Rev. 1
June 2010
DebugMenuAppNote.odt

Application Note

Abstract

This application note and accompanying software detail the basic debugging tools that users must include in their own development and fielded systems. Failure to include this basic functionality will hinder debugging efforts.

This note details the steps required to configure and run the debug menu on the WinPath Packet Processor.
The example software runs under Linux or Winmon. The underlying functionality is simple to port to other operating systems.

Table of Contents

1.0 Introduction

This application note and accompanying software detail the basic debugging tools that users must include in their own development and fielded systems. Failure to include this basic functionality will hinder debugging efforts.

This note details the steps required to configure and run the debug menu on the WinPath Packet Processor. The example software runs under Linux or Winmon. The underlying functionality is simple to port to other operating systems.

2.0 Terminology and conventions used in this document

The following conventions are used throughout the Application Note

Code examples are provided using courier font.

Example: `error_code = WP_IntEnable();`

When used in document text other than code examples, reserved user-visible function names, data structures, types and constants are indicated in bold, universal font format.

Example: **WP_DeviceDelete( )**

3.0 Module Description

The debug menu contains the main debug menu, and several sub-menus which are sorted based on subject. This section will describe the menu interface, the properties of each menu, and instructions on how to use the menu system.

3.1 Menu Interface

The menu interface consists of a header and a list of enumerated options. The header section displays the name of the program, the menu navigation hot keys, and the name of the current menu. An example menu is shown below

```
**************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Gigeloopback Functions
**************************************************************************
1. Send a test packet to enet 1
2. Send a test packet to enet 2
3. Poll for a packet on enet 1
4. Poll for a packet on enet 2
5. Print Statistics for both ports
6. Utilities and Debug
7. Quit the program
# Enter your command:
```

Options in the menu interface are enumerated, the user can make their selection by entering the menu option number followed by <enter>. To provide arguments to functions which require parameters, input the selected option number followed by the required parameters. If the required parameters are unknown, provide an empty argument to print the parameter format to the screen. To navigate quickly between menus, the user can use the menu hot keys which are shown in the menu header.

3.2 Main Debug menu

The Main Debug menu contains the most commonly called debug functions and links to the sub menus. Users using the menu interface can jump to this menu from any location with the 'd' hotkey.

```
**************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Main Debug menu
**************************************************************************
1. Fatal error menu
--------------------------------------------------------------------------
2. map   (WinUtil: Memory Map)
3. md    (WinUtil: Memory Display)
4. mm    (WinUtil: Memory Modify)
--------------------------------------------------------------------------
5. Get table from TA base  <baseIndex> <entrySize> <entryIndex>
6. tab   (WinUtil: Print TA Bases)
--------------------------------------------------------------------------
7. WP_Display Cmd
8. WinUtil(free text)
--------------------------------------------------------------------------
9. Performance menu
10. Memory & Locks menu
11. Display,Log & Versions menu
12. Advanced WinUtil Cmd menu
--------------------------------------------------------------------------
13. Set Winpath ID 0-4
--------------------------------------------------------------------------
?. Help
```

The Memory Map(MAP) command searches for the memory alias specified by the user and returns the physical address of the matching memory definition. The Memory Display(MD) command displays the contents of a specified memory location. To alter the contents of a memory location directly, the user should use the Memory Modify(MM) command.

The “Get table from TA base” option displays the contents of a specified TA base table entry. A full list of TA base register addresses can be displayed with the “tab” option.

The “WinUtil(free text)” option allows the user to call any WinUtil command directly. This function is described in more detail in section 3.7 of this application note.

The “Set Winpath ID 0-4” option allows the user to alter the WinPath ID settings of the menu to suit their board settings. This WinPath ID is passed in as a parameter to functions such as WP_Display. The “WP_Display Cmd” option calls an API function that can display information about the system. This function is described in more detail in section 3.6 of this application note.

3.3 Fatal Error menu

The Fatal Error menu contains a comprehensive set of bus error debugging tools. A example Fatal Error menu is show below:

```
**************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Fatal Error menu: Bus Error Functions
**************************************************************************
1. Display all info
```

```
-------------------------------------------------------------------------
2. Print registers DTCR1 - DTCR4
3. Print Bus ECC registers
-------------------------------------------------------------------------
4. Print TA PC current value
5. Print AA PCT configuration
-------------------------------------------------------------------------
6. wfc reg <all>   (WinUtil: WMM Check Corrupted FIFO by HW)
7. fmum            (WinUtil Cmd: Check FMU put error)
-------------------------------------------------------------------------
8. hbm             (WinUtil: Bus Error Check)
9. tb dump         (WinUtil: Wingine Trace Dump)
10. trsa           (WinUtil: TRS Serial Allocation)
11. tast           (WinUtil: Print Current TRS status)
12. trs            (WinUtil: TRS content Dump)
-------------------------------------------------------------------------
13. taot           (WinUtil: Print TA Order Registers)
14. taom           (WinUtil: Order Monitor Display)
15. Versions and Build times
-------------------------------------------------------------------------
?. Help
```

The “Display all info” option will execute all the other functions listed and dump the result to the screen. If operating under linux, the user can dump the error log generated to a local file. Bus error issues that cannot be resolved by the user should generate this error log before contacting Wintegra technical support.

Printing Debug Trace Control Registers (DTCRs) provides information on the settings of the traces and WinGines, such as which trace serials should be active and the conditions to stop the WinGines.

The Error Correction Code (ECC) registers provide information on the status and settings of the ECC handlers. For more information about ECC handling, please refer to the “ECC Handling” section of the Hardware Developer's Guide.

The Thread Arbiter Program Counters(TA PCs) maintain the return program counters of the last task switch to enter the TA. In the event of a bus error, the TA PCs will hold the program counter of the task that caused the bus error. If printing the TA PCs shows that the TA PCs have halted, a bus error may have occurred. For more information about the thread arbiter, please refer to the “Thread Arbiter (TA)” section of the Hardware Developers Guide.

Users operating the Winpath3 can display the Agent Arbiter Program Counter Table (AA PCT), which maintains a set program counter entries that define the first WinGine instruction to be executed for a newly opened thread. Users operating with the Winpath2 will display the Serial Arbiter Program Counter Table(SA PCT).

The WMM corrupted FIFO check(wfc reg) runs a check for FIFO corruption by traversing the FIFO nodes and checking if the links between nodes are valid. Invalid FIFO links and addresses will be displayed. The “Check FMU put error” option checks for errors that have occurred when the DPS is storing cells and packet descriptors in the FMU FIFOs. For additional information on the FIFO Manager Unit, please refer to the “FIFO Manager Unit” section of the Hardware Developer's Guide.

The Hardware Bus Monitor (HBM) shows memory statuses and reports if a bus error has occurred. In the event of a bus error, the address, size, TRS source and error type will be displayed.

The Trace Buffer allows the user to keep a trace on the flow of the WinGines(new tasks, end tasks, change of flow). The trace buffer dump log can be disassembled to show the sequence of commands that led to the error.

The TRS Serial Allocation(TRSA) displays the allocation of TRS to each active Winfarm and serial. To display the status of a specified TRS, use the Thread Arbiter Status(TAST) menu option. The contents the TRS can be saved to a specified memory location and displayed with the TRS command.

The Thread Arbiter Order Monitor(TAOM) command checks the order monitor for order and coherency problems. Information on the Thread Arbiter Order Table(TAOT) is considered Wintegra Proprietary information and specific details will not be released.

Additional information on the Winutil functions (options denoted by "Winutil: ") can be found in the Debugging WinPath Applications document.

Note: The WMM corrupted FIFO check, and TRS dump should only be done when the TA PC is halted.

3.4 Performance menu

The Performance menu contains functions for displaying performance statistics such as Bus, Wingine and Thread utilizations. The user can also display and change the TRS allocation values.

```
*************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Performance menu
*************************************************************************
1. trsu (WinUtil Cmd: Display Thread Utilization)
2. sysu (WinUtil Cmd: Display Bus Utilization)
3. wgu  (WinUtil Cmd: Display Wingine Utilization)
-------------------------------------------------------------------------
4. WP_SysCmd: Get TRS Allocation
5. WP_SysCmd: Set TRS Allocation
?. Help
```

The TRS utilization(TRSU) displays allocation of TRSs to threads. If there are not enough TRSs to handle new threads, the performance of the WinPath might drop. For more information about TRS Utilization, please reference the "TRS Utilization" section in the WDDI Programmer's Guide.

The WinGine utilization(WGU) shows how much of each WinGine is being used. When usage exceeds 95% on a Wingine, packets can be dropped. For more information about WinGine Utilization, please reference the "WinGine Utilization" section in the WDDI Programmer's Guide.

Users operating the WinPath 3 have an additional System utilization(SYSU) option which displays the Buses, WinGine, WinPath Security Engine(WSE) engines and Parser Classifier Engine (PCE) utilization. Information about these engines can be found in the Hardware Developer's Guide.

The "Get TRS Allocation" option displays the maximum TRS allocation and TRS mask settings for the specified trace. To change these settings, use the "Set TRS Allocation" option. A full list of all available serials and their assigned indices will be displayed if no

arguments are passed in. For more information on these commands, please refer to the "WP_SaEntryGet" and "WP_SaEntrySet" sections of the WDDI API Reference Manual.

3.5 Memory and Locks menu

The Memory and Locks menu contains functions that display memory allocation information. This menu can display information such as the memory used on each memory bank and offers the option to enable or disable memory allocation logging.

```
*************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Memory & Locks menu
*************************************************************************
1. Get table from TA base  <baseIndex> <entrySize> <entryIndex>
2. tab  (WinUtil: Print TA bases)
-------------------------------------------------------------------------
3. Debug base display menu
4. Qnode display menu
5. bmap (WinUtil: Print Bus Mapping Table)
-------------------------------------------------------------------------
6. Print Memory Available
7. Enable WDDI Memory Allocation Log
8. Disable WDDI Memory Allocation Log
-------------------------------------------------------------------------
?. Help


*************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name:  Debug Base Dump
*************************************************************************
1.  Debug Area: only header
2.  Debug Area: full dump
?. Help


*************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Qnode display menu
*************************************************************************
1. WP_QnodeStatus
2. List All Qnode handles
?. Help
```

The "Get table from TA base" option displays the contents of a specified TA base table entry. A full list of TA base register addresses can be displayed with the "tab" option.

The "Debug base display menu" can either display a small section, or the full contents of the debug base(254) . This debug area can be allocated to store temporary data such as TRS dumps.

The "Qnode display menu" displays the number of memory buffers available based on the Queue node(Qnode) handlers provided. A list of all Qnode handlers available can be displayed with the "List all Qnodes" option. For more information about Qnodes, please refer to the "Queue Nodes" section of the WDDI Programmer's Guide.

The “Bus Mappings” option displays the starting and ending addresses of each bus. This information can also be displayed by providing WP_Display with the SYSTEM parameter.

The “Print Memory Available” functions displays the memory available for a specified memory bank. If no memory bank is specified, the function will display the memory available on all buses and banks.

The user can also choose to enable or disable the memory allocation logs from the Memory and Locks menu. However, to unlock this feature, memory allocation logs must be enabled in the wp_debug.h file in the user's api directory.

3.6 Display, Logs, and Versions menu

The Display, Logs, and Versions menu contains options to enable/disable API and WPI logs, access to the WP_Display function, and tool version information.

```
*************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: Display, log & versions menu
*************************************************************************
1. WP_Display Cmd
2. WP_Display System Info
-------------------------------------------------------------------------
3. Set log level
4. Enable API level log
5. Disable API level log
6. Enable WPI level log
7. Disable WPI level log
-------------------------------------------------------------------------
8. WDDI version, build time
9. WinUtil Version, build time
-------------------------------------------------------------------------
?. Help
```

The WP_Display command displays information based on the winpath ID(WPID) and command parameters passed in. The WPID is set to 0 by default and can be changed in the Main Debug Menu. A full list of commands that can be passed to WP_Display will be displayed if no argument is passed in. For more information about WP_Display, please refer to the WDDI API Reference Manual.

For additional information on function call return values, the User can activate the API and WDDI logs. To set the maximum depth of callbacks, the user can use the “Set Log Level” function. However, to unlock this feature, WDDI and API logs must be enabled in the wp_debug.h file in the user's api directory. For more information on API and WDDI logs, please refer to the “Callback Functions” section of the WDDI Programmer's Guide.

The versions and buildtimes of both the WDDI and WinUtils can also be displayed within the Display, Logs, and Versions menu.

3.7 Advanced Winutils menu

The Advanced Winutils menu contains a full set of Winutils functions available.

```
*************************************************************************
  GigE Loopback Test
  Hot keys: !-Main Menu ^-Upper Menu =-Display Cur Menu  d-Debug Menu!
  Name: All WinUtil Commands
*************************************************************************
```

```
1. WinUtil(free text input)
2. trsu     (WinUtil: Display TRS Utilization)
3. wgu      (WinUtil: Display Wingine Utilization)
4. dmap     (WinUtil: Print Device Mapping Table)
5. bmap     (WinUtil: Print Bus Mapping Table)
6. smap     (WinUtil: Print Serial Mapping Table)
7. hbm      (WinUtil: Bus Error Check)
-------------------------------------------------------------------------
8. map      (WinUtil: Print Register Mappings)
9. mc       (WinUtil: Memory Check)
10. md      (WinUtil: Memory Display)
11. mf      (WinUtil: Memory Fill)
12. mm      (WinUtil: Memory Modify)
-------------------------------------------------------------------------
13. tab     (WinUtil: Print TA base registers)
14. taom    (WinUtil: Print TA order monitors)
15. taot    (WinUtil: Print TA order registers)
16. trs     (WinUtil: Print TRS registers)
17. tast    (WinUtil: Print Current TRS status)
18. trsa    (WinUtil: Print TRS Serial Allocation)
-------------------------------------------------------------------------
19. fmum    (WinUtil: Check FMU address errors)
20. freq    (WinUtil: Change Bus frequency)
21. core    (WinUtil: Dump memory partition to file)
22. ver     (WinUtil: Print Version and Buildtimes)
-------------------------------------------------------------------------
23. wfc menu  (WinUtil: WMM FIFO specific menu)
24. tb menu   (WinUtil: Trace specific menu)
?. Help
```

A full set of WinUtil functions is listed in the Advanced Winutils menu. The user can call any of these commands by selecting the associated menu option number. An alternative way to call these WinUtil commands is the “WinUtil(free text input) option. Enter the associated menu option number followed by the WinUtil command name and parameters. An example input for the example menu is shown below:

```
“1 md 0x1d021000 :4”
```

The majority of the WinUtil functions displayed are available from other submenus. For more information on each of these commands, please refer to the Debugging WinPath Applications document.

4.0 Accompanying Material

Five example test programs containing the debug menu is attached to this document. This program can be configured to run under either linux or Winmon. This section will detail the steps required to configure, compile, and run this application. A full list of attached files is shown below.

Filename	Contents
gige_loopback.c	gige_loopback test source code
tdi_aal0_comet.c	tdi_aal0 test source code
hspos_loopback.c	hspos_loopback (SPI3) test source code
upi3_loopback.c	upi_loopback test source code
ima_demo.c	ima_demo source code

wt_debug_menu.c	Debug menu source code
wt_debug_menu.h	Debug menu header file
wt_new_automation.c	Menu interface source code
wt_new_automation.h	Menu interface header file
wt_statistics_functions.c	Statistics functions for gige_loopback test
wt_statistics_functions.h	Statistics functions for gige_loopback test
stats.h	Statistics functions for tdi_aal0 test
stats.c	Statistics functions for tdi_aal0 test
wt_util.h	Utility functions
wt_util.c	Utility functions
wt_config.h	Configurations
makefile.winmon	Makefile for Winmon
makefile.linux	Makefile for Linux
make.list	List of files to compile

4.1 Supported Hardware and Software

This version of the applications accompanying this document are only compatible with the WinPath 2, WinPath3 and WinPath 3 SL boards.

The test programs were tested using the CodeSourcery toolchain under WDDS 4.1. The user may have to install CodeSourcery and WDDS 4.1, and set the required environment variable as shown by the following example code:

```
export WP_PATH_LINUX_MIPS_WINMON_WINPATH3=
~/CodeSourcery/mips-linux/bin:~/CodeSourcery/mips-
elf/bin:~/CodeSourcery/mips-linux/bin
```

4.2 Compiling and running a test application

To run the test application, the user must edit the attached makefiles to suit their system settings. Instructions which must be altered are marked by "`user modify`". The WDDI_DPS_IMAGE in the attached make.list must also be edited to fit the specifications of the user's board. To unlock the WDDI and API logging functionalities, the user must edit the wp_debug.h file in their api directory and change the associated definitions.

Once all changes are applied, the user can compile the program from command line. An example command line is shown below.

example command line for winmon:

```
make -f makefile.winmon example=gige_loopback
WPI_TARGET=mips_winmon   WPI_HW_DEVICE=winpath2 WPI_BOARD=wds
```

example command line for linux:

```
make -f makefile.linux example=gige_loopback
WPI_TARGET=mips_linux WPI_HW_DEVICE=winpath3 WPI_BOARD=wds
```

Once compiled, the user can load the generated file onto their board and run the test program.

5.0 References

-Debugging WinPath Apps with WinUtil
-WDDI 4.1 API Reference Manual
-WDDI 4.1 Programmer's Guide
-WP3 HDG
-WP2 HDG v101

6.0 Glossary

Document History

Ver.	Released	Description
1.0	4th June 2010	First Draft
1.1	19th August 2010	Debug Menu Board Bringup Package Application Note
1.2	24th November 2010	Test applications update

Contact Information

Wintegra Inc.
6850 Austin Center Blvd.
Suite 215
Austin, TX 78731
USA
Tel: +1 (512) 345-3808

Wintegra LTD.
Taya Center
6 Hamasger St.
Ra'anana 43653
Israel
Tel: +972 (9) 743-9998

Wintegra
International House, Stanley Blvd.
Hamilton Int. Technology Park
Blantyre, Glasgow. G72 OBN.
United Kingdom
Tel: +44 (0) 1698-404889

Proprietary Notice