

BAM functions in DTB

There are six new BAM-related functions in DTB version 8.1. They are:

```
int VCSCSIBAMconfigure(long BufSize, long PhaseSize, int Flags, int Phases);
```

```
int VCSCSIBAMclearBuffer();
```

```
int VCSCSIBAMdrive(int ha, int target, int lun, int capture);
```

```
int VCSCSIBAMstartCapture();
```

```
int VCSCSIBAMstopCapture();
```

```
int VCSCSIBAMsaveCapture(CString *fname);
```

General Use of the BAM function

The order of operation to control the capture of a BAM trace in a DTB-enabled application is as follows:

1. Configure BAM
 - this step will define which phases will be captured, what size the capture buffer should be, how much data in a data phase will be recorded, and any operational option flags such as "Stop on Buffer Full"

Use the function `int VCSCSIBAMconfigure(long BufSize, long PhaseSize, int Flags, int Phases);`

with the following parameters:

BufSize is the capture buffer size in MB.

PhaseSize is the amount of data to record, in bytes

Flags can be any combination of or'd values of:

1 = Stop Capture on Buffer Full

2 = Stop Capture on Bus Reset

© Copyright 2009 SCSI Toolbox LLC

Sales: 720.249.2641

General: 303.972.2072

4 = Immediately clear the capture buffer

Phases can be any combination of or'd values of:

```
#define CDB_PHASE      0x00000001 // cdb bytes
#define SENSE_PHASE    0x00000002 // sense data
#define OK_PHASE       0x00000004 // command complete
#define DATA_IN_PHASE 0x00000008 // data in bytes
#define DATA_OUT_PHASE 0x00000010 // data out bytes
#define ATA_PHASE      0x00000020 // ATA task file command
#define ATA_STATUS_PHASE 0x00000040 // ATA task file response
#define ATP_PHASE      0x00000080 // ATA pass through structure
#define SPT_PHASE      0x00000200 // SCSI Pass through struct
#define SRB_PHASE      0x00000400 // SRB structure
#define SRB_STATUS_PHASE 0x00000800 // srb status byte
#define NT_STATUS_PHASE 0x00004000 // nt status
```

Passing a value of 0 for the phases parameter will select the default phases to capture, which are:

```
Default Phases = SENSE_PHASE | OK_PHASE | CDB_PHASE | DATA_IN_PHASE |
DATA_OUT_PHASE | ATA_PHASE |ATA_STATUS_PHASE | SRB_PHASE |
SRB_STATUS_PHASE | RESET_PHASE;
```

2. Define what drive I/O to capture

Use the function `int VCSCSIBAMdrive(int ha, int target, int lun, int capture);`

To define what drive to capture I/O from, and the capture parameter defines whether to set the trace to capture or not capture I/O once the trace capture is started.

Set capture = 1 to turn on capture for this device.

3. Start the capture

Note: You may want to clear the capture buffer before starting a new capture, using the function `int VCSCSIBAMclearBuffer();`

Use the function `int VCSCSIBAMstartCapture();`

To begin capturing

4. Stop the capture

Use the function `int VCSCSIBAMstopCapture();`

To stop capturing.

5. Save the trace

Use the function `int VCSCSIBAMsaveCapture(CString *fname);`

To save the captured trace to your specified file name.

Note: Trace data will be stored in the new BAM version 8.1 data format. You will not be able to read it using BAM version 2 or below.

Some Specific Items

1. The DTB BAM functions allow I/O to/from a specified drive to be captured and saved.
2. At present there are no DTB BAM functions that analyze BAM data
3. Captured data is viewed by using BAM version 8.1, using the File->Open menu.
4. Once you have the capture file open in BAM you can analyze it, save it out as a CSV text file, etc.

© Copyright 2009 SCSI Toolbox LLC

Sales: 720.249.2641

General: 303.972.2072