

Open AT® GTi User Manual

Revision: 001
Date: March 2006

wavecom
Make it wireless

Operating Systems | Integrated Development Environments | Plug-Ins | Wireless CPUs | Services

Open AT® GTi User Manual

Reference: WA_DEV_GTI_UGD_001

Revision: 001

Date: March 31, 2006

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.

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.

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.

Document History

Level	Date	History of the evolution	
001	March 31, 2006	Initial version of Open AT® GTi User Manual	

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.

Table of Contents

1 INTRODUCTION 10
1.1 Overview 10
1.2 Objectives 11
1.3 Software Installation 11
1.3.1 System Requirements 11
1.3.2 Hardware Requirements 11
1.3.3 Software Requirements 12
2 E-SIM DEVELOPMENT TOOLS 24
2.1 e-SIM Overview 24
2.2 RapidPLUS™ 8.1 24
2.3 Other MMI Customization tools 27
2.3.1 Layout Editor 27
2.3.2 Text Resource Manager 28
2.3.3 Multimedia Resource Manager 32
3 WAVECOM OPEN AT® SOFTWARE IDE 33
3.1 Open AT® Software Overview 33
3.2 Open AT® Software Areas of Application 34
3.3 Advantages of using Open AT® Software 35
3.4 Open AT® Software ADL Library 36
3.5 Mandatory Embedded Application Code 36
3.5.1 ADL API 37
3.6 Open AT® Software Running Modes 37
3.6.1 Target Mode 37
3.6.2 Remote Mode 37
3.7 Open AT® Software application size restrictions 37
4 WAVECOM DEVELOPMENT TOOLS 39
4.1 SELIMA 39
4.2 Target Monitoring Tool 40
4.3 Terminal Emulator 43
4.4 Debugger RTE based on VC++ 44
5 WAVECOM SAMPLE MMI 46

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.

5.1 Sample MMI Architecture 46
5.1.1 MMI Modules 47
5.1.2 Service Components 49
5.1.3 Data Containers 50
5.1.4 Widgets 50
5.1.5 Embedded Interface Components 51
5.2 MMI Integration over Wavecom Open AT® Software 51
5.2.1 Basic Services Integration 51
5.2.2 Embedded AT Integration 55
5.2.3 General Services Integration 56
5.2.4 Service Component Integration 57
5.2.5 Embedded Component Integration 67
5.3 Sample Use Case - Outgoing Call Set up 68
5.4 Sample MMI Menu Structure 70
5.5 Sample MMI Workspace Architecture 71
5.5.1 Overall Directory Structure 73
5.5.2 inc - Folder Containing Device and Display Driver Header Files 73
5.5.3 src –Folder Containing Device and Display Source Files 73
5.5.4 ESIM Specific Code Directory 75
5.6 Build and Deployment Procedure 76
5.6.1 Code Generation using RapidPLUS™ 77
5.6.2 Build using ARM Compiler 79
5.6.3 Download the Build on Target Platform 80
5.7 MMI Customization 80
5.8 Screen/Layouts - Screen Sizes, Layouts 81
5.8.1 Multimedia resources – Graphics and Animations 85
5.8.2 Text Resources - Support for different Languages 89
5.8.3 Menus - Add/Modify/Delete Menu Items 90
5.8.4 Keypad 93
5.9 Debugging using RapidPLUS™ and Wavecom tools 94
5.10 Wavecom Sample MMI Limitations 95
5.10.1 Keypad Handler 95
5.10.2 Battery & Charger Handler 95

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.

5.10.3 SMS Handler 95
5.10.4 Phonebook 96
5.10.5 Timer Handler 96
5.10.6 Multiple simultaneous AT commands 96
5.10.7 DOTA impact 96
5.10.8 Language Support 97
5.10.9 SIM Tool kit 97
6 SAMPLE APPLICATION DEVELOPMENT 98
6.1 Requirement 98
6.2 Design 98
6.3 Alarm Clock MMI development 101
6.3.1 Layout Development 101
6.3.2 Creating Menus and Text Resources 103
6.3.3 Adding Multimedia Resources 105
6.3.4 Creating RapidPLUS™ Project 106
6.3.5 Adding RapidPLUS™ development components 106
6.3.6 Adding application modes 107
6.3.7 Debugging Using Simulation 114
6.4 Integration with Open AT® Software Libraries 116
6.4.1 Generate application code 116
6.4.2 Update the C files for embedded components 118
6.5 Build and Deploy 119
6.5.1 Code build using ARM/RTE 119
6.5.2 Download image to target platform 120
6.5.3 Wake up, the alarm is ringing 121
7 REFERENCES 122

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.

List of Figures

Figure 1: Open AT® Software Installation – Startup Screen 13
Figure 2: Open AT® Software Installation – Setup Options 14
Figure 3: Open AT® Software Installation – Destination Folder 15
Figure 4: Open AT® Software Installation – Tools Path 16
Figure 5: Open AT® Software Installation – Cygwin Setup 16
Figure 6: Open AT® Software Installation – Cygwin Setup Screen 17
Figure 7: Cygwin – Download Source 17
Figure 8: Install Cygwin in the Root Directory 18
Figure 9: Cygwin Setup – Local Package Directory 19
Figure 10: Cygwin Setup – Select Packages for Installation 19
Figure 11: Cygwin Setup – Create Icons 20
Figure 12: RapidPLUS™ Installation 21
Figure 13: Dialog box to enter the Rapid product code. 22
Figure 14: RapidPLUS™ installation continues 22
Figure 15: RapidPLUS™ 8.1 – Snapshot 25
Figure 16: Layout Editor 28
Figure 17: Text Resource Manager - Menus 29
Figure 18: Text Resource Manager - Multilingual Support 30
Figure 19: Text Resource Manager - Menu & Text Generator 31
Figure 20: Multimedia Resource Manager - GraphicResources 32
Figure 21: WITHOUT Open AT® Software 33
Figure 22: WITH Open AT® Software 34
Figure 23: Open AT® Software with Custom Commands 34
Figure 24: Open AT® Software Automotive Application 35
Figure 25: Serial Link Manager Port Settings 39
Figure 26: Default PC Serial Link Settings 40
Figure 27: Init Target Command Window 41
Figure 28: The Auto Detect Icon 41
Figure 29: Set Trace Level 42
Figure 30: Set and Request to the Target Window 42

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.

Figure 31: Terminal Emulator 44
Figure 32: Remote Task Environment 45
Figure 33: Remote Task Execution (Remote Mode) 45
Figure 34: Sample MMI integration with Wavecom Open AT® Software 46
Figure 35: Wavecom Sample MMI Design 47
Figure 36: Display Integration 52
Figure 37: Keypad Integration 53
Figure 38: Timer Integration 54
Figure 39: Embedded AT Integration 55
Figure 40: General Services Integration 57
Figure 41: Phonebook Integration 59
Figure 42: SMS Integration 61
Figure 43: Call Control Integration 63
Figure 44: Call Records Integration 65
Figure 45: Network Management Integration 67
Figure 46: Sample Use case Realization – Outgoing Call 70
Figure 47: Sample MMI Menu Structure 71
Figure 48: Overall directory structure 73
Figure 49: Code generation Menu 77
Figure 50: Code generation - Preferences 78
Figure 51: Code generation - In progress 79
Figure 52: HyperTerminal – Build Download 80
Figure 53: Screen Size/Layout Customization - Selecting Layout 82
Figure 54: Screen Size/Layout Customization - Resizing Layout Elements 83
Figure 55: Screen Size/Layout Customization - Adding New Layout Element 84
Figure 56: Multimedia Resource Manager- Folder Structure 85
Figure 57: Multimedia Resource Manager – Settings 86
Figure 58: Multimedia Resource Manager- GraphicResources 87
Figure 59: Multimedia Resource Manager – Updating Bitmaps in RapidPLUS™ 88
Figure 60: TextResources – Adding a language 89
Figure 61 : Menus - Modifying a Menu Item 90
Figure 62: Menus - Modified Menu Item 91
Figure 63: Menus – Adding a Menu Item to an existing menu 92

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.

Figure 64: Clock Application – MMI Design 99
Figure 65: Clock Application - Layout for Idle Screen 101
Figure 66: Clock Application – Layout for WMI_EDITOR 101
Figure 67: Clock Application - Layout used by WDG_LIST 102
Figure 68: Clock Application - Layout for WMI_POP 102
Figure 69: Clock Application - Menu on hidden clock 103
Figure 70: Clock Application - Menu on shown clock 103
Figure 71: Clock Application - Adding string 104
Figure 72: Clock Application - Adding TextResource 104
Figure 73: Clock Application - Alarm Signal bitmap added 105
Figure 74: Clock Application - Creating RapidPLUS™ project 105
Figure 75: Clock Application - Adding User Objects 107
Figure 76: Clock Application - Mode Tree 109
Figure 77: Clock Application - Menu to cancel or set Alarm 110
Figure 78: Clock Application - Alarm Set (WMI_EDITOR) 111
Figure 79: Clock Application - Alarm Set Popup 111
Figure 80: Clock Application - Clock Setting Menu 111
Figure 81: Clock Application - Hidden Clock 112
Figure 82: Clock Application - Set Time Editor 112
Figure 83: Clock Application - Set Date Editor 112
Figure 84: Clock Application - Ringing 113
Figure 85: Clock Application - Opening Debugger 115
Figure 86: Clock Application - Debug Control dialog 115
Figure 87: Clock Application - Setting a breakpoint 116
Figure 88: Clock Application - Breakpoint hit 116
Figure 89: Clock Application - Generate Code Option 117
Figure 90: Clock Application - Code Generation Progress bar 118
Figure 91: Clock Application Workspace 120

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.

1 Introduction

1.1 Overview

Wavecom is a worldwide leader in pre-packaged wireless communication solutions for automotive, industrial and mobile professional applications. Wavecom's solutions include all the software and hardware elements that are necessary to develop innovative wireless devices, as well as tools and necessary associated services needed to bring them to market quickly and easily.

One such solution is the Wavecom Open AT® Software framework. Open AT® Software is at the heart of a growing number of wireless products in the automotive, M2M and mobile professional market segments. The Open AT® software development kit provided by Wavecom helps developers shorten time to market by reducing hardware design, software development, and test and validation cycles. Eliminating extra components and using existing software building blocks helps reduce development costs and shrink hardware footprints. Furthermore, the Open AT® Application Development Layer provides simple, ready-to-use templates for managing communications, messaging, hardware interfaces and more.

In order to reduce the time to market considerably for the customers, Wavecom has further provided an evolution of the existing Open AT® Software solution. The solution is referred to as the Open AT® GTi. Open AT® GTi is an innovative development platform that enables developers of wireless devices to create customized, display-centric products. Recognizing that differentiating products quickly and at the same time keeping costs down are key success factors for our customers, Wavecom has built Open AT® GTi on four cornerstone elements.

- RapidPLUS™ a proven, user-friendly development environment from Wavecom's partner e-SIM Ltd. allowing for quick application building and customization, computer testing and automatic code generation;
- A industrial reference application to get started quickly;
- A wide range of Wavecom's Wireless CPUs to choose from;
- Professional training services and technical support to ease development.

By employing e-SIM's market-proven Man-Machine Interfaces (MMI) technology, already widely used in the mobile handset and automotive markets, development and customization of Open AT® GTi-based products becomes easy and fast, allowing maximum flexibility at minimum risk.

In brief, Open AT® GTi simplifies a very complex process and unlocks the real potential of creative companies who want to quickly differentiate their products by providing display layout flexibility and allowing for other graphic elements like color images to be incorporated.

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.

1.2 Objectives

The objectives of this document are to provide:

- An overview of Wavecom Open AT® Software Architecture
- An overview of the e-SIM MMI development tools used for MMI customization
- An understanding of how Wavecom Sample MMI has been integrated over Wavecom Open AT® Software to create Wavecom Open AT® GTi Architecture
- Help customization of Open AT® GTi MMI using e-SIM MMI customization tools
- Development of a sample application using e-SIM and Wavecom development tools.

1.3 Software Installation

1.3.1 System Requirements

To work on the Open AT® GTi framework, your system must support one of the following operating systems:

- WINDOWS 98
- WINDOWS NT
- WINDOWS Me
- WINDOWS 2000
- WINDOWS XP

Note	Even though the GCC compiler is provided with the Open AT® Software installation, it is not used with the Open AT ® GTi.

1.3.2 Hardware Requirements

The main requirements for the PC to support the Open AT® Software IDE are:

- Pentium 300MHz (or higher) processor with at least 128 MB of RAM and 500 MB free hard disk space.
- A CD ROM drive.
- At least one free COM (serial) port for the communication with the target product.

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.

Currently, Open AT® GTi application is supported on only on Wavecom Q2406B and Q2686H Wireless CPUs.

1.3.3 Software Requirements

Wavecom Open AT® GTi Architecture requires installation of the following software for the application development on the PC:

i. Open AT® Software IDE

ii. e-SIM RapidPLUS™

Wavecom Open AT® GTi requires the following software on the Wavecom CPUs to run the applications:

i. For Q2406B
   a. Open AT® Software v3.10
   b. Wavecom OS 6.56

ii. For Q2686H
   a. Open AT® Software v4.00
   b. Wavecom OS 6.60

1.3.3.1 Installation of Open AT® Software IDE

The steps listed below will guide you through the installation of the Open AT® Software IDE. Refer to the screenshots for references made to specific field labels.

i. Insert the Open AT® Software IDE CD-ROM in the CD drive on your PC. If the Autostart option is enabled on your PC, the installation starts automatically.

ii. If the Autostart option is disabled, and if the auto-installation does not start, you can launch the executable file directly from the root folder of the CD-ROM using Windows Explorer. The executable file is stored in the path: \autorun\autorun.exe

iii. The Open AT® Software Setup screen is displayed. Click **Next** to continue.

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.

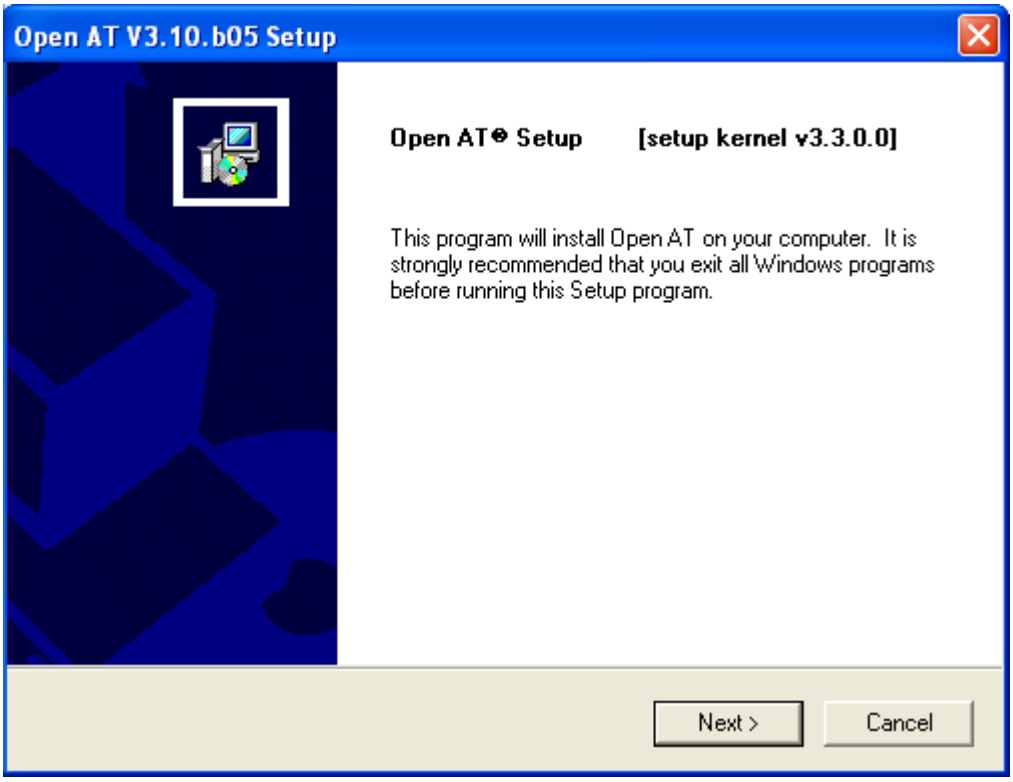


Figure 1: Open AT® Software Installation – Startup Screen

iv. The following components can be installed together or separately:

- Open AT® Software development environment, with libraries, header files and the documentation in PDF format
- Open AT® Software sample applications, with source and binary files with ADS,GCC or a Visual C++ compiler
- Wavecom software generation toolkit, used to build Open AT® Software applications in the target mode
- Cygwin(optional) - this tool is a Linux environment running under Windows
- Select the required components from the Setup options and click **Next** to continue.

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.

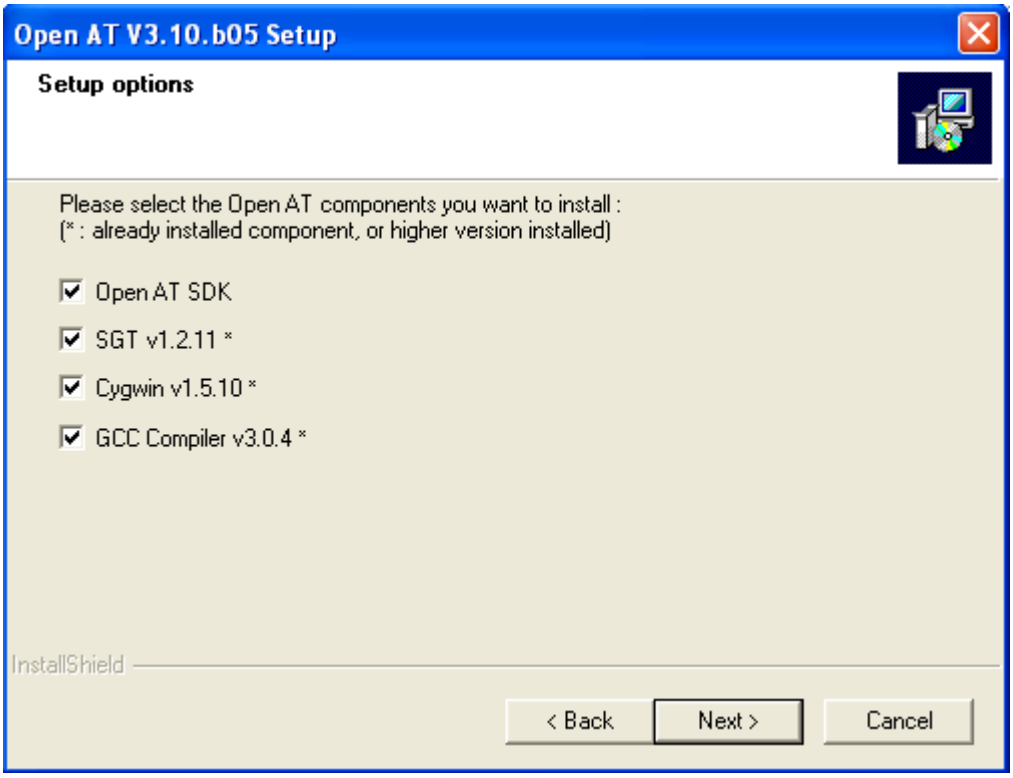


Figure 2: Open AT® Software Installation – Setup Options

v. The Open AT® Software IDE can be installed in any directory on the hard disk.

vi. By default the files are installed in the C drive, in a folder named with the version number of the Open AT® Software IDE. However, if required, you can enter a location that better suits your requirement.

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.

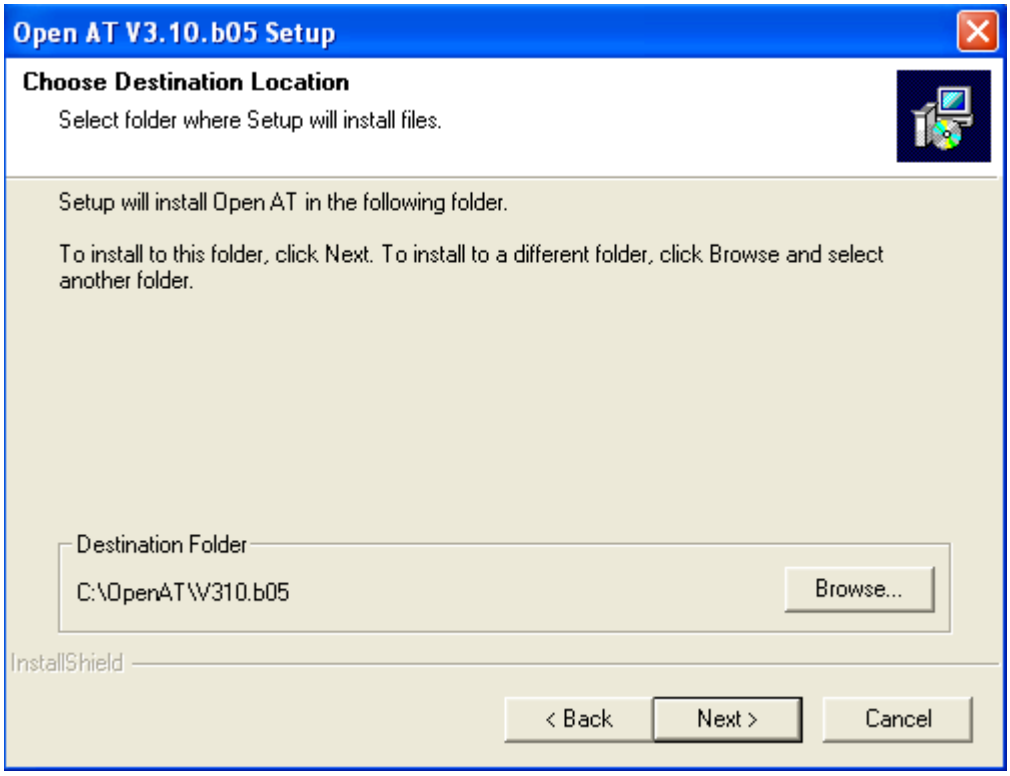


Figure 3: Open AT® Software Installation – Destination Folder

vii. Click **Next** to proceed. The files will be copied under the specified destination folder.

Note	The folder path must not include any spaces in it.

viii. The default location (as shown in the example) is the C drive. However, you can click **Browse** and change this location to suit your requirements. Click **Next** to proceed.

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.

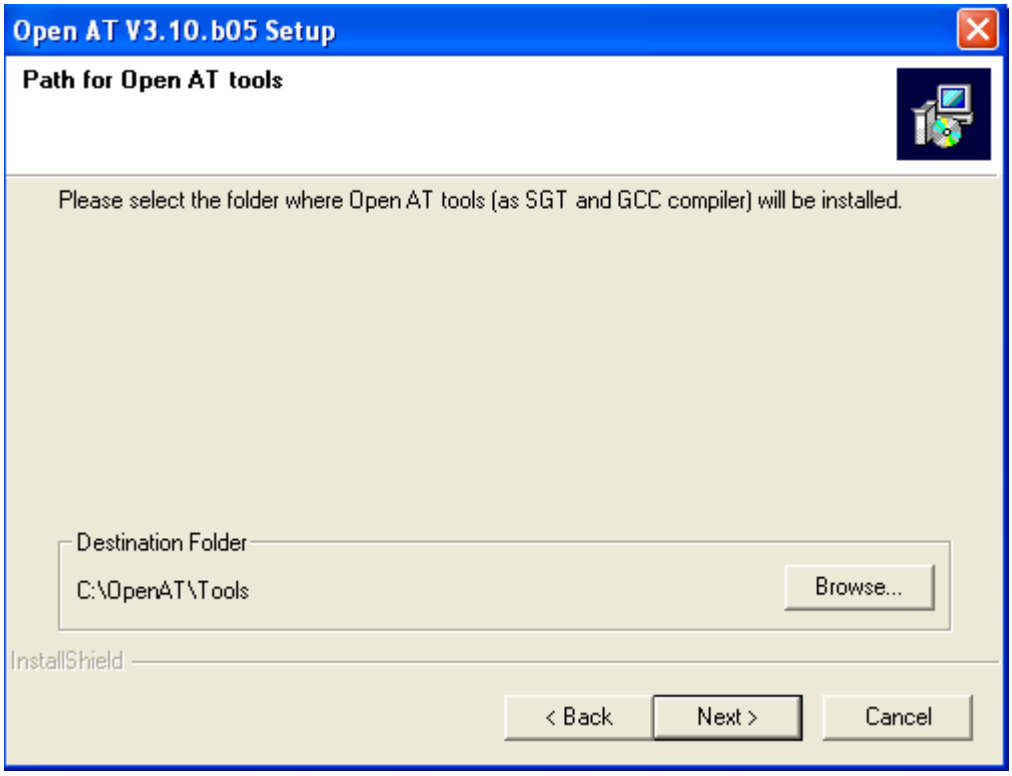


Figure 4: Open AT® Software Installation – Tools Path

Note	The Open AT® Software tools path must not include any spaces in it.

ix. Click **OK** to launch the Cygwin setup.

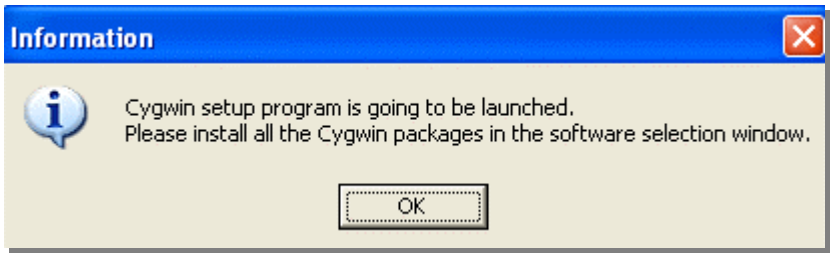


Figure 5: Open AT® Software Installation – Cygwin Setup

x. The Cygwin Setup wizard is displayed. Click **Next** to proceed.

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.

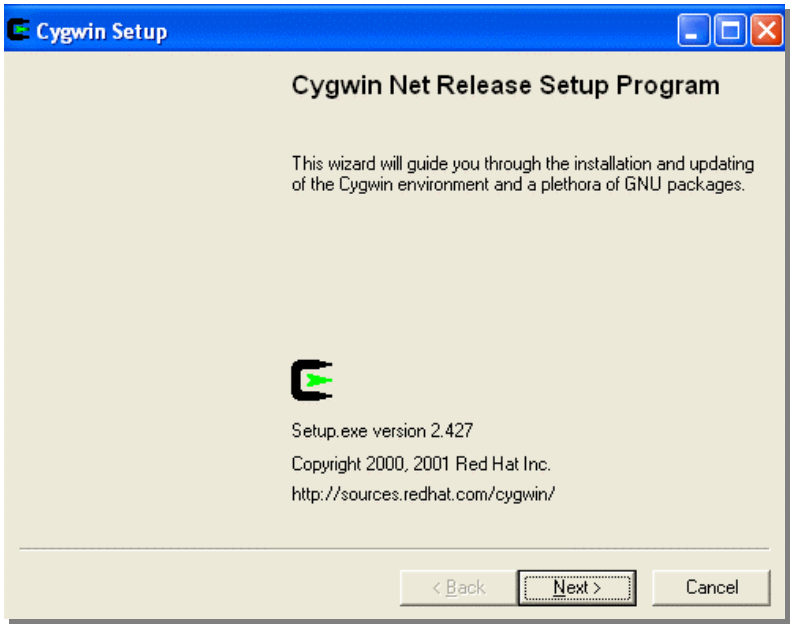


Figure 6: Open AT® Software Installation – Cygwin Setup Screen

xi. The Cygwin Setup – Choose Installation Type screen is displayed. Select the location from where you want to install the files. Select the Install from Local Directory option, and click **Next**.

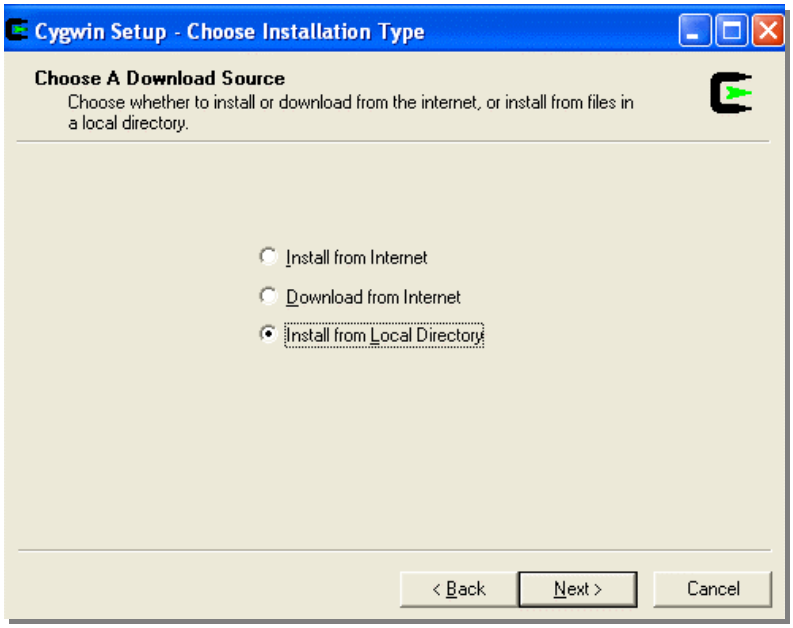


Figure 7: Cygwin – Download Source

xii. Click **Browse** to navigate to the directory from where you want to install Cygwin.

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.

xiii. Select the All Users option in the Install for group box.

xiv. Select the Unix option in the Default Text File Type group box.

xv. Click **Next** to continue.

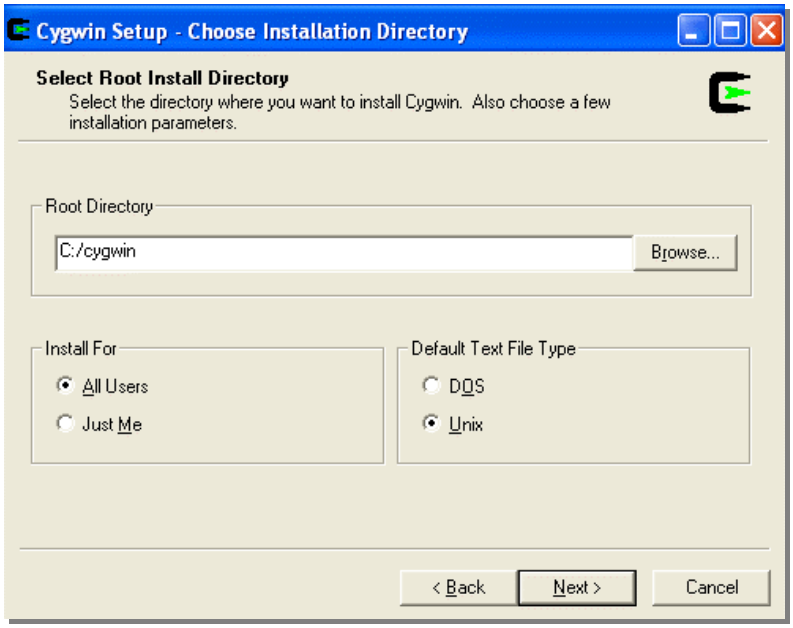


Figure 8: Install Cygwin in the Root Directory

xvi. Select the directory where you want to store the downloaded installation files. If the directory does not exist, the system will create it. Click **Next** to proceed.

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.

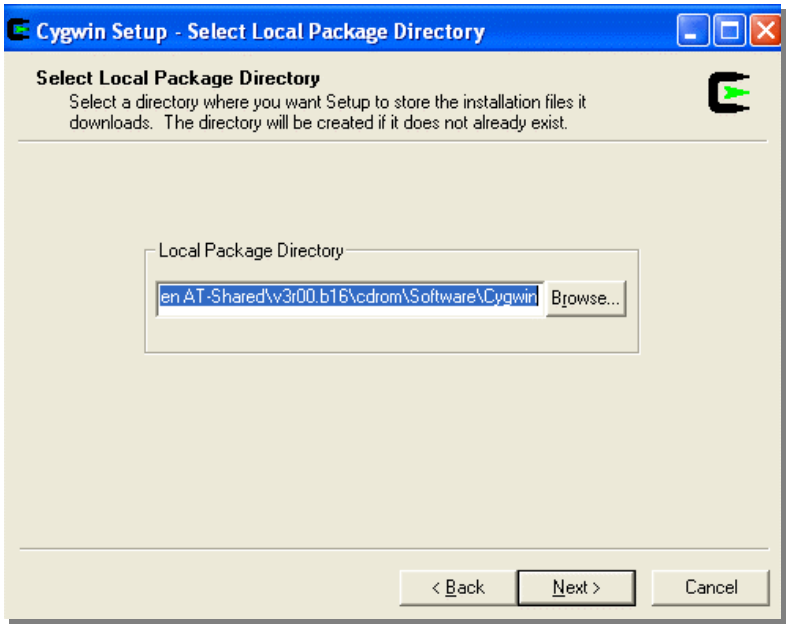


Figure 9: Cygwin Setup – Local Package Directory

xvii. Select the Curr radio button from the Cygwin Setup – Select Packages screen.

xviii. Select the All option to properly install all the packages; otherwise it will display an unspecified error. Click Next to proceed.

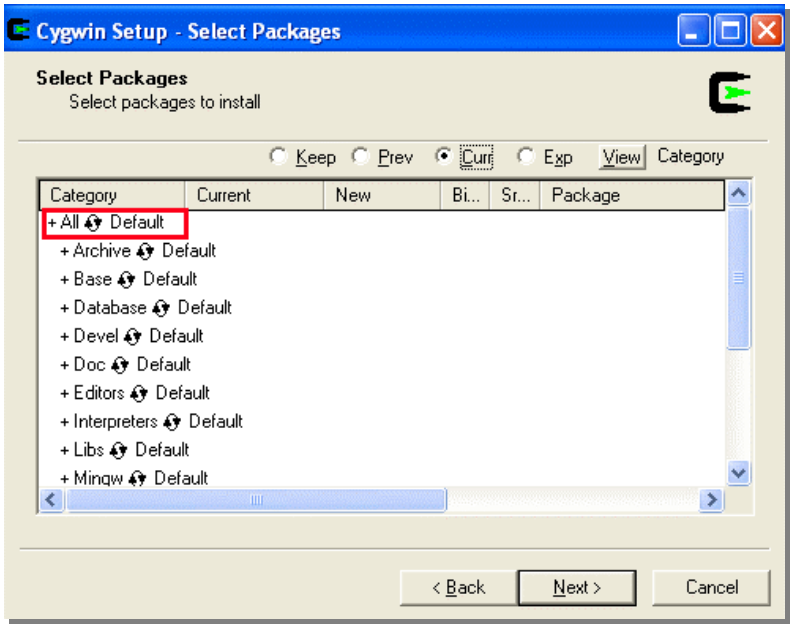


Figure 10: Cygwin Setup – Select Packages for Installation

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.

xix. The Cygwin Setup installs all the packages in the selected directory.

xx. If you want to create icons for convenient access to the Cygwin environment, select both the check boxes on the Create Icons screen. Click **Finish** to proceed.

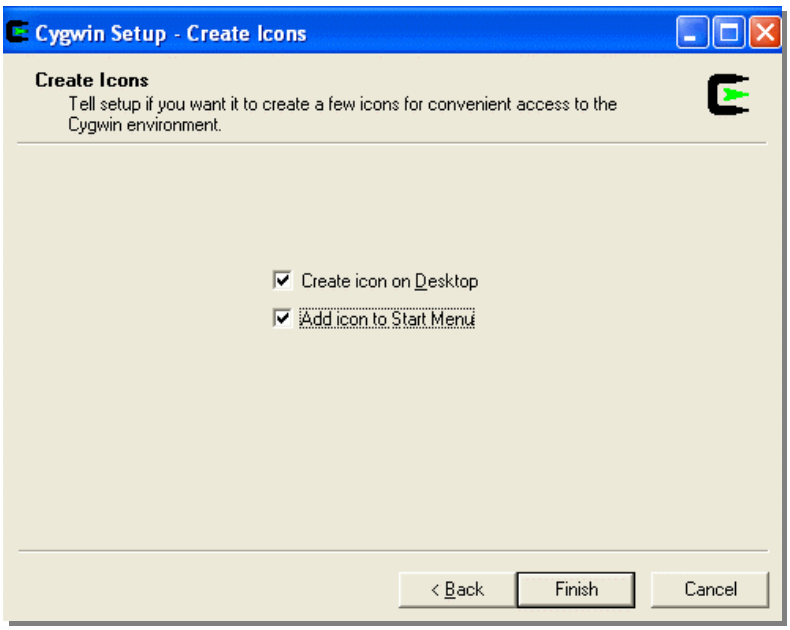


Figure 11: Cygwin Setup – Create Icons

xxi. Click **OK** to complete the process.

xxii. The Setup program copies the Open AT® Software IDE files onto the hard drive.

xxiii. Click **OK** to complete the Open AT® Software installation setup.

1.3.3.2 Installation of e-SIM RapidPLUS™ 8.1

Installing RapidPLUS™ Application

i. Run the setup.exe file from the CD-ROM provided with the Wavecom Open AT® Software IDE.

ii. It will start an install shield wizard, which would guide the user through various steps of the installation as shown in the figure below.

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.

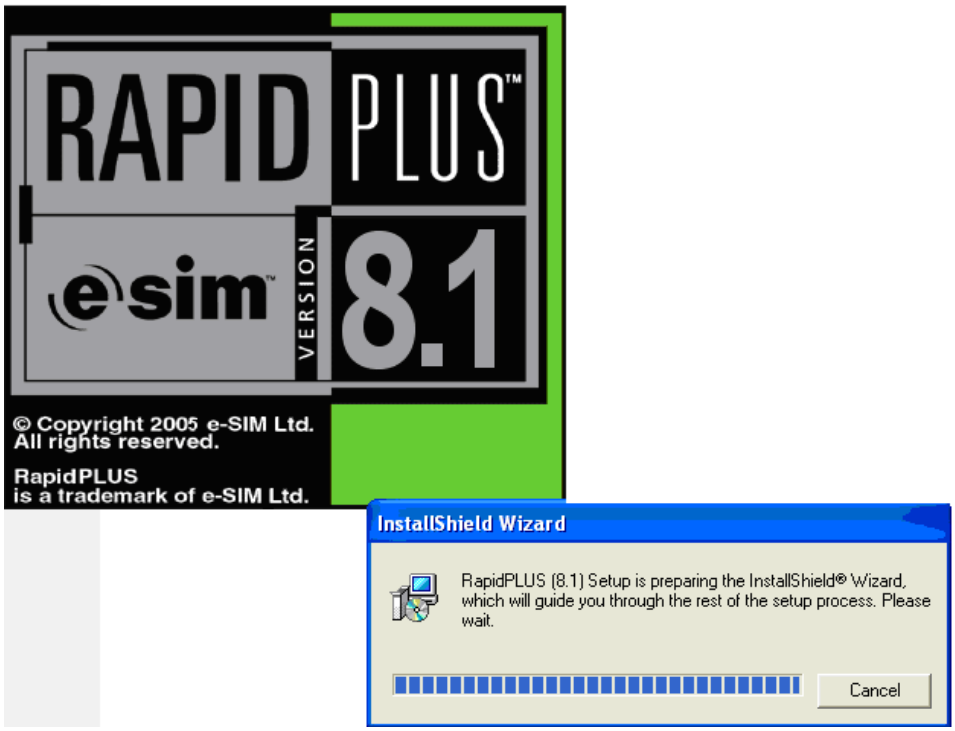


Figure 12: RapidPLUS™ Installation

iii. Once the installation has started, the setup would prompt the user to enter the product key for the RapidPLUS™. The product key is provided with the setup of RapidPLUS™ and should be entered in the text area provided as shown in the figure below.

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.

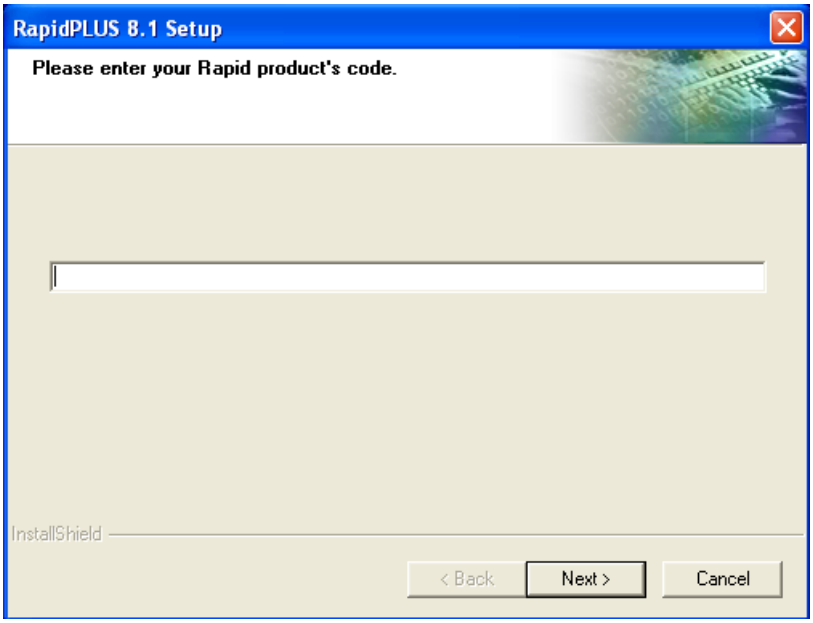


Figure 13: Dialog box to enter the Rapid product code.

iv. Once the Product key has been supplied the installation continues as shown in the next figure and is completed.

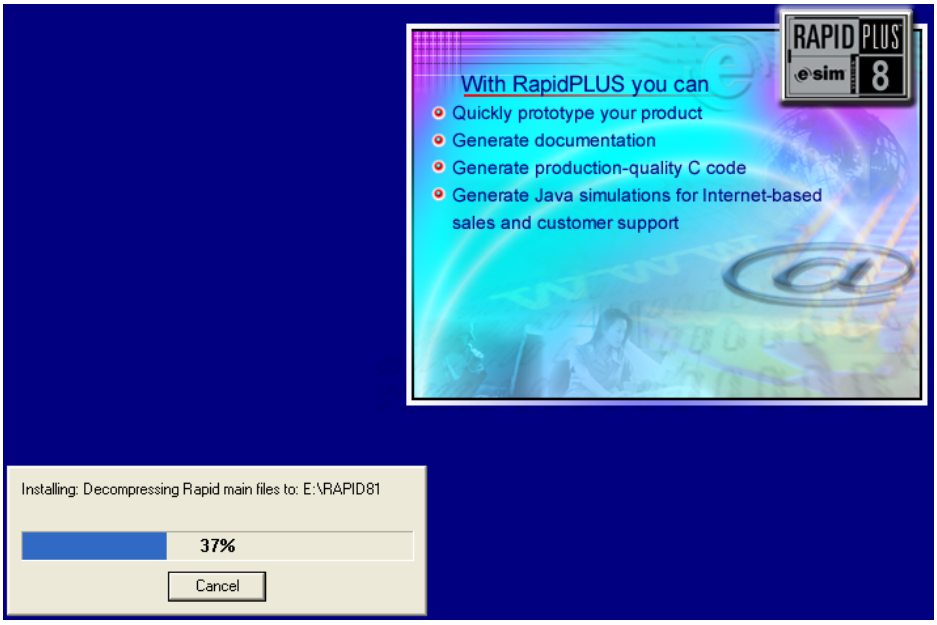


Figure 14: RapidPLUS™ installation continues

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.

Once the RapidPLUS™ Application has been installed in your system follow these steps to request the license from e-SIM.

i. Run the executable ...\RAPID81\FLEXlm\lmtools.exe. This would save the Host ID settings on a file.

ii. This file needs to be sent to e-SIM for the generation of the Rapid License.

iii. E-SIM would then be sending the license.DAT file to the user. This file needs to be copied to the ...\RAPID81\FLEXlm folder.

iv. Once the License file has been copied at the specified location, all the features of RapidPLUS™ such as code generation would be activated.

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.

2 e-SIM Development Tools

2.1 e-SIM Overview

e-SIM (www.e-Sim.com) is the leading provider of complete MMI solutions to the mobile handset industry. Incorporated in millions of handsets and vertical applications on the market and used by major companies, e-SIM's proven MMI technology enables handset manufacturers to create state-of-the-art products and bring them to market quickly and cost-effectively. e-SIM facilitates the attainment of this goal by providing the necessary elements for quick, high-quality MMI development using e-SIM's MMI Reference Design, RapidPLUS™ development and customization tools.

2.2 RapidPLUS™ 8.1

What is RapidPLUS™?

RapidPLUS™, the product of e-SIM Ltd., is a comprehensive software package for the generation of simulations and prototypes of electronic systems. Simulations developed with RapidPLUS™, can be linked with external programs to obtain the ultimate computer based training (CBT) solution. With ANSI-C code generation, RapidPLUS™ enables transformation of the virtual prototype into an executable application that can run on real embedded systems. In a typical multitasking embedded system, the generated RapidPLUS™ application, or the RapidPLUS™ task, would run the system's man-machine interface.

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.

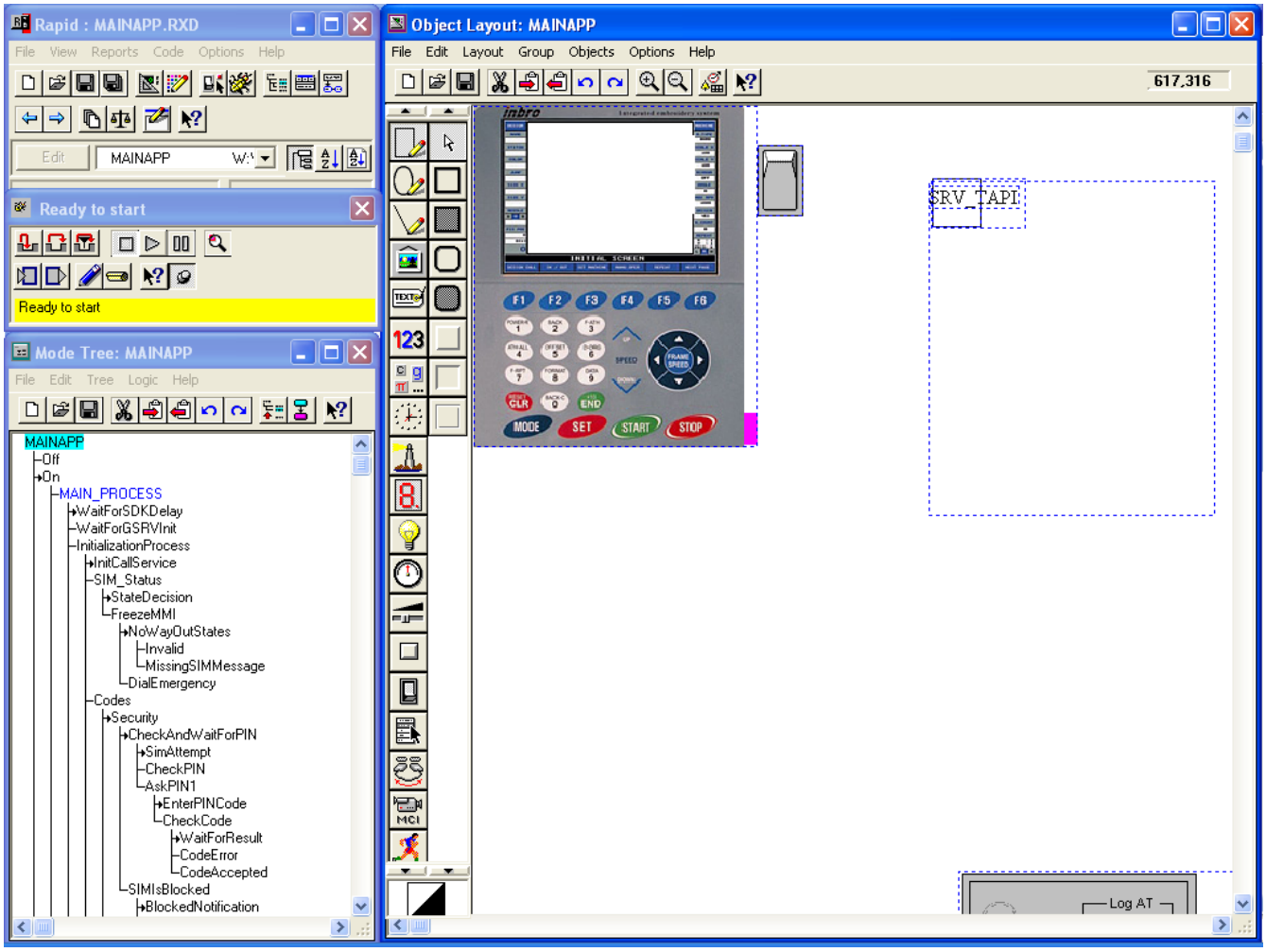


Figure 15: RapidPLUS™ 8.1 – Snapshot

In general terms, RapidPLUS™ is an integrated development environment (IDE) for the development of Man-Machine Interfaces (MMI). The primary uses of RapidPLUS™ include:

- Building simulations
- Generating documentation
- Generating ANSI-C code for embedded systems

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.

RapidPLUS™ draws its power from being based on a strong state-machine. The RapidPLUS™ language semantics may seem familiar to experienced software developers, but it is the concept of state-machine oriented development that needs getting used to. The RapidPLUS™ state-machine is defined by modes, and by transitions between these modes. The most important rule of RapidPLUS™ application development, is building a good set of modes, and defining the correct transitions between them.

RapidPLUS™ Code Generator

The RapidPLUS™ Code Generator produces a header (*.h) and a program (*.c) file for the currently loaded RapidPLUS™ application and each of its user objects. These C source code files translate the RapidPLUS™ objects, modes, transitions, triggers, and activities from their native RapidPLUS™ syntax into C data structures and functions.

RapidPLUS™ User Objects

User objects are RapidPLUS™ applications that have a clearly defined interface that can be used as encapsulated objects inside another RapidPLUS™ application (called the parent application). The parent application sees the user object like any other RapidPLUS™ object. A user object's interface to the parent application is comprised of exported properties, exported events, exported functions, and messages (structure unions). In the code generation context, the parent application represents the embedded system's RapidPLUS™ task while its user objects represent 'real-life' components or modules, such as keypads, switches, displays, or protocol stacks.

While generating the application, the objects can be looked upon at in two ways:

- User object can be generated in its entirety (including its objects and internal logic)
- User object can be generated as an interface.

In the latter case, the Code Generator ignores the objects and internal logic of the user object and generates only the user object's interface to the parent application.

User Defined Objects (UDOs)

A UDO is a discrete unit of functionality, developed as a separate module. It can be used in different applications. UDO, in a way, is a complete lower level RapidPLUS™ application, being used by a higher-level RapidPLUS™ application. It provides functions, properties, and events that can be used in the parent application and it can send messages to and receive messages from the parent application. UDOs can be reused in different applications, thus expanding the RapidPLUS™ objects library with customized objects.

User Defined Interfaces (UDIs)

An interface-only UDO, or UDI, is a UDO that is used to communicate with external modules in an embedded system. Such systems often contain hardware and software modules that need to interact with the RapidPLUS™

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.

application, such as keypad or AT module. Since these are independent modules, the RapidPLUS™ application needs to provide connection points in order to communicate with them. These connection points are known as UDIs. The UDI does not export any implementation but only the component interface when the code is generated. The resulting generated code of a UDI contains empty functions. These are the connection points to the different independent modules. During integration these empty functions are to be filled with the relevant API calls.

User Defined Data Objects (UDDs)

A data container UDO, or UDD, is essentially a UDO that only exports a message and is used for data exchange between two or more other UDOs. The UDD itself contains no objects or logic. Each one of the UDOs that exchange the data contains a holder to the UDD, through which they set the values of the different fields in the UDD's message.

2.3 Other MMI Customization tools

2.3.1 Layout Editor

As the name suggests, Layout Editor is a tool to create and modify MMI layouts. Layout Editor itself has been developed using RapidPLUS™.

Layout Editor is part of the Open AT® GTi build and can be accessed from Reference_Design\libs\ESIM\LayoutEditor\ LAYOUT_EDITOR.RPD.

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.

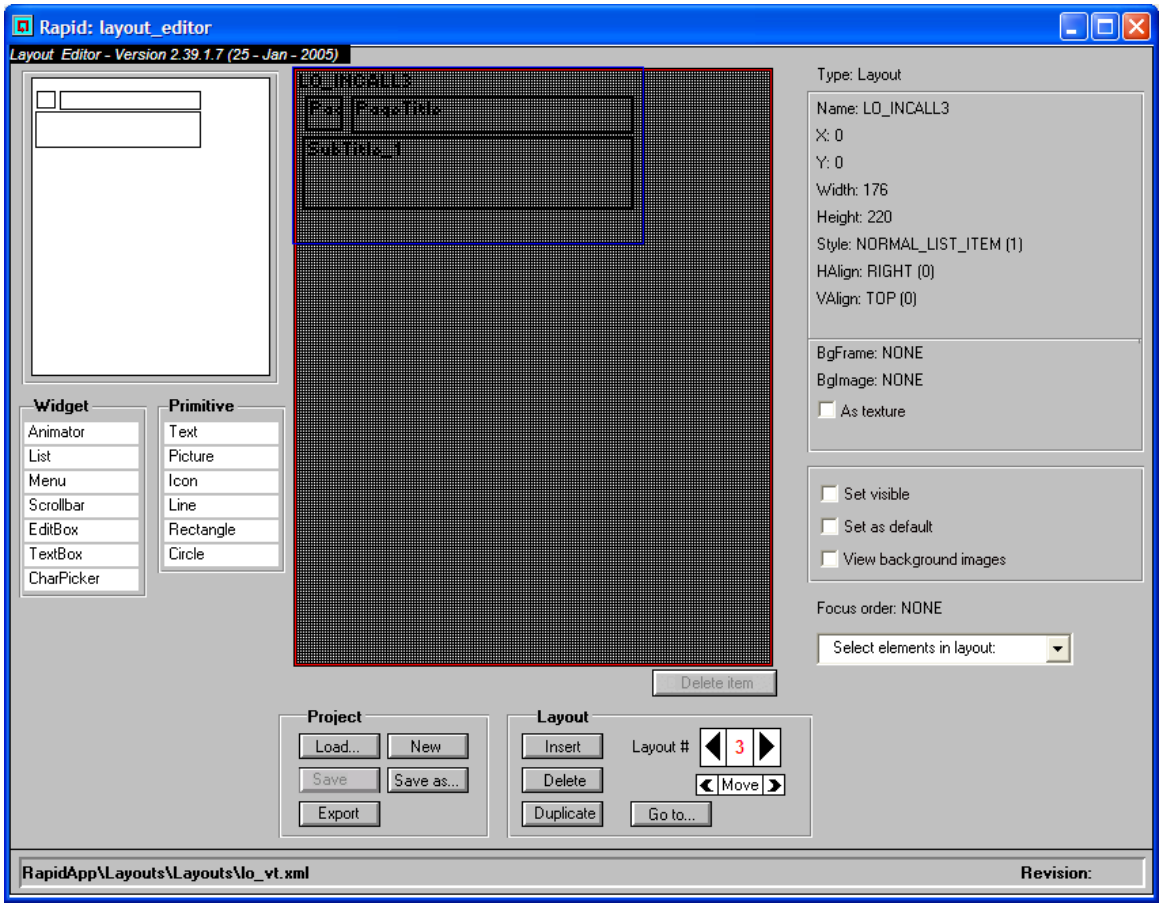


Figure 16: Layout Editor

Layout editor provides options to create new layouts as well as modify and delete existing layouts. Some predefined shapes and widgets[1] are also provided to speed up the layout development process. Layouts once developed are saved in an xml file which is later used by RapidPLUS™ for MMI development.

2.3.2 Text Resource Manager

Text resource manager is the tool provided by e-SIM for adding, deleting and modifying the text string used in the MMI. This tool is also provided along with the Open AT® GTi build.

Text resource manager is an excel file with embedded macros which are used to perform the desired operations. It can be accessed at Reference_Design\libs\ESIM\RapidApp\TextResource.xls in the Open AT® GTi build. TextResource.xls has different worksheets for different operations. The

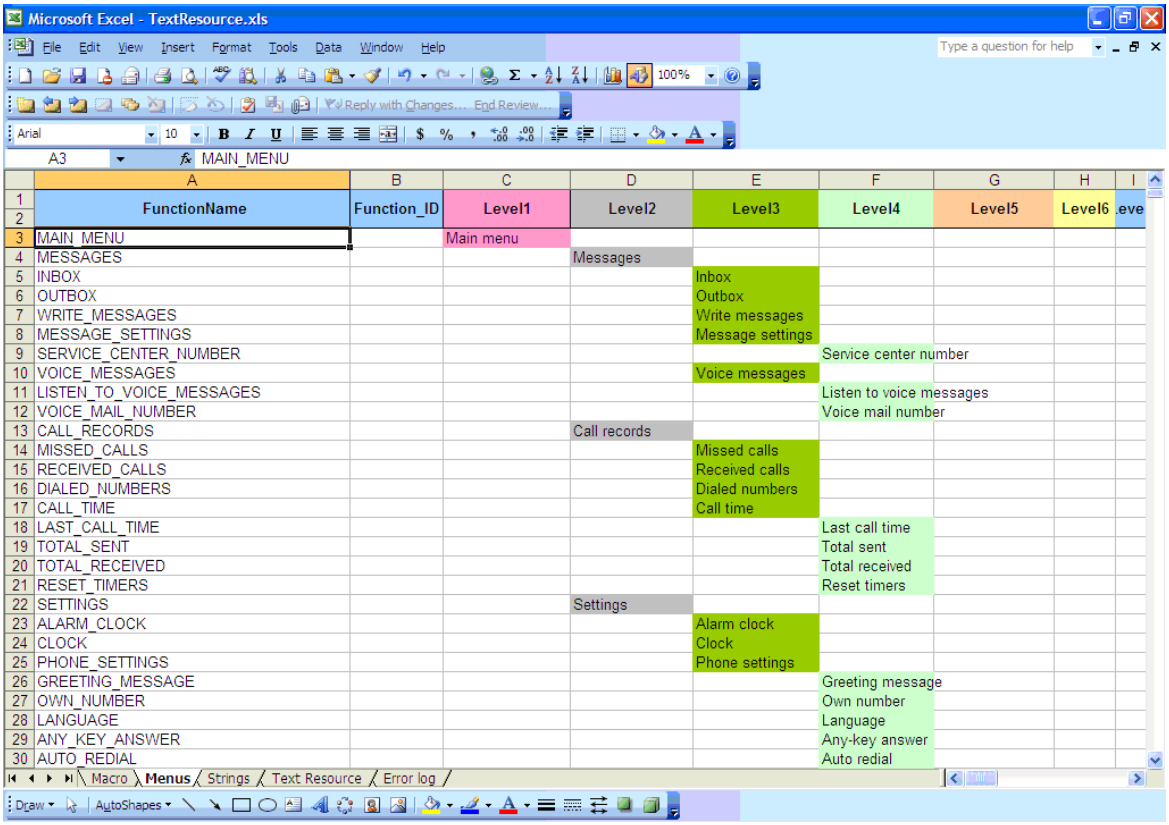
[1] Widgets are reusable MMI controls provided by e-SIM along with the reference MMI. They have been explained in detail in section 5.1.4

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.

worksheets in this excel file are: Macro, Menus, Strings, Text Resource, and Error log.

'Menus' worksheet contains the menu strings shown in the MMI in the same order and sequence as shown in the MMI menu. Level 1 column contains the highest level menu, followed by level 2 child menus and so on. Each unique menu item string gets added to the 'Text Resource' worksheet which contains all the strings used in the MMI along with its translations in all the supported languages.



	A	B	C	D	E	F	G	H	I
1	FunctionName	Function_ID	Level1	Level2	Level3	Level4	Level5	Level6	eve
3	MAIN_MENU		Main menu						
4	MESSAGES			Messages					
5	INBOX				Inbox				
6	OUTBOX				Outbox				
7	WRITE_MESSAGES				Write messages				
8	MESSAGE_SETTINGS				Message settings				
9	SERVICE_CENTER_NUMBER					Service center number			
10	VOICE_MESSAGES				Voice messages				
11	LISTEN_TO_VOICE_MESSAGES					Listen to voice messages			
12	VOICE_MAIL_NUMBER					Voice mail number			
13	CALL_RECORDS			Call records					
14	MISSED_CALLS				Missed calls				
15	RECEIVED_CALLS				Received calls				
16	DIALED_NUMBERS				Dialed numbers				
17	CALL_TIME				Call time				
18	LAST_CALL_TIME					Last call time			
19	TOTAL_SENT					Total sent			
20	TOTAL_RECEIVED					Total received			
21	RESET_TIMERS					Reset timers			
22	SETTINGS			Settings					
23	ALARM_CLOCK				Alarm clock				
24	CLOCK				Clock				
25	PHONE_SETTINGS				Phone settings				
26	GREETING_MESSAGE					Greeting message			
27	OWN_NUMBER					Own number			
28	LANGUAGE					Language			
29	ANY_KEY_ANSWER					Any-key answer			
30	AUTO_REDIAL					Auto redial			

Figure 17: Text Resource Manager - Menus

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.

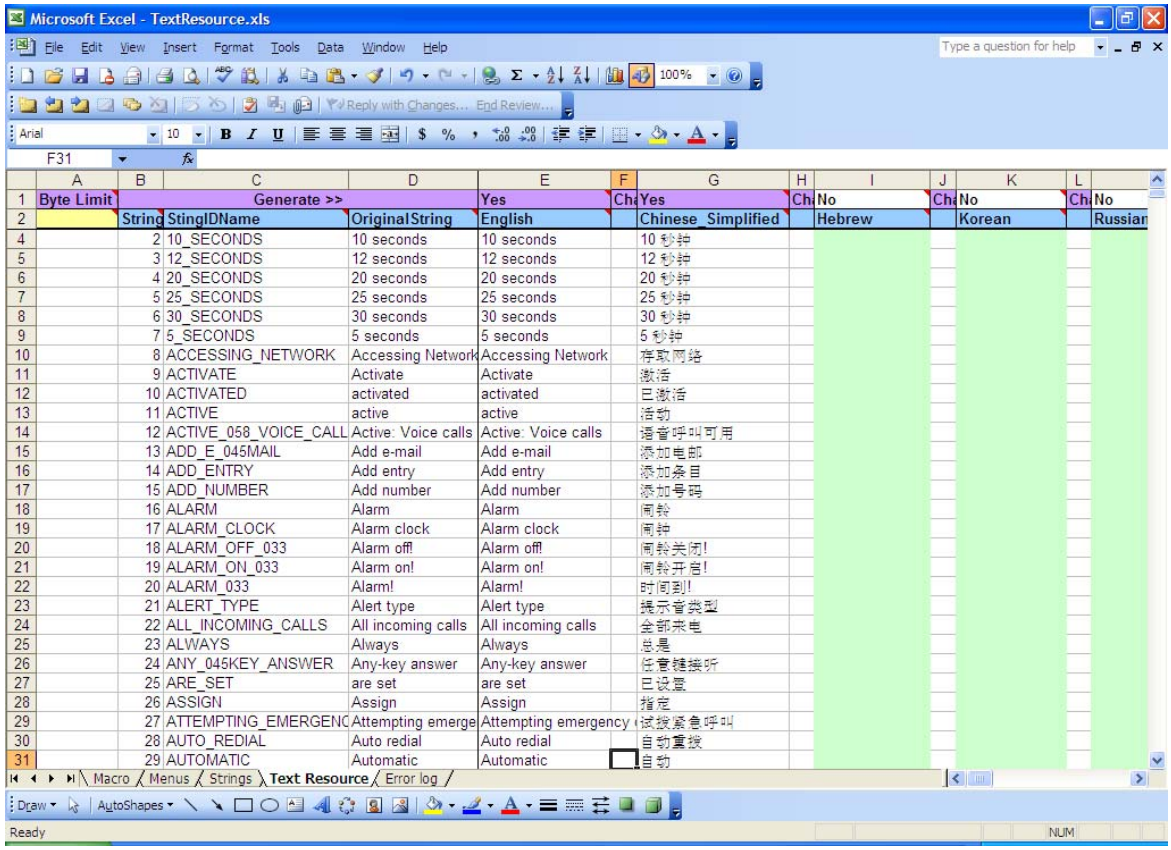


Figure 18: Text Resource Manager - Multilingual Support

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.

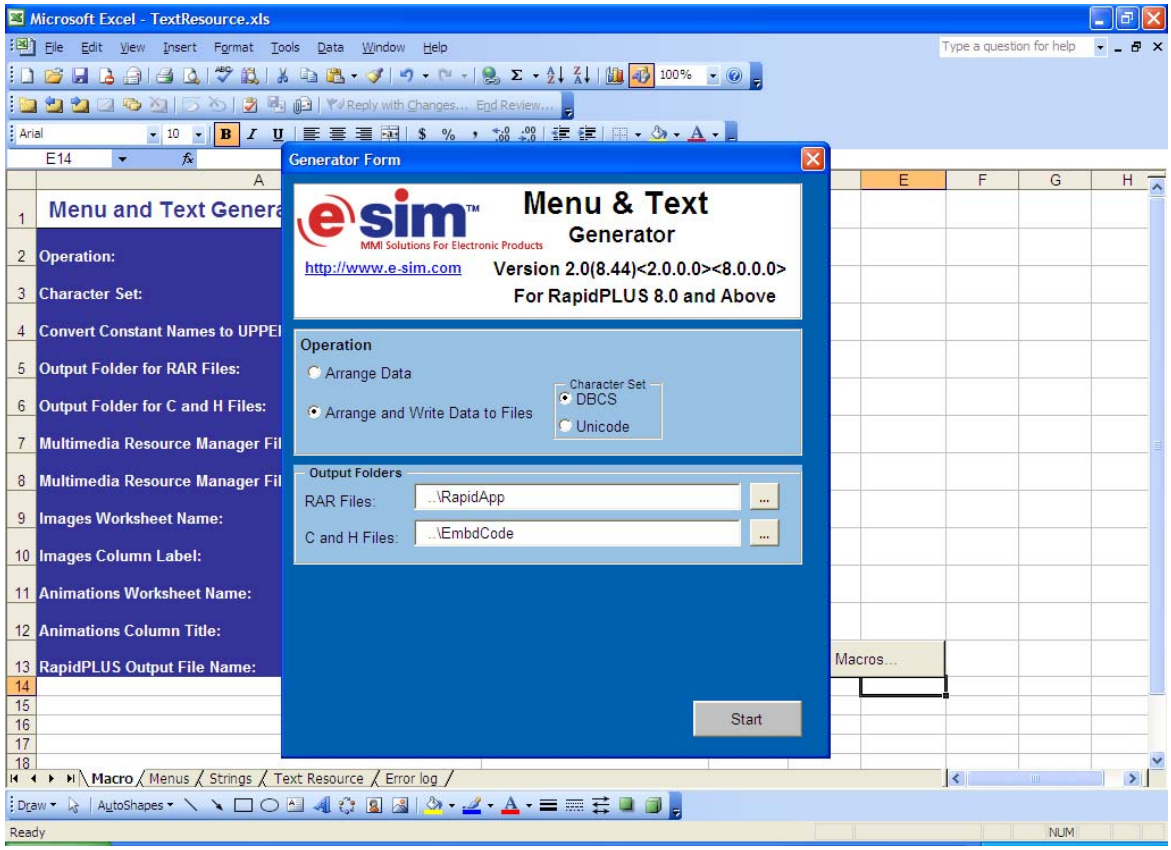


Figure 19: Text Resource Manager - Menu & Text Generator

'Macro' worksheet contains the paths for the generated output folders as well as options related to character set used for the MMI. For Open AT® GTi, DBCS character set is used. Not only is the text resource manager responsible for text strings but it is also responsible for generating the corresponding .c and .h files for the Multimedia Resource Manager as well.

Note	DBCS stands for ***d**ouble-**b**yte **c**haracter **s**et*, a character set that uses two-byte (16-bit) characters rather than one-byte (8-bit) characters. Some languages, such as Chinese, Japanese and Korean, have writing schemes with many different characters that cannot be represented with single-byte codes such as ASCII and EBCDIC. In a single-byte character set, the possible number of binary combinations is 256; the number increases to 65,536 in a double byte character set.

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.

2.3.3 Multimedia Resource Manager

Multimedia manager is a tool provided by e-SIM to manage the multimedia resources like graphics and animations. It is provided along with the Open AT® GTi build and can be accessed at Reference_Design\libs\ESIM\RapidApp\Images\MultimediaResourceManager.xls.

This tool is an MS excel file with some predefined macros to perform the needed operations. This tool provides options to add, delete, and modify the graphics elements and animation in the MMI. All the graphics used in the MMI have been added to the GraphicsResources worksheet in this excel file. The contents for Multimedia Resource Manager get translated to .c and .h files using the macros available in Text Resource Manager (TextResource.xls). Text Resource Manager has been explained in section 2.3.2. These .c and .h files are used by RapidPLUS™ for including all the graphics in the MMI.

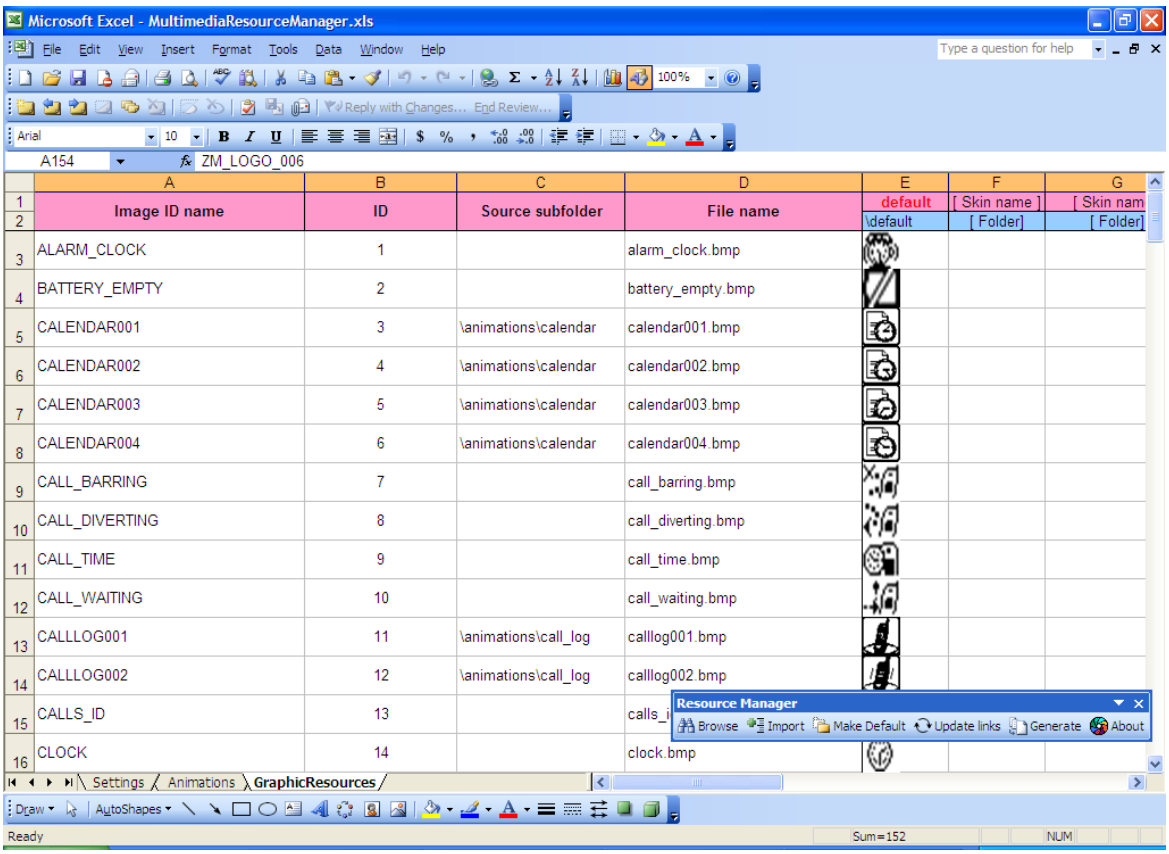


Image ID name	ID	Source subfolder	File name	default	[Skin name]	[Skin nam
				\default	[Folder]	[Folder]
ALARM_CLOCK	1		alarm_clock.bmp			
BATTERY_EMPTY	2		battery_empty.bmp			
CALENDAR001	3	\animations\calendar	calendar001.bmp			
CALENDAR002	4	\animations\calendar	calendar002.bmp			
CALENDAR003	5	\animations\calendar	calendar003.bmp			
CALENDAR004	6	\animations\calendar	calendar004.bmp			
CALL_BARRING	7		call_barring.bmp			
CALL_DIVERTING	8		call_diverting.bmp			
CALL_TIME	9		call_time.bmp			
CALL_WAITING	10		call_waiting.bmp			
CALLLOG001	11	\animations\call_log	calllog001.bmp			
CALLLOG002	12	\animations\call_log	calllog002.bmp			
CALLS_ID	13		calls_i			
CLOCK	14		clock.bmp			

Figure 20: Multimedia Resource Manager - GraphicResources

Note	Before opening up the MultimediaResourceManager.xls, it is required to run a batch file, RegisterActiveX.bat, kept in the same folder. This batch file registers the ActiveX components used by the macros in the xls.

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.

3 Wavecom Open AT® Software IDE

3.1 Open AT® Software Overview

Open AT® Software is an application development framework that helps in the design, development and execution of self-reliant and complete applications, which run on the Open AT® Firmware in the Wavecom Wireless CPUs. With Open AT® Software, wireless application developers have the flexibility and autonomy of creating innovative wireless applications that can be *embedded* on a Wavecom Wireless CPU. This application development environment provides various APIs with which you can easily create applications to perform complex tasks; for example, an automatic meter reading system.

The following image depicts a microcontroller-based application without Open AT® Software. In such a situation, all the applications are external to the Wireless CPU.

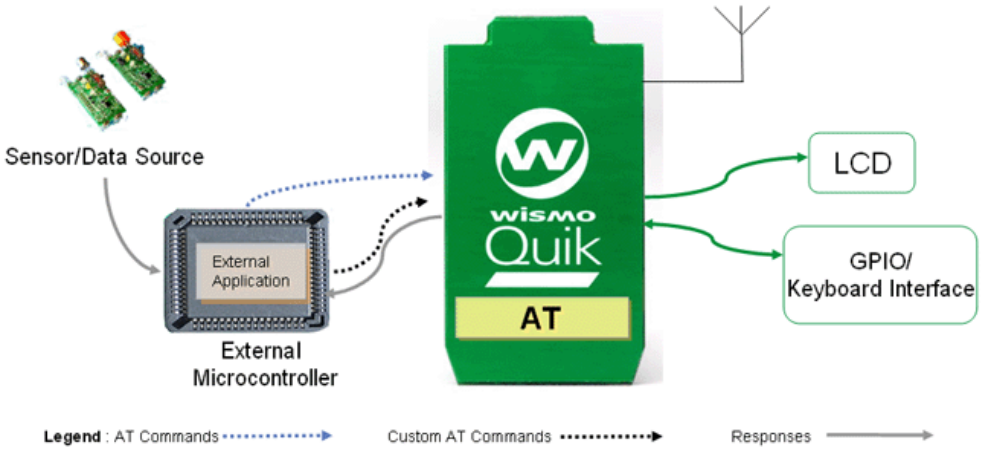


Figure 21: WITHOUT Open AT® Software

In such a scenario, without Open AT® Software, you must have interfaces for the hardware, and also develop custom applications that will be embedded in the external micro-controller to interact with WISMO and with the external devices (by GPIO or BUS interface).

With Open AT® Software, the external micro-controller is not required, and therefore the embedded application takes control of the Wireless CPU and interacts with the external devices. This is depicted in the following image.

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.

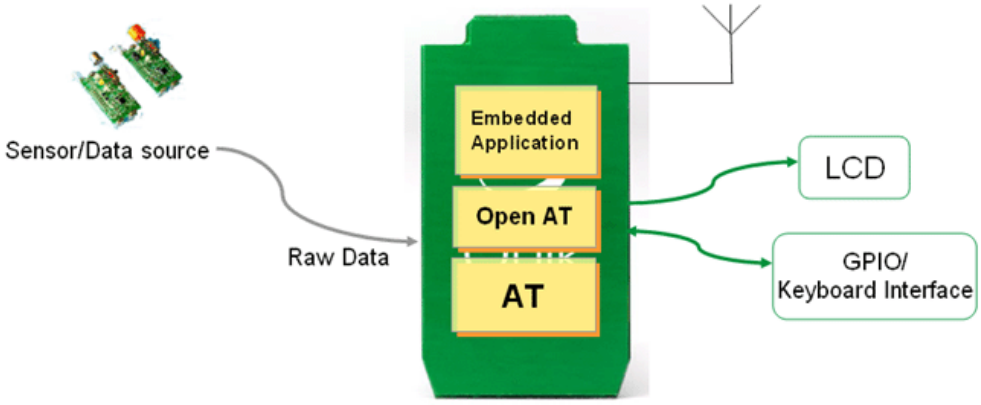


Figure 22: WITH Open AT® Software

Open AT® Software application allows you more flexibility. Open AT® Software allows you to modify the response to basic AT commands and therefore customize according to the functionality required. In addition to the basic AT commands, you can also develop custom commands and add to the available command set. The required functionality can be implemented in these commands.

This provides more power to the wireless developer and eases the programming task. The following figure depicts this advantage with a scenario in which custom commands are used.

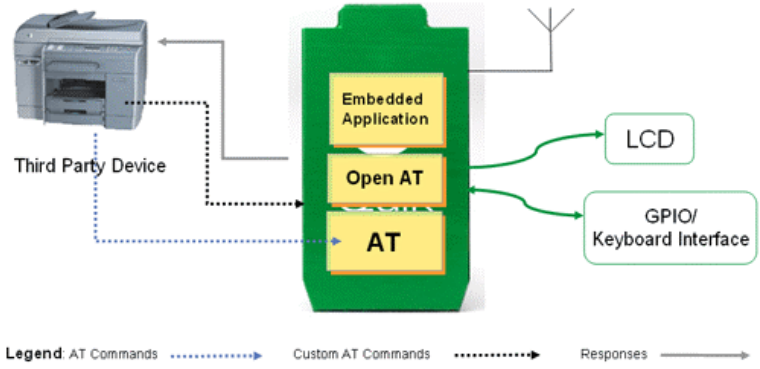


Figure 23: Open AT® Software with Custom Commands

3.2 Open AT® Software Areas of Application

Open AT® Software can provide functionality in a wide variety of applications, such as:

- Systems that use standard GSM and GPRS services for status reporting

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.

- Systems that provide internet connectivity and access to Web services like FTP, SMTP and PING
- Systems that act as TCP/UDP client/server machines and provide services to other nodes
- Systems that provide location information in automotive systems using GPS (Global Positioning System) services
- Applications that send sensor information (read through GPIOs) to remote base locations
- Applications that store sensor data or other vital information in flash (ROM) and help protect critical information for state of the art information processing systems

Domains where Open AT applications can be used include Automotive, Fleet Management, Remote Asset Monitoring, Telemetry and Telematics etc.

The Open AT® Software Automotive Application figure depicts an example of Open AT® Software usage. In case of a motor accident, the sensors fitted in the automobile send the information to the base server, which in turn informs the emergency helpline, prompting it to send help to the accident site. The Open AT® Software application running on the Wavecom Wireless CPU handles this process of collecting and sending the information to the base server.

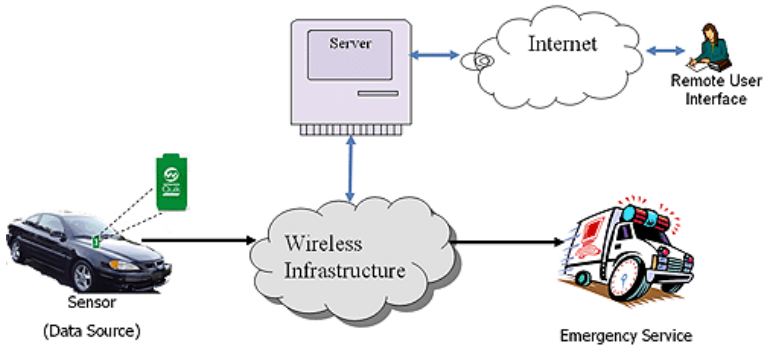


Figure 24: Open AT® Software Automotive Application

3.3 Advantages of using Open AT® Software

Following are the advantages of using Open AT® Software:

- The Wavecom Open AT® Software library helps reducing the development and testing time considerably. You can concentrate on the functionalities of the application instead of the lower level technical details.

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.

- Open AT® Software reduces the cost of the product. Without Open AT® Software, a considerable amount of interfacing hardware and drivers are used to provide external stimuli (information) to the Wireless CPU. But Open AT® Software provides APIs with which the power of the Wireless CPU can be easily manipulated without any external interfacing hardware or software.
- Open AT® Software application is tightly integrated with the core AT software. As the Wavecom Open AT® Software library (on which the application is built) is extensively tested to work with the core AT software, the level of integration and reliability of the embedded application meets user expectations. With Wavecom Open AT® Software library the embedded application can interact efficiently with the core software.
- Open AT® Software application allows you more flexibility; in addition to the basic AT commands, you can also develop custom commands which can add to the available command set. The required functionality can be provided in these commands. Hence, the overall flexibility of programming can be increased.

3.4 Open AT® Software ADL Library

Application Development Library (ADL) was introduced in the V2 release of Open AT® Software. This library counteracts the drawbacks of the previous versions of Open AT Software and is highly recommended for all kinds of applications. ADL mode has the following advantages:

- High-level event processing interface
- Highly modular capabilities
- Plug-in libraries are available only for ADL interfaces (like TCPIP library-eDlib)
- Event processing using user-defined callback functions
- Easily maintainable code
- High ease of programming

Note	Open AT® GTi is available only over the ADL running on Open AT® Operating System.

3.5 Mandatory Embedded Application Code

To develop an Open AT® Software application certain mandatory APIs must be implemented. The following sections describe the mandatory ADL APIs.

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.

3.5.1 ADL API

To define the application call stack size follow the prototype below:

```
u32 wm_apmCustomStack [256];     /* 256 is an example */
const u16 wm_apmCustomStackSize = sizeof (wm_apmCustomStack);
```

The entry point function that is called during the application initialization is:

```
void adl_main ( adl_InitType_e   InitType )
{
}
```

3.6 Open AT® Software Running Modes

There are two execution modes for any Open AT® Software application. The following section describes the target and remote modes of execution.

3.6.1 Target Mode

The application is actually downloaded in the Wireless CPU (programmed in Flash) and executed in the target mode. The application uses the resources of the Wireless CPU to perform the required operation. This mode is used when a standalone application must be deployed in the Wireless CPU. The application is self-reliant and can run without user intervention.

3.6.2 Remote Mode

In the remote mode, the application is executed on the development environment (PC) and uses its own resources like memory and clock. This mode provides the facility of line-by-line debugging. This mode can be used to debug the application by inserting various breakpoints and traces. Dynamic changes in the contents of the memory can be viewed using the tools available in Visual Studio 6 or Visual Studio .NET.

Wavecom provides the RTE (Remote Task Execution) tool (explained in section 4.4) to debug the applications in remote mode.

The following chapters of this document provide more information on the target and remote modes of application development.

3.7 Open AT® Software application size restrictions

The size of the Open AT® Software application image after the build should not exceed the following limits as described for the following Wavecom OS and CPUs.

OS 6.56

ROM: up to 1.5 MB for Open AT ® 3.10

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.

RAM: 128 KB

OS 6.60

ROM: up to 1.5 MB for Open AT ® 4.0

RAM: 256 KB

Since Q2406B Wavecom Wireless CPUs have the version 6.56 OS installed, these CPUs support 128 KB RAM. Whereas Q2686H CPUs with the version 6.60 OS support 256 KB RAM.

The RAM allocated to the application in the Wireless CPU is not only for the application but is also shared with the ADL library and the plug-in (Internet, TCP/IP) if any.

Note	If the Open AT® GTi feature has not been activated on the Wireless CPU, the Open AT® GTi application will not run.

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.

4 Wavecom Development Tools

4.1 SELIMA

The Serial Link Manager (SELIMA) is a tool used to communicate between the PC and Wavecom Wireless CPU. SELIMA acts as a mediator for the different applications on the PC, talking to the Wavecom Wireless CPU through the serial port (COM1).

SELIMA dispatches the data received from the serial link to the Wavecom PC applications (TMT, TE, and RTE). It also multiplexes the frames sent by these applications and forwards them to the Wireless CPU.

The Serial Link Manager is launched when you start the Target Monitoring Tool (TMT), the Terminal Emulator or the Remote Application. This opens the serial port, and the SELIMA icon is displayed on the task bar. Click the **SELIMA** icon on the task bar; proceed to configure the serial port settings.

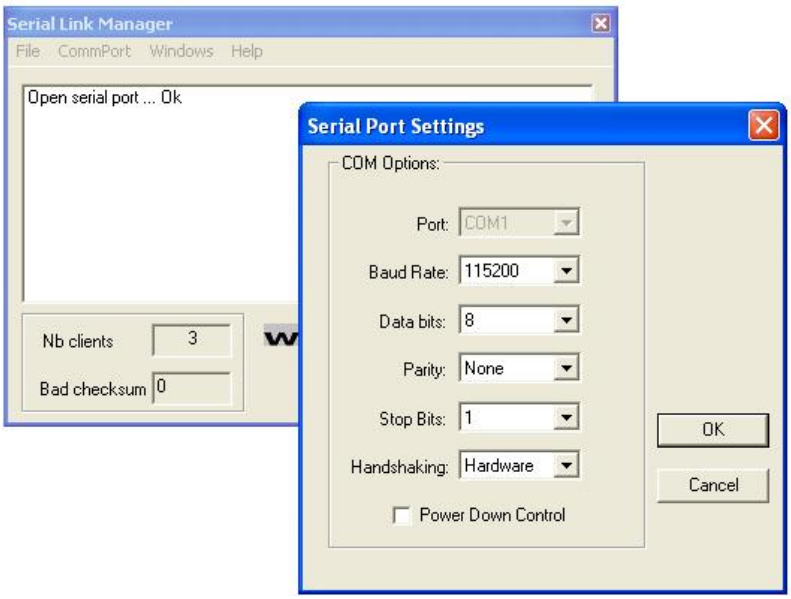


Figure 25: Serial Link Manager Port Settings

Use the Comport option in main menu to manage the serial port. The serial port settings are shown in the Serial Link Manager Port Settings image.

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.

4.2 Target Monitoring Tool

The Target Monitoring Tool (TMT) is used to display the application's trace messages from the target or the Remote Application. To use the TMT, follow the steps mentioned below:

i. Switch on the Wireless CPU. Ensure the target is not in the initialization stage.

ii. Make sure that the selected COM port is not in use by any other application.

iii. Select the Start → Wavecom → Development Toolkit → Target Monitoring Tool shortcut icon.

iv. Check the PC serial link settings. Select the Preferences > Settings menu option. Select the COM tab.

v. Click the **Com** tab. Check default mode parameters. The default values are shown in the Default PC Serial Link Settings image.

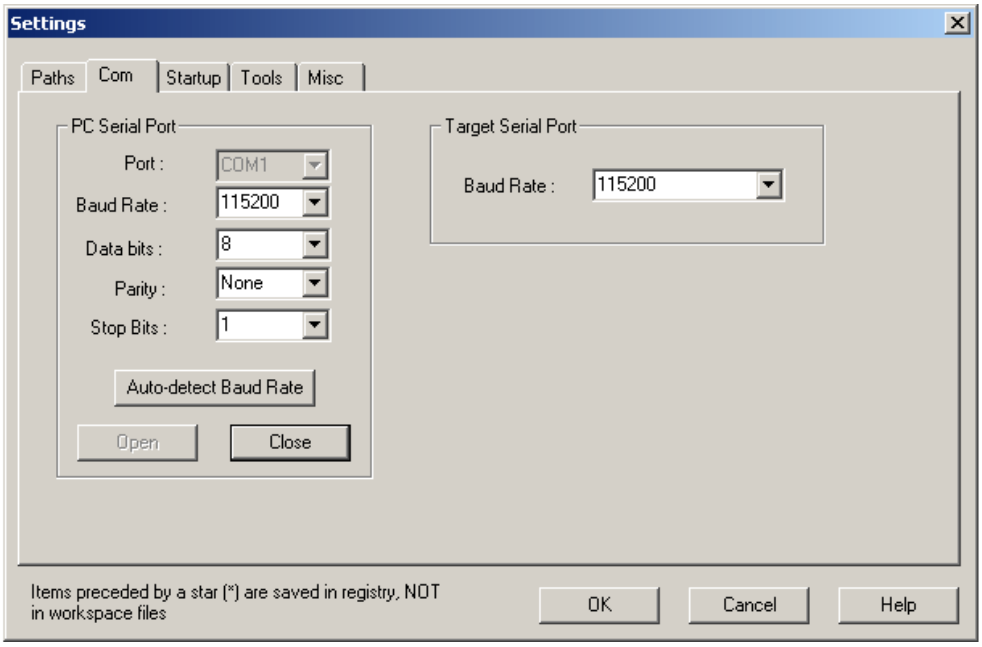


Figure 26: Default PC Serial Link Settings

vi. Select the Commands → Init Target menu option to initialize the target in the remote mode.

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.

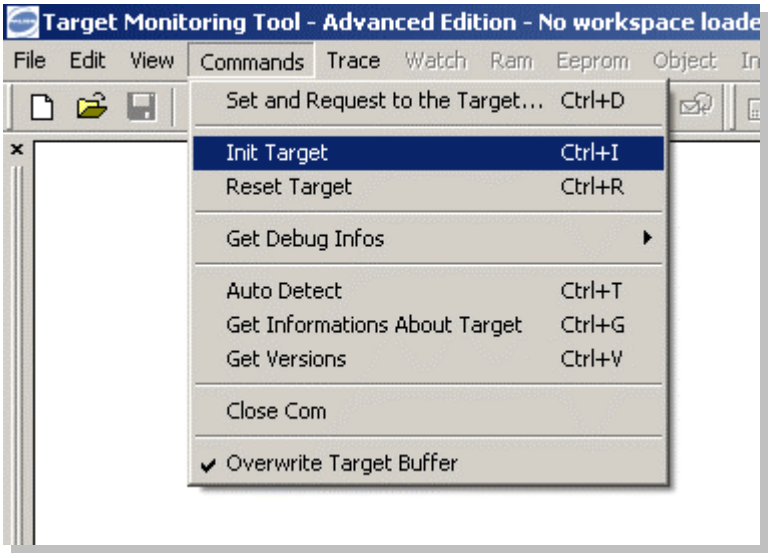


Figure 27: Init Target Command Window

vii. To load the workspace, select the File → Open Workspace menu option.

viii. Run the Auto Detect command to get the same baud rate on your PC and on your target (typically: 115200 bps or 9600 bps).

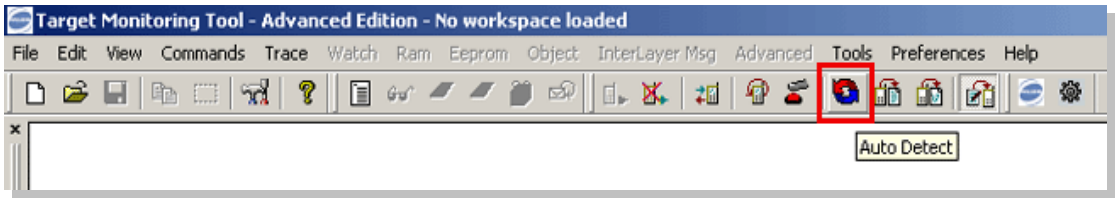


Figure 28: The Auto Detect Icon

ix. The TMT will send frames at different successive baud rates when you select the Auto Detect option. If the target sends a frame back, it implies that the target speed and computer speed match.

x. If the Auto Detect operation is successful, the values in the Baud Rate and Target Speed combo boxes in the Com tab in the Settings dialog box must be the same.

xi. Select the Trace → Set the Trace levels menu options to set the trace levels.

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.

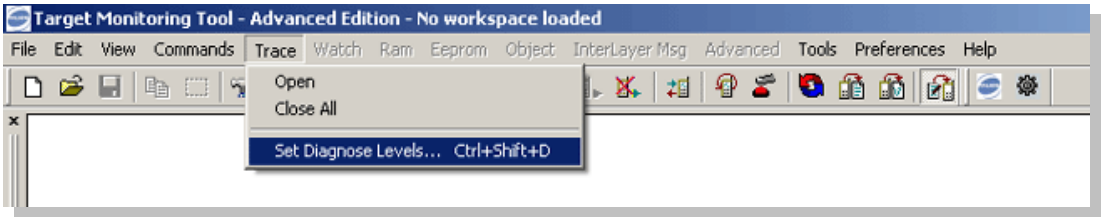


Figure 29: Set Trace Level

xii. The Set and Request to the target window is displayed, from where you can enable the traces. Select the trace parameter and click **Send Level**.

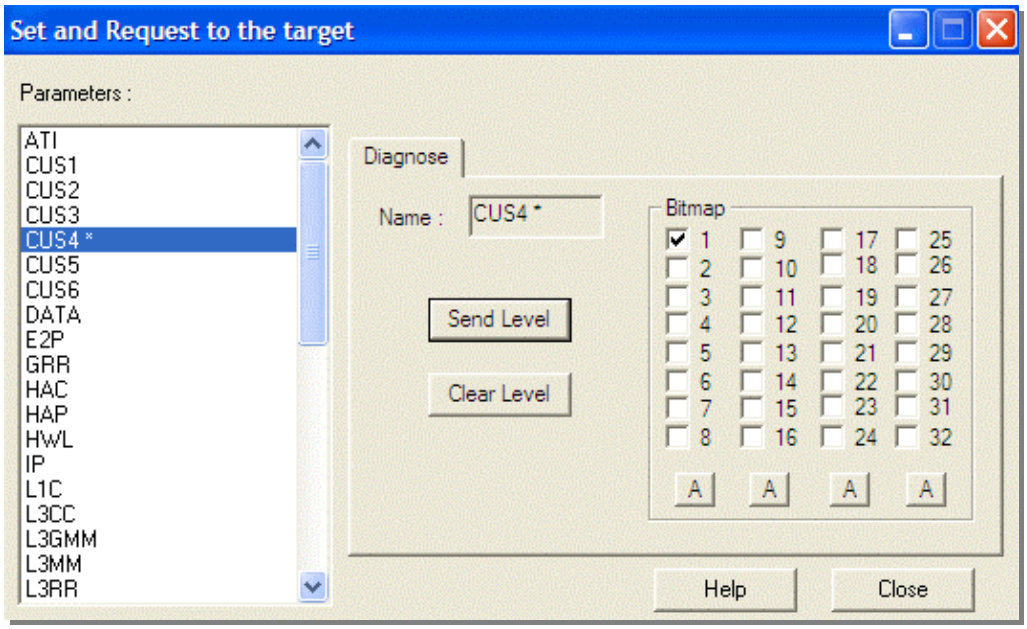


Figure 30: Set and Request to the Target Window

xiii. To check communication with the target:

a. Select the Terminal Emulator option on the TMT toolbar. Select the AT1 window.

b. Type AT and press Enter on your keyboard. A confirmation response must be displayed, otherwise, select the Commands → Init Target menu option (TMT toolbar) and type AT again.

c. Close the Terminal Emulator window.

xiv. Select the Traces → Open menu option in TMT to open the trace window.

xv. Run the application. The required trace levels are displayed in the Target Monitoring window.

xvi. The application can be run in the Debug or Target mode.

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.

xvii. To run the application in the remote mode (Remote Task Environment):

a. Run the application.

b. Select the CUS trace levels from the Remote Task Monitor.

c. Click **Start** on the Remote Task Monitor to start the application.

xviii. To run the application in the target mode:

a. Download the application to the Wireless CPU using the HyperTerminal.

b. Follow the directions given in step 11 to set the trace levels.

c. Run the application using the AT+WOPEN command (AT+WOPEN=1).

4.3 Terminal Emulator

The Terminal Emulator is an AT command terminal. It is used to send and receive string commands to or from a customer task (target or Remote mode).

The terminal emulator connects an external application using the standard V.24 protocol.

The Terminal Emulator can be launched one of the following ways:

i. If the TMT is running, start the Terminal Emulator using the corresponding button in the toolbar. The appropriate toolbar must be selected from View option in main menu.

ii. Select the Start → Wavecom → Development Toolkit → Terminal Emulator shortcut.

The Terminal Emulator provides two windows: one for AT commands, and one for V24 Data flow. For more details refer to the online help of the Terminal Emulator as shown in the figure below.

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.

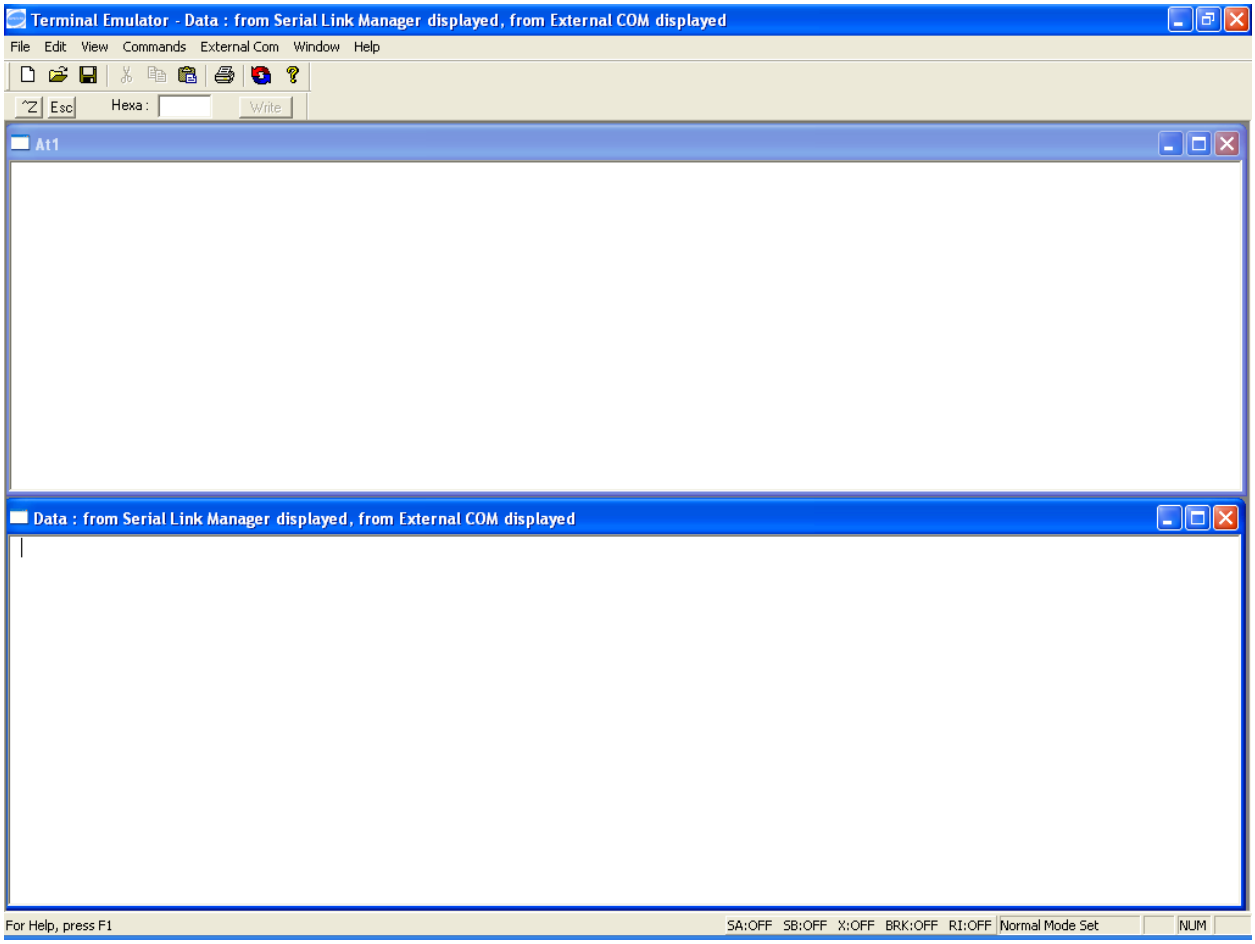


Figure 31: Terminal Emulator

4.4 Debugger RTE based on VC++

In the remote mode, the application is executed on the development environment (PC) and uses its own resources such as the memory and clock. This mode, as the name indicates, is used to debug applications.

The application runs using the Real Time Execution Environment (RTE) which integrates itself with Visual Studio 6 or Visual Studio .NET during the installation of Open AT® Software. This mode can be used to debug the application by inserting various breakpoints and traces. Moreover, dynamic changes in the contents of the memory can also be seen using the tools available in Visual Studio 6.00 or Visual Studio .NET. This mode can be used during the development of the application before making the final binary (which will be downloaded to the Wireless CPU).

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.

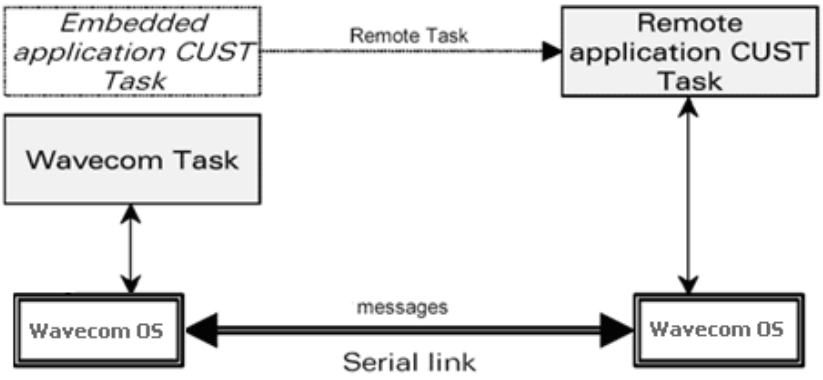


Figure 32: Remote Task Environment

The RTE is used to send the instructions directly to the Wireless CPU, which then executes these instructions and performs the required operation. The Remote Task Execution (Remote Mode) image shows a screenshot of the Remote Task Monitor which is displayed when the Open AT® Software program is run in the remote mode.

Before starting the remote application, set the appropriate CUS level traces from the Remote Task Monitor.

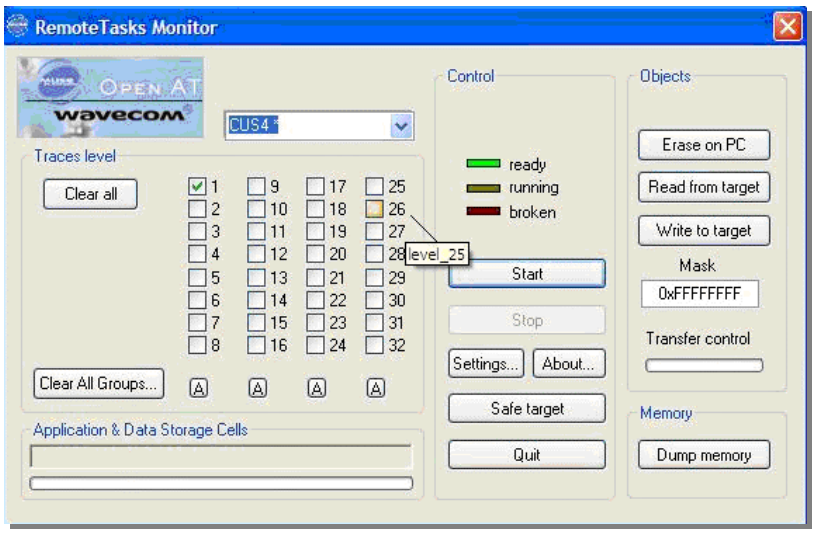


Figure 33: Remote Task Execution (Remote Mode)

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.

5 Wavecom Sample MMI

5.1 Sample MMI Architecture

Wavecom provides the sample MMI implementation that has been developed using Wavecom and e-SIM MMI development tools. Wavecom sample MMI is an embedded Open AT® Software application running over Wavecom Open AT® Software ADL interface. Once integrated over Open AT® Software, ADL takes cares of connecting with the underlying protocol stack.

Wavecom MMI can be used as reference for development of customized MMI over Wavecom Open AT® GTi architecture.

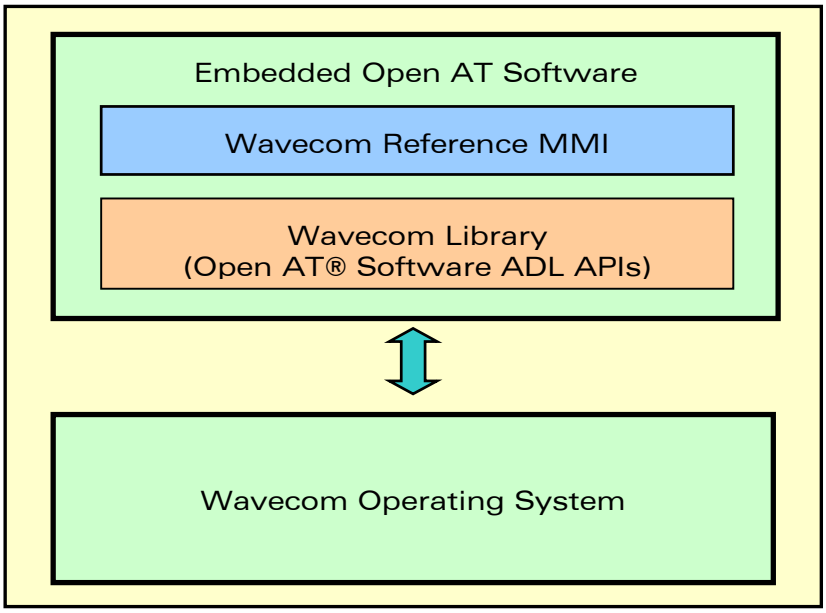


Figure 34: Sample MMI integration with Wavecom Open AT® Software

Wavecom sample MMI consists of three main layers. The layers move from high-level functionality (layer 1) to low-level functionality (layer 3). The main application, running as an embedded AT application, manages the entire project by activating and deactivating the various components according to priorities. It is the umbrella application that contains all the other components.

The various types of functional components are:

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.

- MMI Modules
- Service Components
- Data Containers
- Widgets
- Embedded Interface Component

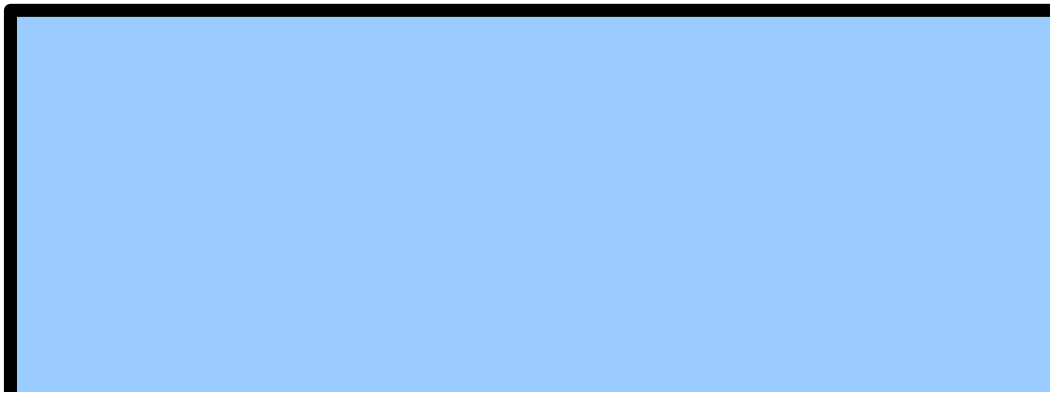


Figure 35: Wavecom Sample MMI Design

5.1.1 MMI Modules

i. Control the main functionalities of the reference design

ii. Handle the MMI: screens, menus, texts, and soft keys.

Only one MMI module can be active at a time and it is responsible for the displays on the screen.

There are two hierarchical levels of MMI modules.

- WMI modules: Reusable MMI components that can control widgets, services, and embedded components.
- HMI modules: Each module handles a subsection of the project MMI.

WMI modules provided along with the Wavecom MMI are:

- WMI_ADD_ENTRY - Encapsulates the adding entry logic. It recognizes if there is already a name similar to the one to add, and asks to replace.

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.

- WMI_EDITOR - Collects the user numeric key inputs and builds text to be used by the phone, to dial, to send data or to create records of phonebook entries. It uses the edit box widget (WDG_EDIT_BOX) to draw text on the display and the character picker widget (WDG_CHARPICKER) to select special characters.
- WMI_MENU - Implements navigation in the menu tree. It is responsible for opening the correct layout, presenting the soft keys, titles, icons, and status strings. It has two instances in the project: WMI_MENU and WMI_MENU_PRIVATE.
- WMI_MENU_PRIVATE – Only HMI_MENU uses the WMI_MENU_PRIVATE instance as a private instance. This is done to separate between navigation on the main menu tree and navigation in small menus, which are required when implementing functions. This separation is required in order to enable resuming from function implementation to the main menu.
- WMI_PB_BROWSE – Serves as a browser in the active phone book. It receives an index of a record and starts browsing from that point. Pressing the alphanumeric keys may cause a search for matching name and may change the list if a match is found.
- WMI_PB_DETAIL – Receives an index of an entry in the current phone book, and displays its details (phone numbers, email names, etc.).
- WMI_POPUP – Supplies popup functionality for the entire project.

HMI modules provided with Wavecom MMI are:

- HMI_CALL – Manages dialing, outgoing calls, active calls, conference calls, on hold calls and swapping calls.
- HMI_IDLE – Handles the activity of phone in idle mode e.g. missed call, new message and time & date information.
- HMI_MENU – Handles menus starting at the main menu. This module manipulates all leaf functions.
- HMI_PBOOK – Handles user interface for phonebook functionality. It can be started in two modes – one for browsing through phonebook and other for opening phonebook menu.
- HMI_REC – Handles user interface for call records functionality. It uses UDD_REC to retrieve data.
- HMI_SMS – Implements the MMI for the messages menu as defined in the file, TextResources.xls.

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.

5.1.2 Service Components

i. Provide specialized functionality to the MMI modules.

ii. Create a flexible buffer between the MMI modules and embedded interface layer.

iii. Manage the project's data and state like phone book list, messages list, active calls, and call records using data containers called UDD. UDD's have been discussed in details later in the document.

iv. Active throughout the lifetime of the project.

Service components provided with Wavecom MMI are:

- SRV_CALL – Implements call functionality. Uses SRV_TAPI to interact with the network.
- SRV_PBOOK – This service component provides access to the various phone books of the device.
- SRV_TAPI – Provides AT implementation for call and many other network related operations. Uses SRV_AT to send the AT commands.
- SRV_SMS – Provides AT implementation for SMS functionality. Uses SRV_AT to send the AT commands.
- SRV_REC – Maintains the call records. Uses EMB_GSRV for all reading and writing call records to persistent storage.
- SRV_KPD – Provides keyboard service to MMI. Uses EMB_KPD for interacting with platform keypad.
- SRV_AT – All other services use this module to send AT command and receive responses. SRV_AT uses EMB_AT to communicate with the platform.
- SRV_PDU – This is used by SRV_SMS to decode and encode SMSs in PDU format.
- SRV_LAYMAN – This service component is responsible for arranging the layouts on the screen.
- SRV_PLOTTER – SRV_LAYMAN uses SRV_PLOTTER to create layouts on the display.
- SRV_DISPLAY – This component is responsible for actually drawing the display elements on the screen.
- SRV_BITMAPS – This service component manages all the bitmaps used in the MMI.
- SRV_ICON – This is the drawing component of the status icons mainly visible in idle screen.

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.

5.1.3 Data Containers

A data container UDO, or UDD, is used for data exchange between two or more other UDOs. It contains no objects or logic.

UDOs that use UDD, contain a holder to the UDD, through which they set the values of the different fields in the UDD's message.

UDDs used in the Wavecom sample MMI are:

- UDD_SMSDATA – Used for data exchange between HMI_SMS and SRV_SMS.
- UDD_CALLS – Holds the data of all active calls in its message interface. It is updated only by SRV_CALL.
- UDD_NET – It holds data of all available operators in a network.
- UDD_REC – It holds call time and duration.
- UDD_SET – It holds user settings.
- UDD_SMS_ILIST – It holds maximum possible No. of SMS and the count of SMS actually stored in list.
- UDD_PDU – Used for data exchange between SRV_SMS and SRV_PDU. It also contains buffers, in which SRV_PDU keeps the temporary results of encoding and decoding PDU strings.

5.1.4 Widgets

i. Provide basic user interface functionality, such as scroll bar and editing functionality.

ii. Easily reusable

Some of the frequently used widgets in the sample MMI are:

- WDG_ANIMATOR – This component provides animation services for the entire project. The animation data is retrieved using a nested component, ANIM_DATA.UDO. Currently the timing of the animation is hard coded in this widget.
- WDG_SOFTKEY – This component handles the two soft keys. It receives the labels to draw and triggers exported events when those keys are pressed. It is the only component that listens to soft key events from the keypad. All other components listen to this widget's events when dealing with soft keys.
- WDG_SCROLLBAR – This component provides two types of scrolling behavior; Scrolls the main menu and page mode lists (bitmap) and scrolls simple lists (text). The first mode displays a scroll bar element on the display and calculates its position according to the displayed list item. The second mode displays the number of displayed list items out of the total number of items in the list.

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.

- WDG_LIST – This component provides list navigation functionality for other components in the project. It is initialized with items to display. The number of items it receives depends on the number of display items that are defined in the Layout Editor. Therefore, if the total number of list items is larger than the number of display items, the owner would need to initialize the list with the next items when it is required to do so.
- WDG_EDIT_BOX – This component draws text on the display inside the widget rectangle. It is used mainly by the editor (WMI_EDITOR) and also by other MMIs that use it directly. WDG_EDIT_BOX has two drawing orientations; from bottom right or from top left. It can draw different cursor types. It also has a special view-only mode. Each time the drawText function is called, the entire text is drawn.

5.1.5 Embedded Interface Components

i. Provide an interface between the RapidPLUS™ components and the functions supplied by Wavecom

ii. There is usually (but not necessarily) a one-to-one relationship to functions with similar names to the IDE functions they call.

iii. Any part of the embedded system, hardware or software that communicates with the RapidPLUS™ application must be represented by an embedded interface component.

The embedded components provided in the Wavecom MMI are:

- EMB_KPD – Provides interface methods to communicate with the platform keypad.
- EMB_AT – Provides interface to send AT commands to the underlying AT modem provided by the platform and receive back the responses.
- EMB_GSRV – Provides interface for general services like battery, audio, backlight, call records, charger etc.
- EMB_TEXTRES – Manages different language text strings used in the MMI.

5.2 MMI Integration over Wavecom Open AT® Software

5.2.1 Basic Services Integration

This involves integration of the following components:

- Display
- Keypad
- Timer

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.

5.2.1.1 Display

The display components in the MMI layer of the sample MMI architecture are responsible for handling the screen display.

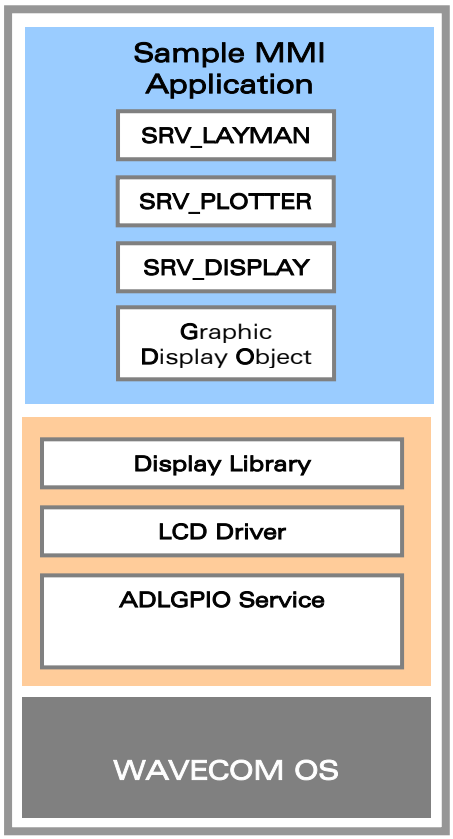


Figure 36: Display Integration

Each screen shot in the MMI conforms to one of the pre-created layouts. Layouts development tool 'Layout Editor' can be used to design and edit the layout elements used in the MMI. The HMI module that is active is responsible for managing and updating the display. All the HMI modules use the services from SRV_LAYMAN to define the components and their appearance on the screen (size, position, style, etc.). For every layout that exists in the MMI, there exists an *initLO_<LayoutName>* method in SRV_LAYMAN. MMI calls a suitable method to display a specific layout.

VideoDriver_BitBlit method is responsible for the integration for RapidPLUS™ GDO with the platform LCD. Platform LCD device driver is implemented using Open AT® Software ADL API. ADL GPIO Service interface and ADL Bus Service interface is used to implement the target display driver.

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.

Every time the display changes, MMI application using the service layer components updates the display buffer maintained by RapidPLUS™ GDO. RapidPLUS™ GDO then calls *VideoDriver_BitBlit* to update the platform display.

5.2.1.2 Keypad

All the MMI components use services of SRV_KEYPAD to provide the keypad functionality. SRV_KEYPAD provides a more convenient interface to the MMI by supplying events for each of the possible key events, including long press.

SRV_KEYPAD encapsulates the single instance of EMB_KPD that simulates the embedded keypad, and generates the keypad user events. EMB_KPD contains the interface between the hardware and the MMI keypad.

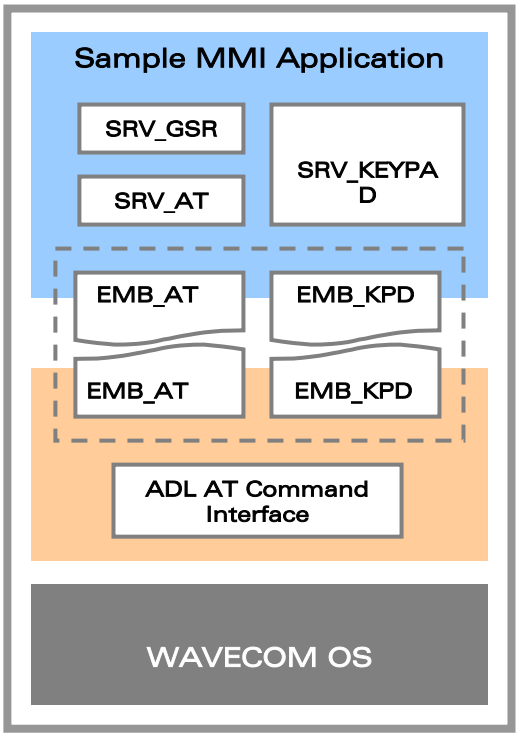


Figure 37: Keypad Integration

ADL interface is used to capture keypad events from the hardware. Embedded application subscribes for keypad events using ***AT+CMER*** command. This command is used for mobile equipment event reporting. To capture the keypad events, application handles the unsolicited responses with ***"+CKEV"***. Along with the key press event, this response contains the ID for the key pressed.

These key press events received from AT are used to trigger EMB_KPD events. The key ID received in AT response is used to set the key code property of EMB_KPD. EMB_KPD in turn updates SRV_KEYPAD accordingly.

SRV_KEYPAD uses services of SRV_GSRV to switch on the platform keypad backlight when any key is pressed. SRV_GSRV uses SRV_AT → EMB_AT to send

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.

the AT command to change the state of a specific GPIO that switches on the keypad backlight.

5.2.1.3 Timer

ADL library method, *adl_tmrSubscribe,* is used to get the timer ticks from the hardware. Every time ADL timer expires, MMI timers are updated using RapidPLUS™ interface method *rpd_PrivUpdateTimer.* RapidPLUS™ internally is responsible for updating all its timers used in various MMI components.

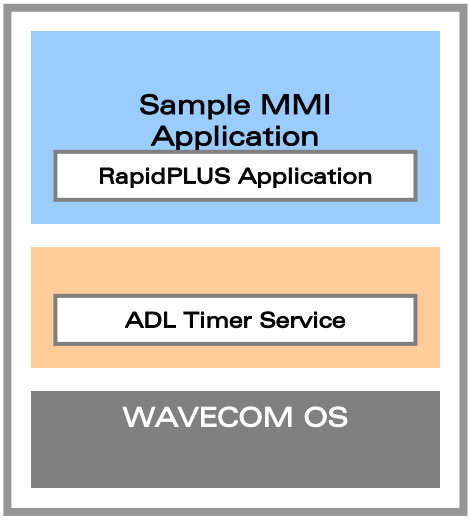


Figure 38: Timer Integration

Method *rpd_PrivUpdateTimer* is implemented inside e-SIM micro kernel. This provides the interface to synchronize the RapidPLUS™ clock using the system clock. ADL library functions are used for this synchronization. ADL timer function, *adl_tmrSubscribe* is used to subscribe a timer proc with the ADL. From within the timer proc, a call to *rpd_PrivUpdateTimer* is made. Every time the timer expires, ADL passes the control to the timer proc which updates the RapidPLUS™ clock with the same time interval as used in adl_tmrSubscribe. Wavecom platform provides two units for the timer expiry. These are 18.5 ms and 100 ms.

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.

5.2.2 Embedded AT Integration

The embedded AT component serves as the AT command channel UDI. Its sole purpose is to connect to the AT service, SRV_AT, to the embedded system. Embedded AT component does not do any processing on AT command itself, it just acts as a dispatcher for AT commands.

In RapidPLUS™ embedded AT is implemented as EMB_AT.UXO. During code generation, this component is generated as UDI, i.e. Interface only. The actual implementation of the interface is done using Open AT® Software ADL APIs provided by Wavecom.

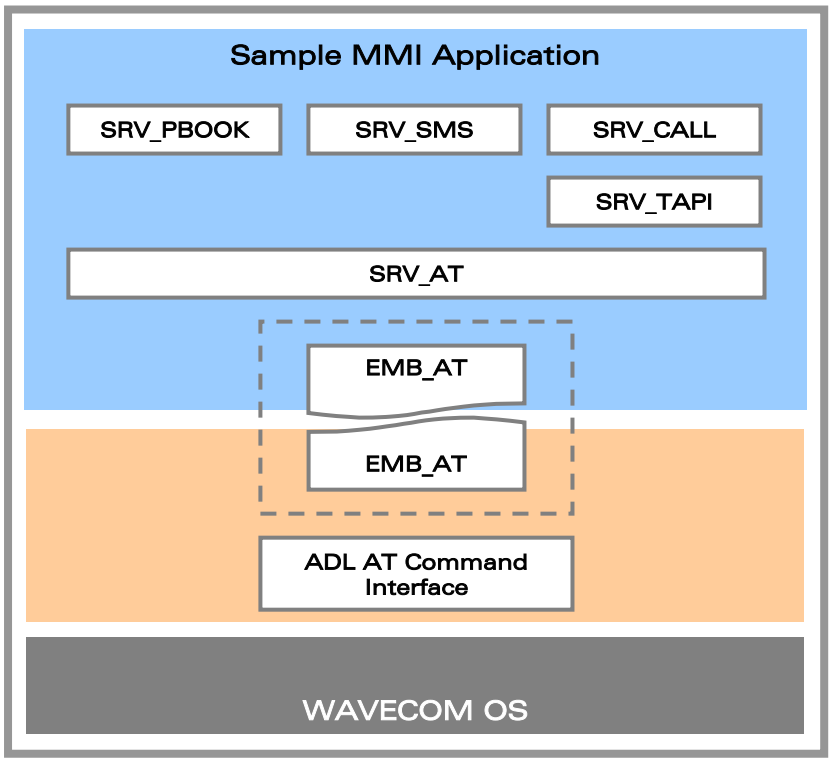


Figure 39: Embedded AT Integration

SRV_AT & EMB_AT, the RapidPLUS™ User Data Objects are responsible for AT integration of sample MMI over Wavecom platform and thus is the most important aspect of the design.

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.

SRV_AT

This component acts as an interface between application services and the underlying Wavecom layer. SRV_AT itself does not understand anything about AT commands. It does not have any ability to parse the AT commands or responses passing through it. Its main use is to provide the service interface of the underlying EMB_AT module, which actually connects to the protocol stack to deliver AT commands and receive AT responses. All traffic to and from the stack must pass through this service which operates as a single instance.

The following services are provided by SRV_AT:

- Registering unsolicited responses.
- Sending commands.
- Receiving responses.

EMB_AT

This component is implemented as a RapidPLUS™ User Interface Object. Its sole purpose is to connect SRV_AT to the underlying protocol stack. It has almost same interface as SRV_AT and thus they have one to one mapping. Since Phonebook, SMS and Calls have been implemented using AT commands, EMB_AT serves as a single point of integration for these modules.

5.2.3 General Services Integration

The general services integration is responsible for the integration of the following services:

- Battery
- Charger
- Backlight
- Clock
- Tone
- Melody
- Mute
- Volume
- Settings
- Call Records

AT commands and ADL APIs are used for the integration of these services.

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.

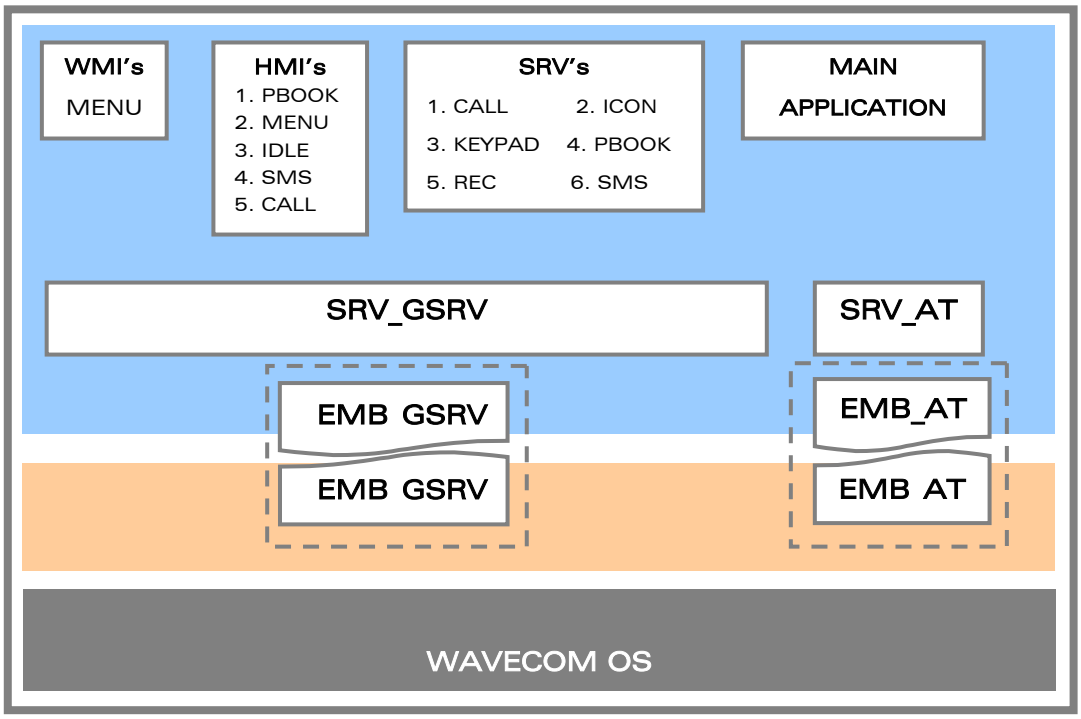


Figure 40: General Services Integration

SRV_GSRV

General services component SRV_GSRV provides an interface to the Service and MMI components above it for interfacing hardware services like battery, charger, backlight, clock etc. SRV_GSRV uses the services of SRV_AT to send various different AT commands to the platform.

EMB_GSRV

Apart from other services, this component is responsible for reading and writing to flash memory. These flash read/write operations are required for reading/writing of User Settings and Call Records. Flash read/write is done using ADL flash read/write APIs provided by Open AT® Software. SVR_GSRV calls the UDI component EMB_GSRV which calls ADL flash APIs for the implementation of its interface.

5.2.4 Service Component Integration

Services component integration includes integration of the following services:

- Phonebook
- SMS
- Calls

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.

- Call Records
- Network Management

As discussed in Section 5.2.2 the integration of these services depends mainly upon the integration of Embedded AT Interface and Embedded General Services Interface.

Referring to Figure 39, Phonebook, SMS and Call Services internally use AT implementation. The required AT commands are sent to AT service component, SRV_AT, which passes the AT command to embedded AT component. Embedded AT component, EMB_AT, once integrated using the Wavecom Open AT® Software ADL APIs, will send the AT command to underlying Wavecom AT parser, ATI. The AT response is redirected to respective modules using the same path. EMB_AT will receive all the responses from underlying layer. SRV_AT will receive the responses from EMB_AT in form of a message and it will pass it on services connected to SRV_AT.

Call Records use EMB_GSRV (refer Figure 40) to read and write the records to the flash memory. Since SRV_GSRV provides general purpose services, it is used by other components like phonebook, SMS and calls also.

5.2.4.1 Phonebook

Sample MMI Phonebook uses SRV_AT/EMB_AT module to communicate with the Wavecom OS. Phonebook implementation is done using the following modules (refer Figure 41):

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.

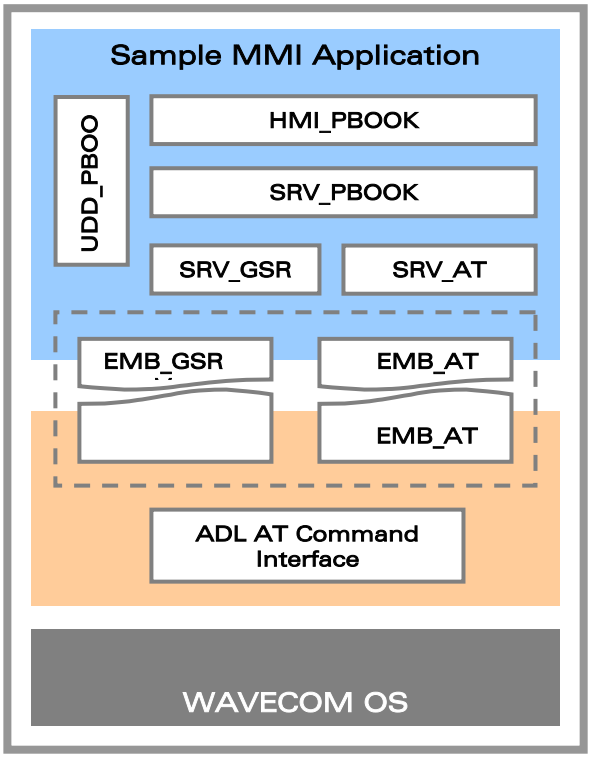


Figure 41: Phonebook Integration

HMI_PBOOK

As discussed HMI modules provide the human machine interface and are responsible for updating the display when they are active. HMI_PBOOK is called by HMI_IDLE when phonebook option is chosen. HMI_PBOOK handles user interface for phonebook functionality. It can be started in two modes, which are defined in MODE_Cs:

- Browsing mode - the entries list of the current phonebook is displayed.
- Menu mode - the phone book menu is opened (pressing right soft key from idle).

HMI_PBOOK uses the services of SRV_PBOOK for all the operations related to Phonebook.

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.

SRV_PBOOK

SRV_PBOOK provides the interface which is used by HMI_PBOOK to do all activities related to phonebook. This service component is responsible for implementing phonebook functionality in terms of AT commands. Depending upon the option chosen in the menu, SRV _AT is responsible for selecting the type of phonebook to be used. The selection is done using the AT commands only. Currently both SIM and memory phonebooks are supported but they cannot be used simultaneously. As discussed above, this module is used for services from SRV_AT to send the commands to the underlying protocol stack. Apart from sending the commands, this module is responsible for listening to all types of responses, both AT command responses and unsolicited responses, from the network. It has the ability to filter and act on the responses valid for Phonebook.

UDD_PBOOK

UDD_PBOOK is implemented as a RapidPLUS™ Data Container Object. Being a data container, it does not provide any interface except to insert and retrieve data. UDD_PBOOK is used by SRV_PBOOK to store the date related to Phonebook.

SRV_GSRV

SRV_PBOOK uses the services of SRV_GSRV for saving the speed dial entries to phone settings. SRV_GSRV internally uses EMB_GSRV for implementation.

5.2.4.2 SMS

SMS Services are also integrated with the Wavecom platform using SRV_AT/EMB_AT module. The sections below briefly explain the RapidPLUS™ modules used by SMS. (Refer Figure 42).

Note	In contrast to the way other AT commands are sent, **SMS** text is sent using the adl API adl_atCmdSendText in the emb_at.c file.

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.

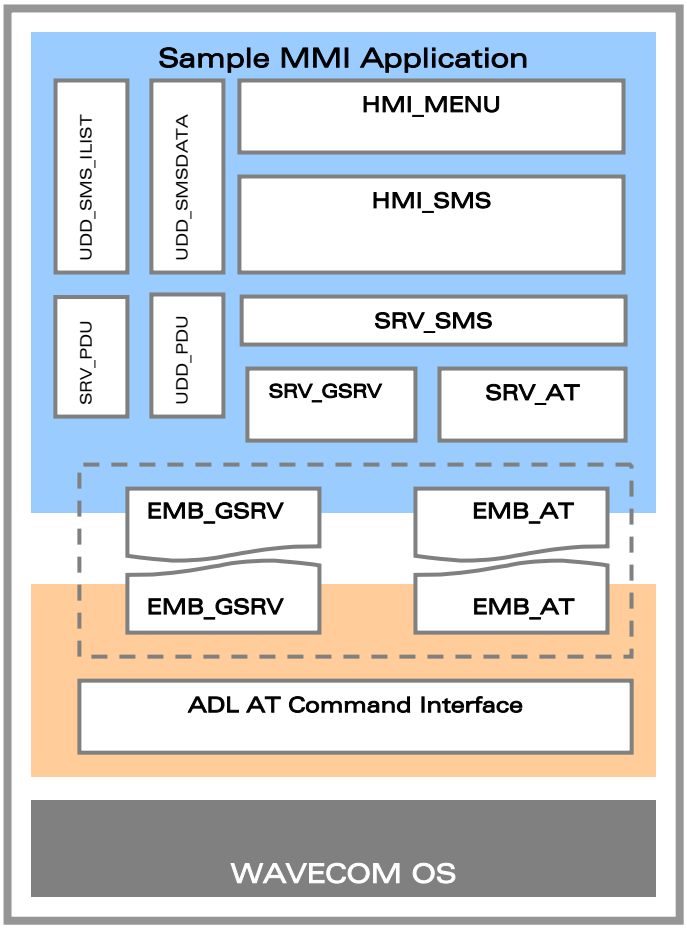


Figure 42: SMS Integration

HMI_MENU

This is an HMI module responsible for providing the Menu display. HMI_MENU handles the menus starting at the main menu. When the user selects an SMS option from the menu, HMI_MENU launches HMI_SMS which in turn is responsible for showing SMS menu.

HMI_SMS

This module implements the MMI for the Messages Menu and is responsible for updating the display when SMS menu is active. HMI_SMS gets/sets messages data from SRV_SMS using UDD_SMS_DATA, a data container that holds the data in a structure.

UDD_SMSDATA

This is a data holder component dedicated to data exchange between HMI_SMS and SRV_SMS. SRV_SMS gets the message from the underlying layer and updates it in this data container from where HMI_SMS takes the inputs.

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.

SRV_SMS

This service component transforms requests from HMI_SMS into AT command form. It sends, stores, and indicates incoming short messages. Most of the operations that SRV_SMS performs are asynchronous; therefore, next operation cannot be requested before Ready_evt is triggered. SRV_SMS uses SRV_AT module to send and receive the SMSs to and from the network. Apart from interacting with HMI_SMS and UDD_SMSDATA discussed above, it stores the SMS list in the UDD_SMS_ILIST which acts as a temporary storage of SMS data. SRV_SMS also interacts with SRV_PDU to convert and decode a message in PDU format.

UDD_SMSILIST

This data container is used to store the list and the content of the SMS received. This is used by SRV_SMS.

SRV_PDU

SRV_PDU provides an interface for PDU creation and decoding when sending and receiving SMS. It is solely used by SRV_SMS. SRV_PDU uses services of SRV_GSRV for the methods related to Unicode character conversion.

UDD_PDU

This component is dedicated to data exchange between SRV_SMS and SRV_PDU. It also contains buffers, in which SRV_PDU keeps the temporary results of encoding and decoding PDU strings.

5.2.4.3 Call Control

Similar to Phonebook and SMS, Call Control also uses services of SRV_AT/EMB_AT module to communicate with the Wavecom OS. The sections below briefly explain the RapidPLUS™ modules used for Call Control Services (Refer Figure 43).

HMI_CALL

This component is responsible for the MMI during active calls. It manages dialing, outgoing calls, active calls, conference calls (multi-party), on-hold calls, and swapping between a held call and an active call. It also manages all in-call options. This module is activated by HMI_IDLE when the user starts dialing a number or when HMI_IDLE receives an incoming call event from SRV_CALL. HMI_CALL, when active, is responsible for showing all the information of the display. This module communicates with SRV_CALL for all the information that it needs to show on the display.

SRV_CALL

This service component is responsible for managing all active calls. It manages the calls data using UDD_CALLS. SRV_CALL is mainly used by HMI_CALL and also by other HMI modules that initiate the calls. As seen in the design, call functionality has been split into two modules, SRV_CALL and SRV_TAPI. This

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.

design keeps the AT implementation out of SRV_CALL and avoids complicating the SRV_CALL module. Thus, SRV_CALL module acts as a service module for the display component, HMI_CALL and internally it communicates with the network using interface provided by SRV_TAPI. SRV_TAPI is the one that actually implements the AT commands related to call handling.

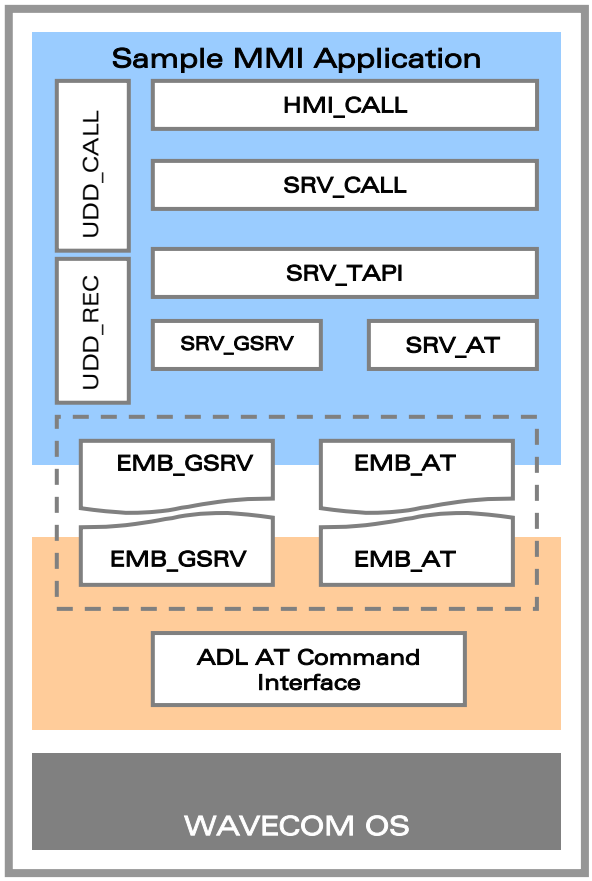


Figure 43: Call Control Integration

UDD_REC

This component is used to transfer data of a single call record. There are two instances of UDD_REC in the application:

- For sending call information from SRV_REC to HMI_REC.
- For sending call Information from SRV_CALL to SRV_REC.

Apart from being used by SRV_CALL, SRV_REC uses this module to show the call records when records menu is active.

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.

UDD_CALL

This component holds the data of all active calls in its message interface. It is updated only by SRV_CALL.

SRV_TAPI

This service component provides AT implementation of several network related functions.

- Modem initialization functions
- Dialing and call manipulation functions
- PIN code functions
- Network services function

SRV_TAPI uses interface provided by SRV_AT to send the AT commands and receive the different type of responses. SRV_TAPI has intelligence to process several kinds of AT responses related to Call handling, network services etc. and then inform the upper services as well as HMI layer about it.

SRV_GSRV

Apart from reading and writing settings to flash memory, SRV_GSRV is also responsible for providing updated date and time to the HMI_IDLE and HMI_MENU components. SRV_GSRV uses SRV_AT to send the command to get the current date and time. SRV_GSRV uses EMB_GSRV to save the settings to flash.

5.2.4.4 Call Records

As discussed in section 5.2.3, Call Records are implemented using flash read/write methods provided by SRV_GSRV. SRV_GSRV internally calls EMB_GSRV for the implementation of the flash read/write functions. EMB_GSRV implements these methods using Open AT® Software ADL flash API. The sections below briefly explain the RapidPLUS™ modules used for Call Records. (Refer Figure 44)

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.

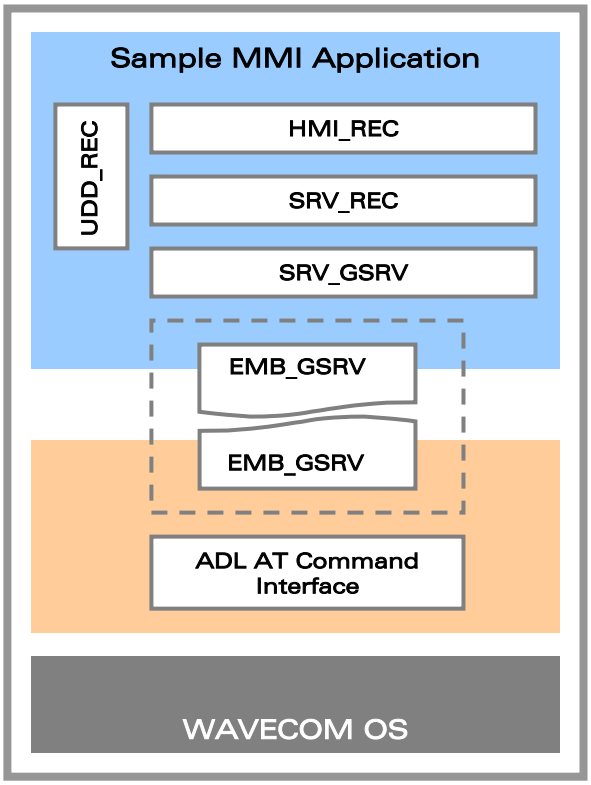


Figure 44: Call Records Integration

HMI_REC

This component handles the user interface for Call Records functionality. It handles all Call Records interaction and all menu functionality under the Call Records menu. The call records data is held in SRV_REC. HMI_REC uses UDD_REC to retrieve the data.

SRV_REC

SRV_REC provides the interface which allows HMI_REC to do all activities related to call records. This service component provides data to the call records menu in HMI_REC. This data includes:

- The number, date, and time of the 10 most recent incoming, outgoing, and missed calls
- Total time of all received calls
- Total time of all calls sent
- Last call time

SRV_REC receives the data from SRV_CALL, sorts the data by time and date, and saves the data using SRV_GSRV. Call records data is not held by this service

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.

component. The service only holds indexes and manages the data that is stored in the flash memory.

SRV_GSRV

SRV_GSRV functions are called by SRV_REC to read and save call records. This component further calls EMB_GSRV functions. EMB_GSRV is used to read and write data to the flash memory. For the implementation of this interface, Open AT® Software flash API will be used.

UDD_REC

UDD_REC has been implemented as a RapidPLUS™ data container object, so it just holds the data but does not provide any interface to read and write data.

UDD_REC holds call related data. It is used to transfer data of a single call record.

There are two instances of UDD_REC in the application:

- For sending call information from SRV_REC to HMI_REC.
- For sending call Information from SRV_CALL to SRV_REC.

5.2.4.5 Network Management

Network Management integration (refer Figure 45) is similar to what has been discussed in the Call Control Integration. All the network related functionality is implemented in the SRV_TAPI component which is responsible for call control functionality as well (refer section 5.2.4.3).

HMI_IDLE

HMI_IDLE is responsible for showing all the network related indications on the idle screen. This includes network strength indicator as well as the operator name. HMI_IDLE calls methods on SRV_TAPI to get this information and then updates the idle screen accordingly.

HMI_MENU

This is an HMI module responsible for providing the menu display. HMI_MENU handles the menus starting at the main menu. All the menus related to Network management and network settings are handled by HMI_MENU itself. For all the network related menu options like set/cancel call diverts, set/cancel call barring, set/cancel call waiting, PIN On/Off/change, network selection, HMI_MENU calls SRV_TAPI component. For reading/writing all these settings to the flash, SRV_GSRV component is used.

SRV_TAPI

This service component provides AT implementation of several network related functions.

- Modem initialization functions – under init mode
- Dialing and call manipulation functions

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.

- PIN code functions
- Network services function

SRV_CALL uses interface methods provided by SRV_TAPI to communicate with the network. SRV_TAPI uses interface provided by SRV_AT to send the AT commands and receive the different types of responses. SRV_TAPI has intelligence to process several kinds of AT responses related to call handling, network services etc. and then inform the upper services as well as the HMI layer about it.

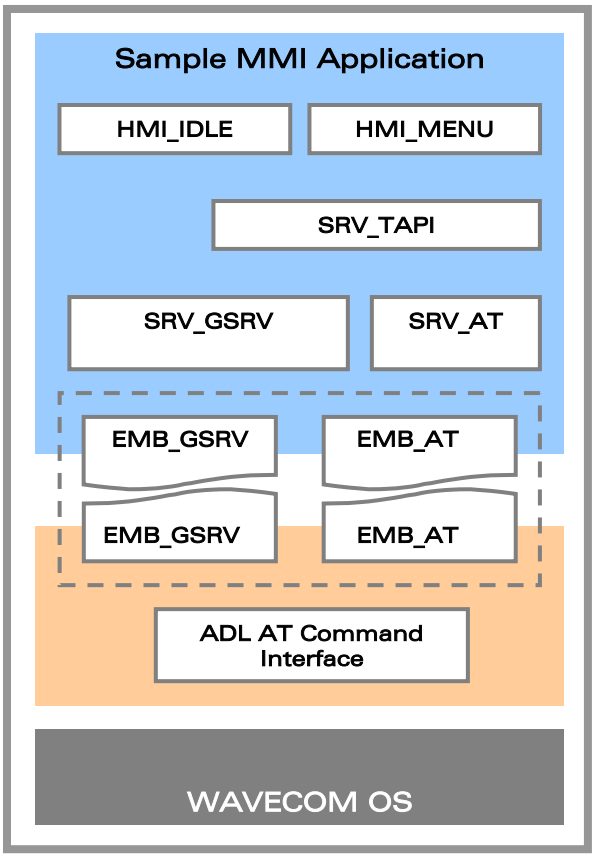


Figure 45: Network Management Integration

5.2.5 Embedded Component Integration

Integration of all these components happens through AT commands only. SRV_GSRV contains the logic of sending the corresponding AT commands and receiving responses for the integration of these components. SRV_GSRV hides

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.

the details about AT commands and provides direct interface methods that can be used to interact with any of these components.

5.2.5.1 Battery

The battery implementation has been done using the AT commands in the SRV_GSRV component. In order to implement the check status of battery, there is a mode in the SRV_GSRV component which is activated whenever the main application calls the SRV_GSRV component to check the battery status. The AT command, AT+CIND, gets the battery status as its response. This response is then captured into a local variable and displayed on the target screen. There are five battery levels (0-5) that could be returned as response.

5.2.5.2 Synthesizer

The Melody synthesizer is implemented in the SRV_GSRV user object. It is implemented using the AT command AT+CRMP. Various melodies available on the target could be played using this AT command. The main application provides the index of the melody that needs to play.

Similarly, there is a separate mode for playing the tones on the press of keys. The implementation is done in the tone mode using the AT command AT+WDTMF. The parameters for this command are provided by the main application component.

5.2.5.3 Backlight

Backlight implementation is done in the SRV_GSRV component. Similar to the other embedded components, it has been implemented using the AT commands. The command used to start and stop the backlight is AT+WIOM.

5.2.5.4 Charger

The charger implementation has been provided in the main application startup file oat_appli.c. The charger and charger events have been subscribed in this file using the adl APIs,

`adl_atUnSoSubscribe, adl_ioEventSubscribe, adl_ioSubscribe.`

Once the charger is plugged, charger plugged event is triggered which causes the control of the application in the SRV_GSRV component to go to the StartCharging mode and vice versa if the charger is unplugged.

5.3 Sample Use Case - Outgoing Call Set up

When user starts dialling a number on the idle screen, the HMI_IDLE calls *startDialing_string* method on HMI_CALL component to handle the dialled

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.

number on the screen. HMI_CALL, after checking for the emergency number, waits for the press of SEND key.

On the press of SEND key, HMI_CALL extracts the number from the editor widget handling number input and it calls SRV_CALL:*dialTo* method with the dialled number as the parameter to the method. SRV_CALL:*dialTo(..)* method sets the CallNumber_Str property on the SRV_TAPI and then calls *tapi_dialTo* method on SRV_TAPI module (refer Figure 46)

SRV_TAPI is responsible for converting the dial number request to AT command and send the command to SRV_AT for execution. To send the AT commands and receive the responses synchronously, SRV_TAPI implements the AT dispatcher.

AT dispatcher is implemented as a concurrent node to *"Functions"* node in SRV_TAPI. When a command is given to AT dispatcher, it goes into the *"WaitForDispatcher"* state where it waits for SRV_AT to be *ready*. When *ready*, it goes to *"TryToSend"* state from where it calls a SendAT method on SRV_AT. If it returns error code as *busy*, it goes back to *"WaitForDispatcher"* state and repeats the process till the error code is *not equal to busy*. Once command is sent, AT dispatcher goes to *"WaitForResponse"* state.

SRV_AT calls *sendAT* on EMB_AT. Since EMB_AT is generated as a UDI it holds the key to the entire AT integration. sendAT maps to c-code function which calls Open AT® Software ADL API to send the AT Command, *atdXXXXXXXXXX*;, to Wavecom AT parser for execution.

On receiving the response from the underlying AT parser, EMB_AT informs SRV_AT using the *send message* interface. EMB_AT encapsulates the response in a message structure and sends it to SRV_AT. SRV_AT extracts the response from the message and triggers the response events, *response_Evt* and *final_Evt* to inform SRV_TAPI. In this case, SRV_TAPI waits for *final_Evt*. On receiving the indication for *final_Evt*, AT dispatcher in SRV_TAPI goes back to idle state and SRV_TAPI sends *OK_Ev* to SRV_CALL.

Apart from the responses to AT commands, AT parser sends some unsolicited responses also. When call connects, underlying AT parser sends unsolicited response with string "%CPI". To get the indication about the call connect, SRV_TAPI has subscribed for this message. On receiving an unsolicited response with "%CPI", it sends CallConnected_Ev to SRV_CALL. SRV_CALL passes the same event to HMI_CALL to enable to perform any changes on the display, if needed.

SRV_CALL maintains the list of active calls in the data container, UDD_CALLS. On getting *CallConnected_Ev* from SRV_TAPI, it updates the list and sends *CallStatusUpdate_Ev* to HMI_CALL to update active calls list on the display.

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.

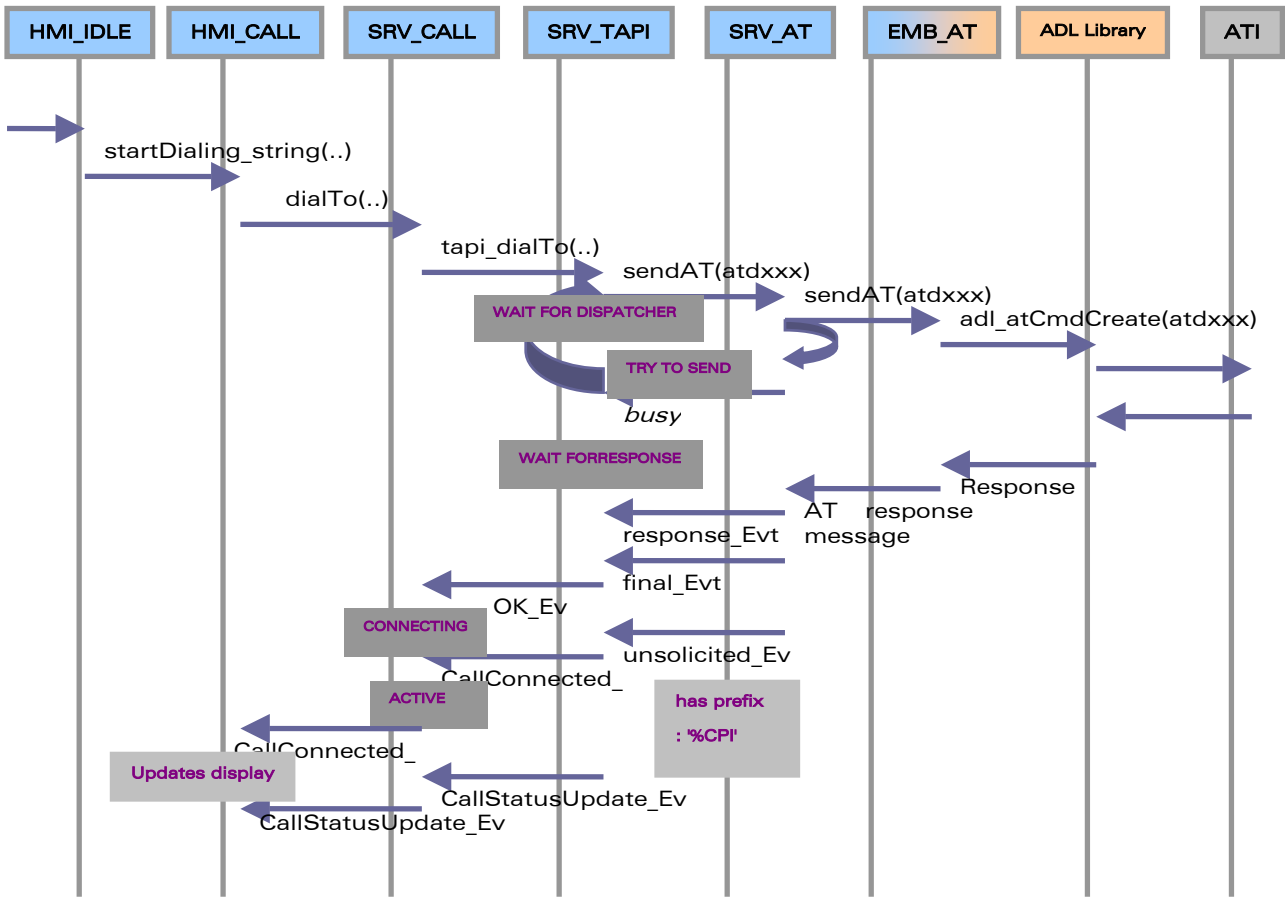


Figure 46: Sample Use case Realization – Outgoing Call

5.4 Sample MMI Menu Structure

Sample MMI consists of various menus and submenus. The menu structure is specified in the textresource.xls file. A snapshot of the main menu for the sample application has been shown in the image Figure 47.

In order to add a menu or submenu to the existing menu, developer can add a row in the excel file in the level as required. E.g. a user wants to add another submenu say "NewMenu" to the main menu. The menu can be added by simply inserting another row and adding a menu in the second column.

After adding the menu, user needs to save the excel file and then generate the Text & Menu Generator macro. This would update the necessary user objects in the application and a new menu is shown in the application.

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.

Level1	Level2	Level3
Main menu		
	Messages	
		Inbox
		Outbox
		Write messages
		Message settings
		Voice messages
	Call records	
		Missed calls
		Received calls
		Dialed numbers
		Call time
	Settings	
		Alarm clock
		Clock
		Phone settings
		Sound settings
		Security settings
	Network	
		Call diverting
		Call barring
		Network selection
		Caller ID
	Lock keypad	
	Phonebook	

Figure 47: Sample MMI Menu Structure

5.5 Sample MMI Workspace Architecture

The highest directory level in the build is Reference_Design. Under Reference_Design, Open AT® GTi workspace architecture is divided into three main directories:

- **inc -** This directory contains Wavecom header files for the LCD device and display drivers,
- **src -** This directory contains Wavecom source files for the LCD device and display drivers and also the main source file of the Open AT® GTi project.
- **libs -** Contains three main folders, ESIM, Device and Display. The ESIM would contain the customized code for the application; device and display would have the code for the custom device and display of LCD drive.

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.

Note	Before starting with the application development, make sure that the GTi folder containing the libraries for Open AT® GTi development has been copied to the **Plug-in** subfolder inside the Open AT® Software Framework folder.

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.

5.5.1 Overall Directory Structure

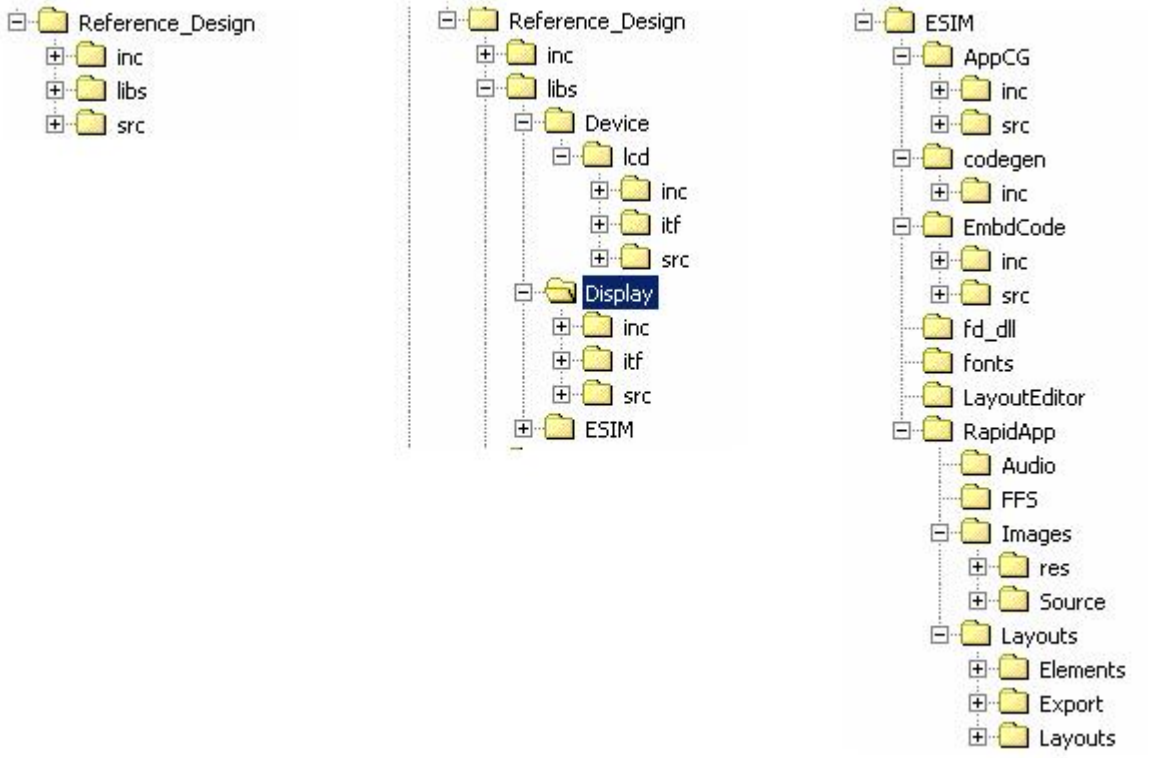


Figure 48: Overall directory structure

5.5.2 inc - Folder Containing Device and Display Driver Header Files

The inc folder contains the header files related to the Wavecom specific LCD display and device drivers.

5.5.3 src –Folder Containing Device and Display Source Files

The src folder contains the source files related to the Wavecom specific LCD display and device drivers.

It also contains the main application file which contains the entry point of the application.

Note	Src folder contains the *oat_appli.c* which has the start point *adl_main* for Open AT® Software embedded application

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.

All the projects contained in the Reference_Design.dsw would have two variants: *<ProjectName>.dsp* and <*ProjectName>_rte.dsp*. <ProjectName>.dsp contains the files required for the target build (ARM build) of the application and <ProjectName>_rte.dsp contains the files required for compiling the build using MS Visual Studio 6.0. The rte build is used for debugging the entire application. Projects included in Reference_Design.dsw are:

- Reference_Design.dsp
- Reference_Design_rte.dsp
- lcd.dsp
- lcd_rte.dsp
- display.dsp
- display_rte.dsp
- ESIM.dsp
- ESIM_rte.dsp

To make the target build, Reference_Design.dsp needs to be compiled which further has the dependencies set for all other projects. Reference_Design.dsp uses *Reference_Design.mak* file to compile the build using ARM. To create the debug build, Reference_Design_rte.dsp has to be compiled.

The target build binary is generated in..Reference_Design**arm\out** folder. Out folder gets created when the target build is compiled for the first time. Debug build is generated in **Reference_Design\rte\vc6\debug** folder. rte folder and all its subfolders get created when the debug build is compiled for the first time.

libs The libs folder contains the following folders:

- Device
- Display
- ESIM

Device: The device folder contains LCD folder which further contains the following folders:

- inc
- itf
- src

Note	▪ **Device** folder contains subfolders with drivers for various devices used by the application. Currently it contains only **lcd** folder for the lcd driver files. ▪ **lcd\src, lcd\inc and lcd\itf** folders contain required .c and .h files needed for the lcd driver to work. LCD driver has been written using Open AT® Software ADL APIs only.

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.

Project files required to open this project, lcd.dsp and lcd_rte.dsp exist in device\lcd folder. lcd.dsp uses ***lcd.mak*** file kept in the same folder to make the target build. lcd.dsp and lcd_rte.dsp are already included in Reference_Design workspace Reference_Design.dsw.

Display: The display folder contains the following folders:

- inc
- itf
- src

Note	▪ **Display** folder contains the files required for the display module. Display module provides a layer of glue between MMI application and the underlying LCD driver. ▪ **src, inc and itf** folders contain .c and .h files required for the implementation of display module.

Project files required to open this project, display.dsp and display_rte.dsp, are present in the display folder. display.dsp uses ***display.mak*** file kept in the same folder to make the target build. display.dsp and display_rte.dsp are already included in reference design workspace Reference_Design.dsw.

5.5.4 ESIM Specific Code Directory

The **ESIM** folder includes:

RapidApp: This folder contains the following folders:

- Audio
- Images
- Layouts
- FFS

Note	▪ All the MMI files containing RapidPLUS™ objects are placed in RapidApp folder. All these files are xml files. MainApp.RXD is the project file which can be used to open the project in RapidPLUS™. ▪ Audio folder contains all the midi files used by the MMI. ▪ Images folder contains all the graphics used by sample MMI. ▪ Layouts folder contains the information related to the layouts used in the MMI ▪ FFS folder contains phonebook data file and phone memory data file. These files are of no relevance for the target build but are used in the simulation build.

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.

appCG: The appCG folder contains the following folders:

- inc
- src

Note	▪ appCG folder contains the code generated from the sample MMI application. ▪ RapidPLUS™ code generator generates .c and .h files in Reference_Design\libs\ESIM \appCG folder. Source and include files are separated using post build script **post-build.bat** kept in the same folder. Source and include files are separated in appCG\src and appCG\inc respectively.

CodeGen: The CodeGen folder contains only the **inc** folder.

Note	▪ CodeGen\inc contains the .h files required by RapidPLUS™ for the code generation.

Embdcode: The Embdcode folder contains the following folders:

- inc
- src

Note	▪ Embedded code folder contains the .c and .h files required by RapidPLUS™ for the code generation. This folder also contains the .c and .h files generated for the language strings.

fd_dll and lib: These folders contain dll files and lib files required by e-SIM RapidPLUS™.

fonts: This folder contains the fonts used by Sample MMI application.

LayoutEditor: This contains a tool, 'Layout Editor', used for modifying the screen layouts. The process to modify the screen layouts has been covered in section 5.8.

Arm: This folder contains the make folder containing all the make files required to build the generated code.

5.6 Build and Deployment Procedure

Build procedure involves the following steps:

i. Code generation from the MMI xml files using e-SIM RapidPLUS™

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.

ii. Building the code using ARM compiler

iii. Downloading the image to the target platform using Xmodem (use 1K Xmodem for Q2686H Wireless CPU) protocol in the HyperTerminal.

5.6.1 Code Generation using RapidPLUS™

Following are the steps for RapidPLUS™ Code Generation:

i. Open the Wavecom Reference MMI workspace Reference_Design\libs\ESIM \RapidApp\mainapp.rxd using RapidPLUS™.

ii. Select the MAINAPP project from the list of projects in the drop-down.

iii. On the Menu bar, open the generate code dialog from Code → Generate Code menu shown in Figure 49.

iv. As shown in the Figure 50, the user can change various preferences for the code generation such as the user objects that need to be generated as interfaces only and those that need to be generated as full objects or data objects. Other settings such as the application queue size and the path for the .c and .h files is specified in the preferences section.

v. Once the user clicks on the 'Start' button for the code generation, the code generation status dialog box appears on the screen and the percentage completion progress bar is displayed on the screen as shown in the Figure 51.

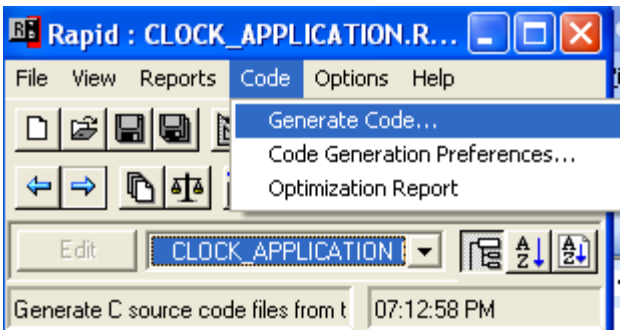


Figure 49: Code generation Menu

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.

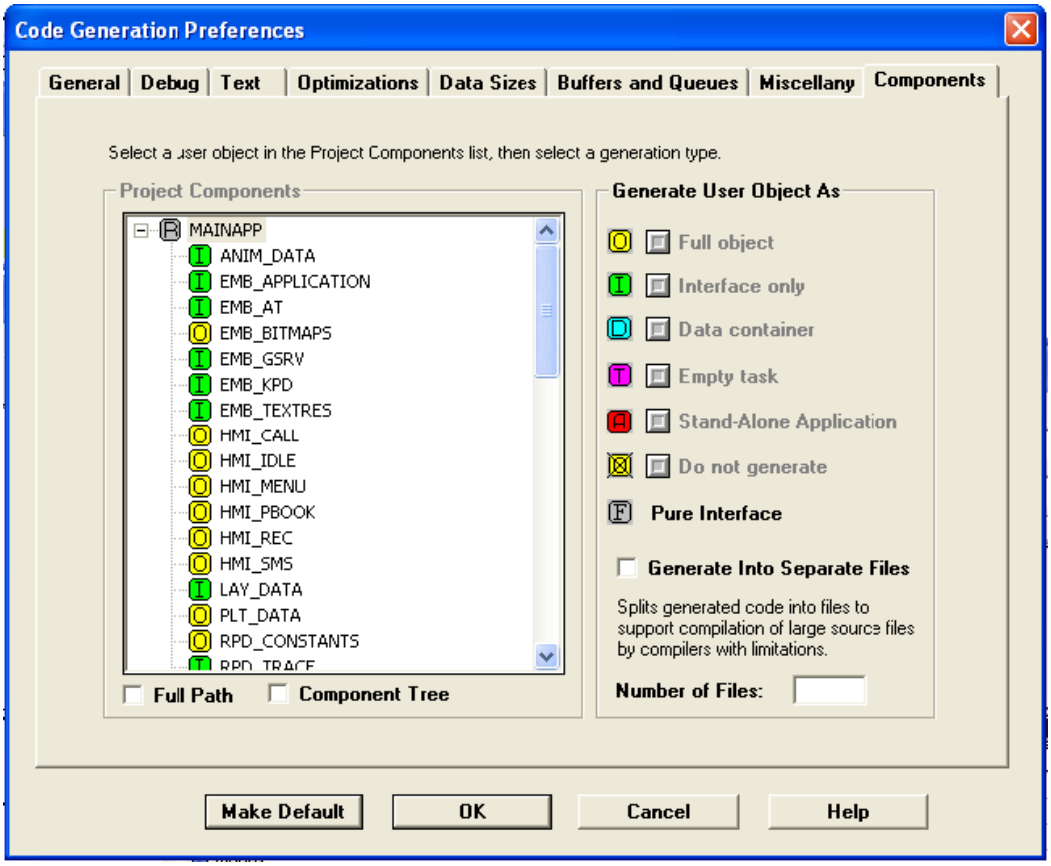


Figure 50: Code generation - Preferences

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.

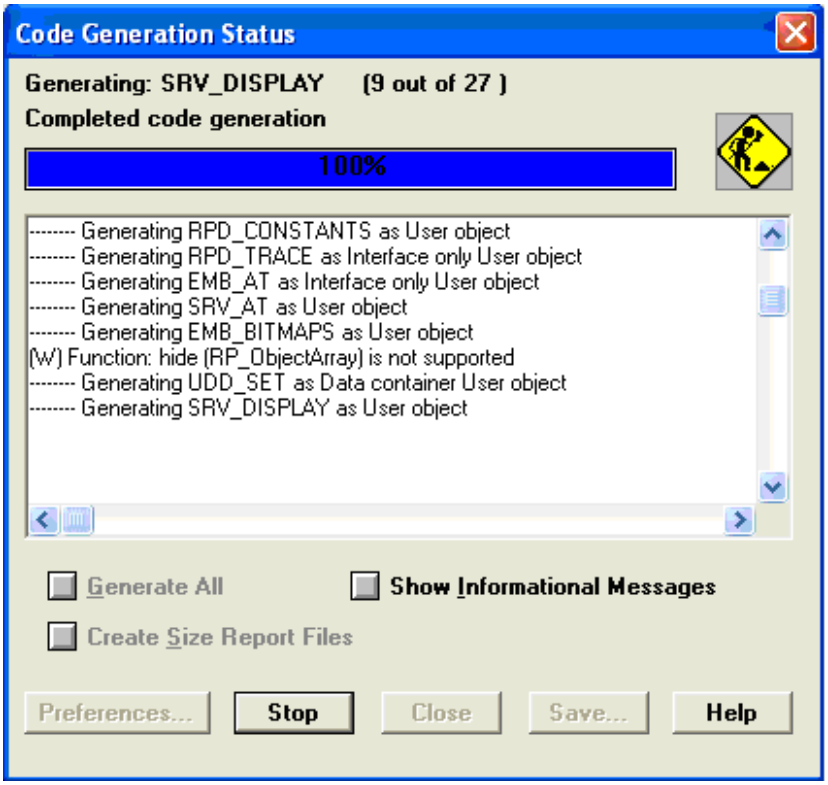


Figure 51: Code generation - In progress

5.6.2 Build using ARM Compiler

Following are the steps to make a target build:

i. Open the project workspace using MS Visual Studio.

ii. Make a default project Reference_Design.

iii. From the Menu, go to Build → Rebuild All. This would build the target build using the make file Reference_Design.mak kept in the same folder.

iv. Once the build is complete, the image file 'arm_Reference_Design_H.wpb.dwl', gets created in ...\arm\out folder.

Note	The ARM compiler used is ARM® Developer Suite (ADS) version 1.2 http://www.arm.com/support/downloads/ads12.html Build 837 is recommended for Q2406. Build 848 is recommended for Q2686.

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.

5.6.3 Download the Build on Target Platform

Following are the steps to download the target build:

i. Open the Hyperterminal and use the AT+WDWL command to set the modem in the download mode.

ii. Select the transfer protocol to Xmodem (use 1K Xmodem for Q2686H Wireless CPU).

Browse to the folder Reference_Design\arm\out and select the image file and start the download using the send button as shown in Figure 52.

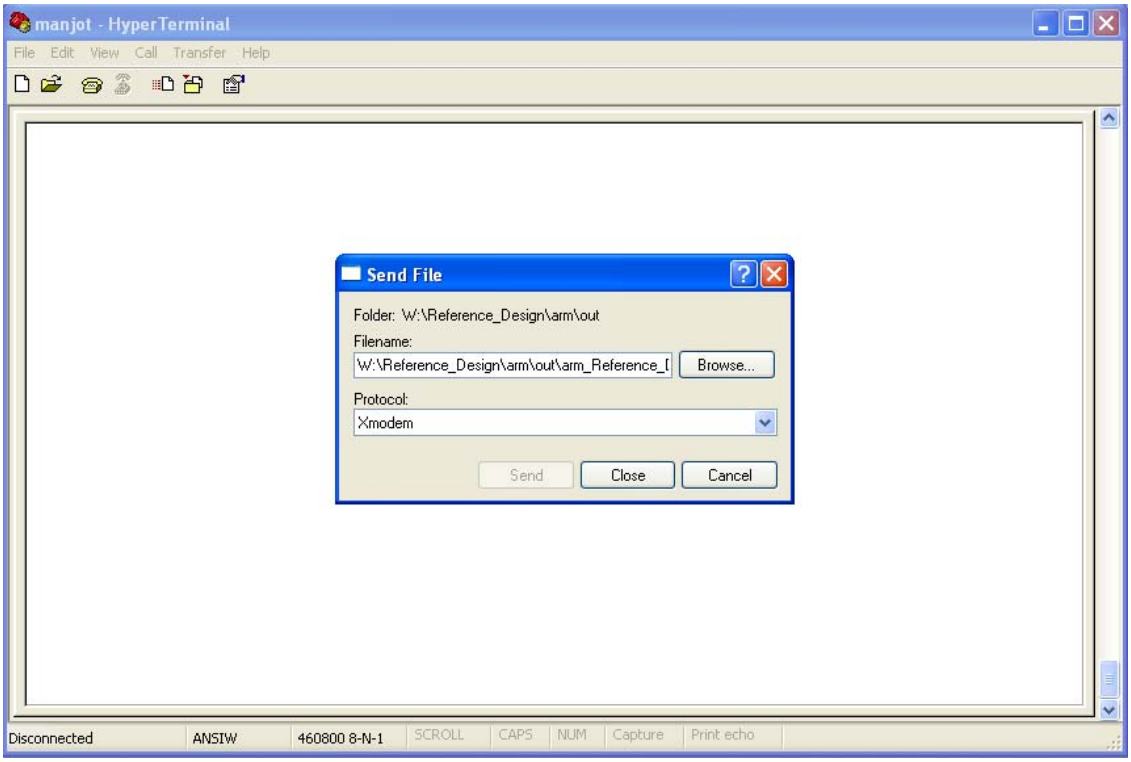


Figure 52: HyperTerminal – Build Download

Note	The application can also be downloaded over the Air. But there are restrictions on the size of the application which can be downloaded.

5.7 MMI Customization

Open AT® GTi provides wide range of possibilities for the customization of MMI to suit the market needs with very short time to market. Wavecom MMI can be customized using e-SIM and Wavecom MMI customization tools. Not only do

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.

these tools allow the MMI developers to change the graphics, ring tones and other multimedia content but also allows them to change the screen sizes and layouts to provide user friendly MMI.

The range of possibilities available for MMI customization is:

- Screen sizes and layout
- Graphics, Animations, Ring tones and other multimedia content
- Multi-lingual support to ensure wider market for the MMI
- Feature enhancements to existing service components like Phonebook, SMS, Call Control and Call Records
- Platform specific applications like Melody Composer, Bitmap Editor, Currency Converter, World Clock etc.
- Provide new and exciting games supported on the specific platform.

Any changes in the MMI involve some basic steps to ensure that MMI changes get reflected on the target platform:

- Every time any change is done using e-SIM RapidPLUS™, it is a MUST to do the code generation to enable the update of .c and .h files.
- RapidPLUS™ generates both .c and .h in the same folder Reference_Design\libs\ESIM\appCG. All the .c files must be separated to Reference_Design\libs\ESIM\appCG\src folder and all the .h files must be separated to Reference_Design\libs\ESIM\appCG\inc folder. This is done using a batch file which runs after the code generation (refer section 5.5.4)
- Once the code generation is complete, the code must be compiled using ARM compiler. If no errors, the target image gets created in the ...\arm\out folder (refer section 5.5.2)
- Target image can be downloaded to the target platform using Hyperterminal Xmodem (use 1K Xmodem for Q2686H Wireless CPU) protocol
- The MMI can be started by running the embedded application using the AT command AT+WOPEN=1
- If the MMI settings structure is modified, it is advisable to clear the existing MMI settings using AT+WOPEN=3. If required, existing target image can be removed using AT+WOPEN=4.

5.8 Screen/Layouts - Screen Sizes, Layouts

Open AT® GTi provides mechanism to conveniently modify the screen sizes and layout design of the MMI using the Layout Editor tool. Layout Editor (refer section 2.3.1) can be used to modify the layouts and then RapidPLUS™ can be used to do other necessary changes in the layout before generating the

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.

corresponding C code. The generated code can then be compiled using ARM compiler.

This section explains the basic steps to perform any changes in the screen sizes and layouts.

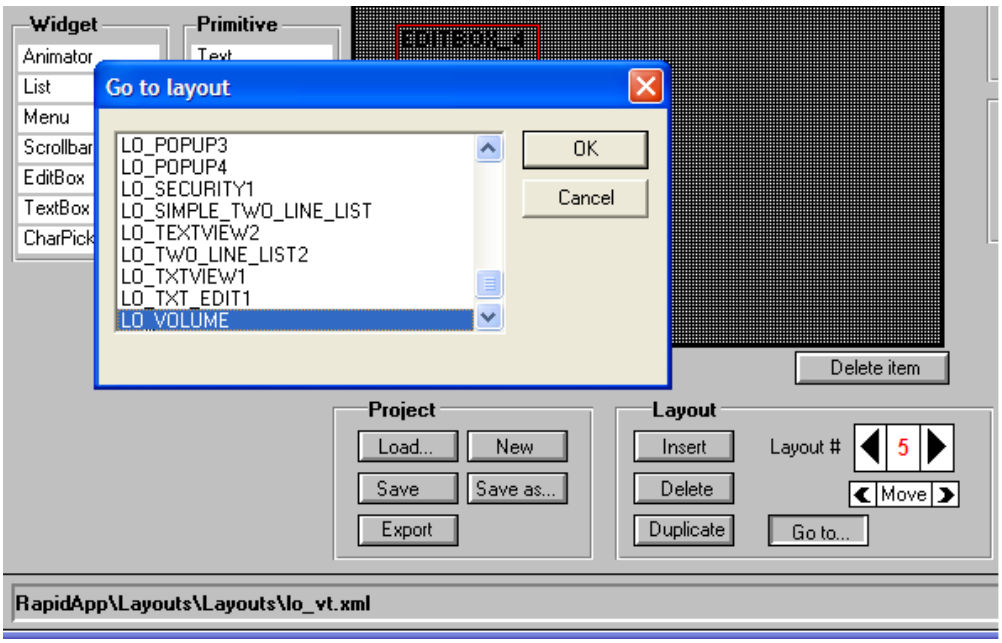


Figure 53: Screen Size/Layout Customization - Selecting Layout

Selecting the layout to be edited

i. Open the Layout Editor from Reference_Design\LayoutEditor\LAYOUT_EDITOR.RPD (refer Figure 16)

ii. 'Load' button can be used to load the project workspace containing information about all the existing layouts being used for the MMI. lo_vt.xml is the project workspace file for the existing MMI and this file can be accessed at the path... \RapidApp\Layouts\Layouts in the Open AT® GTi build.

iii. Existing Layouts can be accessed using arrows keys along with 'Layout #' label. New layouts can be inserted using 'Insert' button. Any existing layout can be deleted using 'Delete' button. Any layout can be duplicated (and then modified) using **Duplicate** button. **Go to** button opens a dialog box to enable the developer to select the layout to be modified from the list of layout IDs (refer Figure 53).

iv. Once the layout to be edited is selected, necessary modifications can be done in the size and placement of elements in the layout.

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.

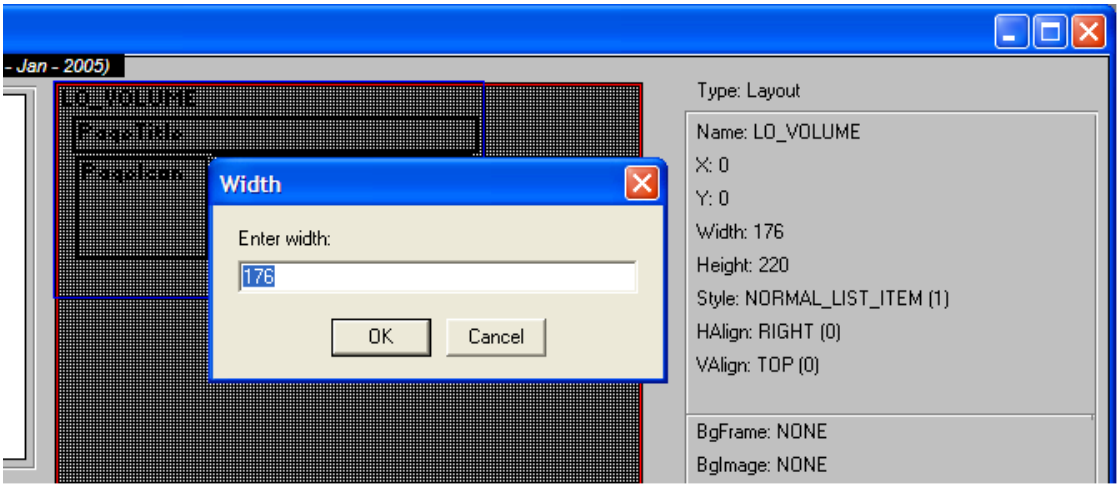


Figure 54: Screen Size/Layout Customization - Resizing Layout Elements

Moving and Re-sizing the layout elements

i. Any layout element can be selected and moved using the mouse pointer.

ii. Size of any layout element can changed by selecting the layout element and clicking on the elements in the layout and then clicking on the value of the Width or Height on the editor. Clicking on any of those values would open a dialog box in which the new value can be entered. After entering the new value, click **OK** on the dialog to update that element in the Layout Editor display with the updated value.

iii. Clicking on the current style of the element would open a dialog box with the list of all the available styles available in the Layout Editor. Click **OK** on the dialog to select a new style.

iv. Similarly, other parameters shown related to the layout element can be updated.

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.

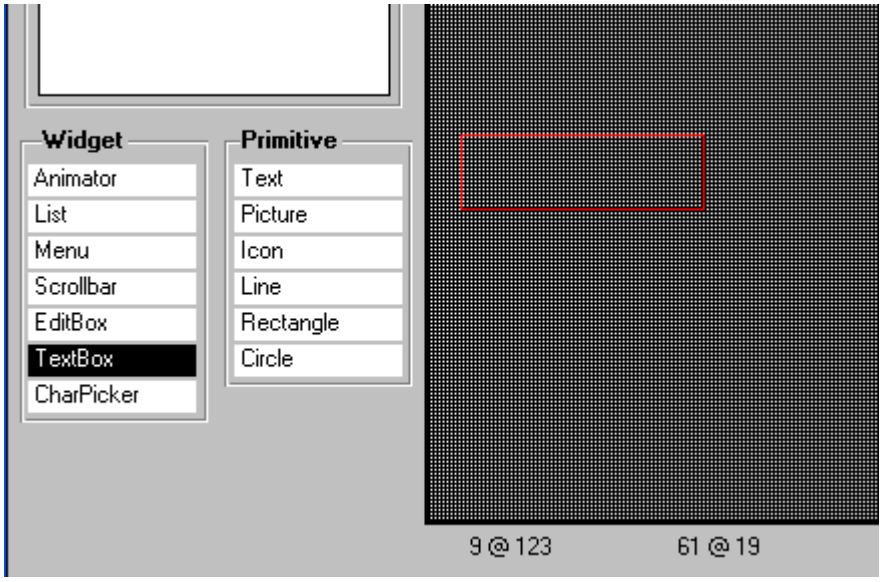


Figure 55: Screen Size/Layout Customization - Adding New Layout Element

Adding new elements to the existing layout

i. New elements can be added to the existing layout from the available list of widgets and primitives (refer Figure 55)

ii. Once added, the size, ID, and position can be changed using the steps mentioned above.

Saving and exporting the layout changes

i. All the layout changes done can be saved to the project file using the **Save** button on the layout editor.

ii. Saving the layout changes updates the existing project file lo_vt.xml.

iii. Upon saving, layout changes can be exported to the .rar and .h files in the ...\RapidApp\Layouts\Export. The files lo_vt.c and lo_vt.h are updated after the export is complete. These files are updated in the EmbdCode folder. These files need to be copied manually to the EmbdCode\src and EmbdCode\inc directory. These files are used by RapidPLUS™ while generating the code for the MMI.

Changes reflected in RapidPLUS

Once the Export has been performed using the layout editor, the xml and c source files which have been updated in the EmbdCode folder are used by the SRV_LAYMAN user object for starting the layout on the screen.

Generating Code and Making the target build

After completing all the changes needed in RapidPLUS™, user can follow the standard build procedure: code generation, code compilation and build download (refer section 5.6).

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.

5.8.1 Multimedia resources – Graphics and Animations

Multimedia resource manager (refer section 2.3.3) is used to add and modify graphics elements and animations to the Wavecom MMI.

This tool is based on an excel template that contains macros for generating IDs for all of the bitmaps. These generated IDs are used by several RapidPLUS™ components. All of the project bitmaps and animations are organized in a separate folder ...RapidApp\Images as shown in the figure below. Res folder is also known as the destination folder from which the application imports the bitmaps. On the other hand Source folder is the source of bitmaps where the user can add, edit or remove the bitmaps. Similarly animations are kept in a subfolder Animations and animation editing is done in this folder.

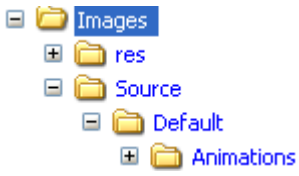


Figure 56: Multimedia Resource Manager- Folder Structure

This section explains the basic steps needed for the customization of graphic elements (Bitmaps and Animations)

5.8.1.1 Updating an existing Bitmap/Animation

Following are the steps to update an existing animation or bitmap:

i. Open the MultimediaResourceManager.xls file in the ...\RapidApp\images folder.

ii. In the settings tab of the file make sure that various components and the source as well as destination folders are set to the default path as shown in the Figure 57.

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.

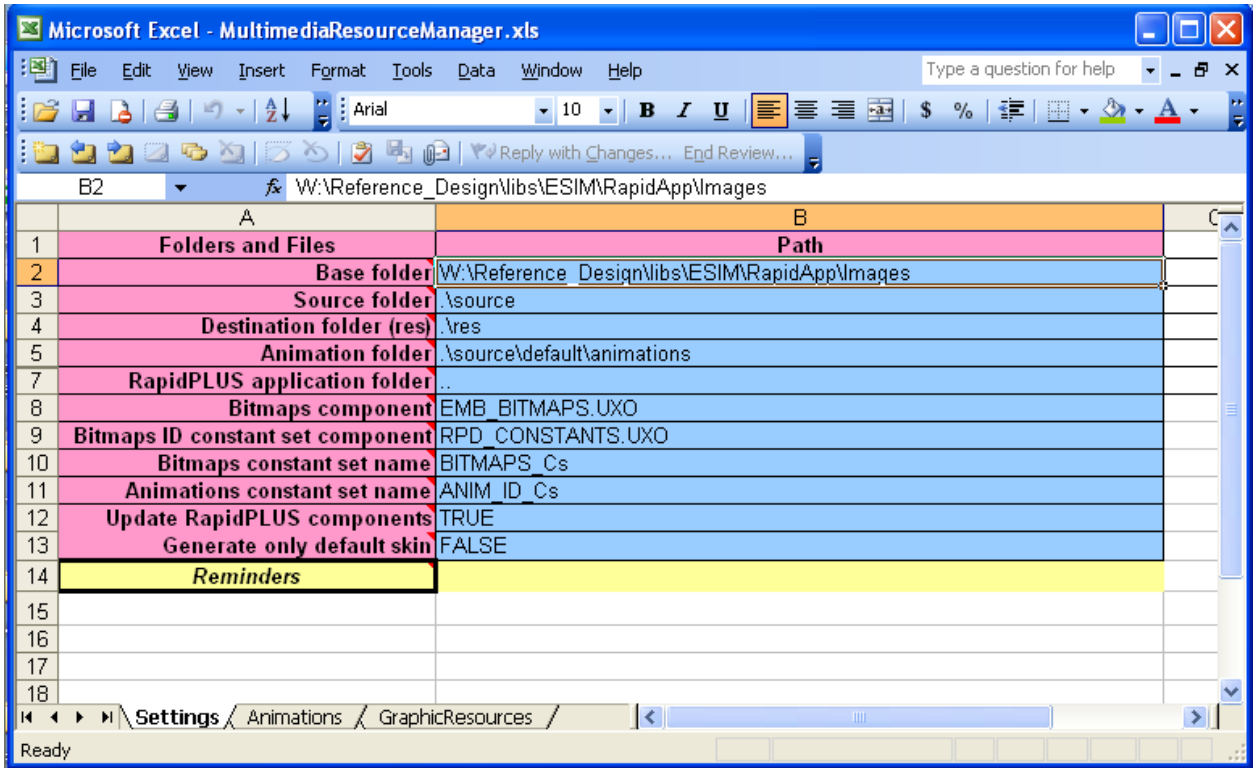


Figure 57: Multimedia Resource Manager – Settings

iii. Edit the image that needs to be changed using any image editor in the source folder and save the image.

iv. Select the GraphicResources tab as shown in Figure 58, and use the import facility of the Resource Manager to import the updated images from the source folder.

v. Once the updated bitmaps have been loaded, import the bitmaps into the RapidPLUS™ Application. The bitmaps are imported into EMB_BITMAPS user object which contains an array which has all the bitmaps that need to be updated by importing the modified bitmaps into the array.

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.

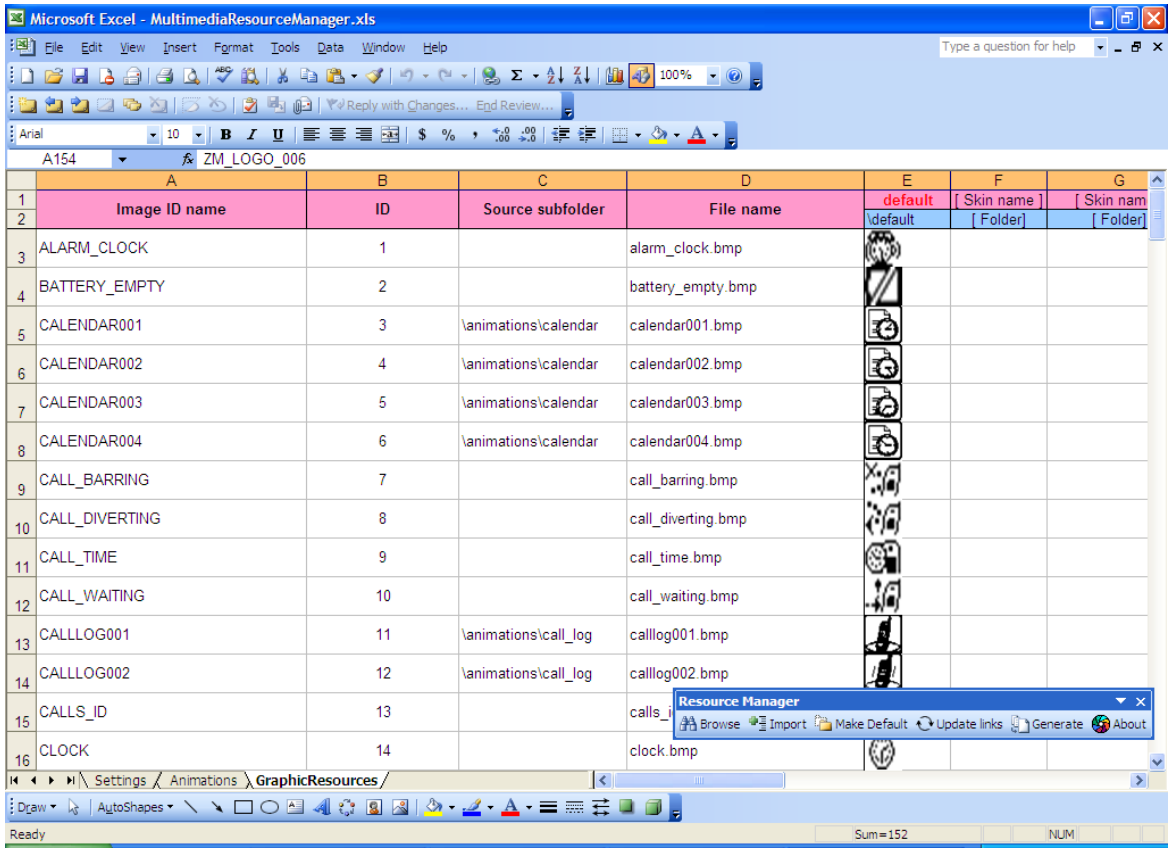


Image ID name	ID	Source subfolder	File name	default	[Skin name]	[Skin nam
				\default	[Folder]	[Folder]
ALARM_CLOCK	1		alarm_clock.bmp			
BATTERY_EMPTY	2		battery_empty.bmp			
CALENDAR001	3	\animations\calendar	calendar001.bmp			
CALENDAR002	4	\animations\calendar	calendar002.bmp			
CALENDAR003	5	\animations\calendar	calendar003.bmp			
CALENDAR004	6	\animations\calendar	calendar004.bmp			
CALL_BARRING	7		call_barring.bmp			
CALL_DIVERTING	8		call_diverting.bmp			
CALL_TIME	9		call_time.bmp			
CALL_WAITING	10		call_waiting.bmp			
CALLLOG001	11	\animations\call_log	calllog001.bmp			
CALLLOG002	12	\animations\call_log	calllog002.bmp			
CALLS_ID	13		calls_i			
CLOCK	14		clock.bmp			

Figure 58: Multimedia Resource Manager- GraphicResources

Note	If the modified bitmap is a part of the set of animation bitmaps (in the ...source\Default\Animations folder) then the changes would be reflected in the animation.

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.

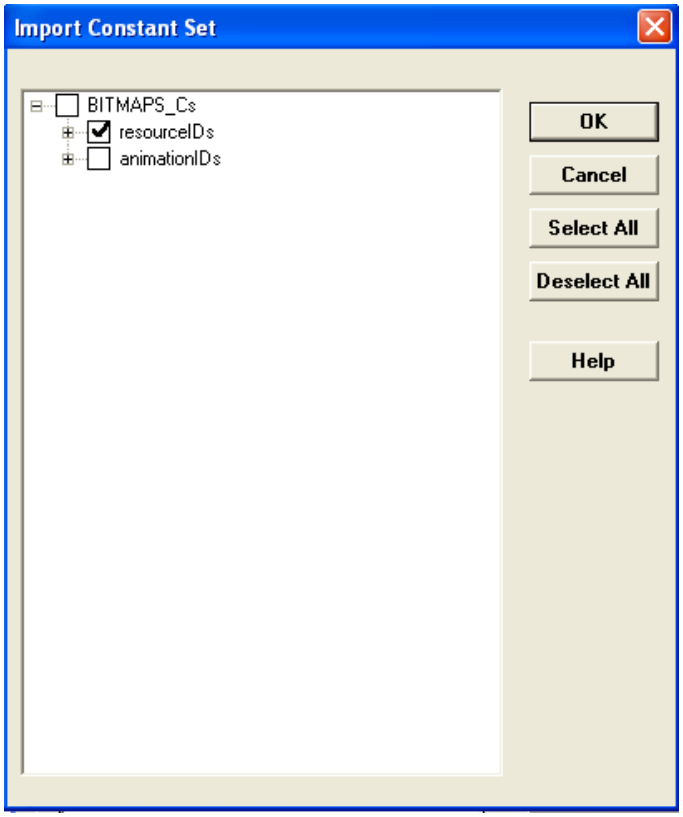


Figure 59: Multimedia Resource Manager – Updating Bitmaps in RapidPLUS™

5.8.1.2 Adding a new Bitmap/Animation

Following are the steps to add a new bitmap/Animation:

i. Open the MultimediaResourceManager.xls file and select the GraphicResources tab. The screen as shown in Figure 58 would be displayed.

ii. Create a new Bitmap image and copy the image to the ...\source folder. Or in case of animations to the ...\source\Default\Animations

iii. Using the resource manager dialog control, import the images from the source folder. This step would also create a new image id. In addition to these, this step also copies the new image to the res folder.

iv. Update the bitmaps array in EMB_BITMAPS user object by importing the new bitmap added.

v. Update the constant set BITMAPS_Cs and ANIM_ID_Cs with the latest bitmap and animation IDs after the addition of the new bitmap as shown in Figure 59

After following the steps the user would be able to use the new bitmap/Animation in the Rapid Application.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement.

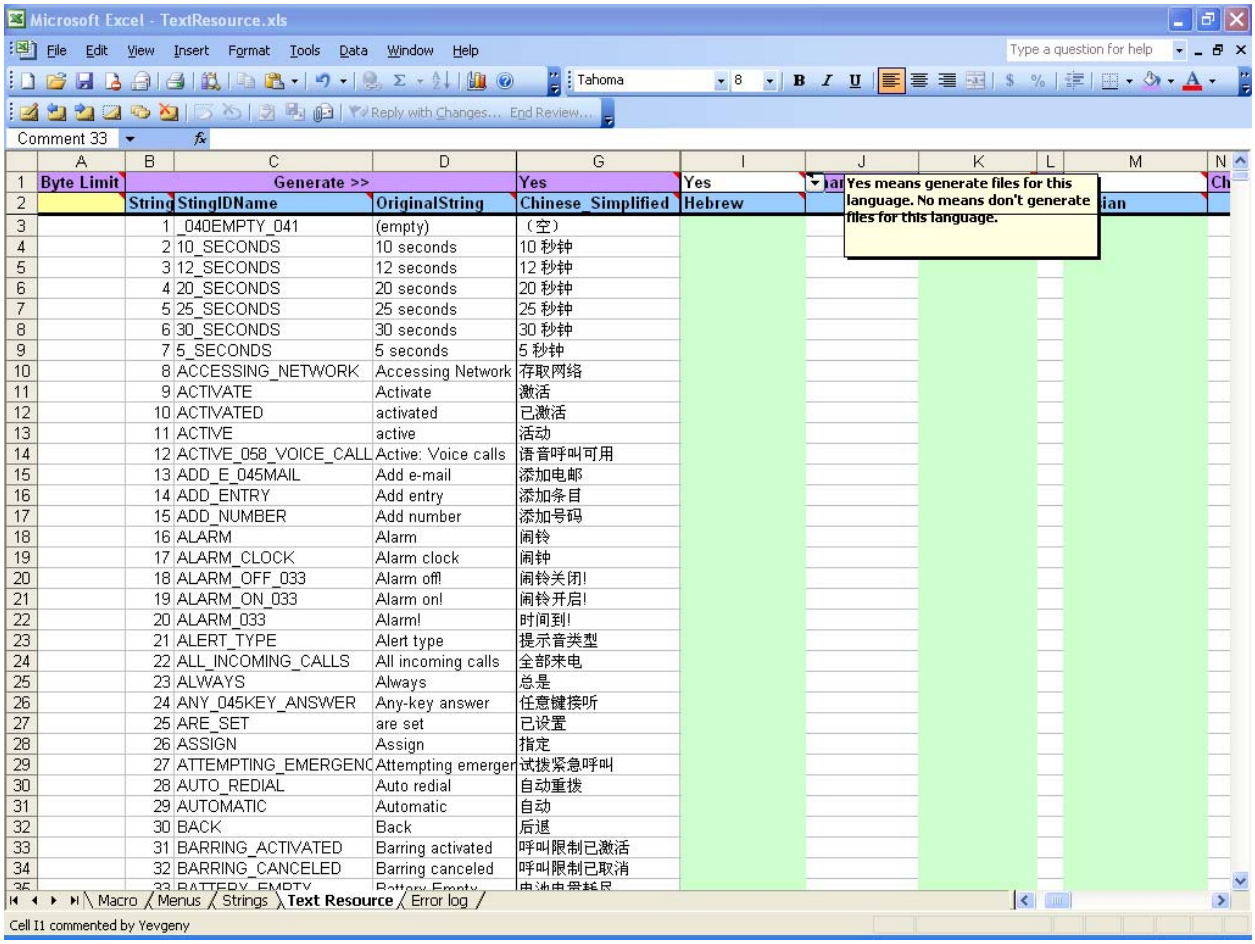
Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

5.8.1.3 Deleting an existing Bitmap/Animation

In order to remove a Bitmap/Animation from the RapidPLUS™ project, remove the bitmap in the source folder. The other steps are same as described for adding a new bitmap in section 5.8.1.2 above.

5.8.2 Text Resources - Support for different Languages

User can select different languages for the MMI. This is done by using the Textresource.xls file. In the textresources tab, user has the option to add labels of any language and then setting the generation flag for that language to yes. The particular language would be generated and updated in the EMB_TEXTRES user object. Once this user object is updated, the language can strings can be used in the RapidPLUS™ application.



	A	B	C	D	G	I	J	K	L	M	N
1	Byte Limit		Generate >>		Yes	Yes					Ch
2		String	StingIDName	OriginalString	Chinese_Simplified	Hebrew				ian	
3		1	_040EMPTY_041	(empty)	（空）						
4		2	10_SECONDS	10 seconds	10 秒钟						
5		3	12_SECONDS	12 seconds	12 秒钟						
6		4	20_SECONDS	20 seconds	20 秒钟						
7		5	25_SECONDS	25 seconds	25 秒钟						
8		6	30_SECONDS	30 seconds	30 秒钟						
9		7	5_SECONDS	5 seconds	5 秒钟						
10		8	ACCESSING_NETWORK	Accessing Network	存取网络						
11		9	ACTIVATE	Activate	激活						
12		10	ACTIVATED	activated	已激活						
13		11	ACTIVE	active	活动						
14		12	ACTIVE_058_VOICE_CALL	Active: Voice calls	语音呼叫可用						
15		13	ADD_E_045MAIL	Add e-mail	添加电邮						
16		14	ADD_ENTRY	Add entry	添加条目						
17		15	ADD_NUMBER	Add number	添加号码						
18		16	ALARM	Alarm	闹铃						
19		17	ALARM_CLOCK	Alarm clock	闹钟						
20		18	ALARM_OFF_033	Alarm off!	闹铃关闭!						
21		19	ALARM_ON_033	Alarm on!	闹铃开启!						
22		20	ALARM_033	Alarm!	时间到!						
23		21	ALERT_TYPE	Alert type	提示音类型						
24		22	ALL_INCOMING_CALLS	All incoming calls	全部来电						
25		23	ALWAYS	Always	总是						
26		24	ANY_045KEY_ANSWER	Any-key answer	任意键接听						
27		25	ARE_SET	are set	已设置						
28		26	ASSIGN	Assign	指定						
29		27	ATTEMPTING_EMERGENC	Attempting emerger	试拨紧急呼叫						
30		28	AUTO_REDIAL	Auto redial	自动重拨						
31		29	AUTOMATIC	Automatic	自动						
32		30	BACK	Back	后退						
33		31	BARRING_ACTIVATED	Barring activated	呼叫限制已激活						
34		32	BARRING_CANCELED	Barring canceled	呼叫限制已取消						

Figure 60: TextResources – Adding a language

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.

Following steps explain the process of adding a new language:

i. Open the file Textresource.xls and select the textresource tab.

ii. Set the generation flag column for the language to Yes and add the labels under the specified language as shown in the Figure 60.

iii. Once all the labels have been added, generate Texresource.xls file using the Menu and Text generator.

iv. Import the new language ID into the LANGUAGE_Cs constant set in the EMB_TEXTRES user object. This ID would now be used by the Rapid Application for setting the language.

5.8.3 Menus - Add/Modify/Delete Menu Items

New Menus can be added and existing menus can be modified and deleted using the textresource.xls.

5.8.3.1 Changing an existing Menu Item

Whenever the user needs to change the menu item for any menu in the Wavecom MMI, the textresource.xls file needs to be updated.

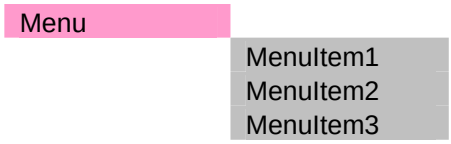


Figure 61 : Menus - Modifying a Menu Item

Following steps need to be followed:

i. Open the Textresource.xls file in the ...\RapidApp folder with the macros enabled.

ii. Select the menu tab in the textresource.xls file.

iii. In the menu tree that is displayed, select the menu item that needs to be renamed and change the name of the menu item as required. For e.g. as shown in Figure 61, rename the first menu item (MenuItem1) and change the name of this menu item to "ModifiedItem" (as shown in Figure 62). Save the Textresource.xls file.

iv. Select the macro tab and click the macro button. Text and Menu generator screen would appear as shown in the Figure 19.

v. Press the **Start** button after making sure that the correct options have been selected or default options as shown in the Figure 19.

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.

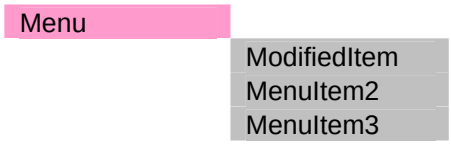


Figure 62: Menus - Modified Menu Item

vi. On completion of the menu generation, the constant set STRING_ID_Cs in the EMB_TEXTRES.UXO would be updated with the new string.

5.8.3.2 Adding a new Menu Item to an existing menu

Following are the steps to add a new Menu Item in an existing menu:

i. Open the TextResource.xls file and select the menu tab.

ii. From the menu tree, select the menu item above which a new Menu Item has to be added.

iii. Right click on the selected menu item and select the insert option to insert a new row in the excel sheet.

iv. Add the menu item along with the following fields given below (refer Figure 63 which shows an example of new Menu item added under Sound Settings Menu):

   a. **LeftSK**: String for the Left Soft key that would be displayed when this menu item would be selected

   b. **RightSK**: String for the Right Soft key that would be displayed when this menu item would be selected

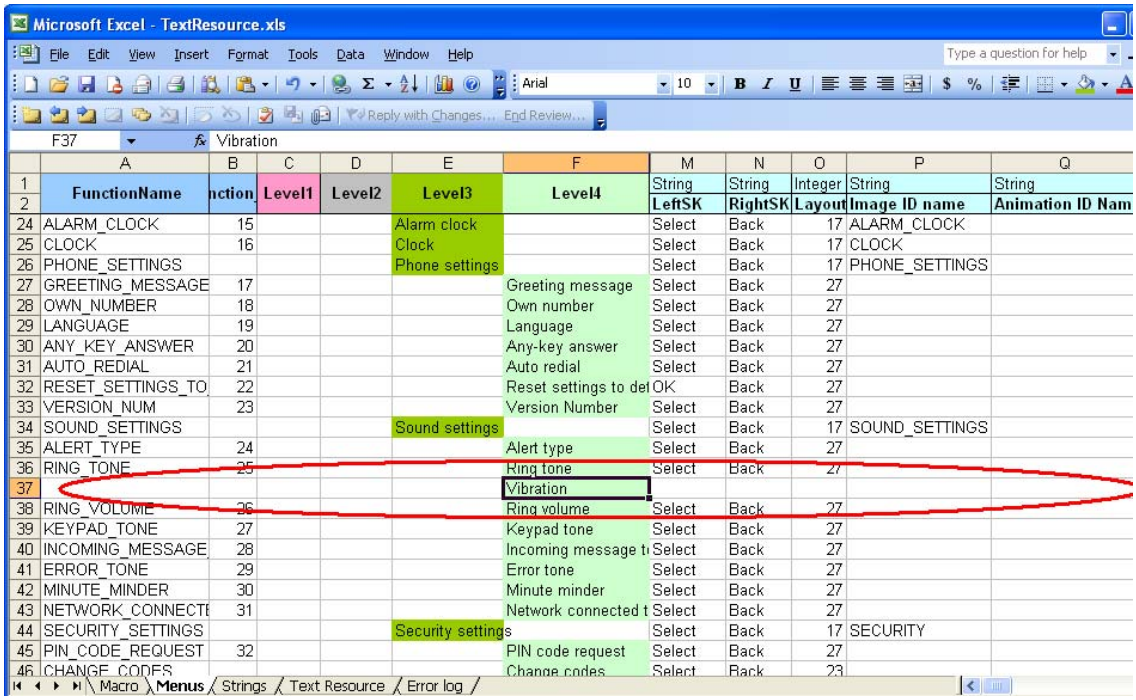
   c. **Layout**: Numeric ID of the Layout that would be started on selection of this menu item.

   d. **Image ID**: Numeric ID of the image that we need to associate with the menu item.

   e. **Animation ID**: Numeric ID of the animation that we need to associate with the menu item.

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.



	A	B	C	D	E	F	M	N	O	P	Q
1	FunctionName	nction	Level1	Level2	Level3	Level4	String	String	Integer	String	String
2							LeftSK	RightSK	Layout	Image ID name	Animation ID Nam
24	ALARM_CLOCK	15			Alarm clock		Select	Back	17	ALARM_CLOCK	
25	CLOCK	16			Clock		Select	Back	17	CLOCK	
26	PHONE_SETTINGS				Phone settings		Select	Back	17	PHONE_SETTINGS	
27	GREETING_MESSAGE	17				Greeting message	Select	Back	27		
28	OWN_NUMBER	18				Own number	Select	Back	27		
29	LANGUAGE	19				Language	Select	Back	27		
30	ANY_KEY_ANSWER	20				Any-key answer	Select	Back	27		
31	AUTO_REDIAL	21				Auto redial	Select	Back	27		
32	RESET_SETTINGS_TO	22				Reset settings to def	OK	Back	27		
33	VERSION_NUM	23				Version Number	Select	Back	27		
34	SOUND_SETTINGS				Sound settings		Select	Back	17	SOUND_SETTINGS	
35	ALERT_TYPE	24				Alert type	Select	Back	27		
36	RING_TONE	25				Ring tone	Select	Back	27		
37						Vibration					
38	RING_VOLUME	26				Ring volume	Select	Back	27		
39	KEYPAD_TONE	27				Keypad tone	Select	Back	27		
40	INCOMING_MESSAGE	28				Incoming message t	Select	Back	27		
41	ERROR_TONE	29				Error tone	Select	Back	27		
42	MINUTE_MINDER	30				Minute minder	Select	Back	27		
43	NETWORK_CONNECTI	31				Network connected t	Select	Back	27		
44	SECURITY_SETTINGS				Security settings		Select	Back	17	SECURITY	
45	PIN_CODE_REQUEST	32				PIN code request	Select	Back	27		
46	CHANGE_CODES					Change codes	Select	Back	23		

Figure 63: Menus – Adding a Menu Item to an existing menu

v. Go to the Macro tab and start the macro generation process.

vi. On completion of the Text and Menu Generation Macro, Function_INDEX_Cs and FUNCTION_ID_Cs constant sets in the EMB_TEXTRES user object are updated.

Note	The constant sets in the EMB_TEXTRES are updated from the LoadTo_ConstantSet.h, which is updated whenever the Text and Menu generator macro is run with certain changes in menus and textresources.

5.8.3.3 Creating a new Menu (including menu items)

Following are the steps to create a new menu:

i. Adding a new Menu in the Textresource.xls file.

ii. Adding menu items under the menu.

The new menu is implemented in the RapidPLUS™ application using WMI_MENU user object to start the submenu.

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.

Adding a new Menu and its menu items

i. Follow the steps given in section 5.8.3.2 to create a menu item.

ii. In order to add a submenu to the menu item, insert the new menu items as shown in the Figure 62. i.e. menuitem1 would be placed column+1 and row+1 relative to the Parent menu item. Further menu items should be added by incrementing the rows in the same column of the excel worksheet.

iii. Once the menu items have been added this menu can be started by calling the start function of WMI_MENU. The parameters to the function call are the function index of the first menu item.

- E.g. WMI_MENU start: emb_textres_h.FUNCTION_INDEX_Cs.IX_SHOW_CLOCK

Menu customization is also explained in section 2.3.2

5.8.3.4 Files updated by the Textresource.xls Macro

The following points list the files which are updated by the Menu and Text Generator for RapidPLUS™.

i. Datafiles updated for the RapidPLUS™ project:
   a. CodePage.rar
   b. Menu_Index.rar
   c. Menu_Unicode.rar
   d. LoadTo_ConstantSet.h

ii. Datafiles updated for the embedded system:
   a. cLng_Chinese_Simplified.c
   b. cLng_English.c
   c. LoadTo_ConstantSet.h
   d. xl_IndexArray.c
   e. xl_menu.h

5.8.4 Keypad

Before we go into customization of Keypad, let us first understand what all methods have been used to integrate the Keypad and which files have been involved (refer section 5.2.1.2). MMI application subscribes for all the keypad events using an ADL API *adl_atUnSoSubscribe.* This can be seen in *DoInit* method in the file oat_appli.c.

```
adl_atUnSoSubscribe ("+CKEV:", CKEV_Handler);
```


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.

+CKEV are the messages that are subscribed for. These messages will be handled in CKEV_Handler method in the same file.

CKEV_Handler calls methods processKeyDown and processKeyUp for processing the key inputs. These methods are implemented in ...\emdbcode\src\app_api.c file.

The processKeyDown method interacts with EMB_KPD which is implemented as a RapidPLUS™ UDI object. On every keypress, code_Int property of EMB_KPD is set corresponding to the code of the key pressed and a key press event is triggered.

On getting the key press event from EMB_KPD, SRV_KPD (implemented as a RapidPLUS™ UDO) informs the above MMI about the key press and code of the pressed key.

MMI keypad can be customized as per the keys available on the target keypad. Keypad customization involves:

i. Adding\deleting\modifying key codes in the constant set KEY_CODES_Cs in EMB_KPD.UXO

ii. Update the switch case in the processKeyDown method in app_api.c to include a new case or modify the existing case accordingly.

iii. Add\update a corresponding event in SRV_KPD (SRV_KPD.UXO) so that the updated key can be recognized by the MMI.

5.9 Debugging using RapidPLUS™ and Wavecom tools

Open AT® Software applications are debugged using the E-SIM simulation mode, RTE (Remote Task Execution) mode and Target Mode.

E-SIM Simulation Mode: This is the debugging that is done in the first stage of the application development in the RapidPLUS™ environment. In this mode the application is debugged on the development platform (PC) itself. The Wireless CPU is connected to the serial port and is used as a modem for sending and receiving AT commands. The application debugging at this stage is least expensive in case of large applications. Application developer should try to identify any application or logical errors at this stage itself. This mode is explained in more detail in section 6.3.7.

 Note	Although the mode is called Simulation mode, it cannot be run on a standalone basis without Wavecom Wireless CPU.

RTE Mode: Once the application is successfully built using the RTE project settings, the application can be debugged in the remote mode. In remote mode, the application runs on the PC and uses its resources such as memory and clock

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.

with the serial link setup with the Wavecom CPU, hence we are able to set the breakpoints for debugging in this mode. (More on RTE is explained in section 4.4)

TARGET Mode: In the target mode, the application runs completely on the target and uses its resources. Debugging in this mode is possible only using the target monitoring tools such as TMT and TE. These tools have been described in section 4.2 and 4.3 respectively and are tools provided to assist the debugging of the application especially in the Target mode. TMT is used for watching the traces and other parameters to monitor the state of the target application. Similarly TE is used to start stop the application, and also to display the commands sent to the target and responses received from the target.

Note	Traces can only be displayed on the TMT screen provided certain flags are uncommented in the **DebugAPI.c** file in the…\EmbdCode folder. The changes that need to done are explained in section 6.4.2.2.

5.10 Wavecom Sample MMI Limitations

5.10.1 Keypad Handler

The number of keypad events passed to E-SIM engine is limited. When the user presses too many keys, adl_atUnSoUnSubscribe is called to unsubscribe further key press event. In about 0.6 seconds, it will recall adl_atUnSoSubscribe to subscribe again. This blocking is done in oat_appli.c. The function Keypboard_TimerHandler has the logic to prevent excess key press event being passed to ESIM.

5.10.2 Battery & Charger Handler

If the user wants to use the same board for simulation and is running the MMI on hardware, the user must turn off the unsolicited battery status before switching the use. For example, when the user is using the board simulation, the indication (AT+WBCM=0, 0) must be turned off using hyper terminal before switching the OAT GTi application on (AT+WOPEN=1). When the MMI is running on the board, user must turn off the Wireless CPU by pressing down the power off button before the turning off the application (AT+WOPEN=0). The application itself will send AT+WBCM=0, 0 to the Wireless CPU.

5.10.3 SMS Handler

SMS handler has to be handled differently than the other service handler. The text must be sent with ADL API adl_atCmdSendText rather than using the AT command AT+CMGS for sending the SMS text.

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.

5.10.4 Phonebook

Wavecom OS supports two phonebooks (ME and SIM) but only one can be activated at the same time. If a name A1 is in ME phonebook but SIM phonebook is selected, then if we receive a call from A1 his name will not be displayed.

5.10.5 Timer Handler

Timer ticks sent by Open AT® Software are not accurate. e-SIM system clock, running as a part of Open AT® Software application, is based on the tick returned by Open AT® Software. Due to inaccuracy of timer ticks, it is necessary for the MMI to obtain the accurate time from the CPU via AT command periodically. This has been implemented in SRV_GSRV.

5.10.6 Multiple simultaneous AT commands

Open AT ® GTi application is sending AT commands to Wavecom OS to perform some actions. If an external HOST also sends AT commands to Wavecom OS, there might be some interaction problems.

In case of two commands of a same class (Phonebook, SMS), one will be rejected. However, it is highly recommended to avoid sending AT commands at the same time from Open AT ® GTi application and from external HOST to avoid interaction issues.

5.10.7 DOTA impact

Open AT® software platform offers integrated download over the air (DOTA) capability. The 'Download over the Air' service provides the functionality to upgrade existing/running embedded applications or Wavecom core software remotely over the air.

DOTA Type I is 'Download Over The Air' of **Open AT® GTi application**.

DOTA Type II is 'Download Over The Air' of **Wavecom OS**.

5.10.7.1 Q2406B (based on OS 6.56)

With both Chinese and English, the size for Wavecom Sample MMI (Version 250 – B&W LCD and Q2406B) is 820KB. It includes Unicode2GB and GB2Unicode table. Those tables have a size of 120KB.

With English only (no Chinese fonts and GB2Unicode tables), the size for Wavecom Sample MMI (Version 250 – B&W LCD and Q2406B) is 456KB.

Hence DOTA I is possible based on Q2406B / X55 but DOTA II is **not** possible as Open AT® Software application shall exceed 256 KB in this case.

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.

5.10.7.2 Q2686H (based on OS 6.60)

Q2686H memory size can have up to 1.5 MB for Open AT ® application. Hence DOTA I is possible based on Q2686H/OS6.60 but DOTA II is not possible as Open AT® application shall be exceed 256 KB in this case.

5.10.8 Language Support

Open AT® GTi does not support languages like Thai, Hebrew and Arabic that require supplementary software development tools.

5.10.9 SIM Tool kit

Wavecom Open AT® GTi does not support SIM Toolkit functionality.

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.

6 Sample Application Development

This section is aimed towards understanding application development on Open AT® GTi and various steps involved in creating a simple application including development, code generation, debugging and deployment.

6.1 Requirement

The requirements for creating a simple application are taken as:

- Creating a clock application for displaying the clock and date on the screen.
- User should be able to set the date and time.
- User should be able to set an alarm.
- Alarm should ring at the set time by playing a melody and showing an alarm pop-up on the display.
- User should have an option to stop the alarm.

6.2 Design

The process of integrating the Alarm Clock application MMI with Wavecom Open AT® Software platform is based on the similar architecture as the Wavecom Sample MMI partly re-using the already integrated AT layer as well. Same workspace provided with Wavecom Sample MMI is used for the development of Alarm Clock Application.

The integration of an MMI over the hardware platform involves:

i. Basic integration - Basic services include Keypad, Timer, Display, Backlight and Synthesizer

ii. Embedded interface component - The components that act as a glue between MMI and the hardware platform

iii. Service components - The MMI layer components that are responsible for providing functionality of the implemented features.

Basic Integration involves integration of platform keypad with the RapidPLUS™ MMI application keypad, integration of RapidPLUS™ graphical display object (GDO) with the platform LCD and RapidPLUS™ timer with the system timer. To speed up the development process, the sample MMI code has been used and customized for the Alarm application.

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.

Embedded interface component involves AT as well as flash read & write components. AT integration from the Wavecom Sample MMI has been used for the development of this Alarm Clock application. Flash read write is being done through EMB_FLASH component developed for this application.

Service components include various MMI components that are required for drawing the layouts, updating the display, showing the menus, showing the pop ups etc. Existing MMI components can be re-used to speed up the development process. The Alarm Clock application has been developed as a RapidPLUS™ application which uses various other generic MMI components like SRV_DISPLAY, SRV_LAYMAN, WMI_MENU, WMI_EDITOR etc.

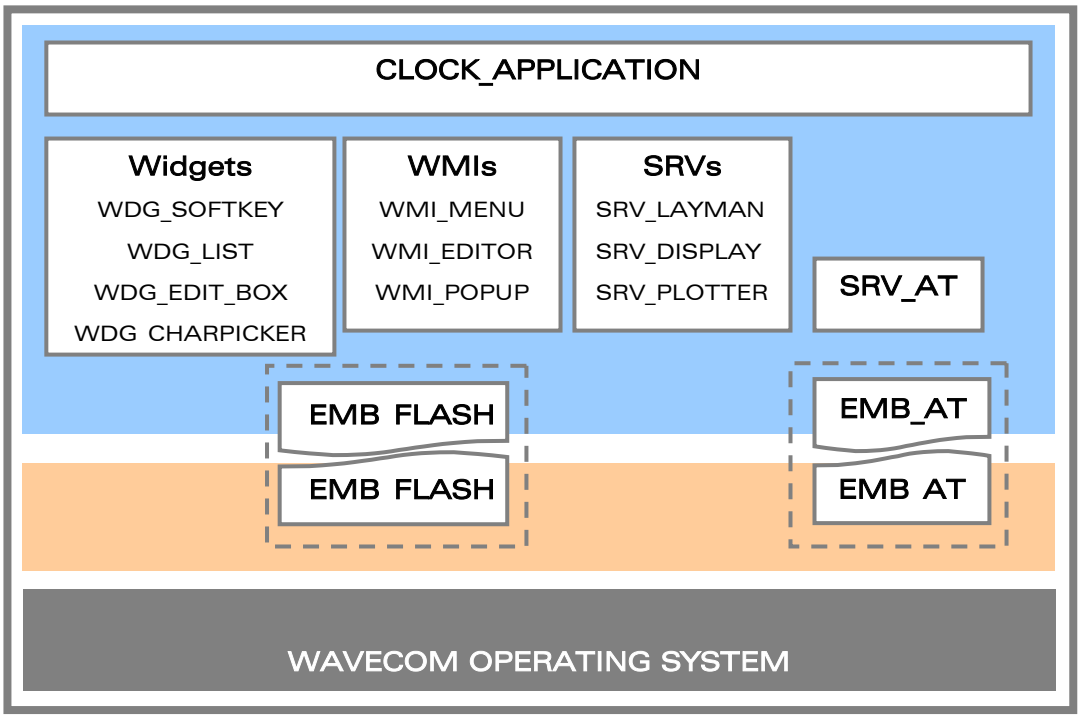


Figure 64: Clock Application – MMI Design

The Alarm Clock application uses the following generic MMI components from the Wavecom Sample MMI:

- SRV_LAYMAN

This module is responsible for creating and destroying layouts on the screen. It enables the opening of more than a single layout at a given time and creates widget instances as required. It allocates memory and initializes the widgets. After the client requests layout initialization, the client receives the widget holder and communicates directly with the widget.

- SRV_PLOTTER

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.

This component (referred to as the Plotter) supplies the means to draw the primitive elements (text, bitmaps, and shapes) defined in the Layout Editor.

- SRV_DISPLAY

The SRV_DISPLAY component provides all the drawing functionality required by the MMI Application. It wraps the EMB_DISPLAY component (which actually performs all the drawing functionality and wraps EMB_BITMAPS). The physical embedded display resides outside the RapidPLUS™ domain and is accessed via API calls.

- EMB_TEXTRES

The embedded interface component EMB_TEXTRES is a text and menu resource. The menus and strings are defined in TextResource.xls.

- RPD_CONSTANTS

The RPD_CONSTANTS component is a collection of constant sets that are used to assign sizes of elements or common enumerations throughout the MMI Application.

- WMI_POPUP

The WMI_POPUP module encapsulates the functionality of a popup window.

- WMI_EDITOR

This component processes end-user inputs and edits the output text using WDG_TEXTBOX.

- WMI_MENU

The WMI_MENU module allows navigation in the menu tree. The menu tree is defined in the Menu and Text Generator tool (TextResource.xls).

- RPD_TRACE

The RPD_TRACE component provides functionality equivalent to the 'C' assert () function.

- WDG_LIST

The WDG_LIST component controls the display of items that are of the same data type (e.g., contacts in the phonebook). The items are displayed as a collection of vertically arranged fields with the first field initially highlighted. When the number of fields to be displayed exceeds the number of fields viewable on a single screen, a scroll bar appears.

- WDG_SOFTKEY

This component is used to process the operations and events on softkeys e.g. Left softkey pressed event and right softkey pressed event.

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.

6.3 Alarm Clock MMI development

e-SIM MMI development tools are used for developing the Alarm Clock application MMI (Refer section 2)

6.3.1 Layout Development

Layout development for the Alarm Clock is done using the Layout Editor tool. Alarm clock application contains 4 application screens:

- Idle screen
- Clocking setting screen
- Menu Screen
- Alarm pop-up screen.

6.3.1.1 Idle Screen Layout

To display the current time and date on the idle screen, this layout contains two primitive type text fields as shown in the figure below.

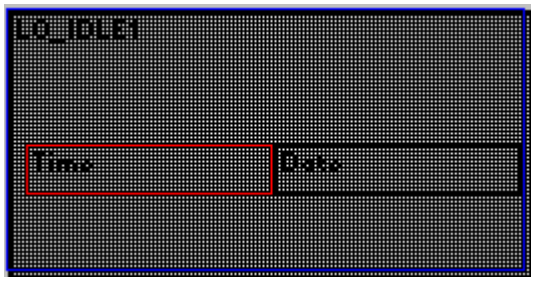


Figure 65: Clock Application - Layout for Idle Screen

6.3.1.2 Date and Time Setting Layout

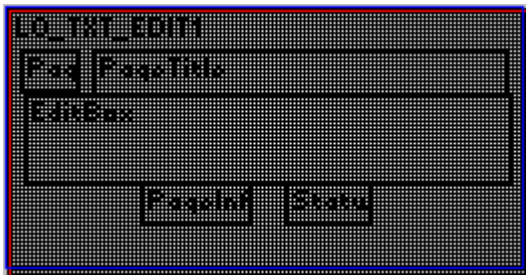


Figure 66: Clock Application – Layout for WMI_EDITOR

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.

As shown in the figure above, this layout would contain the following controls:

i. Edit Box Widget control. (This is the control that would be used by the WMI_EDITOR)
ii. Text and picture controls for displaying any additional text or picture.

6.3.1.3 Menu List Layout

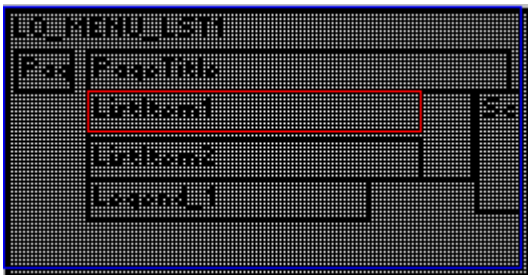


Figure 67: Clock Application - Layout used by WDG_LIST

For this layout making a list control, the following items/controls are required.

i. List widget Control (List)
ii. List Items (Maximum of 2 list items are allowed for one view). In addition there should be List Item for List highlighted overlapping these list item controls. This option is needed to represent the list with colors inverted when the item is selected.
iii. Scroll bar widget control on the right part of screen.
iv. Page Title (text) and Page Icon (Picture) to provide info about the page.

6.3.1.4 Popup Layout

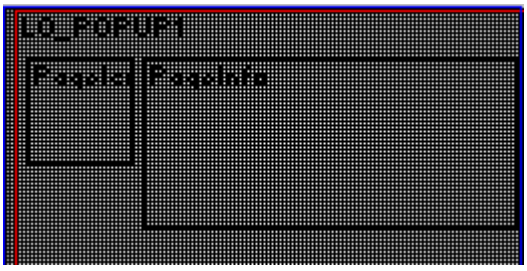


Figure 68: Clock Application - Layout for WMI_POP

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.

For this layout the following controls are needed.

i. PageInfo control(Text) and PageIcon control(picture)

ii. Depending upon the context of the popup i.e. Alarm Set, Time Set, Date Set etc, the popup layout is started and PageInfo control is populated with the respective values.

Once all the layouts are created in the Layout Editor tool, these layouts should be exported using the export option. It will then update the lo_vt.xml file in the Rapid/Layouts/layouts folder. RapidPLUS™ uses the updated layouts from this xml file in the application (Refer section 2.3.1 for the use Layout Editor). The files lo_vt.c and lo_vt.h are updated after the export is complete. These files are updated in the EmbdCode folder. These files need to be copied manually to the EmbdCode\src and EmbdCode\inc directory.

Note	In case a new Layout is added, a new init method by name initLO_<LayoutName> is generated in the SRV_LAYMAN component. This method is used to initialize the new layout added to the project. If an existing layout has been modified, this new function is not required.

6.3.2 Creating Menus and Text Resources

This section explains various steps needed to create menus and text resources for the Alarm Clock application.

6.3.2.1 Creating Menus

There are two menus in the Alarm Clock application as shown below. As is clear from the figures below, the choice of menu to be started would depend upon whether the clock is hidden or shown.

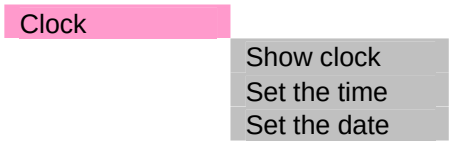


Figure 69: Clock Application - Menu on hidden clock

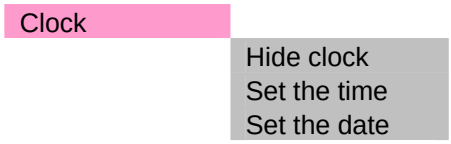


Figure 70: Clock Application - Menu on shown clock

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.

The menus are created in the textresource.xls file by selecting the menu worksheet. Once the menu data has been added to the textresource.xls file, the generation is done using the resource generation macro provided in that excel worksheet (refer section 2.3.2)

6.3.2.2 Creating Text Resources

Text resources are all the text strings that are used in the application. All the text strings should be updated in the Strings worksheet in the Textresource.xls file.

E.g. for adding an alarm text resource in the application, following fields have to be populated in the textresource table given below:

i. Update the Strings worksheet in the textresource.xls file. Under the Strings column a string needs to be added as shown below.

Group	String
	Alarm on!

Figure 71: Clock Application - Adding string

ii. Add alarm ID name and original string to the Text Resource worksheet.

iii. Update the Chinese conversion of the text in case the application uses Chinese fonts.

C	D	E	F	G
Generate >>		Yes	Char	Yes
StingIDName	OriginalString	English		Chinese_Simplified
CLOCK	Clock	Clock		时钟
HIDE_CLOCK	Hide clock	Hide clock		隐藏时钟
SET_THE_DATE	Set the date	Set the date		设置日期
SET_THE_TIME	Set the time	Set the time		设置时间
SHOW_CLOCK	Show clock	Show clock		显示时钟

Figure 72: Clock Application - Adding TextResource

6.3.2.3 Menu and Text Generation

e-SIM text and menu generator Macro is run after making the changes in the previous steps. EMB_TEXTRES user object is updated with the latest menu and text additions done in the textresource.xls file. These text resources could then be used in the Alarm Clock application.

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.

6.3.3 Adding Multimedia Resources

The icons and other bitmaps needed for the Alarm Application are specified in this file.



Figure 73: Clock Application - Alarm Signal bitmap added

As described in section 2.3.3 above once the multimedia bitmap is added, the update links option should be used to update the user object EMB_BITMAPS in the Application. We would be able to use the alarm icon in the Application.

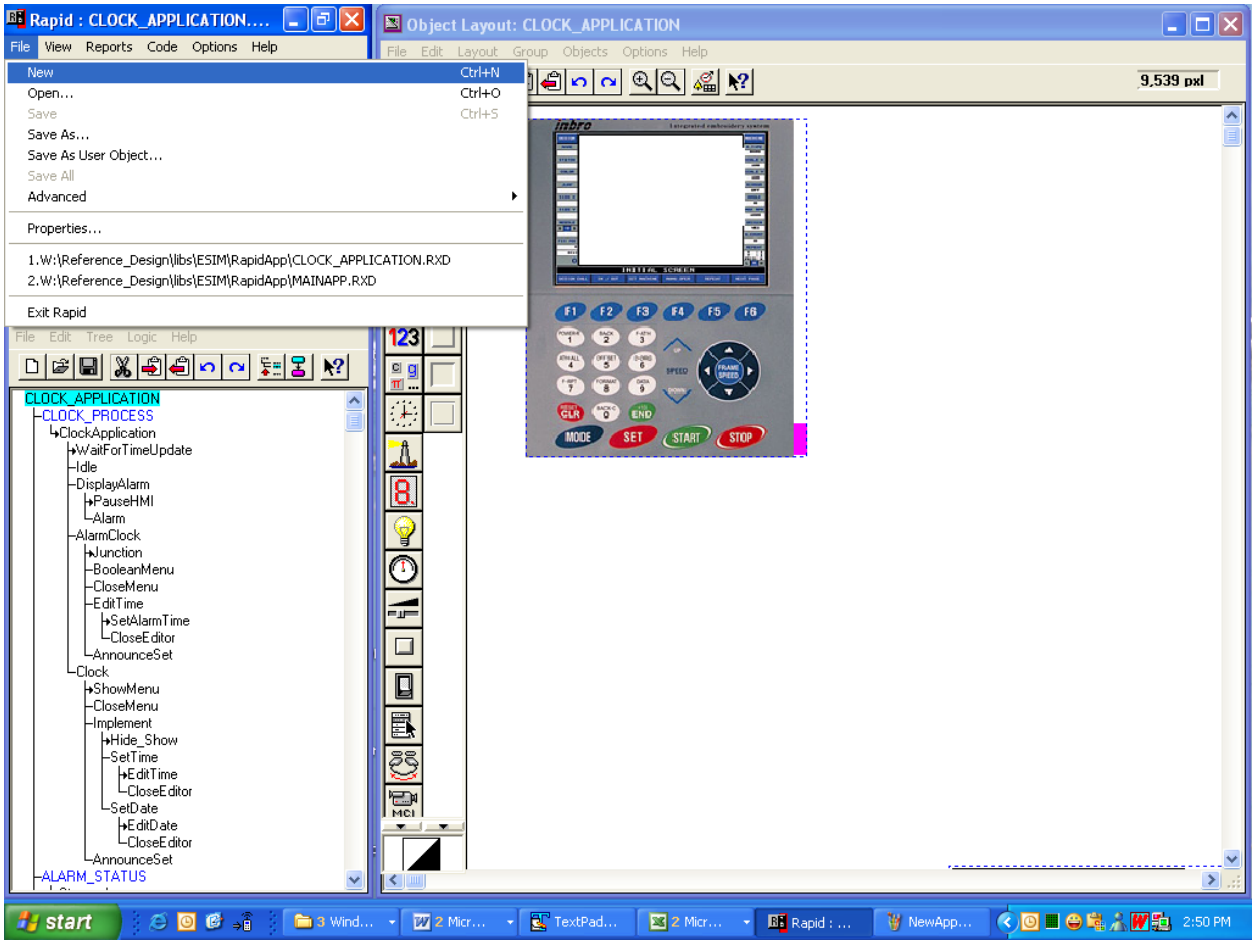


Figure 74: Clock Application - Creating RapidPLUS™ project

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.

6.3.4 Creating RapidPLUS™ Project

Open Rapid PLUS and go to File → New menu item and press Enter. This would create a new Rapid Application. Save the new application as CLOCK_APPLICATION.RXD.

6.3.5 Adding RapidPLUS™ development components

The development components such as textresources, multimedia resources and layouts have already been updated with the strings, graphics and layouts as per section 6.3.1, 6.3.2, 6.3.3.

The following folders and files in Wavecom Reference MMI RapidApp should get updated after all the changes are done in the above sections.

- Layouts Folder (Files updated by Layout Editor)
- Images Folder (MultimediaResourceManager.xls and folders containing the Multimedia Images)
- TextResource.xls
- TextResource.xml
- Textresource.xsd

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.

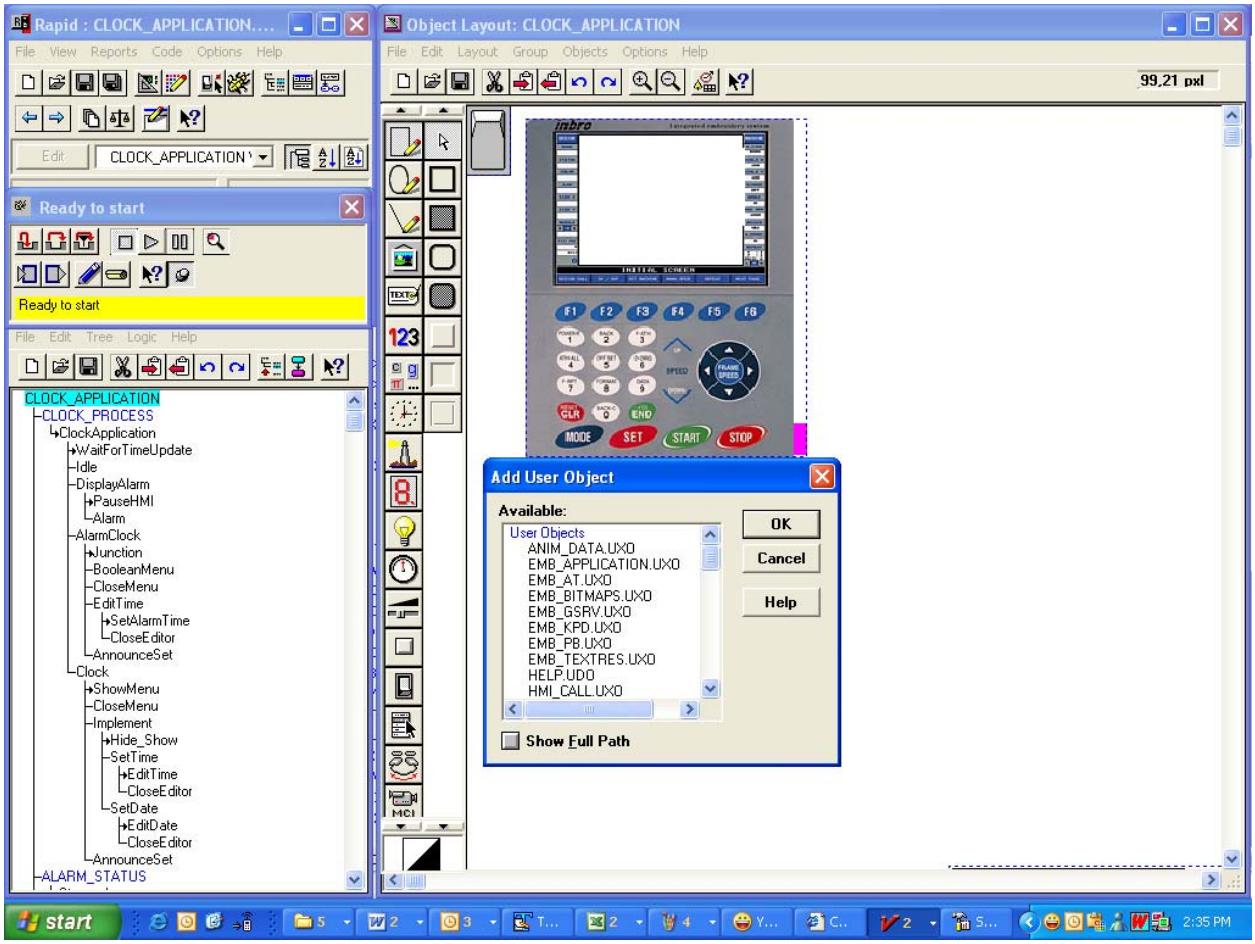


Figure 75: Clock Application - Adding User Objects

All the reusable components mentioned in the list have to be added to the new project, one-by-one, using the 'Add User Object' menu option on the RapidPLUS™ Object Layout Window. The figure above shows how to add the RapidPLUS™ objects to the application.

Note	The RapidPLUS™ objects appearing in the 'Add User Object' window are those that exist in the same folder as the project file i.e. CLOCK_APPLICATION.rxd.

6.3.6 Adding application modes

The application logic for the Alarm Clock has various modes in the RapidPLUS™ application. Each mode represents a state of the alarm clock and is associated

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.

with entry activities, transitions and exit activities. The transitions between the various modes take place on the trigger of an event or on a condition being true.

As shown in the Figure 76, there are 5 concurrent modes in the mode tree.

- CLOCK_PROCESS
- Main logic of clock application. (Setting the alarm, time and date and also displaying the MMI for Alarm ringing)
- ALARM_STATUS
- Takes care of Alarm status. i.e. starting and stopping the melody and setting and resetting the alarmOn variable.
- CLOCK_UPDATE
- Internal timer for the application that would update the clock after the clock initialization from the target.
- CLOCKINIT_SETTINGS
- Initializing the clock once and updating the local variables with current date and time whenever the main application sends a request to change the time and date.
- AT
- AT Dispatcher for sending AT commands for setting time, getting time, starting and stopping melody etc.

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.

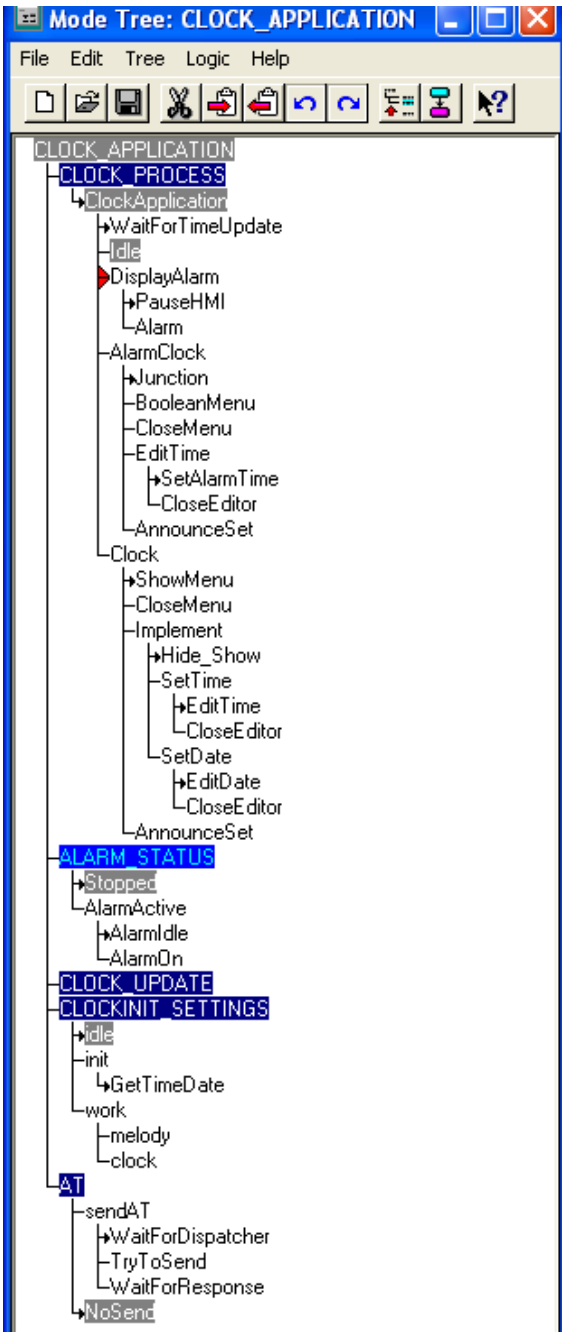


Figure 76: Clock Application - Mode Tree

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.

6.3.6.1 Clock Process Mode

The default mode for this mode is ClockApplication (refer Figure 76). It contains three main modes and their sub trees.

- DisplayAlarm
- AlarmClock
- Clock

6.3.6.1.1 Display Alarm Mode

The control would come to this mode in case the alarm goes off. It has 2 child modes. (PauseHMI and Alarm)

i. In the PauseHMI mode any other layout that is open at that time is paused using the WMI_EDITOR, WMI_MENU or WMI_POPUP pause function.

ii. In the Alarm mode, Alarm layout is started using SRV_LAYMAN start function. Once the user presses left Soft Key the Alarm layout would be stopped and the application goes to the idle mode.

6.3.6.2 Alarm Clock Mode

Control comes to this mode from the idle mode whenever the Alarm left soft key is pressed.

If the alarm is already set then the control would go to the **BooleanMenu** mode and prompt the user to either cancel the alarm (Off) or set another time (On) option from the BooleanMenu as shown in the figure below.

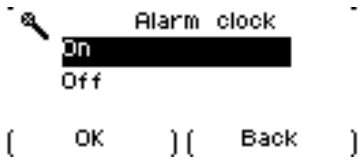


Figure 77: Clock Application - Menu to cancel or set Alarm

Note	In the action of the transition to AlarmClock/EditTime mode, WMI_EDITOR is started with the previous set alarm time displayed in the editor.

If the alarm is not set already then the control does not go to the BooleanMenu mode and goes to EditTime mode. This mode has two child modes: SetAlarmTime and CloseEditor mode.

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.

The default mode is SetAlarmTime. In this mode WMI editor layout is started where the user is able to set a new time for the alarm or exit from the layout using back Soft Key. WMI Editor Screen is shown in the figure below.

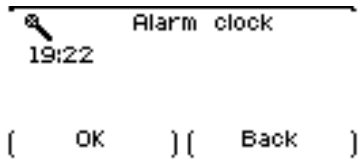


Figure 78: Clock Application - Alarm Set (WMI_EDITOR)

In case the user presses the back Soft Key WMI Editor Layout is stopped and the control goes back to the idle screen. However if the user sets a new time and presses OK i.e. Left Soft Key the control would go to the CloseEditor mode. In this mode WMI editor is stopped, UDD_SET user data object is populated with the new time settings and the control goes to the AnnounceSet Mode. In this mode WMI_POPUP popMessage API is used to display a popup control to announce that the new alarm time has been set as shown in the figure below.

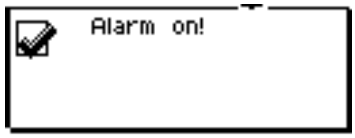


Figure 79: Clock Application - Alarm Set Popup

6.3.6.2.1 Clock Mode

The **clock** mode has child modes which perform the functions to set the date and time and to hide or show the clock.

The default mode under the clock mode is the **ShowMenu** mode. In the entry activity of this mode WMI_MENU start function is called. This function starts an options menu as shown in the figure below.

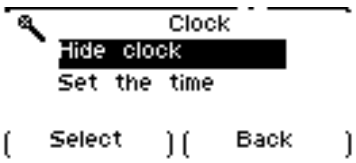


Figure 80: Clock Application - Clock Setting Menu

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.

Depending upon which menu item is selected by the user the control goes to the following three modes.

i. Hide_Show

The control comes to this mode in case the user has selected the menu item to hide or to show the clock. In the entry activity of this mode UDD_SET ClockHide variable is set to False if clock hide has been selected and vice versa. The hidden clock screen would look as shown in the figure below.

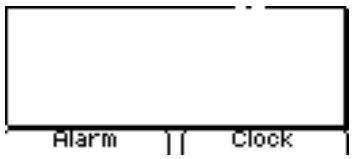


Figure 81: Clock Application - Hidden Clock

ii. SetTime

On selecting the 'set the time' menu item, the control goes to this mode and current time is updated into the local variables using the **self getTime_Hour:** function in the action of the transition from **CloseMenu** mode to **SetTime** mode. In the entry activity of the **SetTime** mode the WMI_EDITOR is started and user is able to set the time as shown in the figure below.

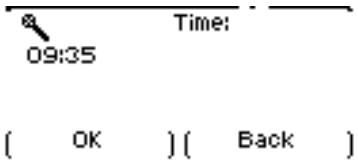


Figure 82: Clock Application - Set Time Editor

iii. SetDate

On selecting the 'Set the Date' menu item, the control goes to this mode and current date is updated into the local variables using the self getDate_Day: function in the action of the transition from CloseMenu mode to SetDate mode. In the entry activity of the SetDate mode the WMI_EDITOR is started and user is able to set the Date as shown in the figure below.

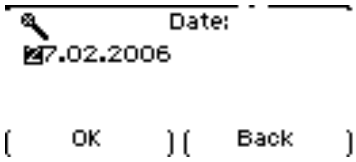


Figure 83: Clock Application - Set Date Editor

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.

6.3.6.3 Alarm Status mode

Alarm Status has 2 modes.

Stopped and Alarm Active

i. Stopped
This is the default mode when the alarm is not set. Whenever the alarm is set i.e. UDD_SET alarmOn property is set to true, the control goes to the AlarmActive mode.

ii. Alarm Active
This mode has 2 child modes. The default mode is Alarm Idle. In this mode the alarm is set but in the idle state. It is waiting for the current time to match the time set for the alarm in the UDD_SET time property.

In addition to calling the melody function, the local variable AlarmOn_Int is set to true. Once it is set to true the control goes from the ClockApplication mode to DisplayAlarm mode (refer Figure 76) which displays the alarm on screen as shown below.

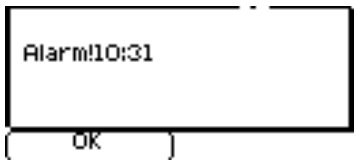


Figure 84: Clock Application - Ringing

6.3.6.4 Clock Update Mode

The mode is used to start a second timer. At every tick of this timer the CurrentTime variable is updated by 1.

6.3.6.5 Initialize Clock Settings Mode

Initialize Clock Settings has the modes to initialize the clock and then update the clock settings like setting time, date and playing melody by preparing AT command strings to be sent for updating the settings for the target.

Idle

This is the default mode. On the start of the application Started event is triggered which causes the application to go to init mode.

When the application has already started, the control goes to the Melody or Clock modes to either play the melody or Set the time or date for the clock.

Init

When the application is started the initial time is updated into the local time variables used in the application. i.e. CurrentTime and CurrentDate.

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.

GetTimeDate

This mode is used to send an AT command AT+CCLK? (command gives time and date of the target) and updates the local time variables with the current date and time. There after the date and time values are updated using a second timer in the CLOCK_UPDATE concurrent mode.)

Work

This mode is used to update the clock settings.

Melody

The mode is used for sending the Melody AT command AT+CRMP to the AT channel for playing the melody when the alarm goes off.

Clock

This mode is used for setting the date and time. It sends the AT+CCLK command for setting the date and time. The parameters for the command are taken from the user inputs for setting a particular date or time.

6.3.6.6 AT Mode

This concurrent mode is also known as the AT dispatcher. The AT dispatcher contains modes which would keep track of sending the AT commands to the AT Channel. SRV_AT is the user object already provided by Wavecom, which handles the concurrent sending of AT commands to the AT channel. When command is given to AT dispatcher, it goes into the *"WaitForDispatcher"* state where it waits for SRV_AT to be *ready*. When *ready*, it goes to *"TryToSend"* state from where it calls a SendAT method on SRV_AT. If it returns error code as *busy*, it goes back to *"WaitForDispatcher"* state and repeats the process till the error code is *not equal to busy*. Once command is sent, AT dispatcher goes to *"WaitForResponse"* state.

6.3.7 Debugging Using Simulation

Starting the debugger

As shown in the figure below user can start a debug session by pressing the CTRL+G key or from view menu.

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.

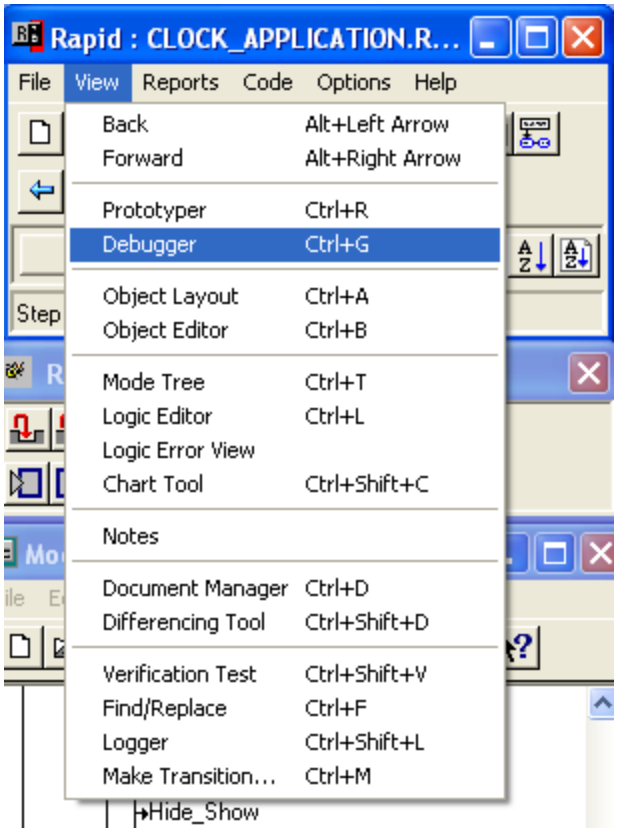


Figure 85: Clock Application - Opening Debugger

Starting the Application in Debug Mode

As shown in the Figure 86, the debugger contains the following buttons. User can start the simulation by clicking the **Play** button.

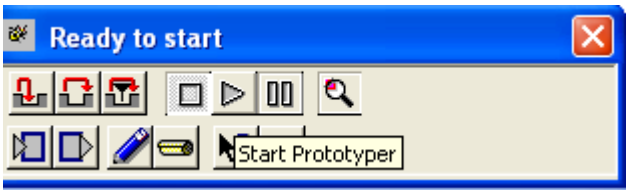


Figure 86: Clock Application - Debug Control dialog

Setting the breakpoints

As described in the Figure 87, user can set a breakpoint by right clicking on a particular line of code and selecting the breakpoint option.

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.

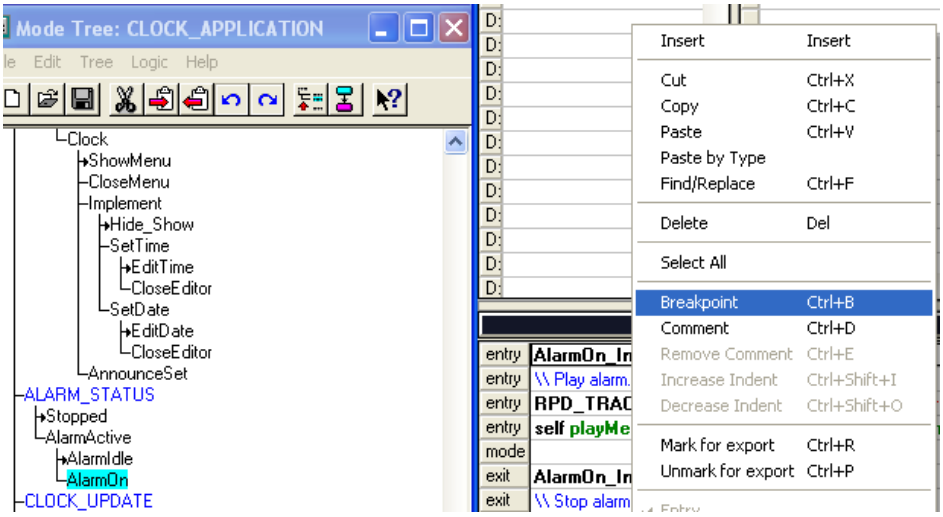


Figure 87: Clock Application - Setting a breakpoint

Hitting the Breakpoint

As shown in the Figure 88, the line of code where a breakpoint has been set is shown in yellow color once the breakpoint is hit at that point. User can inspect the values of all the variables at that time when the breakpoint has been hit.

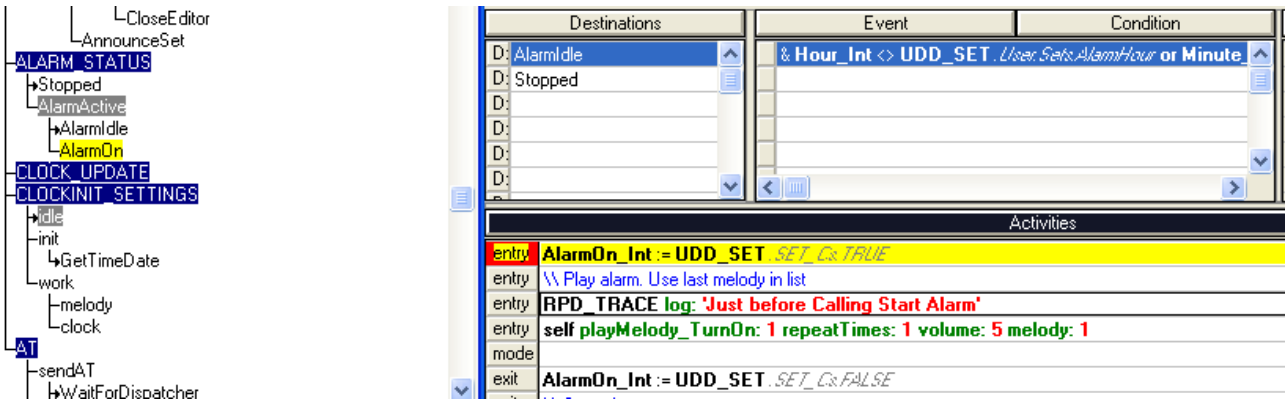


Figure 88: Clock Application - Breakpoint hit

6.4 Integration with Open AT® Software Libraries

6.4.1 Generate application code

Once the application has been tested in the simulation mode, it could be generated using the Rapid PLUS™ generate Code facility. This facility would

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.

generate the .c and .h files for the ClockApplication.RXD file and other UXO files used in the Clock Application. The generate code is done as shown in the figure below.

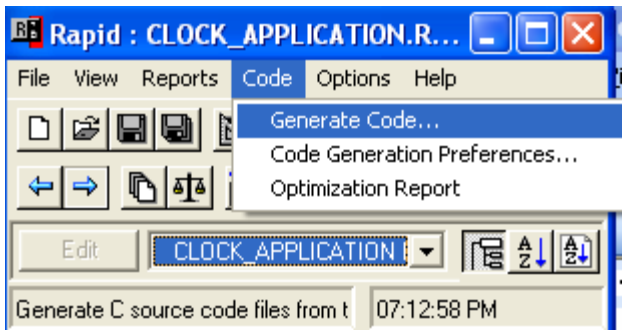


Figure 89: Clock Application - Generate Code Option

On clicking the Generate Code menu option shown above the user would be taken to a new dialog box control (refer Figure 89). On pressing the start button, the code generation would begin as shown in the Figure 90.

The generated code would be placed in the ...\AppCG folder as .c and .h files. The generated .c files should be copied to the ...\AppCG\src folder and the generated .h files should be manually copied to the...\AppCG\inc folder.

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.

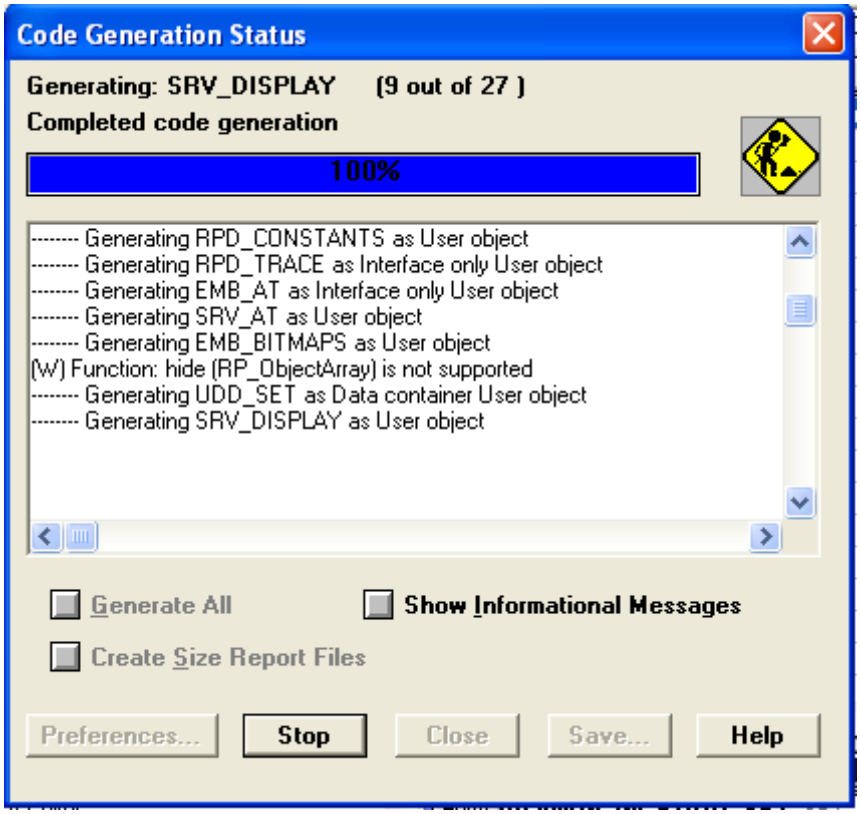


Figure 90: Clock Application - Code Generation Progress bar

6.4.2 Update the C files for embedded components

6.4.2.1 EMB_FLASH

Functions

`loadAlarmClockSettings: <sets>`

and

`saveAlarmClockSettings: <sets>`

These functions are generated as UDI (User Defined Interface) in the emb_flash.c file and are updated with the calls to read and write the Alarm status in the flash memory of the target platform.

In the initialization of the embedded flash component, the flash to be read and written is subscribed using the adl API **adl_flhSubscribe**. For reading the Alarm status when the target is switched on, flash has to be read using the API

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.

adl_flhRead. Similarly for writing the Alarm status (i.e. either ON or OFF) the adl API adl_flhWrite is used.

6.4.2.2 EmbdCode\DebugApi.c

This is an embedded file which comes with the Wavecom Reference MMI. In order to look for traces in the Target Mode using the Wavecom Tools such as TMT and TE, certain changes that need to be done in the function getFlags are as described below.

In this function the flags for transition summary and transition detail are commented by default.

|**fDBGTransitionsSummary** /* 0x0004 */

|**fDBGTransitionsDetail** /* 0x0008 */

These flags should be uncommented so that we are able to debug the ClockApplication by watching the traces for transitions summary and transitions detail.

6.5 Build and Deploy

The application code that has been generated in the earlier section 6.4.1 needs to be built using the ADS ARM Compiler version 1.2.

6.5.1 Code build using ARM/RTE

Open the Reference_Design.dsw provided in the Reference_Design folder of the Wavecom reference MMI. This workspace would contain the following projects.

- Reference_Design files
- Reference_Design_rte files
- LCD files
- LCD_rte files
- Display files
- Display_rte files
- ESIM files
- ESIM_rte files

All the c source and header files that have been generated as part of code generation should be included in the ESIM as well as ESIM_rte project in the reference workspace as shown in the Figure 91. The Reference_Design.mak file also needs to be updated with the updated .c files.

Once the workspace has been updated select Reference_Design as active project and use the Rebuild ALL option to build the project. Make sure to set the

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.

dependency of the Reference_Design project on LCD, Display and ESIM projects). This would build the target build for the clock application using the make file Reference_Design.mak kept in the same folder.

Note	Make sure to update the Reference_Design.mak file with the .c files associated with the alarm clock application. Remove the .c files not needed for the Clock Application.

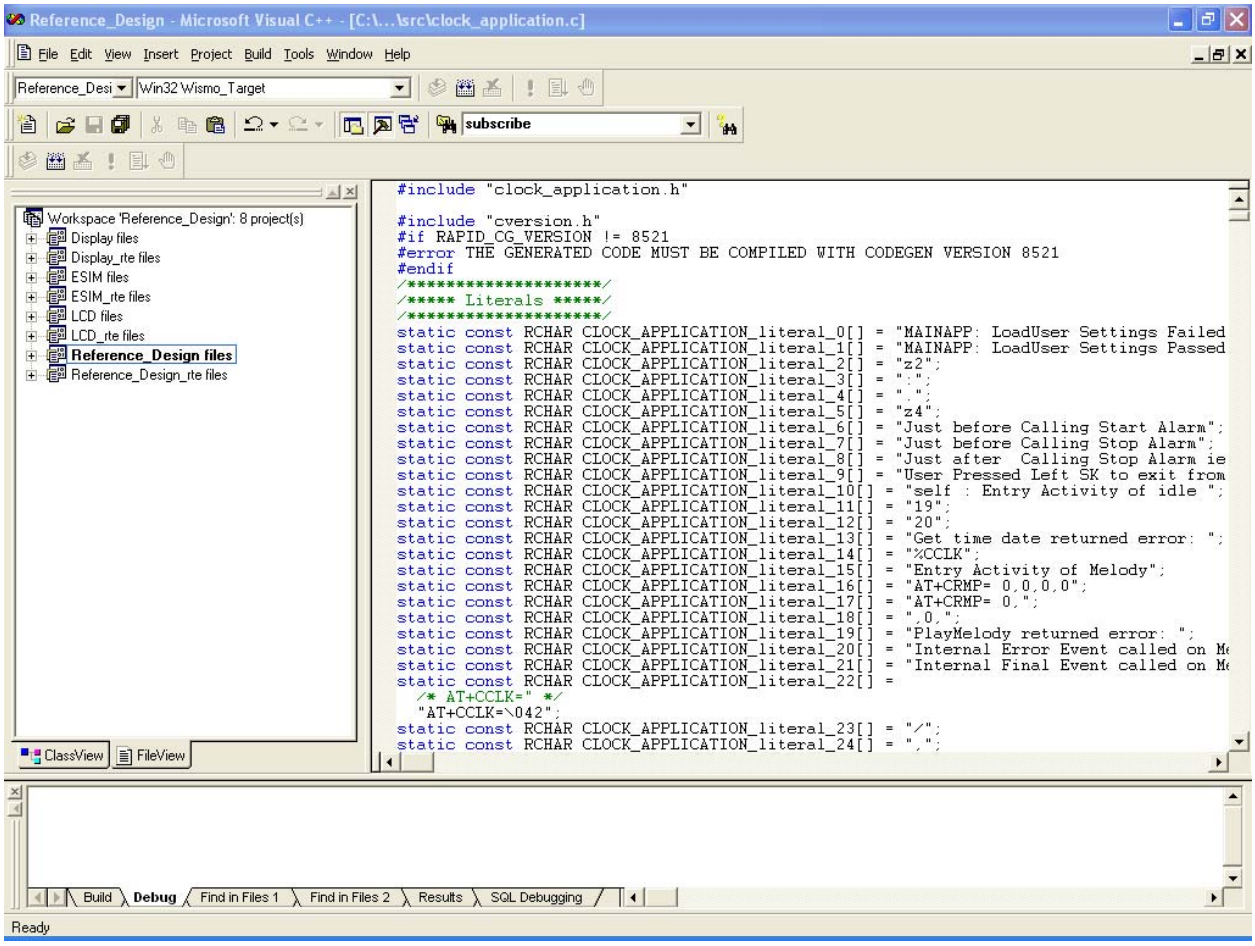


Figure 91: Clock Application Workspace

6.5.2 Download image to target platform

Once the Target build is completed, the folder ...\Reference_Design\arm\out, should be populated with the latest ClockApplication Binaries. Use the Xmodem (use 1K Xmodem for Q2686H Wireless CPU) protocol in the Hyperterminal to

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.

download the Application Image (arm_Reference_Design_H.wpb.dwl) to the Wavecom Target.

6.5.3 Wake up, the alarm is ringing

Once download of the application binary is complete, reset the target and start the application using AT+WOPEN=1. Set the alarm and get up early in the morning.

This document is the sole and exclusive property of WAVECOM. Not to be distributed or divulged without prior written agreement.

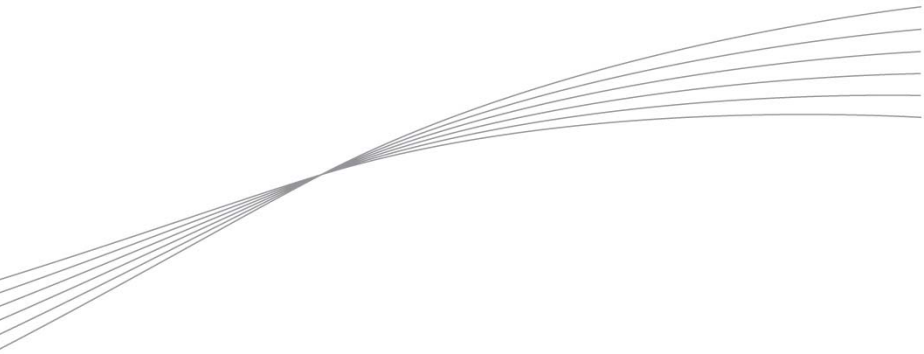
Ce document est la propriété exclusive de WAVECOM. Il ne peut être communiqué ou divulgué à des tiers sans son autorisation préalable.

7 References

- TM_Tools_Manual_for_Open_AT_V310_rev008.pdf
- TU_Tutorial_for_Open_AT_v310_rev007.pdf
- GS_Getting_Started_with_Open_AT_v310_rev010.pdf
- BUG_Basic_Development_Guide_for_Open_AT_v310_rev012.pdf
- AUG_ADL_User_Guide_for_Open_AT_V310_rev007.pdf
- AT_Commands_Interface_Guide_for_X55_release_rev002.pdf
- ATA_Appendixes_to_AT_Commands_Interface_Guide_for_X55_-_rev002.pdf
- Open AT® Course Material
- Open AT® Presentation
- Wavecom OpenAT GTi GDS.doc
- Software Design Document 1.1.doc
- RapidPLUS™ Basics - Course Book 5.0.0.pdf
- Generating Code for Embedded Systems 8.0.pdf
- Wavecom website: www.wavecom.com
- e-SIM website: www.e-SIM.com

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.



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. - Unit 201-207, 2nd Floor, Bio-Informatics Centre – No.2 Science Park West Avenue - Hong Kong Science Park, Shatin - New Territories, Hong Kong