Technical Note TN1098

ATA Device Software Guide Additions and Corrections

CONTENTS

This Technote lists errors and additions to the *ATA Device Software Guide*

We include both corrections to the original guide, as well as some minor additions for ATA Manager 4.0. ATA Manager 4.0 was introduced with the PowerBook 3400.

This Technote is directed at developers who wish to call the ATA Manager directly. Normally, applications would never call the ATA Manager directly; instead, they use the appropriate driver for the device in question. The ATA Manager is helpful for developers who are creating ATA device drivers, and for specialized applications which install such device drivers.

Updated: [May 18 1998]

Errors in the *ATA Device Software Guide*

Throughout. All mentions of the ATA-2 standard should be updated. The ATA Manager supports ATA-3 devices as well.

Page 4: *The ATA disk driver usually has a driver reference number of -54 (decimal) but may also have a different reference number if -54 is taken when the driver is loaded. The driver name is* `.ATDISK`. *Like all Macintosh device drivers, the ATA disk driver can be called by using either the driver reference number or the driver name,* `.ATDISK`.

You should never rely on the driver reference number. Use the `OpenDriver` call with the driver name to retrieve the driver reference number, and use the driver reference number returned by the `OpenDriver` call for all subsequent driver calls. The name of the driver is `.ATADisk`, not `.ATDisk`.

Page 6: *The* `open` *routine should not be called to open the ATA disk driver...*

The `open` routine is harmless. You may call `open` to get the driver reference number for the ATA driver.

Page 9: *The* `verify` *control function...*

The `verify` control function does nothing, and returns `noErr` if valid parameters are passed to it.

Page 9: *The* `format` *control function...*

The `format` control function does nothing, and returns `noErr` if valid parameters are passed to it. If you need to do a low-level format of a drive, you need to consult the ANSI specification.

Page 11: *The* `return drive characteristics` *function returns information about the characteristics of the specified drive, as defined in* Inside Macintosh, Volume V.

The `return drive characteristics` function returns information about the characteristics of the specified drive, as defined in Technote DV 525, Disk Driver Q&As

Page 38: *ATA_RegAccess*

The ATA_RegAccess function does not function properly with ATA Manager version 4.0.0 or 4.1.0. It does function correctly in ATA Manager version 3, and in ATA Manager version 4.1.1, which ships as part of Mac OS 8. To get the ATA Manager version number, use the ATAMgrInquiry function call (documented on page 56). The version is returned in the MgrVersion field.

```
MgrVersion.majorRev = $04
MgrVersion.minorAndBugRev = $00 or $10 are versions which do NOT work.

MgrVersion.majorRev = $04
MgrVersion.minorAndBugRev = $11 does work.
```

Page 48:

Device Config Structure in ATA 4.0

Several of the fields of this structure are obsolete starting with ATA Manager 4.0.

```
struct ATADevConfig
{
     SInt32     ataConfigSetting;       //
<->: Configuration setting*/
     UInt8      ataPIOSpeedMode;        //
<->: Device access speed in PIO Mode
     UInt8      reserved;               // padding
     UInt16     atapcValid;             //
<->: Obsolete with ATA 4
     UInt16     ataRWMultipleCount;
// Reserved for future (not supported yet)
     UInt16     ataSectorsPerCylinder;
// Reserved for future (not supported yet)
     UInt16     ataHeads;               //
// Reserved for future (not supported yet)
     UInt16     ataSectorsPerTrack;
// Reserved for future (not supported yet)
     UInt16     ataSocketNumber;        //
<--: (No longer supported with ATA 4)
     UInt8      ataSocketType;          //
<--: Specifies the socket type
     UInt8      ataDeviceType;          //
<--: Specifies the device type (get config only)
     UInt8      atapcAccessMode;        //
<->: Obsolete with ATA 4
     UInt8      atapcVcc;               //
<->: Obsolete with ATA 4
     UInt8      atapcVpp1;              //
<->: Obsolete with ATA 4
     UInt8      atapcVpp2;              //
<->: Obsolete with ATA 4
     UInt8      atapcStatus;            //
<->: Obsolete with ATA 4
     UInt8      atapcPin;               //
<->: Obsolete with ATA 4
     UInt8      atapcCopy;              //
<->: Obsolete with ATA 4
     UInt8      atapcConfigIndex;       //
<->: Obsolete with ATA 4
     UInt8      ataSingleDMASpeed;      //
<->: Single Word DMA Timing Class
     UInt8      ataMultiDMASpeed;       //
<->: Multiple Word DMA Timing Class
     UInt16     ataPIOCycleTime;        //
<->: Cycle time for PIO mode
     UInt16     ataMultiCycleTime;      //
<->: Cycle time for Multiword DMA mode
     UInt16     Reserved1[7];           // Reserved for future
};
typedef struct ATADevConfig ATADevConfig;
```

ataConfigSetting

This field is used to specify general device configuration information. In general, `ataConfigSetting` is used for things which might be device-configurable, but which might only be known at the ATA driver level. Some of the bits which were defined in previous versions of the ATA Manager are now obsolete. In the current implementation of ATA Manager 4.0, only one bit is used:

Bits 5-0	Reserved	Should be 0.
Bit 6	ATAPIPacketDRQ	1=Check for Interrupt DRQ on ATAPI command DRQ 0=Default-Check only for assertion of command packet DRQ
Bits 31-7	Reserved	Must be 0.

ataPIOSpeedMode

Device access speed in Polled I/O (PIO) Mode.

atapcValid

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

ataRWMultipleCount

Reserved for future (not supported yet).

ataSectorsPerCylinder

Reserved for future (not supported yet).

ataHeads

Reserved for future (not supported yet).

ataSectorsPerTrack

Reserved for future (not supported yet).

ataSocketNumber

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

ataSocketType

Specifies the type of socket in which this ATA bus is located. Current socket types are:

```
1 (kATASocketInternal)
2 (kATASocketMB)
3 (kATASocketPCMCIA)
```

ataDeviceType

Specifies the type of device in this ATA bus. Starting with ATA Manager 4.0, this field only returns one of three device types:

```
0 (kATADeviceUnknown)
1 (kATADeviceATA)
2 (kATADeviceATAPI)
```

Older versions of the ATA Manager had a device type defined for PCMCIA devices. This old device type (`0x03`) will no longer be returned by ATA Manager 4. This value will remain reserved in the future so that old drivers will not get confused. If a driver or other client wants to know if a device is a PC Card device, it should check the `ataSocketType` field instead.

atapcAccessMode

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager to support a different type of accessing mode for PC Card devices. This mode was never implemented, and is not supported.

atapcVcc

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

atapcVpp1

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

atapcVpp2

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

atapcStatus

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

atapcPin

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

atapcCopy

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

atapcConfigIndex

This field is obsolete with ATA Manager 4.0. It was used in previous versions of the ATA Manager when the ATA Manager did much of the PC Card socket configuration. This functionality is now handled by other software.

ataSingleDMASpeed

This field is used to select the timing mode to be used for Single-Word DMA accesses.

ataMultiDMASpeed

This field describes the timing mode used for Multi-Word DMA accesses.

ataPIOCycleTime

This field describes the cycle time to be used for Polled I/O (PIO) accesses.

ataMultiCycleTime

This field describes the cycle time to be used for Multi-Word DMA accesses.

Page 56: *ATA_MgrInquiry*

In the `ATA_MgrInquiry` call (page 56) the fields `ataPIOMaxMode`, `ataSingleDMAModes`, and `ataMultiDMAModes` are obsolete. This functionality has been moved to the `ATA_BusInquiry` function (documented on page 38). This move was necessary because the ATA Manager can support multiple buses, each of which can support different mode sets. For example, PCMCIA ATA cards can only support Polled I/O (PIO) data transfers, whereas internal drives on a PowerBook 3400 can support DMA transfers.

Back to top

Detecting the Presence of the ATA Manager

Some early ROMS could indicate that the ATA Manager exists without proper hardware. This would cause a crash. Therefore, you need to test for the presence of ATA hardware before detecting the presence of the ATA Manager. The following code snippet demonstrates how to detect the ATA Manager properly.

```
// ---------------------------------------------
// returns true if this machine has ATA Manager
// ---------------------------------------------
Boolean ATAManagerPresent(void)
{
    UInt16  configFlags;
    Boolean ATAIsHere = true;

    configFlags = LMGetHWCfgFlags();

    if (!(configFlags & 0x0080))
        ATAIsHere = false;
    // (see Inside Mac VI 3-8 for TrapAvailable)
    if (ATAIsHere && TrapAvailable(kATATrap))
        ATAIsHere = true;
```

```
    return ATAIsHere;
}
```

Back to top

History of the ATA Manager

ATA Manager 4.0 vs. ATA Manager 3.1

ATA Manager 4.0 is PowerPC-Native. ATA Manager 4.0 introduces the concept of the ATA Interface Module or AIM, a plug-in hardware abstraction layer similar to the SIM of SCSI Manager 4.3. The `ATA_DeviceConfig` function has some fields which are obsolete, starting with ATA Manager 4.0.

ATA Manager 3.1 vs. ATA Manager 3.0

The major change from version 3.0 to version 3.1 was the addition of bus-specific transfer timing information to function `ATA_BusInquiry`, rather than general system timing information through `ATA_MgrInquiry`. No new functions were added to the interface.

Altered functions for ATA Manager 3.0 are:

- `ATA_ManagerInquiry`
- `ATA_BusInquiry`

The ATA Manager reports its overall data transfer capabilities via the `ATA_ManagerInquiry` function. For ATA Manager 3.1 and higher, the `ATA_BusInquiry` function further separates and specifies the transfer capabilities of the individual buses.

ATA Manager 3.0 vs. ATA Manager 2.0

The major feature added to ATA Manager 3.0 was the support of DMA I/O operations to the device. No new functions were added to the interface, although several functions now accept or report DMA-specific information.

Expanded functions for ATA Manager 3.0 are:

- `ATA_ManagerInquiry`
- `ATA_SetConfiguration`
- `ATA_GetGonfiguration`

ATA Manager 2.0 vs. ATA Manager 1.0

The following list outlines several major features which were added in ATA Manager 2.0:

- Support ATAPI protocol
- Support for PCMCIA devices
- Support hot plug/removal devices through Card Services
- Dynamic device driver loading and purging
- Client callback messaging/event notification system
- Support for location icon and string

New functions were added and existing functions were expanded to support new features listed above. Consequently, different `ataPBVers` values may result in different responses for a given function. Refer to individual functions for potential differences in response.

New functions for ATA Manager 2.0 are:

- `ATA_EjectDrive`
- `ATA_GetDevConfig`
- `ATA_SetDevConfig`
- `ATA_DriverLoad`

Expanded functions for ATA Manager 2.0 are:

- `ATA_RegAccess`
- `ATA_DrvrRegister`
- `ATA_DrvrDeregister`
- `ATA_FindRefNum`
- `ATA_MgrInit`
- `ATA_MgrShutDown`

In version 1.0 of the ATA manager, the `ataPBActualTxCount` field is not updated for `ExecIO` operations. This is also mentioned in the Power Macintosh 5200/75 and Power Macintosh 6200/75 Computers developer note.

Back to top

ATA Manager 4.0

ATA Manager 4.0 is a redesign of the previous ATA Manager (3.1) undertaken in conjunction with the design of the PowerBook 3400. It is our expectation that every new CPU that supports ATA devices will use this new ATA Manager. The API between ATA Manager 4.0 and its clients (typically ATA and ATAPI disk device drivers) is a superset of ATA Manager 3.1. This allows old disk drivers to function correctly with the new ATA Manager.

The ATA Manager prior to ATA 4.0 had knowledge about specific types of ATA bus controllers. When a new CPU was developed that had a different ATA bus controller, the ATA Manager needed to be revised. The ATA 4.0 design contains a hardware abstraction layer called the *ATA Interface Module* or *AIM*. This AIM is a native driver (`ndrv`), but it is not called using the device manager. Instead, it has the driver option `kDriverIsUnderExpertControl` set, meaning that it is completely controlled by an expert (in this case the ATA Manager). Any new CPU project needs only to create a set of AIMs appropriate for the ATA bus controllers present in that project, and leave the ATA Manager itself untouched.

The Operating System's Name Registry is used to contain all of the hardware-specific information related to the ATA bus controller. The ATA Manager will use the Name Registry to locate and load the AIM, and the AIM will use CPU specific information stored in the Name Registry to do its initialization. For example, the base address of the ATA registers will be calculated by `OpenFirmware` or some other piece of system software and will be used by the AIM to talk to the ATA bus controller.

There is currently no documentation available to external developers who wish to write an AIM. Such developers should contact Developer Support at the Contact Us page. Apple may work together with such developers to help develop appropriate documentation.

While ATA Manager 4.0 is PowerPC-native, it still can be accessed using the 68K trap. If you wish to call the ATA manager on Power PC-equipped machines, you need to supply some Mixed Mode glue to call the `ataManager` trap. The following code will work:

```
#include <MixedMode.h>
#include <ATA.h>

#define RESULT_OFFSET(type) \
    ((sizeof(type) == 1) ? 3 : ((sizeof(type) == 2) ? 1 : 0))
#define TBTrapTableAddress(trapNum) (((trapNum & 0x03FF) << 2) +
0xE00)

pascal SInt16 ataManager(ataPB *pb)
{
    #ifdef applec
        #if sizeof(SInt16) > 4
            #error "Result types larger than 4 bytes are not supported."
        #endif
    #endif
    long    private_result;

    private_result = CallUniversalProc(
        *(UniversalProcPtr*)TBTrapTableAddress(0xAAF1),
        kPascalStackBased
         | RESULT_SIZE(SIZE_CODE(sizeof(SInt16)))
         | STACK_ROUTINE_PARAMETER(1, SIZE_CODE(sizeof(pb))),
        pb);
    return *(((SInt16*)&private_result) + RESULT_OFFSET(SInt16));
}
```

Back to top

Additional Event Documentation

The following events were not well-documented in *ATA Device Software* Guide.

`kATAOnlineEvent` (code 1)

This event notifies clients when an ATA or ATAPI device becomes available for use. The event occurs either when a new device is connected to the bus, or when a previously unavailable device becomes available again (as in system wakeup when power is restored to the device).

If the device has a registered driver, only that driver will be notified of the event; otherwise, each registered default driver will be notified until a driver responds favorably (that is, with a `noErr` response to the event). Note that for newly connected devices a driver loaded from the device is given priority.

Drivers should keep track of whether the device coming online is a newly connected device or one that's currently offline (that is, connected but not unavailable). A device should be considered connected until a `kATARemovedEvent` event for the device occurs.

`kATAOfflineEvent` (code 2)

This event notifies the registered driver of an ATA or ATAPI device that the device is now unavailable for use (offline). The device, however, is still connected to the bus and the offline state is assumed to be temporary. This event will occur at system sleep when power is removed.

Currently, this event is generated only when the ATA Manager receives a PM_SUSPEND event (essentially the same as a Power Manager sleep demand event) from the PC Card Manager. Drivers receiving kATAOfflineEvent events most likely will want to maintain control of the device but deny access to the device to its clients. In addition, the driver should note that the device may need to be reconfigured when it comes online again (a kATAOnlineEvent event will be generated when this happens).

kATARemovedEvent (code 3)

This event notifies the registered driver of an ATA or ATAPI device that the device has been removed. The removal may be either controlled (for example, a software eject command to the ATA Manager) or uncontrolled (for example, a forced removal by the user). Note that the device may have been in either an online or an offline state before the removal. If the state was online before the removal, a kATAOfflineEvent event is not generated, since the removal implies that an offline condition had to occur.

kATAResetEvent (code 4)

This event notifies the registered driver of an ATA or ATAPI device that the device has been reset. The device may need to be reconfigured by the driver before it can be used again. This event was created for use with multiple devices per bus (ATA Master/Slave mode), since reset applies to all devices on the bus and not to a specific device. Apple currently doesn't implement multiple devices per bus with ATA, so this event isn't implemented. However, drivers now support this event to prevent problems later on when the event is implemented.

kATAOfflineRequest (code 5)

This event is obsolete. It was defined for the early stages of the PC Card Manager, which would echo the Power Manager sleep events to its clients. The ATA Manager would in turn echo the request to its clients. This event was like the sleep request event. The current PC Card Manager allows only for an event akin to a sleep demand event, which does not permit rejection by the client.

kATAEjectRequest (code 6)

This event notifies the registered driver of an ATA or ATAPI device that a request has been made to eject the device. If the response to the request is 0, the device will be ejected and a subsequent kATARemovedEvent event will be generated when the ejection is successful. The kATAEjectRequest event serves as a protection mechanism to alert drivers of a pending ejection. Drivers will most likely want to reject the request unless they initiated the request, since ejection will remove the device from the bus.

Note also that the kATAResetEvent, kATAOfflineRequest, and kATAEjectRequest events are not currently implemented in the ATA Manager.

Back to top

Obsolete Resources

The following are some other places where the ATA Manager has been documented. These documents should be considered obsolete. They are replaced by the *ATA Device Software Guide* and this technote:

- Macintosh PowerBook 150
- Macintosh Macintosh PowerBook 190
- Macintosh Macintosh PowerBook 5300
- Macintosh Macintosh PowerBook Duo 2300
- Macintosh Macintosh Quadra 630
- Macintosh Power Macintosh 5200 and 6200
- Macintosh Power Macintosh 5260
- Macintosh Power Macintosh 5400

Q&A DV 24, ATA Manager Events Clarified has some additional ATA events. These events are incorporated in this document. This Q&A, which is incorporated in this document, should now be considered obsolete.

Q&A DV 26, Calling ataManager on a Power Macintosh demonstrates how to call the ATA Manager from PowerPC code. This Q&A, which is incorporated in this document, should now be considered obsolete.

The Q&As of develop issue 26 have some additional documentation of some ATA events. These are the same events discussed in Q&A DV 24, ATA Manager Events Clarified. These events are incorporated in this document. The *develop* Journal Q&As should be considered obsolete.

Back to top

Valid Resources

The ATA Device Software Guide and this document should be considered the current definitive documentation on the ATA Manager. All other documentation that references the ATA Manager should be considered suspect or obsolete.

Technote 1094, Virtual Memory Application Compatibility discusses virtual memory and the ATA Manager. This document

does *not* repeat those discussions. You should read Technote 1094 for a further understanding of the interaction of virtual memory and the ATA Manager.

Technote DV 22, CD-ROM Driver Calls discusses ATA CD-ROM driver calls. These calls remain valid. You should read Technote DV 22 if you are trying to use an ATA CD-ROM, such as those which ship on the PowerBook 1400 or PowerBook 3400.

ATA Demo is sample code demonstrating how to call the ATA Manager. You should look at this code for additional details of how to scan the ATA bus.

Back to top

Summary

ATA Manager 4.0 is a superset of ATA Manager 3.0, documented in the *ATA Device Software Guide*. There are some minor errors in that document. We welcome any additions or corrections to this technote or the *ATA Device Software Guide*; please use the Contact Us page.

Back to top

References

Some of the following documents may be available from http://fission.dt.wdc.com/x3t13/x3t13.html or ftp://ftp.symbios.com/pub/standards/io/

ATA Device Software Guide

AT Attachment Interface for Disk Drives, ANSI X3.221-1994, Approved May 12, 1994.

AT Attachment Interface with Extensions (ATA-2), ANSI ASC X3.279-1996, revision 3, proposed American National Standard 948D.

AT Attachment-3 Interface (ATA-3), ANSI ASC X3.298-199x.

AT Attachment-4 Interface (ATA-4), X3T13 draft.

ATA packet Interface for CD-ROMs, SFF-8020, Revision 1.2, June 13 1994.

Western Digital Enhanced IDE Implementation Guide, by Western Digital Corporation, revision 5.0.

Fast ATA Sourcebook, Quantum Corporation, November 1994.

Enhanced Disk Drive Specification, by Phoenix Technologies Ltd., version 1.1, January 95.

Back to top

Change History

01-July-1997	Originally written as Technote 1098 -- ATA Device Software Guide Additions and Corrections.
18-May-1998	Updated to reflect both corrections to the original guide, as well as some minor additions for ATA Manager 4.0.

Back to top

Downloadables

	Acrobat version of this Note (76K)	Download
	*ATA Device Software Guide*	View

Back to top