USB Mass Storage Enabler for DOS

Copyright Notice and Disclaimer

Issue date 14-Aug-2005.
USB Mass Storage Enabler for DOS

APSoft.
Siemensstrasse 1b
84405 Dorfen
Germany

Tel: +49 (0) 8081 955 89 0
Fax: +49 (0) 8081 955 89 11
Internet: http://www.tssc.de

Table of Contents

Introduction

This manual provides an overview of the USB Mass Storage Enabler for DOS. All released versions are collectively named “USB Mass Storage Enabler” or simple “Enabler” throughout this manual

Enabler is a small, but powerful driver supporting any kind of USB adapter such as Adaptec AUA-1420 USB2Connect Adapter, Orange Micro Inc. OrangeUSB - USB 2.0 Hi-Speed, etc.

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 USB Mass Storage Enabler occupies approximately 1 kB of memory or larger, depend from used features.

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 support of USB devices without load additional driver for your device (e.g. ASPICD.SYS for CD-ROM device).

Enabler does not provide support of audio CD disks for USB CD-ROM devices without load additional driver for your device.

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. Number of supported partitions is limited only by number of available drive letters.

Enabler does not provide hot-swap support. All USB devices should be connected to the computer before Enabler is started.

Enabler cannot be loaded in the Windows DOS session.

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

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

C H A P T E R 1

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.

Deployment

In order to use USB devices (such as Hard disk or CD-ROM) you typically will need to load USB Mass Storage Enabler and driver for your device. E.g. usage of CD-ROM requires ASPICD.SYS; usage of USB Disk – ASPIDISK.SYS, etc. Required drivers could be found in the Adaptec EZ-SCSII package.

However USB Mass Storage Enabler may provide support of USB devices without load any additional drivers.

For use CD-ROM you will also need to load MSCDEX.EXE in your AUTOEXEC.BAT.

Let's take an example. We are willing configure Adaptec AUA-1420 card. The Adaptec card consists of two USB 1.1 OHCI adapters and one USB 2.0 EHCI adapter. On our test platform we are using memory at 0xD4000. In such case USB Mass Storage Enabler will use memory as follow:

4kB at 0xD4000 - will be used for OHCI Adapter (PCI function 0)
4kB at 0xD5000 - will be used for OHCI Adapter (PCI function 1)
4kB at 0xD6000 - will be used for EHCI Adapter (PCI function 2)

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

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

USB Mass Storage Enabler may use memory located anywhere in 4GB address space.

Example of config.sys

You can invoke USB Mass Storage Enabler in the any point. For example:

```
SWITCHES  = /F
BUFFERS   = 10,0
FILES     = 99
BREAK     = ON
FCBS      = 16,8
LASTDRIVE = Z
STACKS    = 9,512
DEVICE    = \dos\himem.sys /testmem:off
DEVICE    = \dos\emm386.exe noems
DOS       = HIGH,UMB
DEVICE    = \USBMASS.EXE
```

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.

C H A P T E R 2

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.

CHAPTER 3

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).

Command line options are separated by spaces.

Common Command Line Options

Following options are common for all versions of USB Mass Storage Enabler.

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

Resource Management

`/PI:n`	PCI Interrupt level	(Default: 0x0B)
`/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., USBMASS.INI). 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\usbmass.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.

`/DOSONLY` By default, Enabler assigns drive letters to all found drive partitions regardless the file system. This mode allows e.g. reformat unrecognized file system to one supported by DOS.

If /DOSONLY switch is added to the command line, the drive letters will be assigned only to partitions with file system recognized by DOS (FAT12 & FAT16 under DOS 6.x and FAT12, FAT16 & FAT32 under DOS.

Use this mode if you plan to install third-party file system driver for non-DOS partitions. For example, Symantec Norton Ghost 8.x contains a built-in NTFS driver allowing backup/restore operation to/from NTFS drive under DOS.

`/IB:B:D:F` Ignore PCI-2-PCI bridge. This switch may be used if PCI-2-PCI bridge pass all memory and I/O requests behind. B - bus (0-255), D - device (0-31), F - function (0-7). Alternative way to ignore bridge is via .INI file.

`/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

`/TC:n` Maximal time in sec waiting for device connect signal after powering of USB port (Value 1 - 60, Default 2 sec)

`/TO:n` Time that BIOS need for establish USB 2.0 companion port ownership in sec (Value 1 - 60, Default 2 sec)

`/LA:n` USB 2.0 Latency in microsec (Default: 1000 microsec)

USB 2.0 Latency is time delay inserted after every USB 2.0 transaction. Values 100-60000 are allowed.

ASPI Management

`/NU` Do not move part of resident code to upper memory.

`/I` Ignore (do not install) INT 13 interface support for ASPI hard disk devices. Also no drive letter will be created.

In this case driver for your device is required.

`/IC` Ignore (do not install) support for ASPI CD-ROM disk devices.

In this case driver for your device is required.

`/D:devname` Specifies device name of CD-ROM driver (Default: ASPICD01)

Can be used by MSCDEX.EXE to support ASPI CD-ROM disk devices.

Unloading

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

This page is intentionally blank.

C H A P T E R 4

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 USB CardBus card 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 PCI-2-PCI Bridges and USB card parameters in any combination.

PCI Sections

PCI Section describes PCI Device. Enabler supports any number of PCI Sections in the .INI file.

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’, ‘VGAEnable’, ‘SkipSocket’.

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

For USB adapter, following values are defined: ‘IRQ’, ‘BARMask’, ‘BAR0’, ‘BAR1’, ‘BAR2’, ‘BAR3’, ‘BAR4’, ‘BAR5’, ‘BAR6’, ‘SkipAdapter’.

MemFilter, PreFilter, MemFilter0, MemFilter1

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

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:

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: 200000:1fffff.

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:

```
ISAEnabe = 1                        Set ISA Enable bit
```

VGAEnable

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

Example:

```
VGAEnabe = 0                          Reset VGAEnabl 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 = D8000000h
```

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
```

BARMask

This option allows specifying mask of BARs that will be initialized by enabler. If bit in the mask is set, the Enabler will initialize corresponding BAR. If bit is reset, Enabler will ignore corresponding BAR.

Example:

```
BARMask          = 3                 Set BAR0 and BAR1
```

BAR0 - BAR6

```
These options allow specifying values for card base
address registers (BARs).
```

If Enabler configures BARs by itself then it automatically adjust parent CardBus bridge filters (if required). However if these options are used then Enabler assumes that parent CardBus bridge filters were already set to proper values.

Example:

```
BAR0             = 0xFFEFE000        Set BAR0
BAR1             = 0xFD00            Set BAR1
```

SkipAdapter

Allows to skip (ignore) particular USB adapter . Valid values are 0 and 1

Example:

```
SkipAdapter = 1                      Skip Adapter
```

SkipPort1 - SkipPort16

Allows skip (ignore) particular USB port. It may save time during the loading. Valid values are 0 and 1.

Example:

```
SkipPort2 = 1                        Skip USB port 2
```

TimeoutPort1 - TimeoutPort16

Allows specify maximal time in milliseconds waiting for device connect signal after powering of particular USB port. Valid value should not exceed 65535.

Example:

```
TimeoutPort1 = 30000                    Wait max 30 sec
```

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]
```

```
[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]
```

A P P E N D I X

Additional information

How to get Technical Support for USB Mass Storage Enabler

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

In your request please specify:

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

Please attach to your e-mail following files:

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

Example of INI file

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

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;            USBMASS 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        = 0xFF000000
PrimaryBus      = 2
SecindaryBus    = 0x81
SubordinateBus  = 0x81
MemFilter0      = 0xFFEFE000,4K
MemFilter1      = 0xFAC00000:0xFEBFFFFF
IOFilter0       = 0xFD00:0xFDFF
IOFilter1       = 0xFC00:0xFCFF
LegacyBase      = 0
IRQ             = 11
ISAEnable       = 0
VGAEnable       = 0

; USB Card
[81:0:0]
BARMask         = 1
BAR0            = 0xFD00
IRQ             = 11
```

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.