
Figure 27 : Select the configuration – way 2.....	32
Figure 28 : Unable to set breakpoints .....	33
Figure 29 : Breakpoints window.....	33
Figure 30 : New Open AT <sup>®</sup> project from Visual C++ .NET .....	34
Figure 31 : Open AT <sup>®</sup> Project Wizard is going to be started .....	35

29th November 2005

Figure 32 : Open AT® projects in Visual C++ .NET .....	36
Figure 33 : Preview the HTML help file from solution explorer.....	37
Figure 34 : Application sample HTML help file display .....	38
Figure 35 : Select the WISMO Target configuration.....	38
Figure 36 : Select the configuration – way 1 .....	39
Figure 37 : Select the configuration – way 2 .....	39
Figure 38 : Provide RTE kernel file name.....	40
Figure 39: The Development ToolKit Environment .....	41
Figure 40 : Project Architecture .....	46
Figure 42 : RTE Monitor control steps (1 to 4).....	49
Figure 43 : Trace Level Selection .....	50
Figure 44 : Select Open AT® task.....	50
Figure 45 : A&D cell synchronization .....	51



# 1 Introduction

## 1.1 Purpose

This document is the user's guide for the Open AT<sup>®</sup> Software Development Kit.

## 1.2 References

- I. Open AT<sup>®</sup> Basic Development Guide (ref WM\_ASW\_OAT\_UGD\_00052 revision 001 for Open AT<sup>®</sup> v3.03)
- II. Open AT<sup>®</sup> Getting Started (ref WM\_ASW\_OAT\_CTI\_00005 revision 001 for Open AT<sup>®</sup> v3.03)
- III. AT Command Interface (ref WM\_ASW\_OAT\_UGD\_016 revision 006 for AT x51a version)

## 1.3 Glossary

<b>AT commands</b>	Set of standard modem commands.
<b>Embedded application</b>	User application sources to be compiled and run on a Wavecom GSM product.
<b>Embedded Core software</b>	Software that includes the Embedded application and the Wavecom library.
<b>Embedded software</b>	User application binary: set of Embedded application sources + Wavecom library.
<b>External application</b>	Application external to the Wavecom product that sends AT commands through the serial link.
<b>Target</b>	Open AT <sup>®</sup> compatible product supporting an Embedded Application.
<b>Target Monitoring Tool</b>	Set of utilities used to monitor a Wavecom product.
<b>Remote Application</b>	Set of libraries with which the User application can be run on a PC.
<b>Wavecom library</b>	Library delivered by Wavecom to interface Embedded application sources with Wavecom Core Software functions.
<b>Wavecom Core Software</b>	Set of GSM and open functions supplied to the User.

## 1.4 Abbreviations

API	Application Programming Interface
CPU	Central Processing Unit
FCM	Flow Control Manager
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

## 1.5 Notation conventions

In this document, the following notations will be used :

- Command line : `wmmake <filename>`.
- Environment or makefile variable : **WMATHOME**
- File name : `appli.c`
- Directory name : **AppliName**

## 2 Open AT<sup>®</sup> SDK elements

### 2.1 Open AT<sup>®</sup> SDK directory structure

After the Open-AT<sup>®</sup> SDK installation, the following structure is generated :

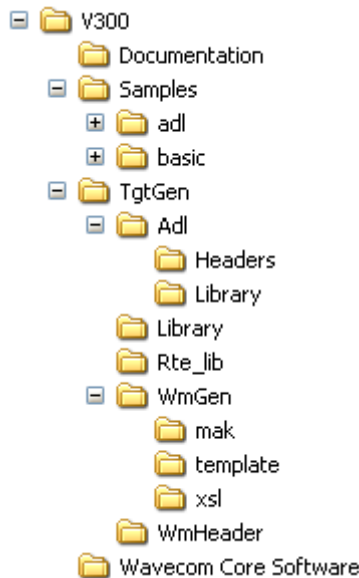


Figure 1 : Open-AT<sup>®</sup> SDK Directory Architecture

These files are installed in a directory referenced by the **WMATHOME** environment variable (Default value : **C:\OpenAT\V303**).

This setting may be changed using the Open AT<sup>®</sup> Settings application.

### 2.2 Compatibility

The current version of Open-AT<sup>®</sup> is compatible with

- the compiler suite "ARM suite ADS 1.2", and
- the arm-elf GCC cross-compiler.

See the ARM or GCC compiler installation procedure in the Getting Started manual (Reference II).

**IMPORTANT WARNING 1 : Wavecom does not guarantee proper operation if software is built with any other compiler/linker versions.**

**IMPORTANT WARNING 2 : The provided GCC compiler runs only on Windows NT/2000/XP versions (it is not supported on Windows 95/98/ME systems).**

## 2.3 SDK elements

### 2.3.1 Documentation

The “\Documentation” directory contains the whole documentation set of the current Open AT<sup>®</sup> installed version.

### 2.3.2 Samples

The “\Samples” directory contains the sources and compiled versions of the samples provided with the current Open AT<sup>®</sup> installed version.

### 2.3.3 TgtGen

The “\TgtGen” directory contains :

- the Open AT<sup>®</sup> libraries for all supported compilers,
- the header files to be included by applications source code
- the Remote Task Environment kernel for this version
- the Wavecom utilities needed to build Open AT<sup>®</sup> applications

### 2.3.4 Wavecom Core Software

The “\Wavecom Core Software” directory contains the AT firmware versions compatible with the current Open AT<sup>®</sup> version, for each supported product.

## 2.4 GCC compiler

The GCC compiler is installed by the Open AT<sup>®</sup> SDK setup in a directory referenced by the **WMGCCHOME** environment variable (default value : **C:\OpenAT\Tools\GCC**).

This setting may be changed using the Open AT<sup>®</sup> Settings application.

29th November 2005

## 2.5 Software Generation Toolkit (SGT)

SGT is the Wavecom compilation environment (based on the `make` program) used to build Target applications. It is installed by the Open AT® SDK setup in a directory referenced by the `SGT_DIR` environment variable (default value : **C:\OpenAT\Tools\SGT\v1.2.11**).

This setting may be changed using the Open AT® Settings application.

## 2.6 Add-on libraries

Add-on libraries (as TCPIP or Orange M2M Connect libraries) may be installed by the Open AT® SDK setup in a directory referenced by the `WMLIBHOME` environment variable (default value : **C:\OpenAT\OtherLibs**).

This setting may be changed using the Open AT® Settings application.

## 3 Open AT<sup>®</sup> Settings

### 3.1 Tool description

The Open AT<sup>®</sup> SDK setup installs an "Open AT<sup>®</sup> Settings" application on the system. The application may be started from the Start menu shortcut (Wavecom\Open AT\Open AT Settings).

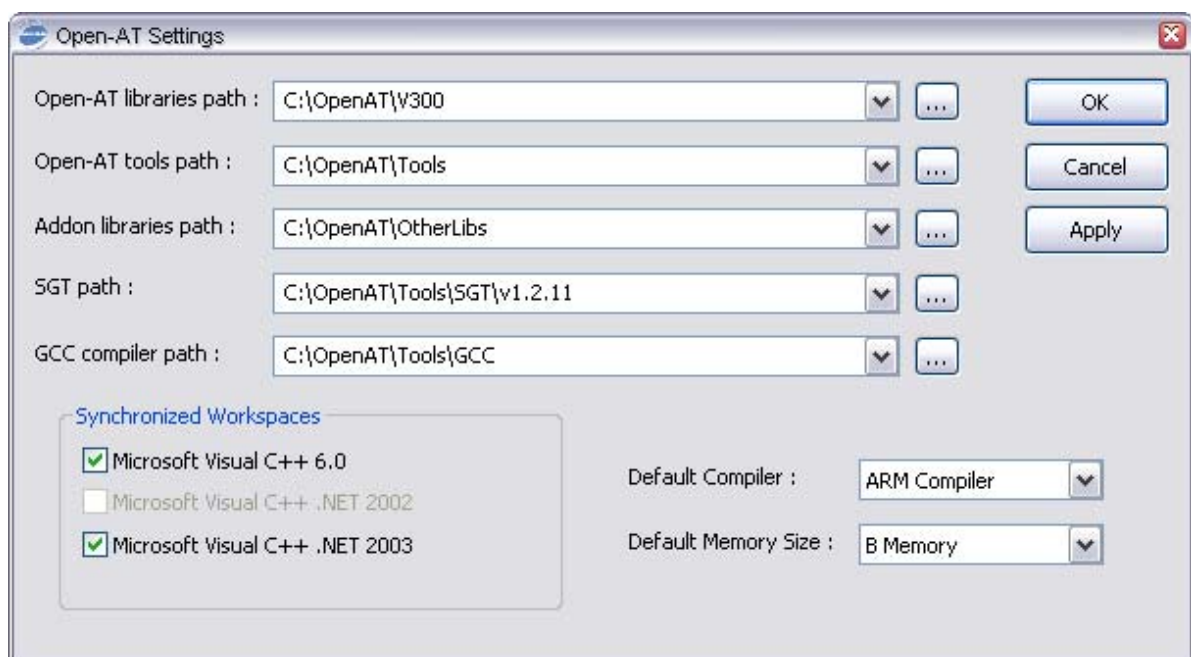


Figure 2 : Open-AT<sup>®</sup> Settings application

This application may be used to change the currently used version of Open AT<sup>®</sup> SDK, SGT, or other paths, but also to set the default compilation options.

## 3.2 Available settings


### 3.2.1 Open AT<sup>®</sup> libraries path

This path is the current used SDK location (where are installed Libraries, Header files and Generation tools).

The Open AT<sup>®</sup> settings application will automatically browse for all installed versions in a provided directory (for example, if **C:\OpenAT\V210b** and **C:\OpenAT\V303** versions are installed on the system, the application will automatically fill the choice list with all found versions).



Figure 3 : Available SDK installed versions sample

The browse button  should also be used to find the required SDK version.

### 3.2.2 Open AT<sup>®</sup> Tools path

This path is the root directory where Open AT<sup>®</sup> related tools (as GCC compiler and SGT) will be installed. Further install processes (with the Open AT<sup>®</sup> SDK setup program) will use this directory to install new tools versions.

The history  and browse  buttons should be used to find the required tools path.

### 3.2.3 Add-on libraries path

This path is the root directory where Add-on libraries (as TCP/IP or Orange M2M connect) are installed. Further install processes (with the Open AT<sup>®</sup> SDK setup program) will use this directory to install new add-on libraries versions.

The history  and browse  buttons should be used to find the required add-on libraries path.

29th November 2005

### 3.2.4 SGT path

This path is the current used SGT version location.

The Open AT<sup>®</sup> settings application will automatically browse for all installed versions in a provided directory (for example, if **C:\OpenAT\Tools\SGT\1.2.4\_c** and **C:\OpenAT\Tools\SGT\1.2.11** versions are installed on the system, the application will automatically fill the choice list with all found versions).

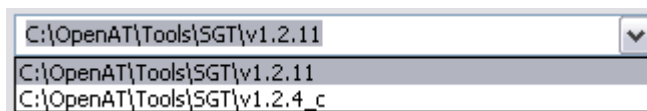



Figure 4 : Available SGT installed versions sample

The browse button  should also be used to find the required SGT version.

### 3.2.5 GCC compiler path

This path is the current used GCC compiler version location.

The history  and browse  buttons should be used to find the required GCC compiler path.

### 3.2.6 Synchronized Workspaces

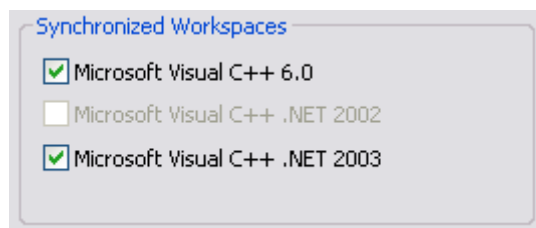


Figure 5 : Synchronized workspaces options

The Synchronized Workspaces checkbox set allows to enable the Open AT<sup>®</sup> workspaces generation for each of the supported environment. If an environment is not currently installed on the system, the check box is disabled (as Microsoft Visual C++ .NET 2002 in the sample figure above).

When the Project Wizard builds an Open AT<sup>®</sup> application workspace, it will build project files for each of the selected environments.



29th November 2005

Currently supported environments are :

- Microsoft Visual C++ 6.0 (<Project>.dsw file is generated) ;
- Microsoft Visual C++ .NET 2002 (<Project>\_2002.sln file is generated) ;
- Microsoft Visual C++ .NET 2003 (<Project>.sln file is generated).

### 3.2.7 Default Compiler



Figure 6 : Default compiler option

This option list allows to select the used compiler (ARM or GCC) when a compilation process is launched for the target application within a development environment.

**IMPORTANT WARNING :** The provided GCC compiler runs only on Windows NT/2000/XP versions (it is not supported on Windows 95/98/ME systems).

### 3.2.8 Default Memory Size

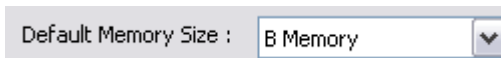


Figure 7 : Default memory size option

This option list allows to select the used memory size (A memory, B memory) when a compilation process is launched for the target application within a development environment.

## 3.3 Applying changes

When changes have been made in the Open AT<sup>®</sup> settings, once the Apply or OK button is pressed, all Development Environments and Cygwin shells have to be closed and restarted, to be updated with these changes.

## 4 Open AT<sup>®</sup> Project Wizard

### 4.1 Tool description

The Open AT<sup>®</sup> SDK setup installs an "Open AT<sup>®</sup> Project Wizard" application on the system. The application may be started from the Start menu shortcut (Wavecom\Open AT\Open AT Project Wizard).

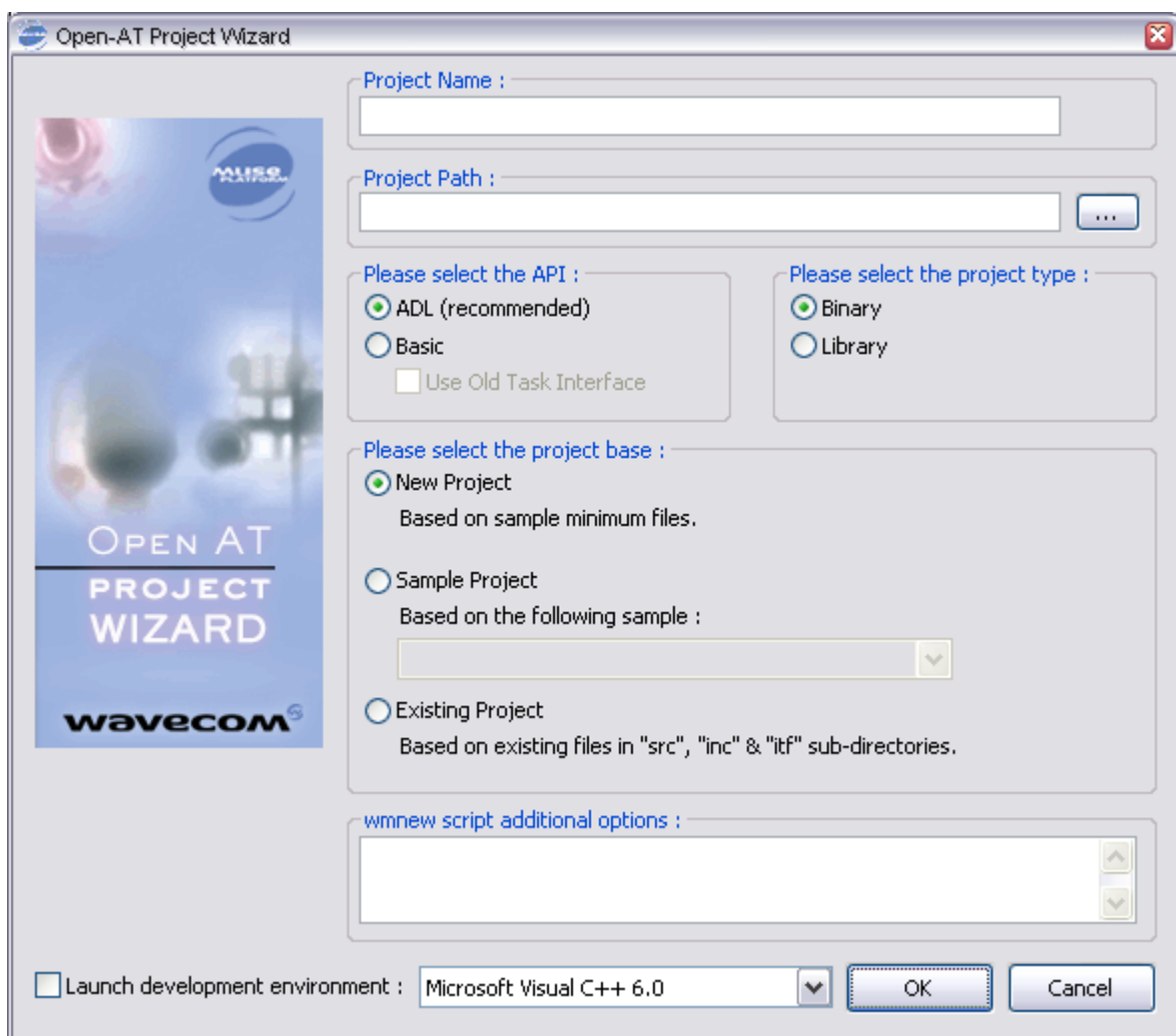


Figure 8 : Open AT<sup>®</sup> Project Wizard

This application has to be used to create, or edit an Open AT<sup>®</sup> project software common settings (saved in the corresponding <Project>.scs file), and also the

29th November 2005

selected synchronized Workspaces in the Open AT<sup>®</sup> Settings application (cf. §3.2.6).

## 4.2 Basic settings

### 4.2.1 Project Name

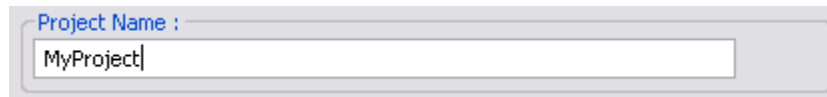


Figure 9 : Project Name

This setting will be the name of the Open AT<sup>®</sup> project (software common settings will be saved in the <ProjectName>.scs file).

When using the Sample base mode, the name will be set to the selected sample name.

When browsing for an existing project directory, if a .scs file is found in the selected directory, this file name will be used as the project name.

### 4.2.2 Project Path

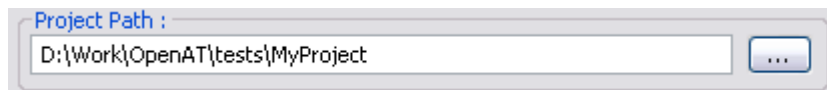


Figure 10 : Project Path

This setting is the path where the Open AT<sup>®</sup> project is located.

The browse button may be used to find the required project Path. If the path selected by the browse button contains a .scs file, the Wizard goes to the Existing Project mode, and reads the project name and the project options from this selected directory.

### 4.2.3 Used API

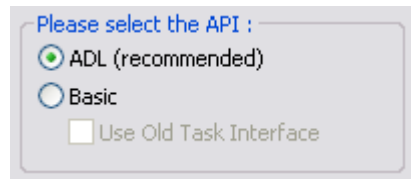


Figure 11 : Project API

This setting is the API used by the current Open AT<sup>®</sup> project. ADL API is strongly recommended (Basic API should be used by advanced users only).

*The "Use Old Task Interface" option applies only to Basic interface binaries : when using this option, the Open AT<sup>®</sup> application has only to implement the old `wm_apmAppliInit` & `wm_apmAppliParser` functions instead of a task table (cf. Open AT<sup>®</sup> Basic Development Guide for more details).*

### 4.2.4 Project Type

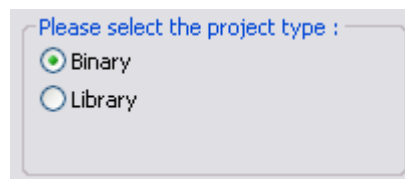


Figure 12 : Project type

This setting is the current project type (binary or library) which will be generated within the Development Environment.

### 4.2.5 Wmnew script additional options

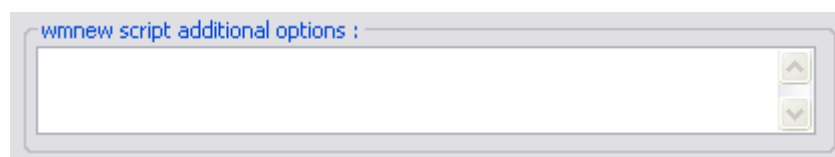


Figure 13 : wmnew additional options

This text field may contains additional options for the `wmnew` script ; please refer the Command Line Scripts chapter for more details. Only described options starting from §8.1.5 chapter (Project global options) may be used ; the other ones are all handled by the Open AT<sup>®</sup> Project Wizard.

This option is available only in New Project & Existing Project modes.

#### 4.2.6 Development Environment launch

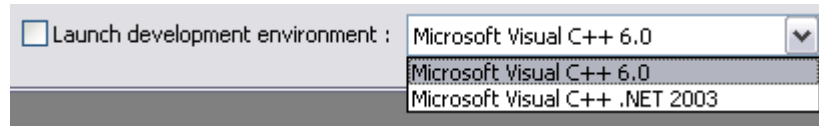


Figure 14 : Available Development Environments

If the “Launch development environment” option is checked, the currently selected environment will be launched with the current project. The drop-down list contains only the installed environments on the system.

#### 4.3 Wizard modes

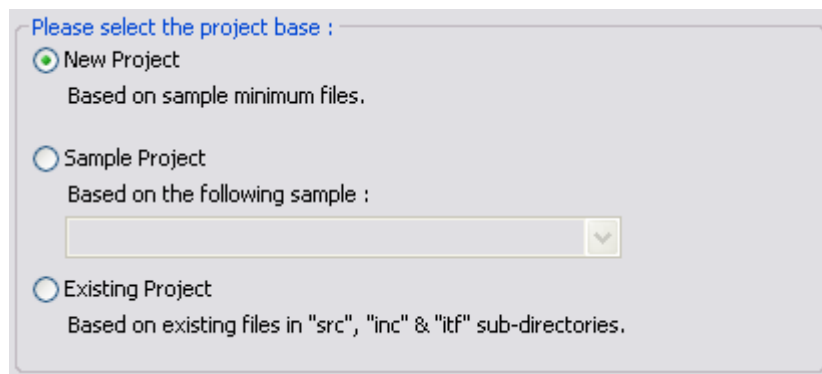


Figure 15 : Wizard modes

The wizard behavior is determined by the selected mode in this option field.

##### 4.3.1 New Project

In this mode, minimum code sample files will be copied in the current directory, before building the project settings.

*The `wmnew` additional option field is available in this mode, to set-up other project settings options.*

### 4.3.2 Sample Project

This mode creates a new project, based on the selected sample in the drop-down list. This list content depends on the selected API and project type.

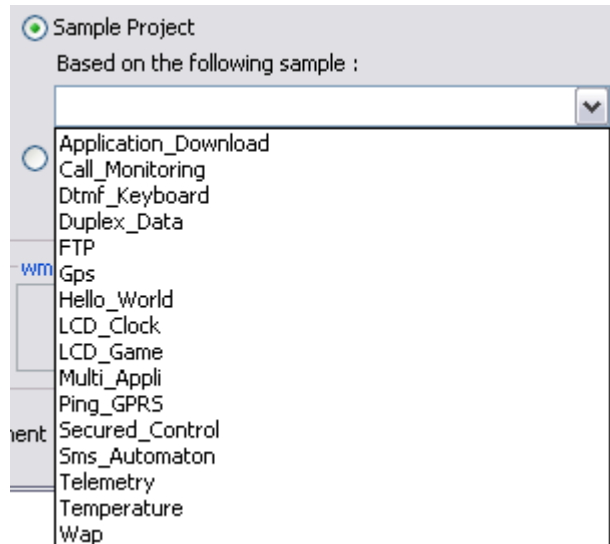


Figure 16 : Samples list

When a new sample is selected in the list, the project name is updated to be renamed with the sample one.

*As the `wmnew` script, if the selected sample depends on one or several sample libraries, this/these library/libraries project(s) will also be copied in the Libraries subdirectory.*

### 4.3.3 Existing Project

In this mode, the Project Wizard is used to update an existing project settings. To edit an existing project, the path browse button has to be used to find the project path. Then, the project name and the `wmnew` option field will be updated with the values read from this directory.

*The `wmnew` additional option field is available in this mode, to set-up other project settings options.*

29th November 2005

## 4.4 Updating existing projects

**Important Warning :** It is strongly recommended always to use the Project Wizard to update existing projects (for adding source files, include paths or others...), rather than hand-editing make files, or modifying projects in development environments.

An existing project may be opened with the wizard by the following ways :

- Browse for the existing project path with the Project Wizard application ;
- Open directly the project from the Windows Explorer (".scs" files are automatically associated with Open AT<sup>®</sup> Project Wizard).

## 4.5 Error messages

The Open AT<sup>®</sup> Project Wizard may display the following error messages.

**"Please enter a name to create the new project..."**

*The project name text field is empty.*

**"Not authorized characters (space, ".", "\$") found in the project name..."**

*The project name field can contain all characters, except spaces (" "), dollars ("\$") and dots (".").*

**"Not authorized characters (space, "\$") found in the project path..."**

*The project path field can contain all characters, except spaces (" ") and dollars ("\$").*

**"Please select a sample to create the new project..."**

*The Project Wizard is in Sample mode, but no sample was selected in the drop-down list.*

**"Please select a path to create the new project..."**

*The project path text field is empty.*

**"Bad format for provided path string..."**

*The project path text field does not contains an absolute path (in Windows format).*

**"Unable to create the folder."**

*A new directory was required to be created, but the creation failed.*

**"Wmnew script error 3 (bad environment settings)"**

*The Open AT<sup>®</sup> environment is not correctly configured in Cygwin. Please refer to the `wmcheck` script documentation for more details.*

**"Wmnew script error 6 ([wmsync] xsltproc package is not installed on the system)"**

29th November 2005

*The xsltproc package, needed by the Open AT<sup>®</sup> Project Wizard, is not installed within Cygwin. Please check the Cygwin installation, or re-install it to solve the problem.*

**"Unable to run the wmnew script ; Open AT<sup>®</sup> or Cygwin may not be correctly installed."**

Cygwin environment should be uninstalled, and re-installed again. If the issue is then not solved, please refer to the wmcheck script documentation for more information.



## 5 Building Open AT® applications

### 5.1 Open-AT® application directories architecture

The typical directory tree structure of an application is shown below.

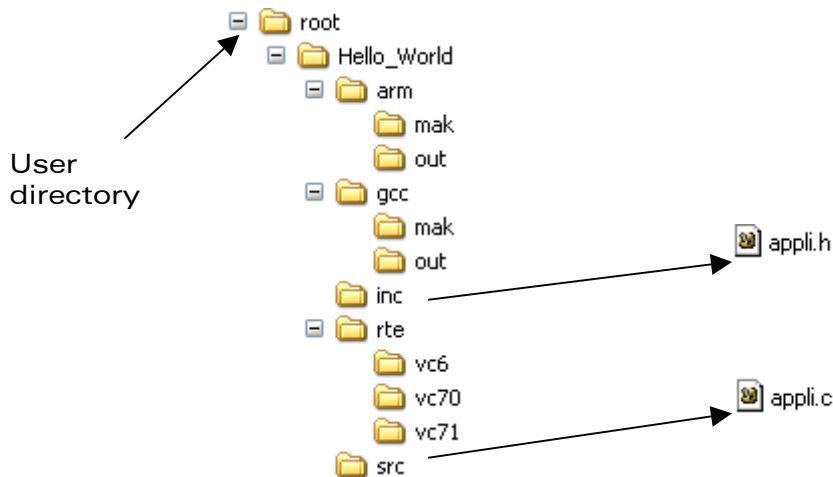


Figure 17 : Open-AT® Project Directory Architecture

The **AppliName** directory is specific to each new application ; it contains the project software common settings (.scs) file, and also the different supported development environments project files. It also contains the following sub-directories :

- ❑ **rte** : contains the Remote Application binaries, generated with one of the supported development environments,
- ❑ **arm/out** : contains the Embedded Core Software ready to be downloaded to the Target, generated with the ARM compiler,
- ❑ **gcc/out** : contains the Embedded Core Software ready to be downloaded to the Target, generated with the GCC compiler,
- ❑ **src**: contains the User Open-AT® sources,
- ❑ **inc**: contains the User Open-AT® header files.

## 5.2 Generated applications for target mode

Open AT<sup>®</sup> projects (applications or libraries) may be generated using a development environment (cf. following chapters), or using the wmmake script from a Cygwin bash shell command line (for advanced users, cf. §8.2).

### 5.2.1 Generated binary files

The generated binary files are :

- for a library :

[compiler]/out/[project].lib

where:

[compiler] is the current used compiler (arm or gcc),

[project] is the project name.

- for an application :

[compiler]/out/[compiler]\_[project]\_[memory].dwl &

[compiler]/out/[compiler]\_[project]\_[memory].wpb.dwl

where:

[compiler] is the current used compiler (arm or gcc),

[project] is the project name,

[memory] is the current used memory size (16, 32 or 32W).

The “.wpb.dwl” is a compressed version of the “.dwl” one (wpb is the acronym for Wavecom Packed Binary).

Both files may be downloaded on the product ; as the “.wpb.dwl” file is smaller than the simple “.dwl” one, downloading this file will be faster.

### 5.2.2 Open AT<sup>®</sup> application download

Please refer to the Tutorial documentation to know how to download Open AT<sup>®</sup> application binaries on WISMO modules.

29th November 2005

## 5.3 Microsoft Visual C++ 6.0 development environment

### 5.3.1 Creating an Open AT<sup>®</sup> application workspace for Visual C++ 6.0

Open AT<sup>®</sup> application projects are created with the Open AT<sup>®</sup> Project Wizard (see corresponding chapter). This wizard may be launched from the Start Menu, or from the Visual C++ "New Project" menu.

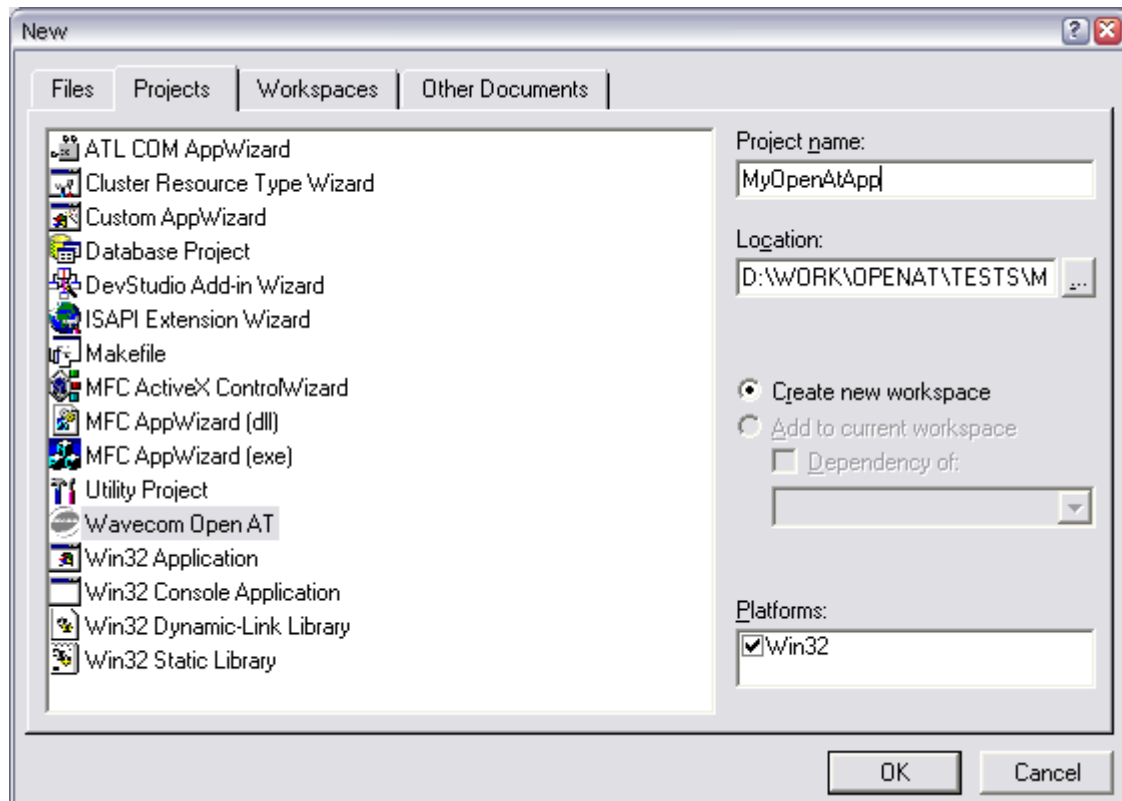
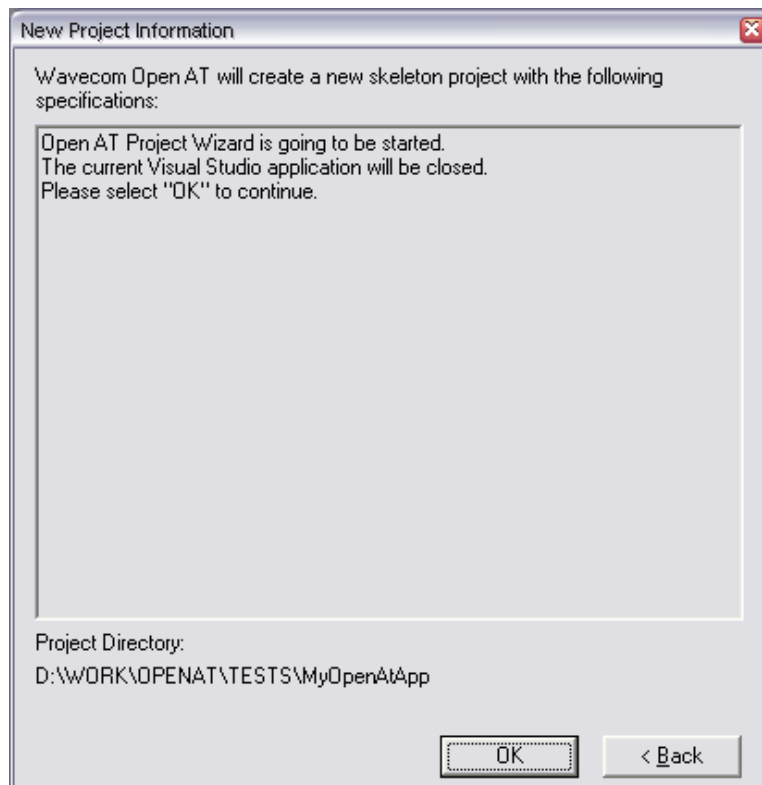


Figure 18 : New Open AT<sup>®</sup> project from Visual C++ 6.0

After have hit the OK button, the following box is displayed.

29th November 2005

Figure 19 : Open AT<sup>®</sup> Project Wizard is going to be started

The Open AT<sup>®</sup> Project Wizard is going to be started. When the OK button will be clicked, the current running Visual C++ session will be closed, and the Wizard will be launched (See the Open AT<sup>®</sup> Project Wizard description for more information).

29th November 2005

### 5.3.2 Visual C++ 6.0 Projects

A Visual C++ 6.0 Open AT<sup>®</sup> workspace contains two projects, for target mode (named <project>) and remote mode (named <project>\_rte).

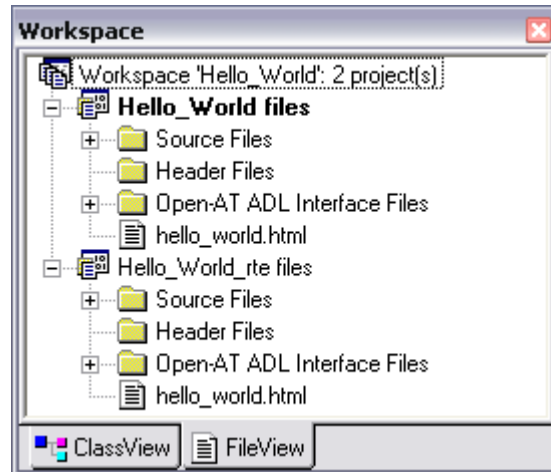


Figure 20 : Open AT<sup>®</sup> projects in Visual C++ 6.0

Each project contains the same files (application sources & headers, the Open AT<sup>®</sup> interface files, and the HTML help file, if the project was created from a provided sample), which may be edited from any of these two projects.

If an Open AT<sup>®</sup> project implements libraries, all the corresponding projects (target and remote) will also be added to the main application workspace.

**As written in the §4.4, it is strongly recommended always to use the Project Wizard to update existing projects (for adding source files, include paths or others...), rather than modifying projects in the Visual C++ environment.**

29th November 2005

### 5.3.3 Displaying HTML help file

When an Open AT<sup>®</sup> project is created from one of the provided samples, the projects include an HTML help file, describing the sample interfaces and processing.

In order to display this file from Visual C++, it has to be opened in the editor, and launched in the HTML browser by selecting the "Preview Page" option in the mouse right-click menu.

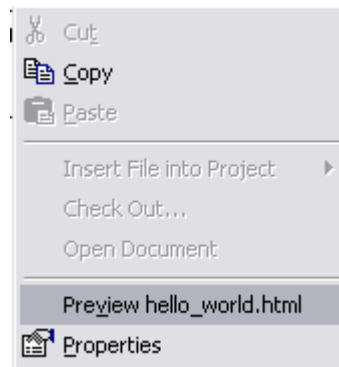


Figure 21 : Preview the HTML help file from editor screen

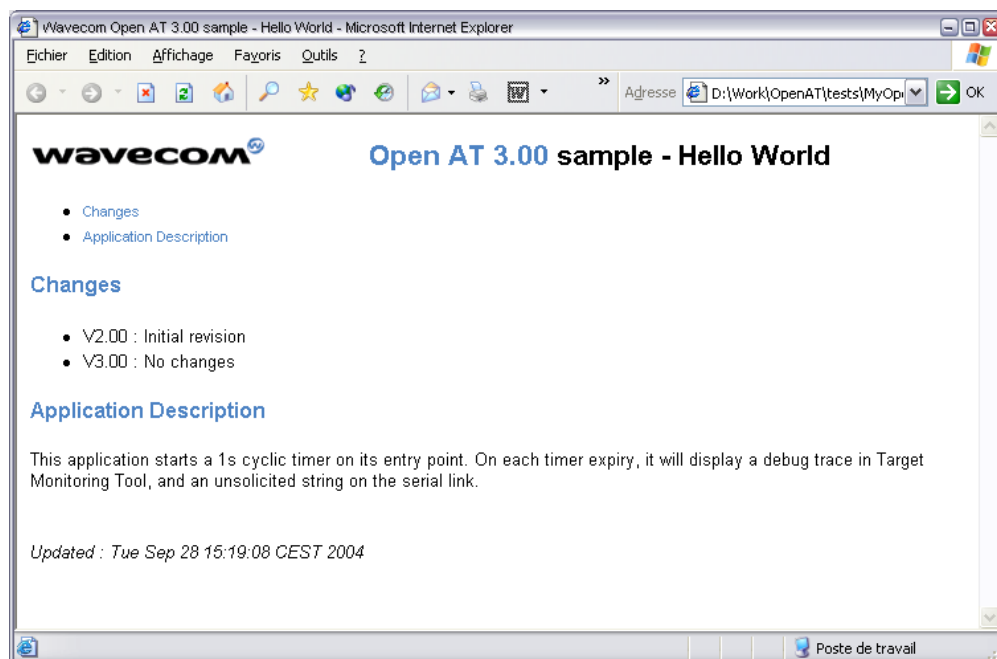


Figure 22 : Application sample HTML help file display

29th November 2005

### 5.3.4 Building application for Target Mode

To build the application for target mode, the target project has to be selected as the active project, from the Project menu :

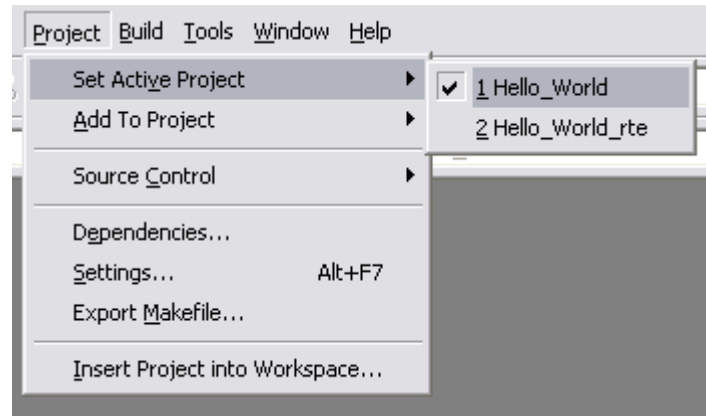


Figure 23 : Select the active project – way 1

Or from the right mouse button click contextual menu :

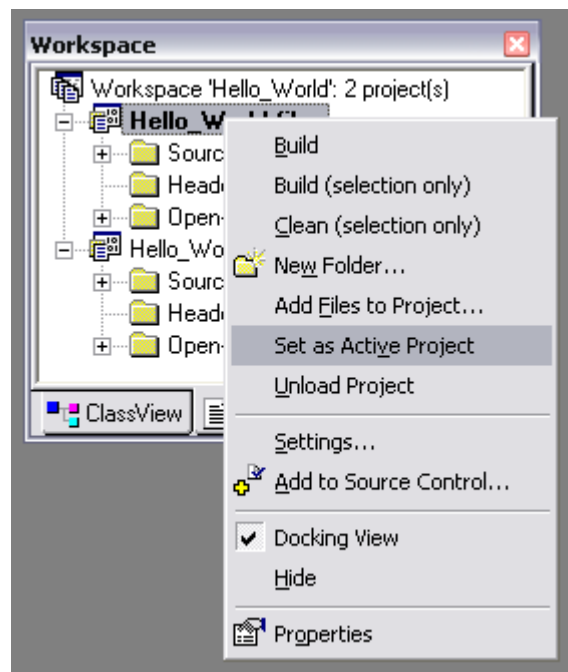


Figure 24 : Select the active project – way 2

29th November 2005

Or from the Build toolbar :

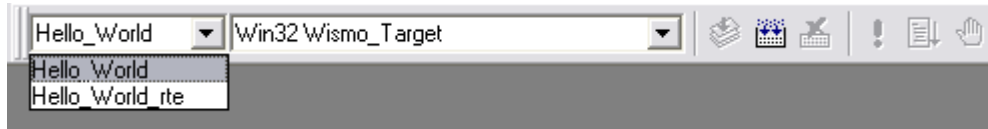


Figure 25 : Select the active project – way 3

Then, the project may be built by launching the Build command, from the Build Menu, the contextual menu, or with the F7 key. The compilation results will be displayed in the output window.

Warning: detected error and warning messages should not be relevant, since the Visual Environment displays these messages when it detects “error” or “warning” strings in the compilation text output (i.e. for example the “no error” string will be detected as an error by Visual Environment).

Error messages which may be displayed by the Software Generation Toolkit are listed in the `wmmake` script description.

### 5.3.5 Building application for remote mode

To build the application for remote mode, the remote project has to be selected as the active project, by the Project menu, the contextual menu or the Build toolbar (as shown above in the previous paragraph).

Then the configuration has to be set. Both Release and Debug mode are available, but the Debug mode should be chosen in order to perform step-by-step debugging in the program. The configuration may be selected from the Build->Set Active Configuration... menu :

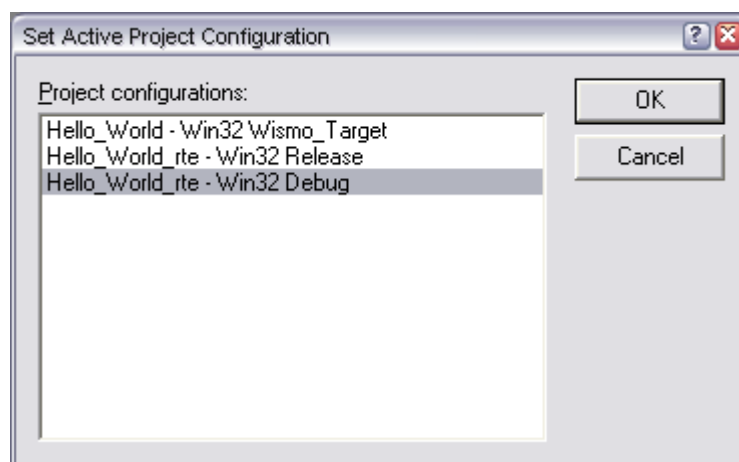


Figure 26 : Select the configuration – way 1



29th November 2005

Or from the Build toolbar :

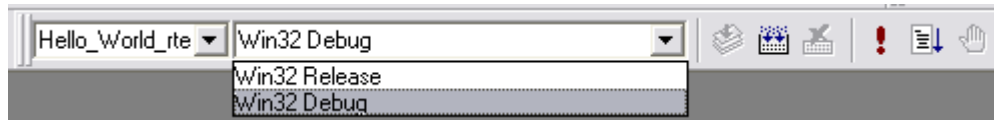



Figure 27 : Select the configuration – way 2

Then, the project may be built by launching the Build command, from the Build Menu, the contextual menu, or with the F7 key. The compilation results will be displayed in the output window.

### 5.3.6 Setting breakpoints

Due to the Remote Task Environment architecture, breakpoints should be set once the RTE graphical interface is launched, just before pressing the Start button (see the Remote Task Environment chapter).

If breakpoints are set before running the application, the Environment will inform the user that the breakpoints cannot be set (see figure 24). The execution will then stop at the beginning of the program : just press Go  to continue running the application.

Once the application is started, the user has to return to Visual environment, and to edit the breakpoints window (from the Edit menu, or with the Ctrl-B key). In this window, all breakpoints are displayed, and may be enabled / disabled as required (see figure 29).

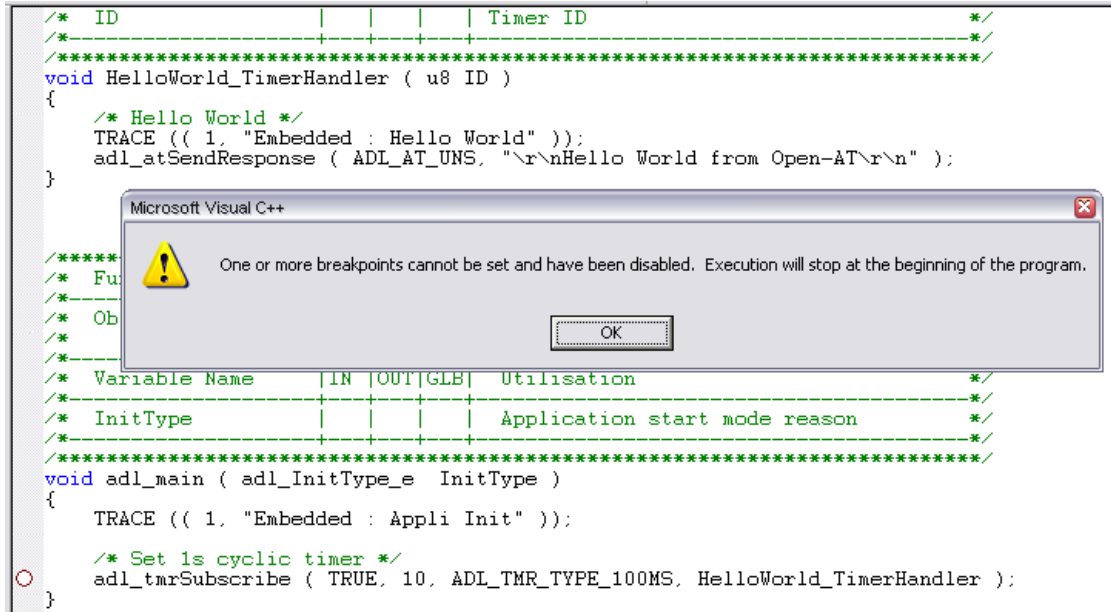


Figure 28 : Unable to set breakpoints

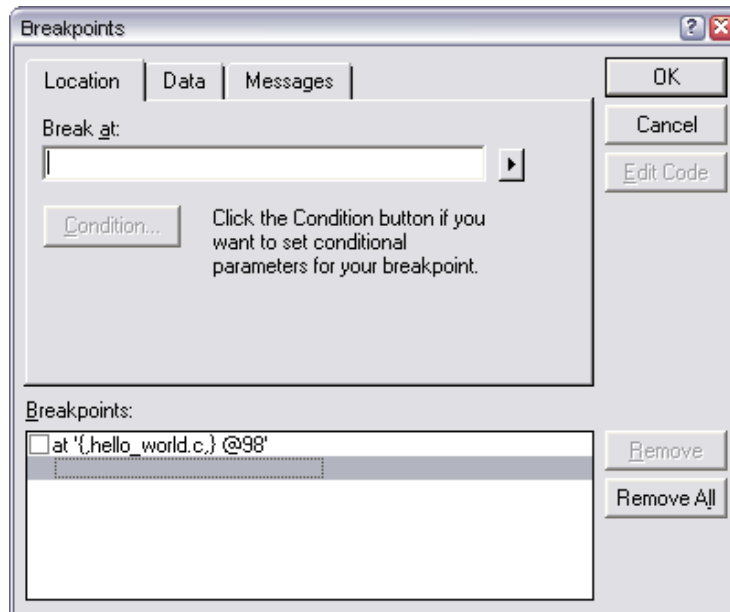


Figure 29 : Breakpoints window

29th November 2005

## 5.4 Microsoft Visual C++ .NET development environment (2002 & 2003 versions)

### 5.4.1 Creating an Open AT<sup>®</sup> application workspace for Visual C++ .NET

Open AT<sup>®</sup> application projects are created with the Open AT<sup>®</sup> Project Wizard (see corresponding chapter). This wizard may be launched from the Start Menu, or from the Visual C++ "New Project" menu.

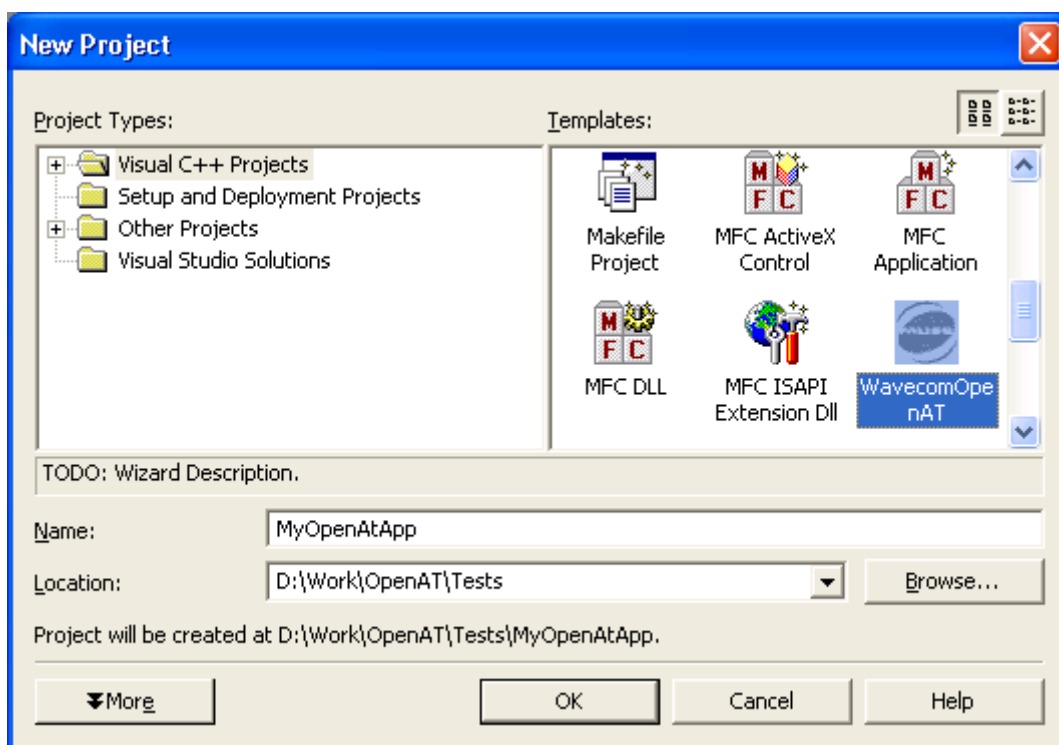


Figure 30 : New Open AT<sup>®</sup> project from Visual C++ .NET

29th November 2005

After have hit the OK button, the following box is displayed.

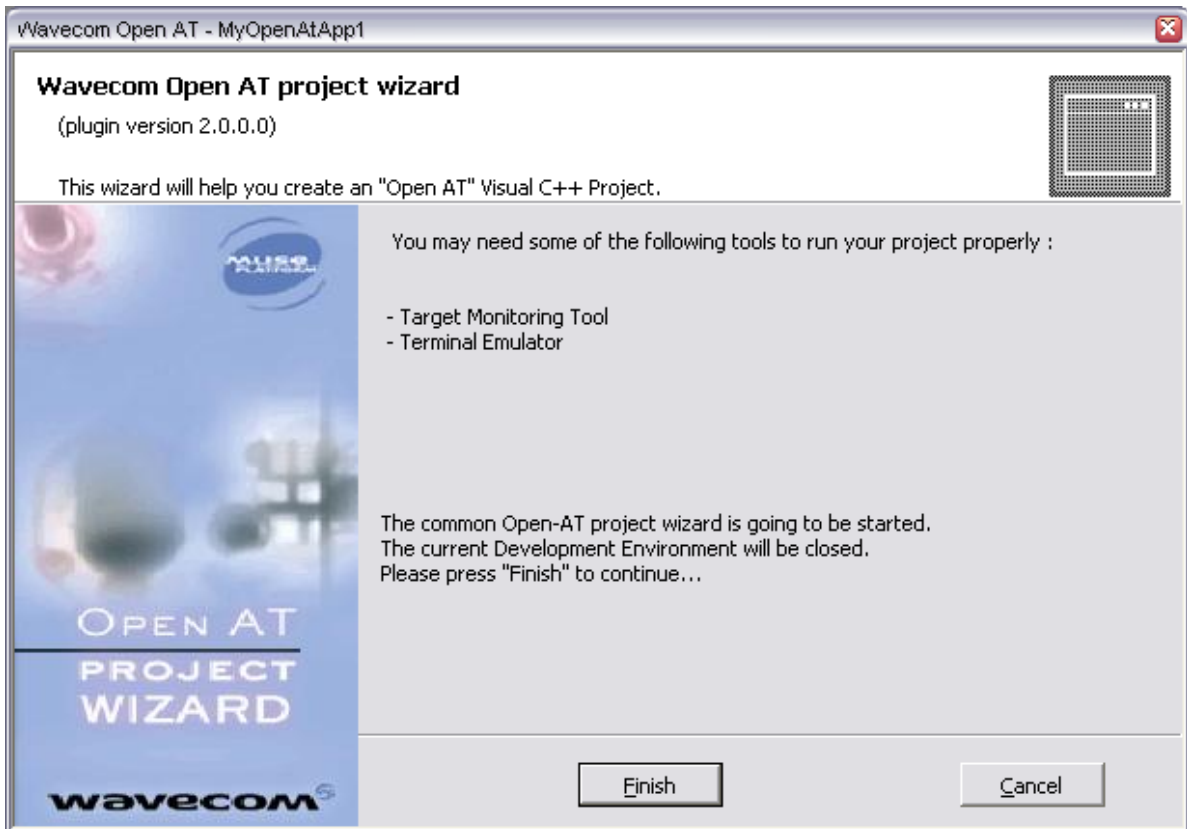


Figure 31 : Open AT<sup>®</sup> Project Wizard is going to be started

The Open AT<sup>®</sup> Project Wizard is going to be started. When the Finish button will be clicked, the current running Visual C++ session will be closed, and the Wizard will be launched (See the Open AT<sup>®</sup> Project Wizard description for more information).

29th November 2005

### 5.4.2 Visual C++ .NET Projects

A Visual C++ .NET Open AT<sup>®</sup> workspace contains two projects, for target mode (named <project>) and remote mode (named <project>\_rte).

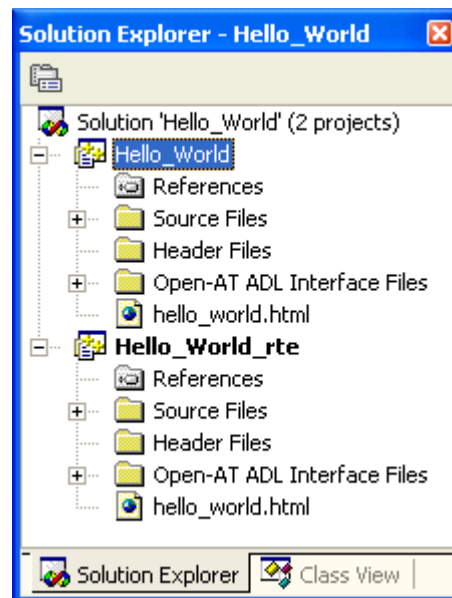


Figure 32 : Open AT<sup>®</sup> projects in Visual C++ .NET

Each project contains the same files (application sources & headers, the Open AT<sup>®</sup> interface files, and the HTML help file, if the project was created from a provided sample), which may be edited from any of these two projects.

If an Open AT<sup>®</sup> project implements libraries, all the corresponding projects (target and remote) will also be added to the main application workspace.

**As written in the §4.4, it is strongly recommended always to use the Project Wizard to update existing projects (for adding source files, include paths or others...), rather than modifying projects in the Visual C++ environment.**

29th November 2005

### 5.4.3 Displaying HTML help file

When an Open AT<sup>®</sup> project is created from one of the provided samples, the projects include an HTML help file, describing the sample interfaces and processing.

In order to display this file in Visual C++ Web Browser, it has to be launched by selecting the "Display in Browser" option in the mouse right-click menu.

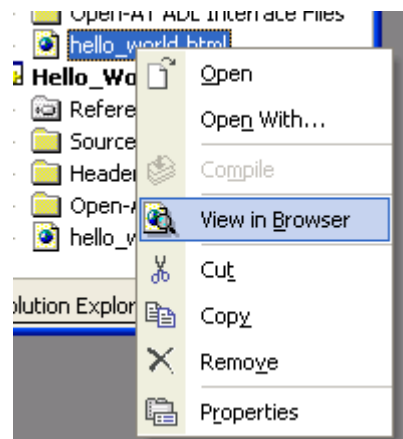


Figure 33 : Preview the HTML help file from solution explorer

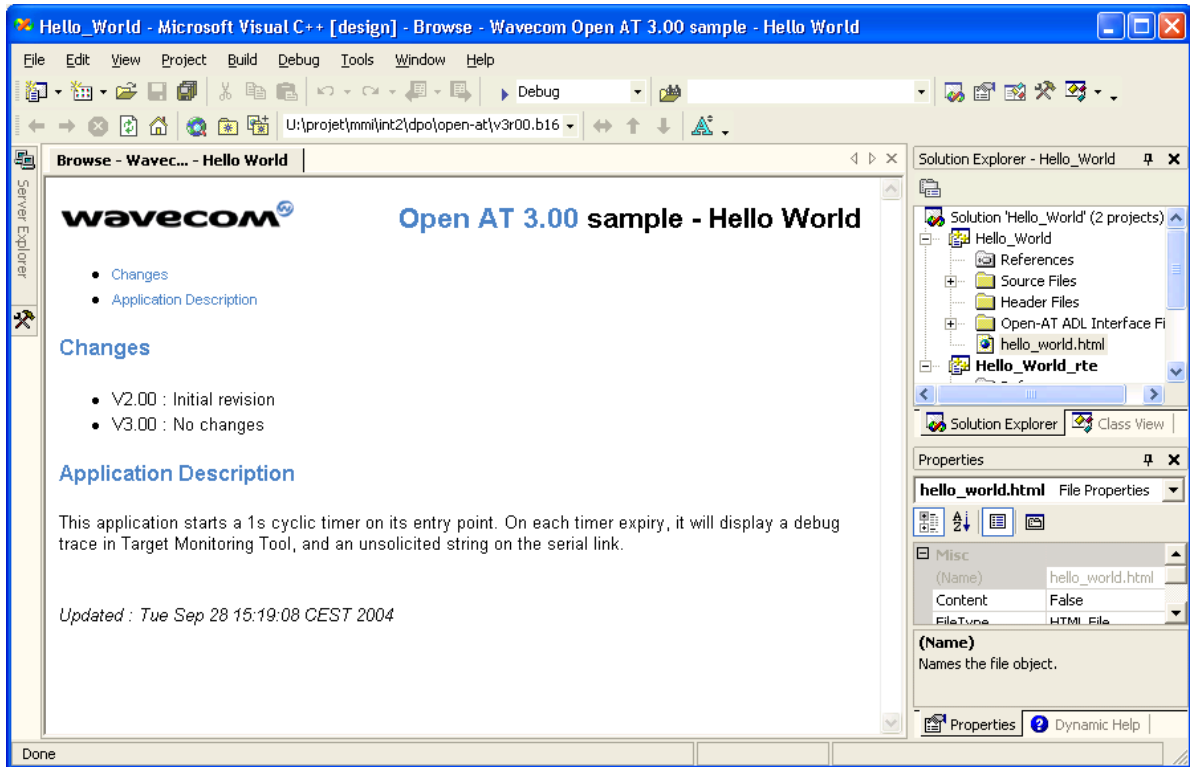


Figure 34 : Application sample HTML help file display

#### 5.4.4 Building application for Target Mode

To build the application for target mode, the target WISMO Target configuration has to be selected as the active one, from the Build toolbar :

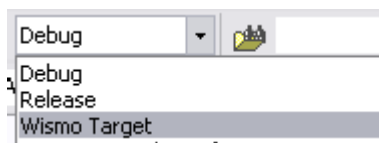


Figure 35 : Select the WISMO Target configuration

Then, the project may be built by launching the Build Solution command, from the Build Menu, the contextual menu, or with the Ctrl+Shift+B key combination. The compilation results will be displayed in the output window.

Error messages which may be displayed by the Software Generation Toolkit are listed in the wmmake script description.

**5.4.5 Building application for remote mode**

To build the application for remote mode, the configuration has to be set. Both Release and Debug mode are available, but the Debug mode should be chosen in order to perform step-by-step debugging in the program. The configuration may be selected from the Build->Configuration Manager... menu :

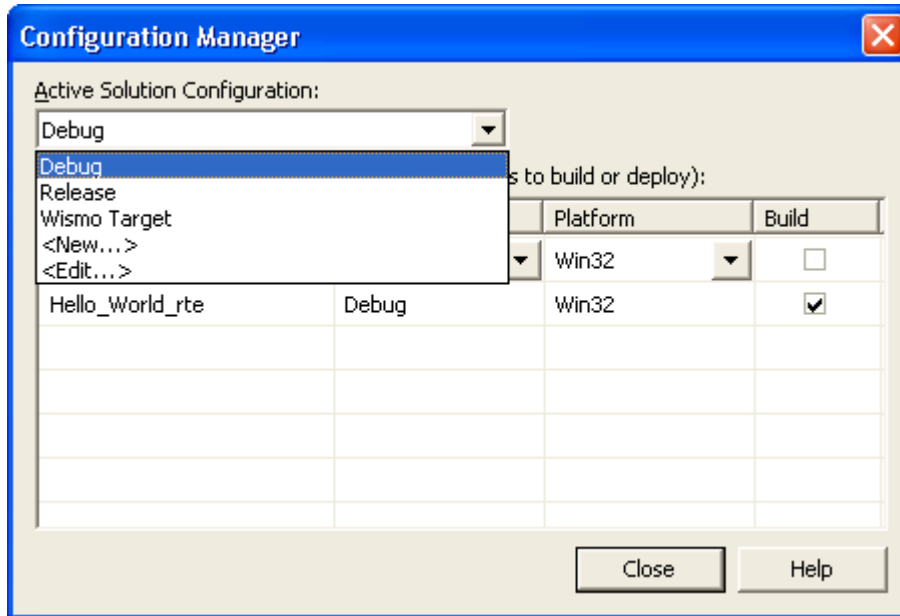


Figure 36 : Select the configuration – way 1

Or from the Build toolbar :



Figure 37 : Select the configuration – way 2

Then, the project may be built by launching the Build Solution command, from the Build Menu, the contextual menu, or with the Ctrl+Shift+B key combination. The compilation results will be displayed in the output window.



29th November 2005

#### 5.4.6 Starting the Remote Task Environment

The first time the RTE project is executed (by the Start command / F5 key), the environment will prompt for the debug executable file.

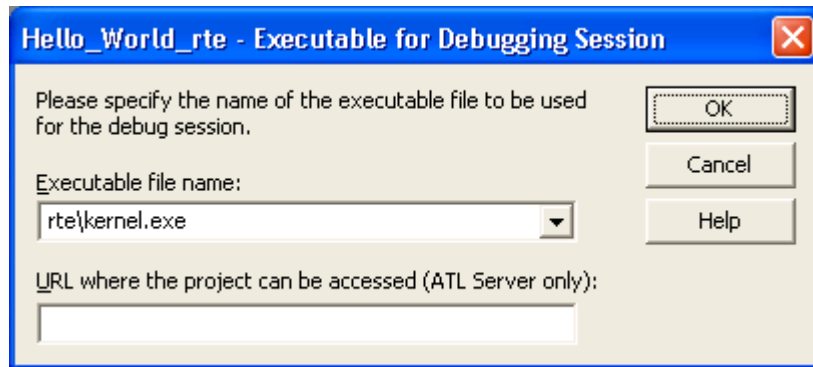


Figure 38 : Provide RTE kernel file name

The file to provide in order to start debug the application is "**rte\kernel.exe**". Further debug sessions (for this project) will not prompt again for this executable name. If a new project is created, or if the current project is updated with the Open AT<sup>®</sup> Project Wizard, the kernel executable name will be prompted again.

## 6 Development ToolKit

### 6.1.1 Overview

The Development ToolKit is a set of 4 tools running under Windows, designed and supplied by Wavecom.

The Development ToolKit environment is presented in Figure 3.

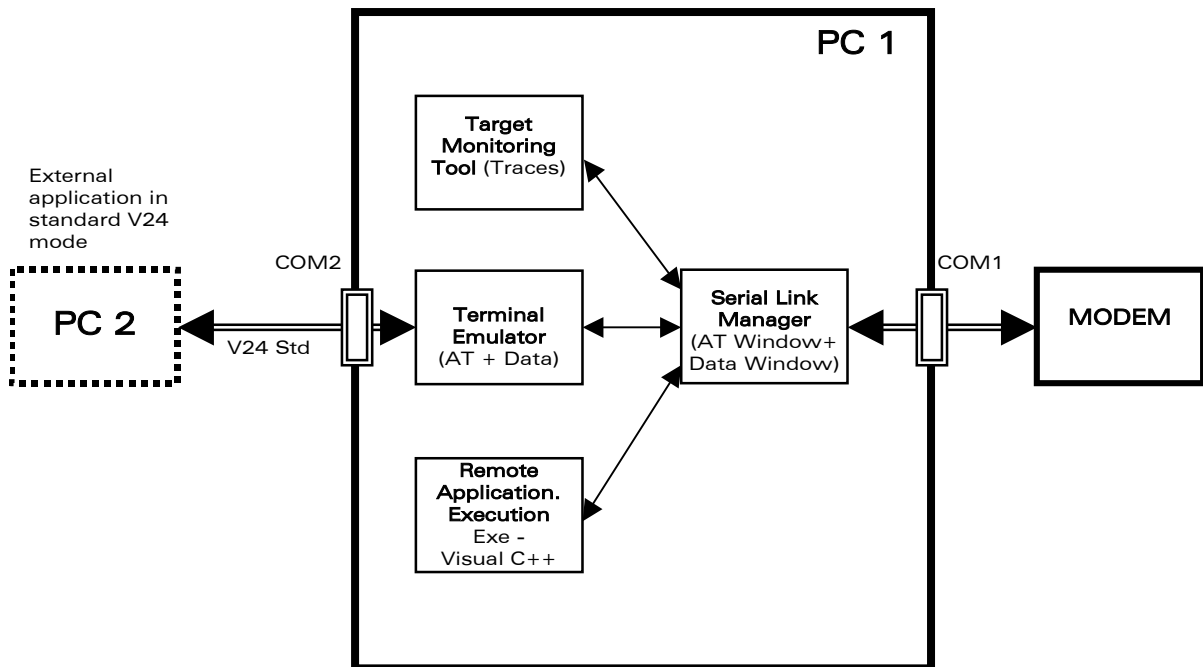


Figure 39: The Development ToolKit Environment

These tools are described here-below.

## 6.1.2 Serial Link Manager

### 6.1.2.1 Description

The **Serial Link Manager** uses the PC serial link (COM1) to communicate with Wavecom products. It behaves like a server between the different applications.

The other serial port (COM2) can be used to connect another PC, using the serial link in nominal mode (see Figure 3).

### 6.1.2.2 Getting Started

The Serial Link Manager is automatically launched as soon as the **Target Monitoring Tool**, the **Terminal Emulator** or the **Remote Application** starts : the serial port is opened and the Serial Link Manager icon is displayed on the task bar. Double-clicking on this icon will launch the Serial Link Manager dialog box.

With this Dialog Box, the user can manage the Serial Port (with the help of the **"CommPort"** menu : open/close the port, set the port configuration). It can be closed with the **"Window" -> "Hide Tray"** command. Refer to the Target Monitoring Tool help for more information on the Serial Link Manager.

Note: the Serial Port configuration used in the Serial Link Manager will be also used in the other programs of the Development ToolKit.

## 6.1.3 Serial Link Manager

The Serial Link Manager dispatches data received from the serial link to the Wavecom PC applications (Target Monitoring Tool, Terminal Emulator and Remote Task Environment). It also multiplexes the frames sent by these applications and forwards them to the product.

29th November 2005

## 6.1.4 Target Monitoring Tool

### 6.1.4.1 Description

The **Target Monitoring Tool** is used to display the Embedded application trace messages (from the Target or the Remote Application).

### 6.1.4.2 Getting Started

In order to use the **Target Monitoring Tool**, please follow these steps:

- 1) Download an Embedded Core Software to the target, using one of the provided samples,
- 2) Make sure the COM1 port is not used by another application,
- 3) Start the **Target Monitoring Tool** application from the Windows Start menu,
- 4) Check the PC serial link settings:
  - Click "**Settings**" in the "**Preferences**" menu,
  - Select the "**Com**" tab,
  - Check default mode parameters :
    - Baud rate : 115200
    - Data bits : 8
    - Parity : none
    - Stop bits : 1

Notes:

The serial port settings can be changed using either the Target Monitoring Tool or the Windows Serial Link Manager.

The serial port speed can be detected by the **Commands->Auto Detect** command of the Target Monitoring Tool.

- 5) Check the communication with the Target :
  - Click the "**Terminal Emulator**" button in the Target Monitoring Tool toolbar, and select the "**AT1**" window.

Type AT and press <Enter>:

A blue Ok response must appear. Otherwise, select "**Init Target**" from the "**Commands**" menu of the **Target Monitoring Tool** and type AT again,

- 6) Close the **Terminal Emulator** window.

29th November 2005

#### 6.1.4.3 Get Traces Sent by the Wavecom Target

- 1) Select **"Open"** from the **"Traces"** menu,
- 2) Select **"Get Information about Target"** from the **"Commands"** menu,
- 3) Select **"Set and Request to the Target..."** from the **"Commands"** menu,  
Select **CUS** from the **Parameter** list,  
Click on all levels from 0 to 31 in **Bitmap**,  
Click on the **"Send Level"** button,  
Click on the **"Close"** button,
- 4) The Trace window then displays the traces sent by the Wavecom product.

#### Notes:

- The traces are strings sent by the embedded application to the Target Monitoring Tool thanks to the `TRACE()` & `DUMP()` functions. This function takes the display level, and the string to display as parameters. See the Development Guide for more information.
- If the serial link speed is set to 9600 bps, some traces may not appear (lost). The user should, to avoid this, use the speed of 115200 bps on the serial link (see the **"AT+IPR"** command in the AT Commands Interface Guide).
- In the case of an embedded application using the FCM API, the traces flow from the Target Monitoring Tool may conflict with the Terminal Emulator data block flow : some of the data or of the traces may be not displayed. When sending many data on an IO flow, the user should limit the number of traces requested to the Target Monitoring Tool.

#### 6.1.4.4 Select the Current Embedded Application Workspace

When an embedded application binary file is generated, a workspace file `MyProject.wks` is also produced.

1. Use the **File->Open Workspace...** menu.
2. Select the `.wks` file the **out/tmt** directory of the current downloaded embedded application (in **arm** or **gcc** directory, according to the used compiler).
3. The Target Monitoring Tool does not need to be restarted.

The Target Monitoring Tool can use this file to display the function's names when the software crashes and leads to a BackTrace. (See the Development Guide for more information on software security.)

The **"BackTraces"** log can be retrieved by using the **Commands->Get Debug Info->Request EEPROM logged errors** menu.

The workspace defines also a diagnose tip file, usable to display tooltip windows according to the current downloaded application, during the selection of trace levels. This file is named `DiagnoseTips.ini` and is copied in the

29th November 2005

**out/tmt/config** directory. The user should update this file with its own tip labels to use them with the target monitoring tool.

## 6.1.5 Terminal Emulator

### 6.1.5.1 Description

The **Terminal Emulator** is an AT command terminal. It is used by the User to send and receive string commands to or from a customer task (on the Target or in Remote mode). It can use either the serial link Nominal mode (the AT window is then open) or the Wavecom Debug mode.

The **Terminal Emulator** connects an External application in standard V24 mode. It transforms a data flow from standard mode to debug mode. An External application can also get information in nominal mode.

### 6.1.5.2 Getting Started

If the **Target Monitoring Tool** is running, start the **Terminal Emulator** using the corresponding button in the toolbar (check **"Toolbar"** in **"View"** menu), or **"AT cmd terminal"** from the **"Tools"** menu.

You can also launch the **Terminal Emulator** using the shortcut from the Windows Start menu.

The Terminal Emulator provides two windows : one for AT commands and one for the V24 Data flow. Refer to the Terminal Emulator **"help"** menu for more information.

Note: in case of an embedded application using the FCM API (cf. the Development Guide), the target should be initialized in Debug Mode (**"Commands"** -> **"Init Target"**) before any FCM operation; otherwise the Terminal Emulator and the Target Monitoring Tool will not run properly. Moreover, if the external serial port (COM2) is used, the **"External Com"** -> **"Open..."** command must be used before any FCM operation.

Note: if there is a data call, or if the serial link is switched into DATA mode with the FCM API, the **"+++"** sequence does not work in the Terminal Emulator.

## 6.1.6 Remote Application Execution

This tool is a set of libraries with which the User application can be executed on a PC by means of Visual C++<sup>®</sup> while communicating with the target through a serial link.

This tool is detailed in the next paragraph.

## 7 Remote Task Environment

### 7.1 Basic Concepts

An Embedded application can be executed from a PC, while communicating with the rest of the Target software through a serial link.

An Embedded application is implemented in a specific task (customer task). This task can be executed on a PC, using the Windows version of RTK. In this case, the customer task is deactivated on the target. This concept is described in Figure 4.

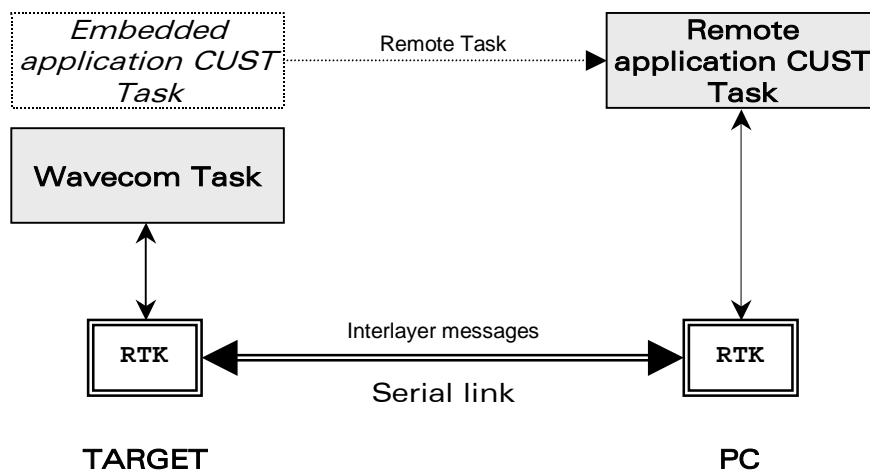


Figure 40 : Project Architecture

Without changing the code, an Embedded Application becomes a Remote Application after recompilation.

The Remote process (development, compilation, link, execution) operates in one of the supported development environments.

## 7.2 Project Creation

Please refer to the “Building Open AT<sup>®</sup> applications” chapter for more details on how to create and build projects in remote mode.

## 7.3 Remote Application Execution

### 7.3.1 Serial Port Configuration

Using the **Serial Link Manager**, the user has to check that the serial port speed is set to the same value as the Target port speed.

The **Auto Detect** command from the **Target Monitoring Tool** can be used to retrieve this speed. In order to get better results, the speed should be kept at **115200 bauds** (default value).

When connecting the target to the computer through the serial link, the connection may not work, because the Target and the computer serial interfaces may use different baud rates.

Selecting **Auto Detect** will then make the **Target Monitoring Tool** send frames with different successive baud rates. If the target sends them back, it means the target speed and the computer speed match. The **Target Monitoring Tool** then stops scanning.

An **Auto Detect** sequence typically lasts five to ten seconds. The **Target Monitoring Tool** may seem slower during this time.

The Terminal Emulator has to be launched before starting the remote application, to check whether the communication with the target is correct or not.



### 7.3.2 Remote Application Launch

To launch the user interface, from the Development Environment menu, select the following command :

**build -> Start Debug -> Go (F5) Visual C++ 6**  
 or  
**Debug -> Start (F5) Visual C++ .NET**

The Development environment then starts in Debug mode, and launches the **Remote Application Execution Tool**.

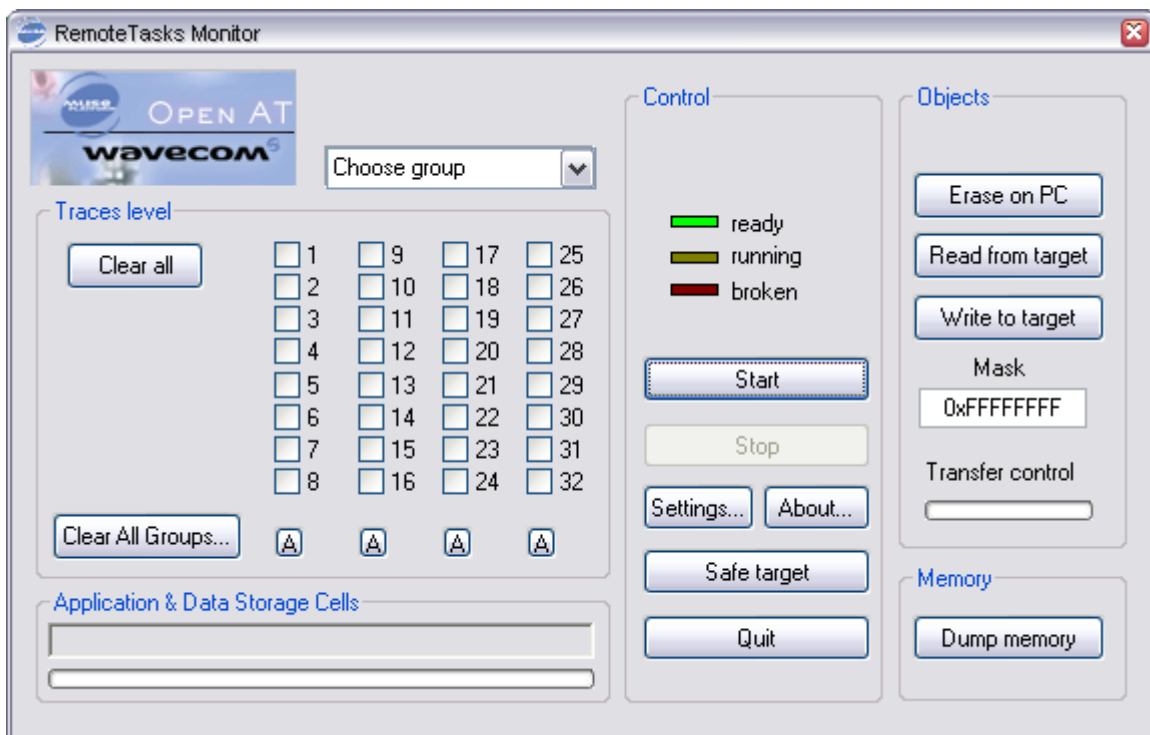


Figure 41 : Remote Application Controller

The remote process is started as soon as the **Start** button is clicked from the Remote Application Controller (see Figure 42 – step 1, then step 2).

The Initialization sequence is described below :

1. Initialization of the serial link (in Serial Link Manager mode),
2. Target configuration,
3. Target RTK Re-initialization,
4. Remote Application Execution creation and activation.
5. The **Target Monitoring Tool** should then display the selected trace levels, showing that the application is running correctly (in Figure 42, the step 3 is reached)

29th November 2005

**Note**

During step-by-step execution, trace messages are not displayed at once by means of the **Target Monitoring Tool**. They are only displayed after an RTK scheduling.



Figure 42 : RTE Monitor control steps (1 to 4)

To stop the remote application execution, the Stop button has to be clicked (Figure 42, step 4). Then, a Flash Object synchronization may be done (if wished), and the RTE monitor may be closed by the Quit button.

If the RTE application does not end by this way (Stop button, and then Quit button), due to an exception during the process, or by using the "Stop Debug" option of the Development Environment, any target Open AT<sup>®</sup> application will still be disabled.

In order to restore the target application normal running mode, the RTE monitor has to be launched again, and the Safe Target button has to be clicked.

**7.3.3 Traces Configuration**

Embedded application trace messages are displayed using the **Target Monitoring Tool**, by the way of a trace window opened from the **Traces** menu.

The Target application traces then appear **in black** and the Remote Application traces appear **in blue**.

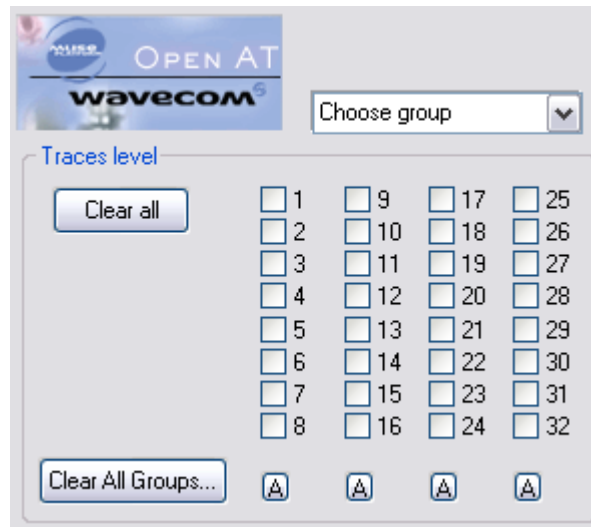


Figure 43 : Trace Level Selection

Each trace instruction corresponds to a trace level. The user has to select this level using the Remote Application Controller before using the **Target Monitoring Tool** to display the traces.

Firstly, the Open AT<sup>®</sup> task has to be selected (ADL applications use always CUS4 task).

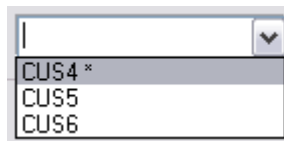


Figure 44 : Select Open AT<sup>®</sup> task

Then each trace level may be selected separately, or a whole column by pressing a **A** button. Trace levels may be also cleared separately, or together, for the selected task ( **Clear all** button), or for all tasks ( **Clear All Groups...** button).


Enabled trace levels may be changed at any time, during or out of the application execution. Changes will immediately be applied in the **Target Monitoring Tool**.

29th November 2005

### 7.3.4 Data Flash Objects synchronization

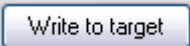
Data Flash Objects are entities in which the User application can save data such as addresses, phone numbers, etc. An object is referred to by a 16-bit hexadecimal identifier (this is NOT a physical address).

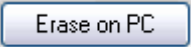
In remote application, these Data Flash Objects are emulated through physical files, located in the **[project path]\rte\objects\** directory.

In order to have the same Data Flash Objects configuration as on the Target, the user has to launch a read synchronization (  button) before starting the Remote Application.

**Note :**

Because both size and number of Data Flash Objects present on the Target, read synchronization may take up to one or two minutes.

After a remote application execution, objects written on PC side may be synchronized to target using the  button.

Finally, synchronized objects on PC side may be erased using the  button.

### 7.3.5 Application & Data Storage Cells synchronization

Application & Data storage cells are fully available within RTE mode. Writing to cells process is completely transparent. On "GetInfo" (reading) process, cells are synchronized and shadowed on PC each time the data are updated. The synchronization process is monitored by the corresponding progress bar in the RTE monitor interface.

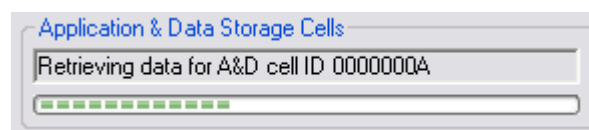


Figure 45 : A&D cell synchronization

## 8 Command-line scripts

Open AT<sup>®</sup> Project Wizard and development environment are graphical interfaces which use the following command-line scripts to respectively create and build Open AT<sup>®</sup> application projects.

### 8.1 Create Open AT<sup>®</sup> projects : wmnew script

The wmnew script may be used from a Cygwin bash shell command line in order to create/update Open AT<sup>®</sup> application projects. This script has to be called from the project root directory. The available options are described below.

#### 8.1.1 Script control

Option	Use
-h	Displays help on wmnew script and exits.
-v	Verbose mode (displays information on each project generation step).
-c	Cleans previously generated workspaces files before process.
-i	Ignores the current directory wmnew.opt file if any (by default, this file content is added to the command line options).
-ns	Does not save the currently required options in the wmnew.opt file (by default, command line options are saved in a wmnew.opt file in the current directory).
-s	Synchronizes the enabled workspaces in the Open AT <sup>®</sup> Settings application (by default, the only generated/updated file is the .scs one)

### 8.1.2 Script modes

Option	Use
[default]	Updates the existing project in the current directory with existing options.
-n	Creates a new project (similar as <code>-sample adl/New_Project</code> option)
-sample SAMPLE	Creates a copy of the required SAMPLE in the local directory ; if the sample depends on some sample libraries, these ones are also copied in a Libraries directory (located in the same directory as the project one). SAMPLE syntax is "[api]/SampleName" (where SampleName is the sub-directory name of chose sample in the \$WMATHOME/Samples/[api] root one ; for example <code>-sample adl/Hello_World</code> )

### 8.1.3 Project interface options

Option	Use
-adl	Uses the ADL interface (default).
-basic	Uses the Basic interface.
-olditf	[Basic interface only :] Uses the V2 and older interfaces task declaration mechanism, instead of the V3 task table.

### 8.1.4 Project type options

Option	Use
-app	Defines an application project (default).
-lib	Defines a library project.

### 8.1.5 Project global options

Option	Use
-name NAME	Defines the project name (for new projects only ; existing and sample projects names are automatically defined).
-flag FLAG	Adds the FLAG compilation flag to the project flags list.
-gssf FACTOR	Sets the GCC stack size factor to the FACTOR value (default 2). This value is the factor with which the source code stack size has to be multiplied for the GCC compiler (since GCC needs more room in the call stack on run time than the ARM compiler).
-product PRODUCT	Defines the used firmware on the target as the PRODUCT one. The required value is the corresponding sub-directory of the "\$WMATHOME/Wavecom Core Software" one, which contents the needed firmware files.

### 8.1.6 Project source paths options

Option	Use
-src SRCPATH	Adds SRCPATH to the source paths list (always contains by default the ./src current sub-directory). All .c files in these directories will be considered as a source file for the project.
-inc INCPATH	Adds INCPATH to the include paths list (always contains by default the ./inc & ./itf current sub-directories). All .h files in these directories will be considered as an header file for the project.
-path FULLPATH	Similar to : -src FULLPATH/src -inc FULLPATH/inc -inc FULLPATH/itf

### 8.1.7 Project object paths & names options

Option	Use
-extobj OBJPATH	Adds the OBJPATH path to the absolute objects paths list. Object files will be browsed exactly in the provided directory.
-rootobj OBJPATH	Adds the OBJPATH path to the root objects paths list. Object files will be browsed in the sub-directory which matches with the currently used compiler (for example arm/out for ARM compiler, gcc/out for GCC compiler, etc...)
-objname OBJNAME	Adds the OBJNAME template to the real object names filter list. Object files will be browsed with trying to match exactly the provided template (including the extension), whatever is the currently used compiler. (for example, the -objname "m*.o" option will browse for all object file names beginning with the "m" letter).
-objpref OBJNAME	Adds the OBJNAME template to the prefixed object names filter list. Object files will be browsed with trying to match the provided template, according to the currently used compiler. (for example, the -objpref "myobject" option will browse for object file names which match the "arm_myobject.o", "gcc_myobject.o" or "rte_myobject.obj", according to the currently used compiler).

29th November 2005

### 8.1.8 Project library paths & names options

Option	Use
-extlib LIBPATH	Adds the LIBPATH path to the absolute libraries paths list. Library files will be browsed exactly in the provided directory.
-rootlib LIBPATH	Adds the LIBPATH path to the root libraries paths list. Library files will be browsed in the sub-directory which matches with the currently used compiler (for example arm/out for ARM compiler, gcc/out for GCC compiler, etc...)
-libname OBJNAME	Adds the LIBNAME template to the real library names filter list. Library files will be browsed with trying to match exactly the provided template (including the extension), whatever is the currently used compiler. (for example, the -libname "mylibraries*.lib" option will browse for all library file names beginning with the "mylibraries" template).
-libpref LIBNAME	Adds the LIBNAME template to the prefixed library names filter list. Library files will be browsed with trying to match the provided template, according to the currently used compiler. (for example, the -libpref "mylibrary" option will browse for library file names which match the "arm_mylibrary.lib", "gcc_mylibrary.lib" or "rte_mylibrary.lib", according to the currently used compiler).

### 8.1.9 Provided libraries options

Option	Use
-samplelib SMPLIB	Adds dependencies to the SMPLIB sample library in the current project. (copy the SMPLIB sample library files in the ../Libraries/SMPLIB directory, and then similar to : -inc ../Libraries/SMPLIB/itf -rootlib ../Libraries/SMPLIB -libpref SMPLIB)
-otherlib OLIB	Adds dependencies to OLIB in the Add-on libraries list. OLIB syntax is [add-on]/[version] ; Where [add-on] is the required add-on directory (located in the \$WMLIBHOME location), and [version] is the version sub-directory.



29th November 2005

## 8.2 Building Open AT<sup>®</sup> projects : wmmake script

The wmmake script may be used from a Cygwin bash shell command line, in order to build an Open AT<sup>®</sup> project. This script has to be called from the project root directory. The script syntax & options are defined below :

```
wmmake <Project> [options...]
```

where <Project> is the project name, without any extension.

### 8.2.1 Script control

Option	Use
-h	Displays help on wmmake script and exits.
-n	Do not browse for libraries projects dependencies. By default, for all libraries with a prefixed name (-libpref option in wmnew script) and a root path (-rootlib option in wmnew script), the wmmake script (re-)builds also the corresponding library project.

### 8.2.2 Clean options

Option	Use
all_clean	Cleans the output directory generated files and exits.
-c	Performs a "all_clean" process, and re-launches the build process.

### 8.2.3 Compiler options

Option	Use
[default]	Builds the project using the compiler defined by the Open AT <sup>®</sup> settings application.
-arm	Builds the project using the ARM compiler
-gcc	Builds the project using the GCC compiler

### 8.2.4 Memory options

Option	Use
[default]	Builds the project using the memory size defined by the Open AT® settings application.
-16	Builds the project for A memory size products
-32	Builds the project for B memory size products
-32W	Builds the project for B memory size P5186 products

### 8.2.5 Debug Options

Please see the ADL Trace Service description for more information about the debug flags.

-debug	Defines the <code>__DEBUG_APP__</code> flag at compilation time (default value).
-fulldebug	Defines both the <code>__DEBUG_APP__</code> & <code>__DEBUG_FULL__</code> flags at compilation time.
-release	Does not define any debug flag.

The following error messages may be displayed by the `wmmake` script.

**"The Open AT® environment is not correctly installed."**

*Please refer to the `wmcheck` script description to find what is not correctly configured in the Open AT® environment.*

**"The ARMDLL environment variable is not set."**

*The ARM compiler is selected in the Open AT® Settings application, but either it is not correctly configured in the ARMDLL environment variable, or it is not installed.*

**"The WMGCCHOME environment variable is not set."**

*The GCC compiler is selected in the Open AT® Settings application, but either the compiler path is not set, or the GCC compiler is not installed.*

29th November 2005

### 8.3 Check current Open AT® environment settings : wmcheck script

The `wmcheck` script may be used from a Cygwin bash shell command line, in order to display the current environment configuration, and detect what is going wrong if other tools (as the Open AT® project wizard, or the `wmmake` script) do not run properly. The script syntax & options are defined below :

```
wmcheck [-n | <output file>]
```

#### 8.3.1 Script control

Option	Use
<code>-n</code>	Do not display the configuration. If a configuration error is detected, an error message will be displayed, and the script return value will be 1 instead of 0.
<code>&lt;output file&gt;</code>	Name of the text file where to copy the configuration dump. This text file should be used to report any configuration problem.

#### 8.3.2 Configuration dump sample

When running the `wmcheck` script, a configuration dump similar to the one below may be displayed.

```
Current target compiler : arm [OK]
Current target memory size : 32 [OK]
Current IDE workspaces synchro list : .dsp .vcproj [OK]
Current SGT directory : /cygdrive/C/OpenAT/Tools/SGT/v1.2.11 [OK]
Current SGT version : v1.2.11 [OK]
Current SGT scripts directory :
/cygdrive/C/OpenAT/Tools/SGT/v1.2.11/script_sgt [OK]
Check make lib script :
/cygdrive/C/OpenAT/Tools/SGT/v1.2.11/script_sgt/make_lib [OK]
Current Open AT directory : /cygdrive/C/OpenAT/V302 [OK]
Current Open AT directory \ (for Windows\ ) : "C:\OpenAT\V303" [OK]
Current GCC compiler directory : /cygdrive/C/OpenAT/Tools/GCC [OK]
Current ARM compiler directory : C:\Program Files\ARM\ADSV1_2\BIN [OK]
Current Add-on libraries directory : C:/OpenAT/OtherLibs [OK]
Current Tools base directory : /cygdrive/C/OpenAT/Tools [OK]
Check wmnew script : /cygdrive/c/OpenAT/V303/TgtGen/WmGen/wmnew [OK]
Check wmmake script : /cygdrive/c/OpenAT/V303/TgtGen/WmGen/wmmake [OK]
Check wmsync script : /cygdrive/c/OpenAT/V303/TgtGen/WmGen/wmsync [OK]
Check wmsetup script : /cygdrive/c/OpenAT/V303/TgtGen/WmGen/wmsetup [OK]
```

29th November 2005

If at least one line of the configuration dump displays an [ERROR] tag, it means that there is a problem in the environment configuration.

In order to solve configuration problems :

- Check the configuration in the Open AT<sup>®</sup> Settings application.
- Try to run the "\$WMATHOME/TgtGen/WmGen/wmsetup" script from a Cygwin bash shell, and restart Cygwin.
- If the problem is not solved, try to install again Cygwin (ie. delete completely the cygwin directory, and re-install it from the Open AT<sup>®</sup> CDROM), and run the "\$WMATHOME/TgtGen/WmGen/wmsetup" script from a Cygwin bash shell.
- If the problem is still not solved, try to re-install completely the Open AT<sup>®</sup> environment.

## 9 TCPIP debug tool

A script tool is provided on Open AT<sup>®</sup> 3.03 SDK, in order to trace IP packets when using the TCPIP Add-on library.

Note: IP packets can be only monitored on the GPRS flow, not on the GSM Data flow.

### 9.1 Enable the traces

In order to display the IP packet in the Target Monitoring Tool, the +ADLDBG command may be used. This command syntax is defined below:

```
AT+ADLDBG=<FlowId>
```

Where <FlowId> is the ADL FCM flow ID to monitor, in order to display IP packets:

<FlowId>	0: no monitored flow (default value on start-up)
	1: GSM Data flow
	2: GPRS flow
	3: V24 UART1 flow.
	4: V24 UART2 flow
	5: V24 USB flow

In order to monitor the IP packets sent and received on the GPRS flow, the AT+ADLDBG=2 command has to be used.

FCM data packets traces are displayed on the level 25 of the CUS4 task, so in order to retrieve the IP packets traces, the CUS4 task's 25<sup>th</sup> level has to be enabled in the "Set and request to the target" dialog box of the Target Monitoring Tool.

### 9.2 Write traces to disk

In order to save retrieved IP packets traces into a file, the user may:

- Either enable the "Saving" checkbox in the Target Monitoring Tool ; the file name is defined by the "New File" options of the right-click menu in the Traces window.
- Or copy all the traces lines of the Traces window, and paste them into a new text file, within a text editor.

29th November 2005

### 9.3 TCPDUMP format conversion

From the Cygwin command line, the `wm_trc2tcpdump.awk` script may now be used to convert the traces text file to the TCPDUMP format. The syntax is:

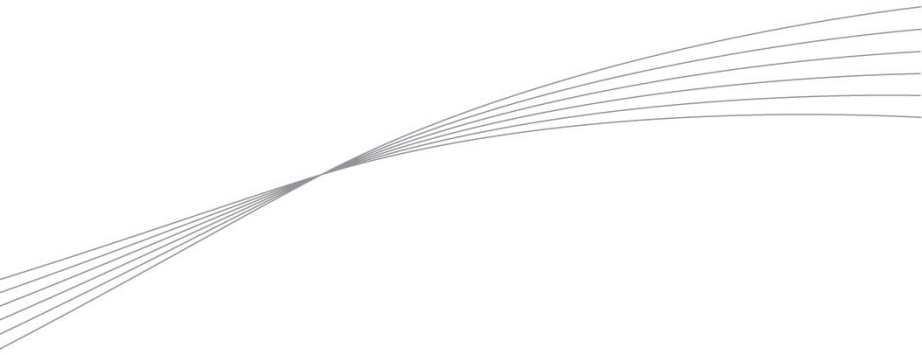
```
Usage: wm_trc2tcpdump.awk <TraceFile> <TcpDumpFile>
```

Where `<TraceFile>` is the text traces file saved from the Target Monitoring Tool, and `<TcpDumpFile>` is the TCPDUMP format destination file. This file will be overwritten without any warning.

If the text traces file does content any IP packets, a warning will be displayed, and the output file will not be written.

### 9.4 IP packets analysis

Now the TCPDUMP format file may be opened by any software able to read this format. The most known one is Ethereal, which may be downloaded for free from the <http://www.ethereal.com> URL.



**wavecom**   
*Make it wireless*

---

WAVECOM S.A. - 3 esplanade du Foncet - 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. - 4/F, Shui On Centre - 6/8 Harbour Road - Hong Kong - Tel: +852 2824 0254 - Fax: +852 2824 025

[www.wavecom.com](http://www.wavecom.com)