# Atola Disk Recycler: Script Command List

The SAS commands will not work on units without SAS support

Rev. 2.0.0

| Command | Description | Example |
|---|---|---|
| Ata | Execute a custom ATA command on the device. | // Perform STANDBY IMMEDIATE command<br>Ata 0xE0 0 0 0 |
| AtaIn | Execute a custom ATA command and transfer data from the hard drive. | // Perform IDENTIFY DEVICE command.<br>AtaIn 0xEC 0 0 1<br>Dump LastResult.Bytes |
| AtaOut | Execute a custom ATA command and transfer data to the hard drive. | bytes = new byte[512]<br>bytes[0] = 0x37<br>// WRITE DMA EXT Command: LBA=1200, write 1 sector<br>AtaOut 0x35 0 1200 1 bytes<br>// Read the same sector and dump contents to verify<br>ReadSectors 1200 1200<br>Print LastResult |
| BadDisk | Stops script execution with Failed status.<br>see also: GoodDisk, SwitchAutoBadDisk | if (LastResult.Error)<br>  BadDisk |
| Capacity | Reads capacity of the device in bytes. | Capacity<br>Print "Disk capacity = " LastResult |
| Checksum | Performs checksum calculation of the entire media or specified range. | Checksum MD5 1000 2000000<br>Print "MD5 Checksum: " LastResult |
| Compare | Compare sectors with the specified pattern. | Compare AA55 0 1000000 |
| Contains | Indicates whether the soughtString occurs within this text. | res = SmartTable<br>Contains res.Text "Head_Flying_Hours"<br>if (LastResult.OK)<br>  Print "Name of SMART attribute 240 is found." |
| Date | Outputs current date and time into the report log. | Date "yyyy-MM-dd HH:mm:ss" |
| DoD7Erase | Performs DoD 5220.22-M 7-pass wiping of the entire media or the interval defined by startLBA and endLBA parameters.<br><br>According to the latest DoD specification:<br>- thr last pass is not random anymore and can be defined by user.<br>- default value is 0.<br>- auto-compare pass addid as enabled by default. Can be disabled via command parameter<br>- first pass character can be specified via parameter. | DoD7Erase 1000 2000<br>if (lastresult.OK)<br>  Print "Sectors 1000-2000 were wiped successfully."<br>else<br>{<br>  Print "An error occurred during DoD erase of the specified range."<br>  Print LastResult<br>} |

| | | |
|---|---|---|
| DoDErase | Performs DoD 5220.22-M 3-pass wiping of the entire media or the interval defined by startLBA and endLBA parameters.<br><br>According to the latest DoD specification:<br>- thr last pass is not random anymore and can be defined by user.<br>- default value is 0.<br>- auto-compare pass addid as enabled by default. Can be disabled via command parameter<br>- first pass character can be specified via parameter. | DoDErase 1000 2000<br>if (lastresult.OK)<br>  Print "Sectors 1000-2000 were wiped successfully."<br>else<br>{<br>  Print "An error occurred during DoD erase of the specified range."<br>  Print LastResult<br>} |
| Dump | Dumps a raw byte array as HEX codes. | ReadDCO<br>Dump LastResult.Bytes |
| GoodDisk<br>see also: BadDisk, Switch | Stops script execution with Success status. | if (LastResult.OK)<br>  GoodDisk |
| Identify | Runs IDENTIFY DEVICE command. | Identify<br>Dump LastResult.Bytes |
| LastLBA | Reads the number of the last LBA of the device. | lastLbaRes = LastLBA<br>prevSector = 0<br>prevSector = lastLbaRes.Number - 1<br>Scan prevSector lastLbaRes.Number |
| LBAErase | Wipes all the device space or for the interval defined by startLBA and endLBA parameters writing LBA numbers in the beggining of every sector. | // Wipes first 200 sectors<br>LBAErase 0 199 |
| Lock | Executes SECURITY SET PASSWORD command with given parameters. See also: Locked, Unlock, SecurityStatus | Lock SecPw123 false true<br>if (LastResult.OK)<br>  Print "Device locked with password SecPw123 with maximum security level." |
| Locked | Reads ID sector to verify if the deviced is locked by password.<br><br>See also: Lock, Unlock, SecurityStatus | res = Locked<br>if (res.OK)<br>  Print "Disk is locked."<br>else<br>  Print "Disk is not locked." |
| Model | Reads model of the hard drive. | m = Model<br>Print m |
| NISTErase | Performs NIST 800-88 wiping of the entire media or the interval defined by startLBA and endLBA parameters. It includes linear overwrite + full verification of written data according to the requirements of the 'National Institute of Standards and Technology: Draft NIST Special Publication 800-88 Revision 1' document. | NISTErase 1000 2000<br>if (lastresult.OK)<br>  Print "Sectors 1000-2000 were wiped successfully."<br>else<br>  Print "An error occurred during NIST erase of the specified range." |
| PatternErase | Performs wiping with specified pattern of the entire media or the interval defined by startLBA and endLBA parameters. | // Wipes sectors 0-200 with HEX pattern 55AA<br>PatternErase 55AA 0 200 |

| | | |
|---|---|---|
| PowerCycle | Powers off and then powers on the hard drive. | Identify<br>PowerCycle<br>Identify |
| PowerOff | Powers off the hard drive. | PowerOff<br>Sleep 1<br>PowerOn |
| PowerOn | Powers on the hard drive. | PowerOff<br>Sleep 1<br>PowerOn |
| Print | Print the value of a variable to the report with a custom prefix. | LastLba<br>Print "Last LBA number = " LastResult.Number |
| RandomNumber | Generates a random integer value bounded by parameter maxValue. Returns generated integer value in LastResult.Number | Random 1000<br>ReadSectors LastResult.Number LastResult.Number |
| ReadDCO | Reads Raw Device Overlay Configuration sector of the device. | ReadDCO<br>Dump LastResult.Bytes |
| ReadMaxAddressDCO | Reads maximum LBA from Device Configuration Overlay. | ReadMaxAddressDCO<br>Print LastResult.Number |
| ReadNativeMaxAddress | Reads native maximum LBA. | ReadNativeMaxAddress<br>Print LastResult.Number |
| ReadSectors | Reads raw sector data from a range of sectors. Max allowed range is 4096 sectors. | ReadSectors 10 1000<br>if (LastResult.OK)<br>  Dump LastResult.Bytes |
| Repair | Performs media recovery. Software bad sectors will be recovered; hardware bad sectors will be reallocated (remapped), provided that the hard drive has spare sectors left on the media. | // Remap all bad sectors in the LBA region 0 - 2 000 000<br>Repair 0 2000000<br>// Another example - simply repair the entire drive<br>// No parameters are needed in this case:<br>Repair |
| Report | Creates custom report based on the specified template file. Parameters are passed to the template like this: {1} - first parameter, {2} - second parameter and so on. | myVar1 = Model<br>myVar2 = "Some text"<br>Report MyTemplate myVar1.Text myVar2 |
| Reset | Performs hard disk reset. | Reset<br>Identify |

| | | |
|---|---|---|
| RestoreDCO | Restores factory Device Overlay Configuration (DCO). | hpaAddress = ReadNativeMaxAddress<br>dcoAddress = ReadMaxAddressDCO<br>if (dcoAddress.Number > hpaAddress.Number)<br>  RestoreDCO |
| Revision | Reads firmware revision of the hard drive. | r = Revision<br>Print r |
| Sas | Execute a custom SCSI command on SAS device | cdb = new byte[6]<br>cdb[0] = 0x1B<br>// START_STOP_UNIT: Spin down<br>Sas cdb<br><br>// Setting up spin-up bit<br>cdb[4] = 1<br>// START_STOP_UNIT: Spin up<br>Sas cdb |
| SasGList | Reads SCSI device GList and checks if result exceeds limit. | |
| SasIn | Execute a custom data-in SCSI command and transfer data from SAS device. | // Issue READ (10) SCSI command for LBA 1<br>read10 = new byte[10]<br>read10[0] = 0x28<br>read10[5] = 0x01<br>read10[8] = 1<br>SasIn read10 512 |
| SasOut | Execute a custom SCSI command and transfer data to SAS device. | // Issue WRITE (10) SCSI command for LBA 1<br>bytes = new byte[512]<br>bytes[1] = 0xDE<br>bytes[2] = 0xAD<br>bytes[3] = 0xBE<br>bytes[4] = 0xEF<br>cdb = new byte[10]<br>cdb[0] = 0x2a<br>cdb[5] = 0x01<br>cdb[8] = 0x01<br>SasOut cdb bytes |
| SasPList | Reads SCSI device primary defect list (P-List). | |
| Scan | Performs media scan of the entire media, or of the specified region. | // Scan the entire media - no parameters are needed:<br>Scan<br>// More sophisticated example:<br>Scan 0 0xF00000<br>if (LastResult.OK)<br>  Print "Media surface up to LBA 0xF00000 is Ok."<br>else<br>  Print LastResult |
| SecurityErase<br>(+ Enhanced Security Erase ) | Performs long-lasting Security Erase command. Wipes the entire media<br>Security Erase = SecurityErase SecurePassword123<br>Enhanced Security Erase = SecurityErase SecurePassword123 true | SecurityErase SecurePassword123 (true)<br>if (lastresult.OK)<br>  Print "Device was wiped succsessfully."<br>else<br>  Print "An error occurred during Security Erase." |

| | | |
|---|---|---|
| SecurityEraseSupported | Reads ID sector to verify if Security Erase or Enhanced Security Erase command is supported by the device.<br>if Security Erase supported = SecurityEraseSupported<br>if Enhanced Security Erase supp. = SecurityEraseSupported true | res = SecurityEraseSupported true<br>if (res.OK)<br>  Print "Enhanced Security Erase command is supported."<br>else<br>  res = SecurityEraseSupported<br>  if (res.OK)<br>  Print "Security Erase command is supported."<br>  else<br>  Print "Security Erase command is not supported." |
| SecurityStatus | Reads ID sector to verify if the device is locked and which level it is locked with.<br>See also: Lock, Locked, Unlock | res = SecurityStatus<br>if (res.Number == 0)<br>  Print "Device is not locked."<br>if (res.Number == 1)<br>  Print "Device is locked with High level."<br>if (res.Number == 2)<br>  Print "Device is locked with Maximum level." |
| Seek | Performs seek test. | // Perform linear seek test within the first 600K sectors;<br>// Timeout = 50 seconds<br>Seek linear 50 0 600000<br>// Perform backward seek test within the last 600K sectors;<br>// Timeout = 50 seconds. We will need to perform some math to<br>// calculate the range we need to test:<br>lastSector = LastLBA<br>firstSector = 0<br>firstSector = lastSector.Number - 600000<br>Seek backward 50 firstSector lastSector.Number |
| Serial | Returns LastResult.Text if command executes successfully. | s = Serial<br>Print s |
| SetCustomField | Sets a value to current custom field. Can be successfully used only from custom field Script Code accessed via Settings->Data. | LastLba<br>SetCustomField LastResult |
| SetExtendedResult | Define Extended result. This is useful when you need to deviate from standard Success/Failed result and want to add a specific comment about the hard drive, for example, to define its Grade. Extended result will be saved into the data-base and shown on the report.<br>Supports color codes (ex. #AAFFAA) and influences port color | Scan<br>if (LastResult.BlockCount > 0) {<br>  SetExtendedResult "Bad sectors found"<br>} |
| SetMaxAddress | Limits the hard drive's capacity to the specified Maximum LBA (Set Max Address) | // Caps disk size to 10000 sectors (5 Mbytes) by means of HPA<br>// Since the very first sector on the drive is 0 and not 1,<br>// we need to set Max LBA to 9999 and not to 10000.<br>SetMaxAddress 9999<br>PowerOff<br>PowerOn<br>LastLba<br>Print "LastLBA = " LastResult |
| SetReadTimeout | This timeout influences all read operations. Sets a timeout value used during any disk read operation. Default is 20 seconds.<br>Returns set timeout value in LastResult.Number. | // Set timeout equal to 2 seconds<br>SetReadTimout 2000 |

| | | |
|---|---|---|
| SetWriteTimeout | This timeout influences all write operations. Sets a timeout value used during any disk write operation. Default is 20 seconds. Returns set timeout value in LastResult.Number. | // Set timeout equal to 3 seconds<br>SetWriteTimout 3000 |
| Sleep | Pause the script for specified number of seconds. | Sleep 10<br>Print "10 seconds elapsed." |
| SmartAttributeRaw | Reads raw SMART value by attribute ID. | res = SmartAttributeRaw 1<br>Print "Raw read error rate value is " res |
| SmartAttributeThreshold | Reads threshold SMART value by attribute ID. | res = SmartAttributeThreshold 1<br>Print "Raw read error rate threshold value is " res |
| SmartAttributeValue | Reads normalized SMART value by attribute ID. | res = SmartAttributeValue 1<br>Print "Raw read error rate normalized value is " res |
| SmartAttributeWorst | Reads worst SMART value by attribute ID. | res = SmartAttributeWorst 1<br>Print "Raw read error rate worst value is " res |
| SmartErrors | Returns SMART error log page as text. | res = SmartErrors<br>Print res |
| SmartLog | Shows SMART log page information depending on specified logType parameter. | SmartLog selftest |
| SmartStatus | Reads hard drive's SMART status | SmartStatus<br>Print LastResult |
| SmartTable | Reads the entire SMART attribute table. | res = SmartTable<br>Print res.Text |
| SmartTest | Performs a SMART test. | res = SmartTest short<br>Print res.Text |
| SpinDown | Spins down the hard drive bringing it into standby mode. | SpinDown<br>SpinUp |
| SpinUp | Spins up the hard drive from the standby mode. | SpinDown<br>SpinUp |
| SwitchAutoBadDisk | Toggles automatic script termination on or off.<br>See also: BadDisk, GoodDisk | SwitchAutoBadDisk true<br>Print "From this point any disk error will terminate the script with BadDisk status" |

| | | |
|---|---|---|
| SwitchAutolog | Toggles automatic logging on or off. | SwitchAutolog true<br>Print "Now all commands will automatically print their Result to the report." |
| SwitchReadCache | Enables/disables HDD read look ahead function.<br>Turns hard drive´s write cache on or off.<br>Returns LastResult.OK if command executes successfully. | res = SwitchWriteCache true<br>if (res.OK)<br>Print "Cached writes are now enabled." |
| SwitchSmart | Turns SMART on or off. | res = SwitchSmart false<br>if (res.OK)<br>  Print "Smart has been disabled." |
| SwitchWriteCache | Turns hard drive's write cache on or off. | res = SwitchWriteCache true<br>if (res.OK)<br>  Print "Cached writes are now enabled." |
| Time | Returns the time in milliseconds since the moment the current script has started. | t = Time<br>Print t |
| TransferRate | Performs transfer rate test. The benchmark is performed for the inner, middle and outer tracks. For more precise results allow 30-60 seconds for testing. 20 seconds is considered sufficient for most applications. | // Measure transfer rate for 20 seconds<br>TransferRate 20<br>Print LastResult |
| Trim (for SSD only) | This command is to be expected in a coming update. It can be used to invalidate all data or the data in a specified range of an SSD. | |
| UnclipHpaDco | Removes max address restrictions from Host Protected Area (HPA) and Device Overlay Configuration (DCO) if supported by HPP. | UnclipHpaDco<br>if (LastResult.Error)<br>  Print "Max address restriction removal error" |
| Unlock | Permanently unlocks device with given parameters.<br>See also: Lock, Locked, SecurityStatus | Unlock SecPw123 false true<br>if (LastResult.OK)<br>  Print "Device was permanently unlocked with password SecPw123." |
| VerifySectors | Verifies a sector range. Max allowed range is 4096 sectors. To verify the entire hard drive, use Scan command instead. | VerifySectors 0 1000<br>if (LastResult.Error)<br>  Print LastResult<br>else<br>  Print "Media surface up to LBA 1000 is Ok." |
| WaitReadyTime | Returns the time in milliseconds spent waiting for the hard drive to become ready after power on. | res = WaitReadyTime<br>Print res |

| | | |
|---|---|---|
| WriteDCO | Write Raw Device Configuration Overlay (DCO). | dco = ReadDCO<br>bytes = new byte[512]<br>bytes = dco.Bytes<br><br>// Decrease one of DCO Max LBA bytes<br>bytes[6]--<br>// Write change to DCO<br>WriteDCO bytes |
| WriteRandom | Writes random data to interval defined by startLBA and endLBA parameters. | WriteRandom 0 1000<br>if (LastResult.OK)<br>  Print "Sectors 0-1000 were filled with random data." |
| WriteSectors | Fills a specified sector range with the predefined data (pattern). Max allowed range is 4096 sectors. To fill/wipe entire drive, use PatternErase command instead. | data = new byte[512]<br>data[100] = 0xFF<br>WriteSectors 0 1000 data<br>if (LastResult.OK)<br>Print "Data was written. |
| WriteTest | Performs write test making 1-sector writes, supports random mode<br>Returns error sector intervals in LastResult.Blocks. | // Perform random write test within the first 600K sectors;<br>// Timeout = 50 seconds<br>WriteTest random 50<br><br>// Perform backward write test within the last 600K sectors<br>// Timeout = 50 seconds. We will need to perform some math to<br>// calculate the range we need to test:<br>lastSector = LastLBA<br>firstSector = 0<br>firstSector = lastSector.Number - 600000<br>WriteTest backward 50 firstSector lastSector.Number |
| WriteTransferRate | Runs performance test executing write operations.<br>A transfer rate test executing write operations. The benchmark is performed for the inner, middle and outer tracks. For a more precise result allow 30-60 seconds for testing. 20 seconds is considered sufficient for most applications  Returns disk write average speed in LastResult.Text. | // Measure write transfer rate for 40 seconds<br>WriteTransferRate 40<br>Print LastResult |