

Wintegra MAC Development Kit

WINPATH™ ACCESS PACKET PROCESSORS

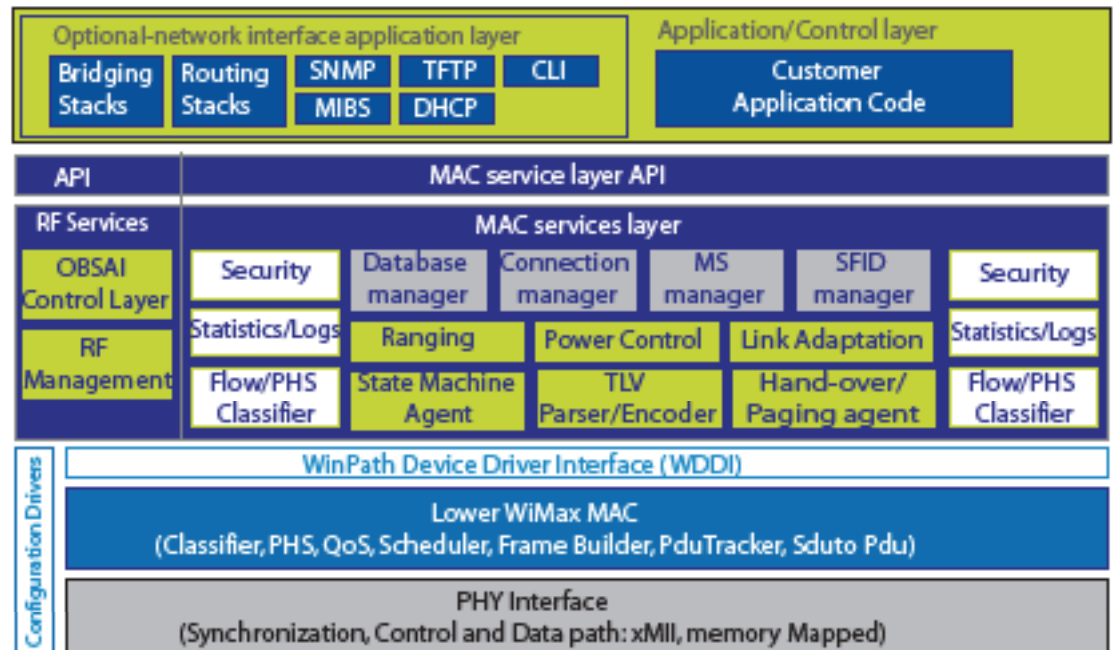
Product Highlights

- Access to DPS source code, including relevant WiMAX Access components and WiMAX transport package.
- Access to the upper MAC source code (ANSI C code)
- A comprehensive technical training which include the following:
 - WDDI MAC specific API ,
 - Services layer design and API.
 - Lower layer MAC and DPS data flow design
- DPS Tool Chain
 - Compiler, Linker, Simulator, Profiler, Analyzer, Debugger
- Object code right for upgrades, subject to the standard wireless-wimax releases
- Full Documentation and online technical support.
- Requires signing and execution of the WDDK source code license agreement.

WMDK- full access to MAC features

The Wintegra MAC Development Kit – includes a source license for the Wintegra WiMAX upper and lower MAC software, as well as the proprietary DPS development tools and training. This WMDK gives full access to all the features of the lower and upper MAC that may be modified by the customer as desired under terms of the WMDK license. The development kit includes: training in the tool chain and architecture of the DPS code, the Upper MAC software and technical support. (Note: The “MAC” designation refers to the fact that WMDK customers receive source code for the MAC software, not Wintegra’s full DPS source code package.)

Wintegra WiMAX services and software stack



DPS source code components:

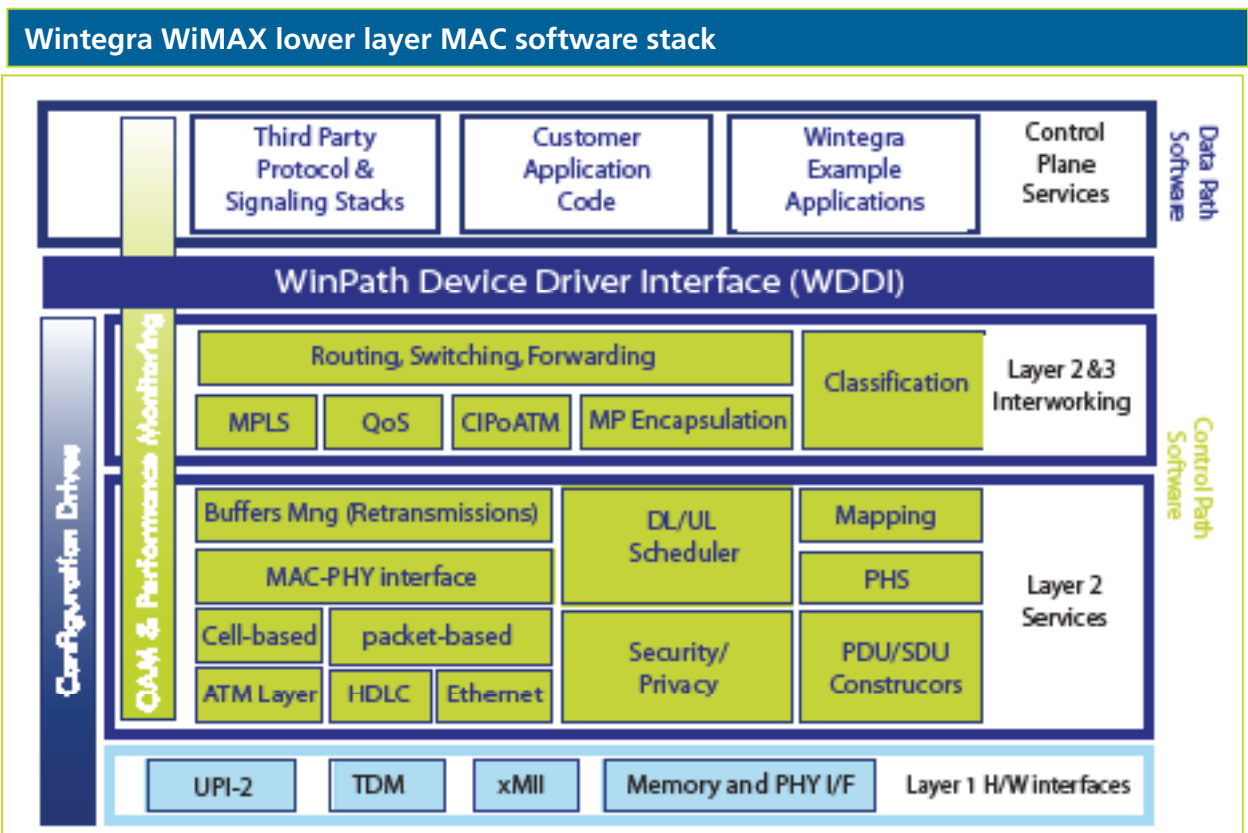
WMDK includes the following protocols for which DPS source code is provided for a WiMAX 802.16e lower MAC, in production quality form.

- Bridging and Routing network interface.
- Per service flow policer and shaper
- Advance QoS and buffer management.
- 802.16e Classifier
- Packet Header Suppression (PHS).
- Transport module handover buffering and multicast
- GRE or IPinIP DPS tunneling protocol (for R6 and R8)
- SDU to PDU builder.
- Uplink and Downlink ARQ
- UL/DL schedulers.
- OFDMA frame mapper and builder.
- Per subscribers measurement module
- Handover and scanning modules.
- MAC-PHY message logger
- Statefull MAC/PHY interface including flow control.
- MAC/PHY ethernet interface

The WiMAX MAC software implementation is divided into two main parts, Data Path processing and the Control Path processing.

The Data Path processing handles all the frame-based and real-time WiMAX-MAC events, such as packet classification, QoS, frame builder, fragmentation, reassembly and scheduling. These tasks are executed on the WinPath engines (WinGines) and are integrated within the standard Data Path Software (DPS). These software tasks are dynamically distributed between the WinPath’s pool of processing engines via WinPath’s SMTMP technology, enabling automatic load-balancing and optimal efficiency of processing hardware resources.

The Control Path processing handles the upper MAC services layer and the less time-sensitive events of the WiMAX-MAC. Those events include resource management, signaling, monitoring, connection set-up and configuration.

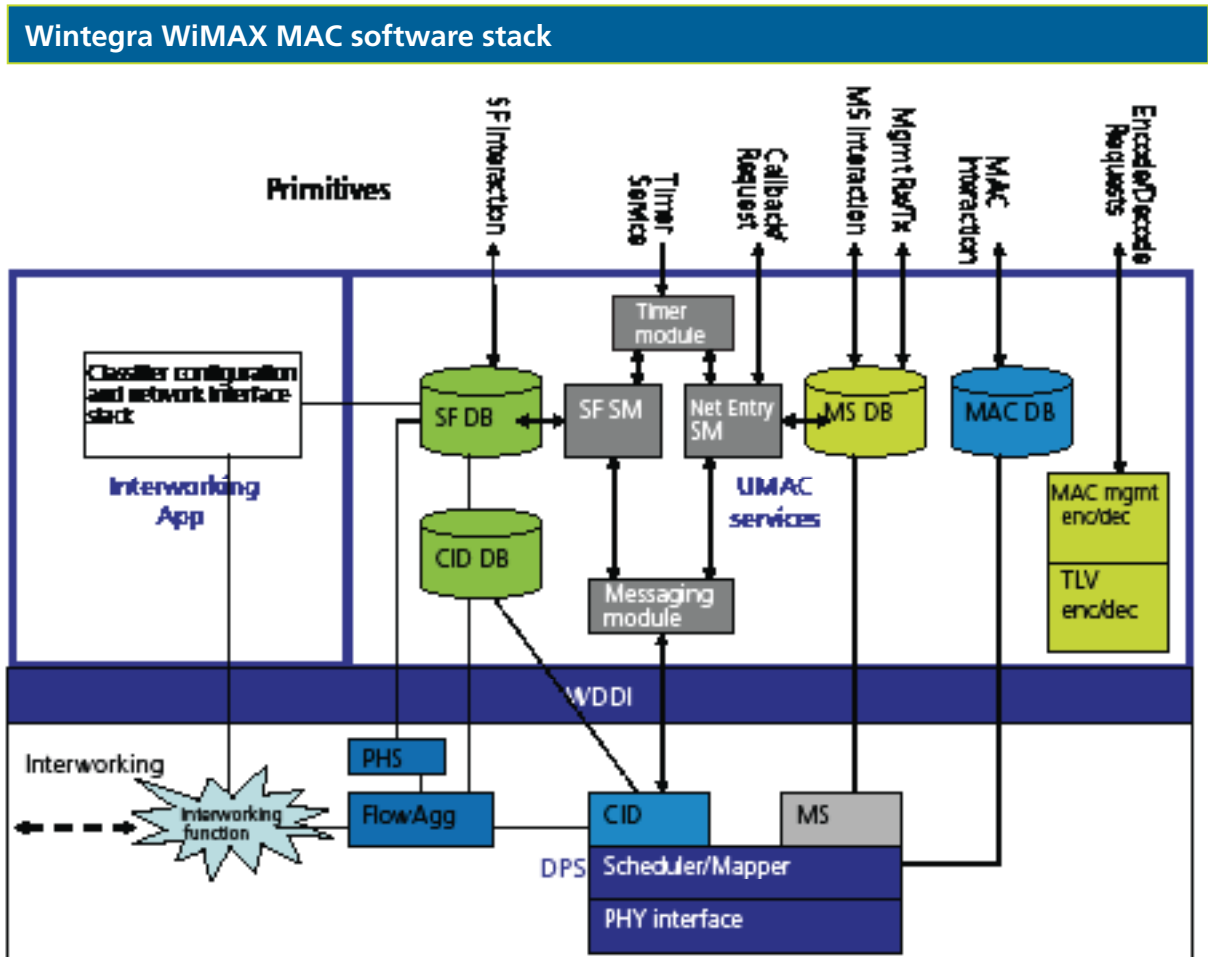


Services layer source code components:

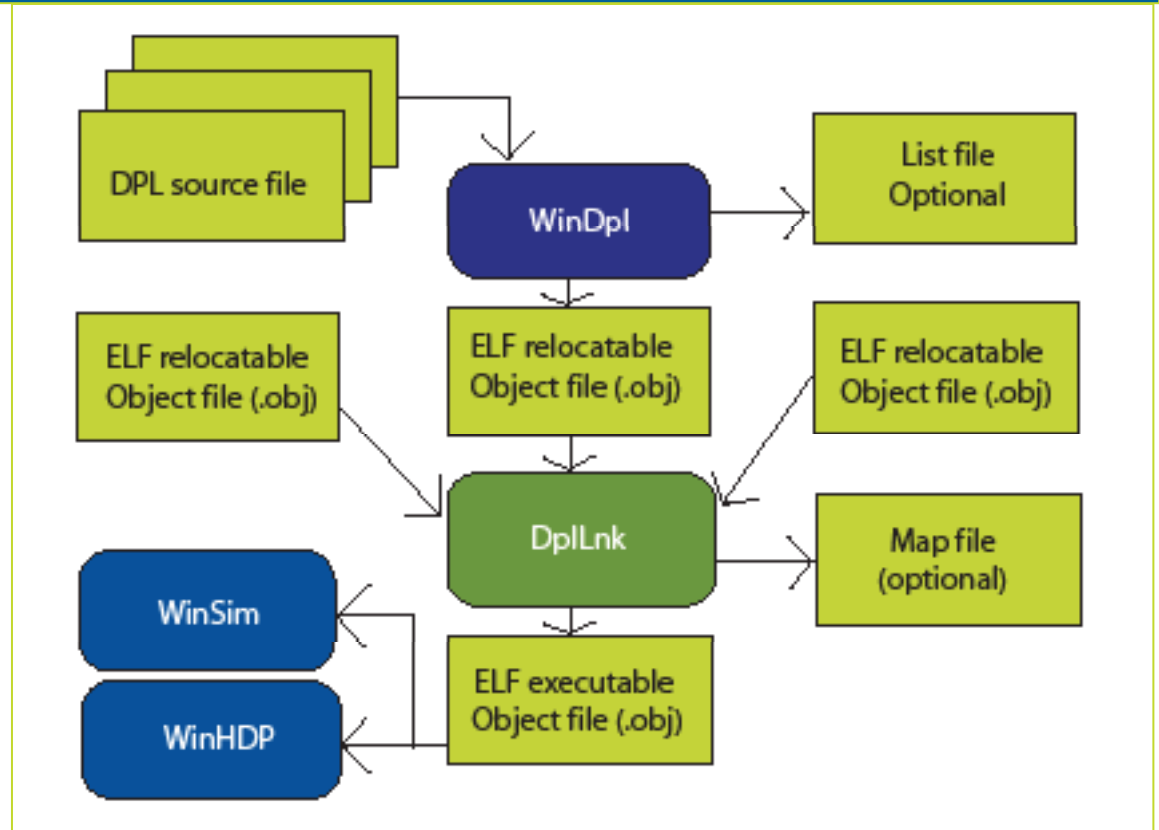
Wintegra WiMAX 802.16e upper MAC - provides a comprehensive set of upper MAC functionality including:

- Global MAC configuration
- Initial and periodic ranging
- Network entry registration
- Dynamic service updates
- Handover and scanning
- Sleep control
- Power control
- TLV parser and encoder

The WiMAX Upper MAC services layer operates in conjunction with the lower MAC services provided by the WiMAX DPS. Together, they provide a full implementation of the WiMAX Base Station MAC functionality defined in IEEE 802.16e. The role of this function in the overall WiMAX system is depicted in the following diagram.



Wintegra DPS development flow



Wintegra DataPath software development environment:

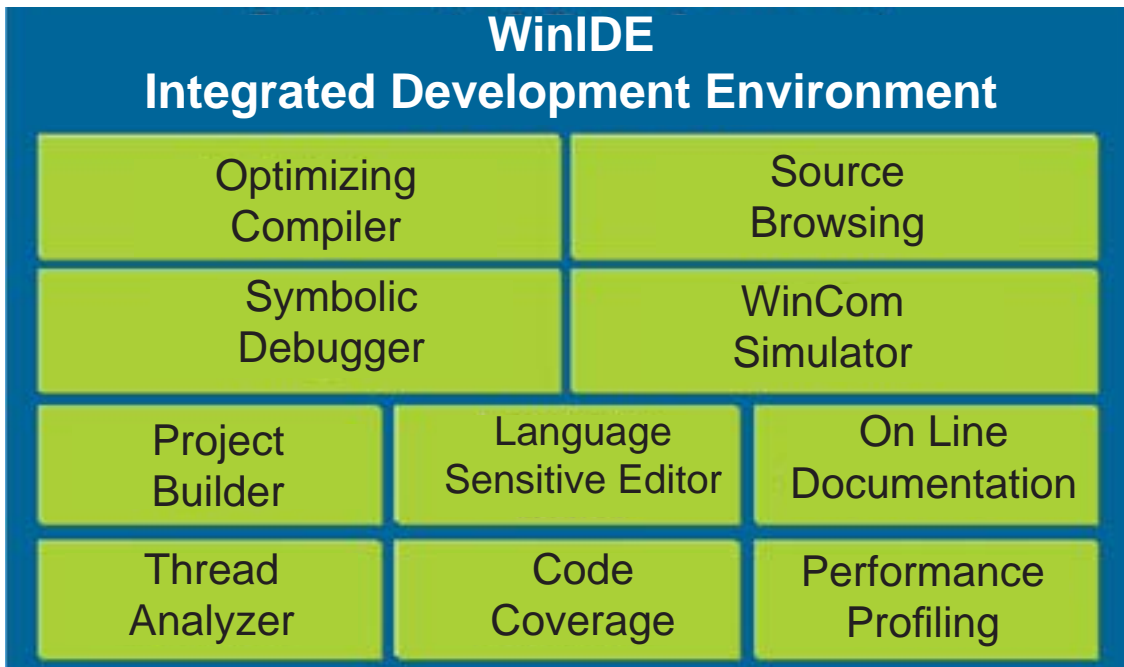
The WinPath Integrated Development Environment (WinIDE) is intended for programmers and developers, using the Wintegra WinPath to modify the lower layer WiMAX MAC and to add their own proprietary features and added value to the 16e MAC provided by Wintegra.

Main components depicted in the above figure:

- WinSim (Wintegra simulator) is a clock accurate simulator which can simulate the exact Winpath Hardware behavior. It allows the programmers to trace the different threads execution in the system, to track the actual utilization of each one of the winpath hardware components and to investigate the code/flow converge of a specific examin module.
- WinHDP: The WINHDP hardware reference design is a complete software-defined platform suitable for wireless communications applications, such as WCDMA and 802.16e compliant Micro/Macro channel card and complete Pico Base Station (BTS) products.
- DPL Source Files: Includes the DPL-language source code.
- WinDPL Compiler: Translates DPL-language source files into machine language ELF Object Files. It also includes a preprocessor that is used automatically by the Compiler to transform the program before actual compilation.
- List File: Lists the input source file along with the actual machine instructions and encodings.
- Relocatable ELF Object File: Contains the machine code representation of the source input, and contains multiple sections representing code, data and control information needed for the linking process.
- Command File: Contains linking commands that modify the operation of the WinDPL Linker.
- WinDPL Linker: Processes ELF Object Files and a WinDPL Linker command file (optional) to create a single ELF executable file and a map file (optional).
- Map File: Outlines the memory map, section allocation table and global symbol table.
- Executable ELF Object File: Executes instructions in the WinPath WinGine processor.

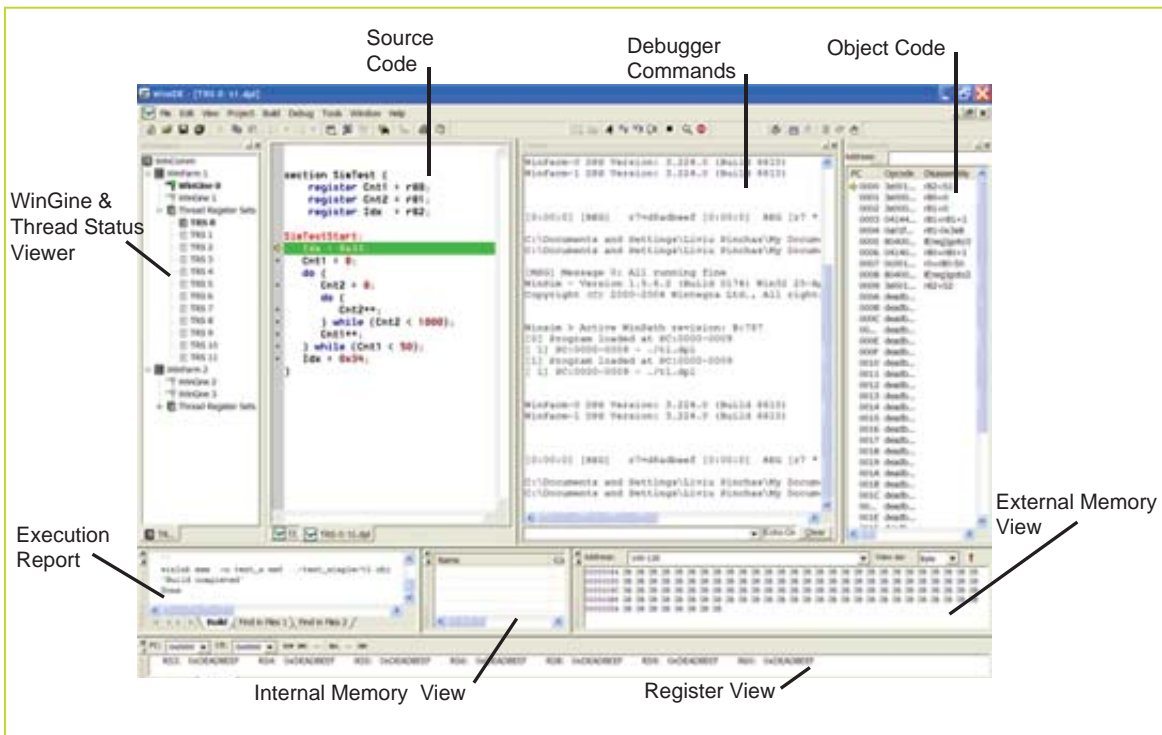
WinIDE tools package

The WinIDE provides a friendly, integrated environment for the development, simulation, code profiling, debugging and analysis of lower MAC applications.

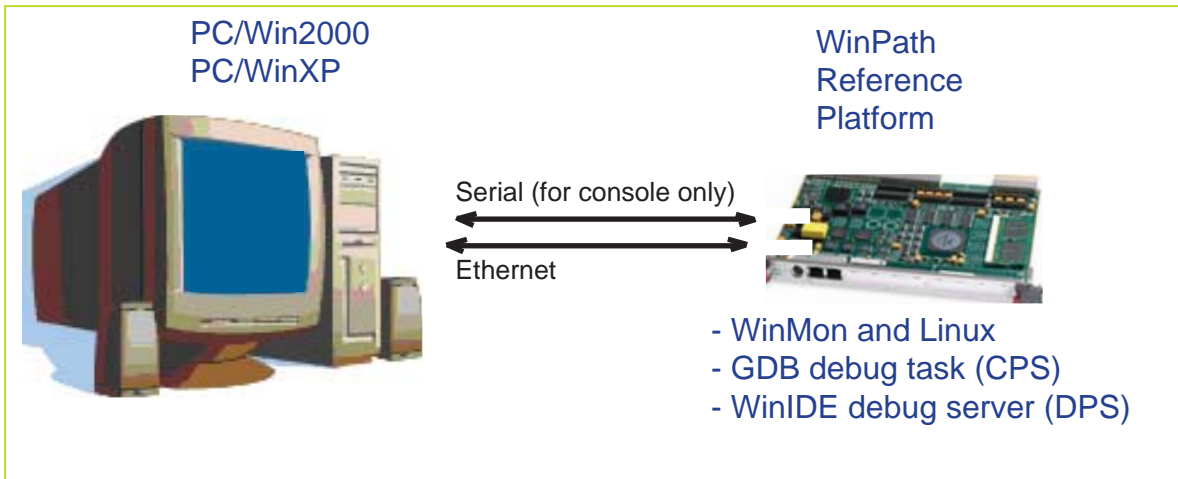


Example for WinIDE simulator and debugger user interface

The programmer can very easily get detailed diagnostics of the specific application and investigate the exact threads and flow behaviors. The actual debugging can be accomplished via offline debugging on the memory dump and Winpath tracing or with standard breakpoints and real-time single step code tracing. The WinIDE allows the user to examine functionality and values of the DPL application as it runs and, at the same time, verify hardware registers values and test the content external memory.



WinHDP debugging scheme



WinIDE enables both debugging platforms, either through the WinSim (simulator) or directly on the WinHDP hardware.

- Upper MAC development and debugging is performed via standard off-the-shelf tool chain (e.g. GNU).
- DPS development and debugging is performed via WinIDE development environment.
- Both upper layer MAC and lower layer MAC debugging are supported simultaneously.

All Rights Reserved

Printed in the United States of America

All information contained in this document is subject to change without notice.

The products in these documents are not intended for use in medical, life saving, or life support applications where malfunction may result in injury or death to persons. Wintegra may make changes to specifications or product descriptions at any time, without notice.

The information supplied by this document is provided on an "AS IS" basis. In no event will Wintegra be liable for damages arising directly or indirectly from any use of the information contained in this document. Wintegra® is registered in the United States Patent and Trademark Office

For more information, see www.wintegra.com.

WMDLPB - 0307 - CW