



# ATA Enabler for DOS

Copyright © APSOFT, 2004

## Copyright Notice and Disclaimer

Unless otherwise noted, this document and the information herein disclosed are proprietary to APSOft. Any person or entity to whom this document is furnished or who otherwise has possession thereof, by acceptance agrees that it will not be copied or reproduced in whole or in part, nor used in any manner except to meet the purposes for which it was delivered.

The information in this document is subject to change without notice, and should not be considered as a commitment by APSOft. Although APSOft will make every effort to inform users of substantive errors, APSOft disclaims all liability for any loss or damage resulting from the use of this document or any hardware or software described herein, including without limitation contingent, special, or incidental liability.

Copyright © 2004 APSOft. All rights reserved.

Issue date 18-February-2004.  
ATA Enabler for DOS

CardWare and the APSOft logo are registered trademarks of APSOft.

All other products and brand names are trademarks and registered trademarks of their respective companies.

APSOft.  
Sonnenstrasse 26b  
85622 Feldkirchen  
Germany

Tel: +49 (0) 89 900 479 0  
Fax: +49 (0) 89 900 479 11  
Internet: <http://www.tssc.de>

## Table of Contents

<b>Introduction .....</b>	<b>V</b>
Features and Limitations .....	v
Enabler Versions .....	vi
ATAENAB .....	vi
CBATA .....	vi
UNATA .....	vi
<b>Installation .....</b>	<b>7</b>
Memory Requirements.....	7
Example of config.sys.....	7
Special Note for Toshiba Notebooks.....	8
Fine-tuning .....	9
<b>Resource management. ....</b>	<b>11</b>
<b>Command Line Options. ....</b>	<b>13</b>
Common Command Line Options.....	13
Resource Management .....	13
Diagnostics and Fine-tuning.....	13
Unloading .....	14
ATAENAB and UNATA Specific Options.....	15
CBATA and UNATA Specific Options.....	15
<b>Initialization File .....</b>	<b>17</b>
Common Section .....	17
Socket Sections.....	18
PCI Sections.....	19
MemFilter, PreFilter, MemFilter0, MemFilter1 .....	19
IoFilter, IoFilter0, IoFilter1 .....	20
IRQ.....	20
ISAEnable .....	20
VGAEnable.....	20
ExCABase .....	20
LegacyBase.....	21
PrimaryBus, SecondaryBus, SubordinateBus .....	21
SkipSocket .....	21
IgnoreBridge.....	21
CISBase ATAloBase ReadVerifyRetry LBAMode DriveLetter.....	22
IdeloBase8 IdeloBase2 SkipAtaConfig .....	22
Resources Section .....	22
<b>Additional information .....</b>	<b>25</b>
How to get Technical Support for ATA Enabler .....	25
Example of INI file.....	26

**This page is intentionally blank.**

## Introduction

This manual provides an overview of the ATA Card Enabler family for DOS. All released versions are collectively named “ATA Enabler” or simple “Enabler” throughout this manual.

Enabler is a small, but powerful driver supporting ATA cards and IDE Hard Disks connected via the PCMCIA-2-IDE PC Card adapter.

Due to the small memory footprint of the resident part the enabler is an ideal solution for use in various disk-cloning schemes. Both, Symantec Norton Ghost and Power Quest Disk Copy, are supported.

---

## Features and Limitations

Enabler supports both, original DOS (e.g. MS-DOS 6.22) as well as DOS included with Windows 95/98. Resident part of ATA Enabler occupies approximately 29 kB of memory (depending of number of ATA cards inserted).

Enabler can be loaded via CONFIG.SYS, AUTOEXEC.BAT, or simply started from DOS command prompt.

Enabler can be unloaded, except if it is loaded from CONFIG.SYS. Besides Enabler prevents second load of itself.

Enabler may provide full LBA support for disks of any size. Large disks are supported.

Enabler may provide full INT 13h support, including extension. Most utilities (FORMAT.COM, FDISK.EXE, Norton Utilities, etc.) are supported.

User may freely assign drive letters to partitions, except if Enabler is loaded from CONFIG.SYS.

In comparison with any competitive product Enabler doesn't need an IRQ. Therefore, user does not need to care about resource conflicts.

Enabler may support simultaneously up to 10 ATA cards (or Hard Disks). Number of supported partitions is limited only by number of available drive letters. Number of physical PC Card adapters is not limited.

Enabler does not provide hot-swap support. The ATA Card (or Hard Disk) should be connected to the computer before ATA Enabler is started.

Enabler will not support ATA Cards without CIS, and ATA Cards with CIS where CISTPL\_CONFIG, CISTPL\_FUNCID, CISTPL\_VERS\_1, or CISTPL\_CFTABLE\_ENTRY are missing.

Enabler configures all cards to ATA mode (16 continuous I/O) or to IDE mode (8 and 2 continuous I/Os). Both primarily IDE and secondary IDE modes are supported.

Enabler cannot be loaded in the Windows DOS session.

Enabler provides a set of flexible configuration options via command line or external .INI file.

---

## Enabler Versions

There are three versions of ATA Enabler released: ATAENAB, CBATA and UNATA. All three versions share common features and differ only by a set of supported PC Card adapters.

---

### ATAENAB

ATAENAB is designed for Intel PCIC compatible PC Card adapters. (E.g., Intel Step A, B and C, Vadem VG-365/465/468/469, Ricoh RF5C266/366/269/369, Cirrus Logic CL-PD6710/6720/6722/6729/6730, Toshiba ToPIC, etc.)

ATAENAB will also work on most CardBus adapters, but only if the CardBus adapter is properly configured by the Computer BIOS. Most of today's notebooks BIOS will provide proper CardBus initialization.

ATAENAB will not work on CardBus adapters that are not initialized by BIOS.

---

### CBATA

Compared to ATAENAB, CBATA is especially designed to work on CardBus adapters.

In case a CardBus adapter is not properly initialized by BIOS, Enabler is able to update the CardBus adapter configuration. Most of the required initialization can be done automatically without user intervention. However, if required, user will be able to control most settings, either via command line parameters or, in complicate cases, via external .INI file

---

### UNATA

UNATA combines the best features of ATAENAB and CBATA in one driver; of course at a price of a slightly larger memory footprint compared to ATAENAB and CBATA.

UNATA is recommended for mixed CardBus/PCIC environments and for Toshiba Laptops where the PC Card adapter mode can be set via BIOS Setup (ToPIC 95/97/100 can work either in PCIC compatible or in CardBus mode).

## Installation

The Enabler may be installed either in CONFIG.SYS, AUTOEXEC.BAT or started from DOS command line. It's possible to relocate enabler into upper memory using DEVICEHIGH or INSTALLHIGH statements in CONFIG.SYS.

The Enabler provides several command line switches allowing fine-tuning of the driver. In more complicate cases you may create an initialization file with very detailed configuration instructions.

It is recommended that you fine-tune CardBus Enabler switches starting CardBus Enabler from the command line. After switches are set, you can place call of CardBus Enabler either to CONFIG.SYS or to AUTOEXEC.BAT.

---

## Memory Requirements

All versions of ATA Enablers require 4 kB of memory to access the ATA Card Information Structure (CIS). This memory is shareable amongst all installed ATA cards.

In addition CBATA and UNATA require 4 kB of memory per CardBus socket to establish the memory access to CardBus registers.

Example: You are using CBATA and want to support 2 ATA cards; then you will need 12 kB of memory (2 x 4 kB for two CardBus sockets + 4 kB CIS access).

Enabler may allocate the required memory automatically (see "Resource Management" chapter for details).

To be available inside of 1<sup>st</sup> MB (required for ATAENAB) memory should not be shadowed by BIOS and should be excluded from memory management by EMM386.EXE.

For CBATA and UNATA there is no need to allocate memory below the 1st MB (memory anywhere in 4GB address space may be used).

---

### Example of config.sys

Let's say we are working with a computer that provides 144 kB of free upper memory from 0xCC000 trough 0xEFFFF. We will be using ATAENAB and 2

ATA cards. As mentioned above, we will need 4 kB of memory for CIS access to support such a configuration. If we use the default memory address (0xD0000), we will split the upper memory into three regions, reducing the biggest available memory block to 124 kB:

0xCC000 – 0xCFFFF	16kB	Free
0xD0000 – 0xD0FFF	4kB	Used by us
0xD1000 – 0xEFFFF	124kB	Free

We can optimize the memory management by shifting our memory window to the beginning or to the end of the available upper memory. For example:

```
...
DEVICE=C:\DOS\HIMEM.SYS /TESTMEM:OFF
DEVICE=C:\DOS\EMM386.EXE NOEMS X=EF00-EFFF
DOS=HIGH,UMB
...
DEVICEHIGH=C:\ATAENAB.EXE <other switches>
...
```

In the above example our memory window is shifted to the end of the available upper memory, address 0xEF00. Please notice that used memory is excluded from EMM386.EXE memory management (via X=EF00-EFFF switch) thus enabling it for use by ATAENAB.

Example above illustrates memory management for memory located inside of 1<sup>st</sup> MB. This is required only for ATAENAB. For CBENAB and UNATA we may use memory located anywhere in 4GB address space. For example:

```
...
DEVICE=C:\DOS\HIMEM.SYS /TESTMEM:OFF
DEVICE=C:\DOS\EMM386.EXE NOEMS
DOS=HIGH,UMB
...
DEVICEHIGH=C:\CBATA.EXE <other switches>
...
```

---

## Special Note for Toshiba Notebooks.

If you are using ATAENAB on Toshiba notebooks please make sure that you set the PC Card adapter either to Auto-Select mode, or to PCIC mode via BIOS setup. The CardBus/PC-Card 16 mode is not supported. When using CBATA please make sure that you set the PC Card adapter to CardBus/PC-Card 16 mode. The PCIC and Auto-Select modes aren't supported. UNATA works in either mode.

---

## Fine-tuning

If your BIOS configure CardBus Adapter properly, no command line parameters are necessary. Otherwise fine-tuning could be required.

The fine-tuning could be performed either in standard installation mode via command line options, or in advanced installation mode via an external INI file.

Following chapters describe both modes in detail.

In both modes several optional parameters require a numeric value. Any numeric value could be entered as decimal (default form) or hexadecimal (with 0x prefix) number. For example: `/PI:10` and `/PI:0x0A` both specify IRQ 10.

In both modes several optional parameters require `<use>` values. Use values may typically be specified as ON, OFF or AUTO in upper or lower case.

**This page is intentionally blank.**

## Resource management.

During the loading Enabler collects information about resources (IRQ, I/O and memory) used by PCI, PnP and Option ROMs. Enabler considers such resources as non-free and excludes them from hardware configuration procedure.

In non-PnP computer Enabler may be not able detect all used resources (e.g. resources used by Legacy ISA devices). In such case you can adjust resource filters manually by adding such resources into [Resource] section of the external .INI file. The list of all detected resources is displayed if Enabler is started with /V+ switch.

Algorithm of resource management is as follow:

1. Collect PCI, PnP and Option ROM resource information.
2. Result is combined with resource information in [Resources] section of .INI file (if file and section exist).
3. Resulting resources collection is used as resource filter.
4. Resource assignment via external .INI file (e.g. bridge filter settings or ExCA address register) overrides assigned resources unconditionally, even if specified resource is unavailable. If resource is unavailable and Enabler is started in verbose mode (with /V switch) - a warning is issued.
5. If specific resource is assigned by BIOS, Enabler keeps such resource (unless otherwise is directed by .INI file).
6. If specific resource is not assigned by BIOS and there is no .INI file settings, Enabler will allocate required resource automatically:
  - a) Command line settings specify preferable resources to be used if available.
  - b) If preferable resources are not available or if no preferable resources specified via command line, any available (according to filters set during the loading) resource will be used.

Note: For I/O and memory pools Enabler use preferable resource from bottom up (in other words, minimal possible address is used first).

Intel 16-bit PCIC compatible (non-CardBus) PC Card adapters are limited by using memory inside of 1st MB. PCI and CardBus adapters may use memory anywhere in 4GB address space.

## Command Line Options.

Command line options can be specified in any order, either in upper case or in lower case. Typical parameters start with slash (/) or dash (-) followed by one or two letters (e.g. /H or /VR).

Some command line options require an additional value. In this case parameters should be followed by colon (:), or equal (=) then value (e.g. /M:0xEC000 or /IO=150).

Command line options are separated by spaces.

---

### Common Command Line Options

Following options are common for all versions of ATA Enablers.

- /H | ?    Print online help
- /E        Silent mode: No display output.

---

### Resource Management

- /M:n     Start address of memory pool     (Default: 0xD0000)
- /IO:n    Start address of I/O pool            (Default: 0x340)

---

### Diagnostics and Fine-tuning

- `/F=filename` Specifies the initialization file pathname.
- By default the initialization file has the same name as the Enabler, but with extension .INI (i.e., ATAENAB.INI, CBATA.INI and UNATA.INI correspondingly). Enabler will be looking for the .INI file in the same subdirectory where enabler itself is located.
- Using /F parameter user may override path and name of the default initialization file. (E.g., /F:E:\Settings\ata.ini)
- `/V[:<file>]` Verbose mode: Configuration steps will be outlined on console or to file (if file pathname is specified).
- `/FL` Flush log file to disk immediately after each update (may be useful in case of system hang).
- `/ID` Displays identified drive information.
- `/Ln:Letter` Specifies the drive letter assigned to partition 'n', where n could be in range of 0 to 9 (E.g., /L0:M /L1:K /L2:Z)
- Drive letters selected by the user can only be taken into account if the Enabler is loaded in TSR mode (e.g., from AUTOEXEC.BAT or command line), and will be ignored if the device driver is loaded from CONFIG.SYS.
- User may assign more than 10 drive letters using the .INI file.
- `/LS` Disables use of LBA mode.
- Some old ATA PC cards (e.g., Seagate Technology ST7050P) support LBA mode improperly. If you've detected numerous errors during read/write of such a PC card, try to disable the LBA support.
- `/VR` Enable numerous scanning by Read Verify command during formatting.
- On some ATA cards not all 'bad' sectors are detected during the format process. In this case try to enable numerous scanning by Read Verify command during formatting.
- This parameter, if specified, applies to all drives supported by ATA Enabler.

---

## Unloading

---

`/UL` Unload resident part (not supported, if driver is loaded via config.sys)

---

## ATAENAB and UNATA Specific Options

Following command line options are for ATAENAB and UNATA only. Using them the user may override the default base address of the PC Card adapter or override the default socket number.

In case of UNATA: Parameters only apply to non-CardBus sockets.

`/B:n` PCMCIA adapter base I/O address (Default: 0x03E0)

`/S:n` PCMCIA socket number (Default: 0)

Please notice, that if you want to support more than one non-CardBus socket using ATAENAB or UNATA, you will have to create an INI file.

---

## CBATA and UNATA Specific Options

Following command line options are for CBATA and UNATA only. Using them the user may control the initialization of CardBus sockets.

On most of recently made notebooks the BIOS will configure the interrupt mode properly. You should not specify `/PI`, `/MI`, `/MX` and `/OZ` options unless you are sure that the interrupt mode is not set properly.

`/PI:n` PCI Interrupt level (Default: 0x0B)

`/IB:B:D:F`, `<use>` Ignore PCI-2-PCI bridge. This switch may be used for subtractive-decode PCI-2-PCI bridges (bridges which pass all memory and I/O requests behind).  
B - bus (0-255), D - device (0-31), F - function (0-7).

ON Bridge is ignored (i.e. bridge is subtractive).

OFF Bridge is not ignored (even if physically subtractive).

AUTO Bridge's decoding mode should be read out of bridge

`/FI` Force updating of PCI IRQ routing table for socket (even if selected IRQ is already in table).

`/BS` Performs PCI bus scan and termination. Information will be displayed on console

- `/EC: <use>`    **ON**            Use external power switch clock for adapter.  
                 **OFF**            Use power switch clock generated by PCI clock
- Note: This switch is ignored for all PC Card adapters except Texas Instruments PCI121x/122x and TI PCI125x/14xx/44xx families.
- `/IM: m`    Set interrupt mode. Possible values are:
- |             |                                     |   |
|-------------|-------------------------------------|---|
| <b>Par</b>  | TI1130/1x31:                        | use parallel ISA-type interrupts  |
|             | TI12xx/14xx/44xx:                   | use parallel ISA and PCI interrupts   |
|             | CL-6832, OZ6832:                    | use External-Hardware Interrupt mode  |
|             | Ricoh RB5C478:                      | use parallel interrupt mode   |
| <b>Ser</b>  | TI1130/1x31:                        | use serialized interrupt type scheme  |
|             | TI12xx/14xx/44xx:                   | use serialized ISA and PCI interrupts   |
|             | CL-6832, OZ6832:                    | use PC/PCI Serial Interrupt protocol  |
|             | Ricoh RB5C478:                      | use serialized interrupt mode   |
| <b>PPci</b> | TI 12xx/14xx/44xx, CI-6832, OZ6832: | use parallel PCI interrupts only  |
| <b>ISPP</b> | TI 12xx/14xx/44xx only:             | use serialized ISA and parallel PCI interrupts. Will be ignored for all other adapters. |
| <b>Pway</b> | CL-6832, OZ6832:                    | use PCI/Way Interrupt Signaling mode. Will be ignored for all other adapters            |
- `/MX: <use>`    Initialize the multiplex IRQ routing register. This switch is only used on TI PCI122x/125x/14xx/44xx adapters and is ignored for all other adapters.
- `/OZ: 97: <use>`    O2Micro PC Card adapter only: set/reset PC97 IRQ bit
- `/OZ: IL: <use>`    O2Micro PC Card adapter only: set/reset ISA Legacy bit

## Initialization File

The initialization file is a text file that contains special settings. These settings allow you to configure computer hardware components so your CardBus adapters and/or ATA cards become functional.

Any text in the .INI file starting after the semicolon (;) to end of the line, is interpreted as a comment.

Using the INI file you may set both, system-wide and per-socket parameters in any combination.

There are three types of sections defined: Common, Socket and PCI.

---

### Common Section

Only one [Common] section can be defined in INI file. The Common Section defines ATA configuration parameters that apply to all ATA cards in the system. Following parameters are defined: 'CISBase', 'ATAIoBase', 'ReadVerifyRetry', 'LBAMode', 'DriveLetterN' (where N is a number between 0 and 9).

These parameters represent an alternative for command line options /M, /IO, /VR, /LS and /Ln correspondingly.

Example:

```
[Common]
CISBase = 0xEC00
ATAIoBase = 0x800
ReadVerifyRetry = Off
LBAMode = On
DriveLetter0 = M
DriveLetter1 = K
DriveLetter2 = Z
```

## Socket Sections

Socket Sections is interpreted only by ATAENAB and UNATA and can only be used for non-CardBus sockets. Both enablers support up to 8 non-CardBus sockets. The socket configuration of such sockets should be described in sections [Socket0] to [Socket7].

Following values are defined for Socket Sections: 'LegacyBase', 'Socket', 'CISBase', 'ATAIoBase', 'ReadVerifyRetry', 'LBAMode', 'SkipSocket', 'DriveLetterN' (where N is a number between 0 and 9), 'IdeIoBase8', 'IdeIoBase2' and 'SkipAtaConfig'.

Parameters in Socket Sections represent an alternative for command line options /B, /S, /M, /IO, /VR, /LS and /Ln correspondingly.

There is no corresponding command line option for 'SkipSocket' parameter. It allows to skip (ignore) particular socket. Valid values are 0 and 1.

There is no corresponding command line options for 'IdeIoBase8', 'IdeIoBase2' and 'SkipAtaConfig' parameters. 'IdeIoBase8' and 'IdeIoBase2' are IDE 8 and 2 continuous I/O ranges. 'SkipAtaConfig' allows to skip (ignore) ATA config (16 continuous I/O) and use only IDE config (if any). Valid values are ON and OFF.

Parameters specified in Socket Sections override parameters specified in Common Section and Command Line. Using them you may control initialization order and execution mode. For example, you may enable LBA mode by default, but disable it for selected socket.

Example:

```
[Socket0]
LegacyBase = 0x3E2
Socket = 1
CISBase = 0xEC00
ATAIoBase = 0x800
ReadVerifyRetry = Off
LBAMode = On
DriveLetter0 = M
DriveLetter1 = K
DriveLetter2 = Z
IdeIoBase8 = 0x180
IdeIoBase2 = 0x386
SkipAtaConfig = ON           Skip ATA (use IDE) config
SkipSocket = 1              Skip this socket

[Socket1]
LegacyBase = 0x3E0
```

```

Socket = 0
LBAMode = Off
DriveLetter0 = Y
SkipSocket = 0

```

Please note, that [Socket1] definition omits several settings (such as ATAIoBase). In this case settings from [Common] section or command line will be used.

---

## PCI Sections

PCI Sections is interpreted only by CBATA and UNATA and can only be used for CardBus sockets and PCI-2-PCI bridges. Both enablers support any number of CardBus sockets.

All section names are relative to original PCI subsystem configuration. You can see that configuration if you switch program to verbose mode. (/V key) or use the /BS switch.

In order to specify the required configuration of the PCI device, you need to know the location of the device. This location is specified in Bus:Device:Function form. Location 0:A:1 describes the first function of PCI device number 10 ("A" in hexadecimal format) located on PCI bus 0. Initialization file section used for the configuration of such a device should have the name [0:A:1].

Value names depend on device type.

For CardBus adapters, following values are defined: 'IoFilter0', 'IoFilter1', 'MemFilter0', 'MemFilter1', 'ExCABase', 'LegacyBase', 'IRQ', 'ISAEnable', 'VGAEnable', 'CISBase', 'ATAIoBase', 'ReadVerifyRetry', 'LBAMode', 'SkipSocket', 'DriveLetterN' (where N is a number between 0 and 9), 'IdeIoBase8', 'IdeIoBase2' and 'SkipAtaConfig'.

For PCI-to-PCI bridges, following values are defined: 'IoFilter', 'MemFilter', 'PreFilter', 'IRQ', 'ISAEnable', 'VGAEnable', 'IgnoreBridge'.

These parameters define the CardBus or PCI-to-PCI bridge memory filter base and limit (or size) both memory filters (prefetchable and non-prefetchable).

The CardBus base and size values should have 4K alignments according to Yenta specification and limit (maximum address) should be aligned to 4K minus one.

Following formats could be used to specify bridge memory filter ranges:

---

**MemFilter,  
PreFilter,  
MemFilter0,  
MemFilter1**

BaseAddress:Limit. This means that we define the first and exactly the last integer number in a range (e.g., 0xA0000000:0xA0000FFF specifies 4K ranges). If you wish to disable positive decoding of PCI-to-PCI bridge's filter range, please specify something like: 200000:1ffff.

If you prefer to specify filter size rather than filter max address you should use BaseAddress,Size format, where Size by default is a literal decimal number. In this case you also could use 'M' or 'K' postfixes (e.g., 0xA0000000,1M specifies 1M memory filter starting at 0xA0000000 address).

Attention! All values are treated as hexadecimal even if there is no "0x" prefix.

Examples:

```
MemFilter0=A0000000h:A0000fffh      4K memory range
MemFilter0=A0000000h,4K             4K memory range
MemFilter0=A1000000h:A1001fffh,Pre 8K pref. memory
MemFilter0=A1000000h,8K,Pre        8K pref. memory
```

---

**IoFilter,  
IoFilter0,  
IoFilter1**

This is the bridge I/O range base and limit (or size). For CardBus base and size values must have double-word alignment.

Examples:

```
IoFilter = 0xD000:0xD00F           16-bytes I/O range
```

---

**IRQ**

CardBus interrupt line settings. According to PC/AT architecture this value must not exceed 15.

```
IRQ = 10                           10th interrupt
```

```
IRQ = 10                           10th interrupt
```

---

**ISAEnable**

Control setting of ISA Enable bit in Bridge Control Register. Valid values are 0 and 1.

Example:

```
ISAEnable = 1                       Set ISA Enable bit
```

---

**VGAEnable**

Control setting of VGA Enable bit in Bridge Control Register. Valid values are 0 and 1.

Example:

```
VGAEnable = 0                       Reset VGAEnable bit
```

---

**ExCABase**

This option is used to mountain CardBus socket register/ExCA registers base address register. Available values: any 32-bit hexadecimal number aligned to 4K boundaries.

Example:

```
ExCABase = D800000h
```

---

### LegacyBase

This is PC Card 16-Bit IF legacy mode base address. Available values: any 16-bit hexadecimal number aligned to double-word boundary.

Example:

```
LegacyBase = 3e0h
```

---

### PrimaryBus, SecondaryBus, SubordinateBus

These options control bus number assignments for PCI-to-PCI and CardBus bridges. Available values: any decimal number not greater than 255.

Example:

```
[0:A:0]                                TI PCI1131 socket
PrimaryBus      = 0
SecondaryBus    = 1                Set sec. bus to 1
SubordinateBus  = 3                Set sub. bus to 3
```

```
; Now socket 0 of TI PCI1131 CardBus adapter accepts
; I/O requests to buses from 1 to 3 inclusive
```

```
[0:A:1]                                TI PCI1131 socket
PrimaryBus      = 0
SecondaryBus    = 4                Set sec. bus to 4
SubordinateBus  = 3                Set sub. bus to 3
```

---

### SkipSocket

Allows skip (ignore) particular CardBus socket . Valid values are 0 and 1

Example:

```
SkipSocket = 1                        Skip socket
```

---

### IgnoreBridge

Allows ignore PCI-2-PCI bridge. This option may be used for subtractive-decode PCI-2-PCI bridges (bridges which pass all memory and I/O requests behind). Valid values are ON, OFF and AUTO. 'ON' means that bridge is ignored (i.e. bridge is subtractive). 'OFF' means that bridge is not ignored (even if physically subtractive). 'AUTO' means that bridge's decoding mode should be read out of bridge.

Example:

```
IgnoreBridge = ON                    Ignore bridge
```

---

**CISBase**  
**ATAIoBase**  
**ReadVerifyRetry**  
**LBAMode**  
**DriveLetter**

Card entries 'CISBase', 'ATAIoBase', 'ReadVerifyRetry', 'LBAMode' and 'DriveLetterN' (where N is a number between 0 and 9), define configuration of ATA card.

Example:

```
[0:A:1]                                Card in Socket 1
CISBase = 0xD4000000
ATAIoBase = 0x800
ReadVerifyRetry = Off
LBAMode = On
DriveLetter0 = M
DriveLetter1 = K
DriveLetter2 = Z
```

---

**IdeloBase8**  
**IdeloBase2**  
**SkipAtaConfig**

Card entries 'IdeloBase8', 'IdeloBase2' and 'SkipAtaConfig' define configuration of ATA card for IDE mode.

Example:

```
[0:A:0]                                Card in Socket 0
CISBase = 0xD4000000
IdeIoBase8 = 0x1A0
IdeIoBase2 = 0x3A6
SkipAtaConfig = ON
ReadVerifyRetry = Off
LBAMode = On
DriveLetter0 = 0
```

---

## Resources Section

The Enabler INI file offers user an extended control over resource allocation. Using [Resources] section of the Enabler INI file user can include or exclude specific resources for use by Enabler.

During the loading Enabler collects information about resources (IRQ, I/O and memory) used by PCI, PnP and Option ROMs. Enabler considers such resources as non-free and excludes them from hardware configuration procedure. In non-PnP computer Enabler may be not able detect all used resources (e.g. resources used by Legacy ISA devices). Such undetected resources may cause resource conflict. In such case you can adjust resource filters manually by adding resources into this [Resources] section for the Enabler's INI file.

Only one [Resources] section can be defined in INI file.

To add resource you should define its description in separate line.

Any resource may be either excluded or included to hardware configuration procedure. To exclude resource its description should start from 'XMEM' (for memory), 'XIO' (for I/O) or 'XIRQ' (for IRQ). To include resource its description should start from 'MEM' (for memory), 'IO' (for I/O) or 'IRQ' (for IRQ).

For memory and I/O you also specify base address and size of resource range separated by commas. For IRQ you specify only IRQ number separated by comma.

Finally for any resource you may specify sharable attribute separated by comma. There are next possible attributes:

- E - Resource is available for exclusive (non-shared) use.
- D - Resource is available for dynamic-shared use.
- T - Resource is available for time-shared use.

You may specify sharable attribute for any resource, either excluded or included. However for excluded resource it makes no sense (since excluded resource is unavailable for any using) and will be ignored.

If no shared attribute specified, resource is assumed as an exclusive use resource.

Example:

```
[Resources]
XMEM = 0xD0000, 0x400           ; Exclude memory
                                ; [0xD0000-0xD03FF]
XIO = 0, 0x100                 ; Exclude I/O [0-0xFF]
IRQ = 5, D                     ; Include IRQ 5 with dynamic
                                ; share
XIRQ = 6                       ; Exclude IRQ 6
XMEM = 0x100000, 0x7FF0000     ; Exclude memory
                                ; [1MB - 2GB]

; Exclude memory [0xE0000-0xE7FFF] except [0xE1000-0xE2FFF]

XMEM = 0xE0000, 0x8000, E      ; Exclude [0xE0000-0xE7FFF]
MEM = 0xE1000, 0x2000, E      ; Include [0xE1000-0xE2FFF]
```

**This page is intentionally blank.**

## Additional information

---

### How to get Technical Support for ATA Enabler

Please send an E-mail in English (we can't process technical support questions in any other language) to [technical support](mailto:support@tssc.de) ([support@tssc.de](mailto:support@tssc.de)).

In your request please specify:

1. Description of your problem.
2. The memory card name and manufacturer **exactly** as it appears on the card.

Please attach to your e-mail following files:

1. Please run ATA Enabler with /V+[:<file>] /ID switches (plus all parameters you normally use), capture the output to the file and attach it to your e-mail. E.g.  
C:\>UNATA /V+:LOG.TXT /ID  
or  
DEVICE = UNATA.EXE /V+:LOG.TXT /ID
2. CONFIG.SYS
3. AUTOEXEC.BAT
4. Enabler initialization file if one exists.

---

## Example of INI file

Following sample shows an example of UNATA initialization file for the system with one CardBus and one PCIC compatible adapter

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;          UNATA Enabler. Sample of Initialization file
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; PCI-2-PCI Bridge between bus 0 and bus2
[0:1E:0]
PreFilter=0x60000000:0x60FFFFFF

; CardBus socket 0
[2:B:0]
ExCABase=0xD2000                                ; Set ExCA address to 0xD2000
CISBase=0xF0000000
ATAIoBase = 0x150
IdeIoBase8 = 0x1A0
IdeIoBase2 = 0x3A6
SkipAtaConfig = OFF
ReadVerifyRetry=Off
LBAMode=On
DriveLetter0=M
DriveLetter1=K
DriveLetter2=Z

; PCIC Socket 1
[Socket0]
LegacyBase=0x3E0
Socket=1
CISBase=0xD3000
ATAIoBase=0x140
ReadVerifyRetry=Off
LBAMode=On
DriveLetter0=N
DriveLetter1=O
```

**This page is intentionally blank.**

# **CUSTOMER LICENSE AGREEMENT**

APSoft thanks you for selecting one of their products for your computer. This is the APSOFT Customer License Agreement which describes APSOFT's license terms.

**- PLEASE READ THIS NOTICE CAREFULLY -**

**DO NOT DOWNLOAD OR USE THE SOFTWARE UNTIL YOU HAVE READ THE LICENSE AGREEMENT. BY ACCEPTING TO DOWNLOAD THIS SOFTWARE, APSOFT WILL ASSUME THAT YOU HAVE AGREED TO BE BOUND BY THIS STANDARD AGREEMENT. IF YOU DO NOT ACCEPT THE TERMS OF THIS LICENSE, YOU MUST RETURN THE PACKAGE UNUSED AND UNOPENED TO THE PARTY FROM WHOM YOU RECEIVED IT.**

**Grant of License.** APSOFT grants to you and you accept a license to use the programs and related materials ("Software") delivered with this License Agreement. This software is licensed for use on one computer at a time. You may run this software on as many machines as you like provided there is no possibility that it can be run on more than one machine at a time. The software should never be installed on the hard drive of more than one computer. If the software is installed on a network hard drive, access must be restricted to a single user. You agree that you will not transfer or sublicense these rights. If you use the Software on more than one computer at a time, you must license additional copies or request a multi-user license from APSOFT.

**Term.** This License Agreement is effective from the day you receive the Software, and continues until you return the original magnetic media and all copies of the Software to APSOFT. APSOFT shall have the right to terminate this license if you violate any of its provisions. APSOFT or its licensors own all rights, title, and interest including all worldwide copyrights, in the Software and all copies of the Software.

**Your Agreement.** You agree not to transfer the Software in any form to any party without the prior written consent of APSOFT. You further agree not to copy the Software in whole or in part, except for backup copies, unless APSOFT consents in writing. Each of these copies must have a label placed on the magnetic media showing the program name, copyright, and trademark designation in the same form as the original Software. You will use your best efforts and take all reasonable steps to protect the Software from unauthorized reproduction, publication, disclosure, or distribution, and you agree not to disassemble, decompile, reverse engineer, or transmit the Software in any form or by any means. You understand that the unauthorized reproduction of the Software and/or transfer of any copy may be a serious crime, as well as subjecting you to damages and attorney fees.

**Disclaimer.** APSOFT MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE OR MERCHANTABILITY, AND APSOFT SHALL NOT BE LIABLE FOR TORT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES SUCH AS LOSS OF PROFITS OR LOSS OF GOODWILL FROM THE USE OR INABILITY TO USE THE SOFTWARE FOR ANY PURPOSE. SOME STATES MAY NOT ALLOW THIS DISCLAIMER SO THIS LANGUAGE MAY NOT APPLY TO YOU. IN SUCH CASE, OUR LIABILITY SHALL BE LIMITED TO THE REFUND OF THE APSOFT LIST PRICE.

**Updates.** APSOFT will do its best to notify you of subsequent updates released to the public or major corrections and the price for which they may be obtained. All updates and corrections which are provided to you shall become part of the Software and be governed by the terms of this license agreement.

**Miscellaneous.** This is the only agreement between you and APSOFT and it cannot and shall not be modified by purchase orders, advertising or other representations of anyone, unless a written amendment has been signed by one of our company officers.