



Mode/Diagnostic Page Application Note

The SANBlaze Virtualun product features a powerful method to create user defined mode and diagnostic pages for use with emulated targets. This document provides some examples and tips on how to create custom pages for your own use.

How it works: User definable mode and diagnostic pages are created as follows:

A text file must be created that contains the page contents. The file must be saved in a file location that describes the Port/Target/LUN that is affected, as well as extension for Sub Mode and page control, if required, as described below.

Once these pages are created and stored in the file system, a Mode Sense or Select can be used to access the page data.

NOTE: In the command line interface, alias is used instead of target. For example, a LUN described as Port0alias1lun2 references Port 0, Target 1, LUN 2

Mode Pages:

Defining the proper Page File name and directory location:

Mode pages are stored as text files on the Virtualun system. They need to be stored in the following directory:

/virtualun/modepages/port<X>/ (where X is the port number being accessed)

First, determine what alias/lun the page is for, the page number, whether you want to access a sub mode page or just a normal page, and if the page is being accessed using the Page Control bit in ModeSense(6) cdb.

For example, to use alias5 lun2 on port 2 and access page 0x20, the mode page data file would be created as follows:

/virtualun/modepages/port2/ alias5lun2.page20

To access submode page 4 for this page, the file name and location would look like:

/virtualun/modepages/port2/alias5lun2.page20sp4

If the ModeSense(6) cdb being sent to the emulated drive has the page control bit set to 1 or 2, a “pcX” must be appended to the end of the filename. This allows for different pages to be returned by the emulated drive depending on the page control passed in.

For example, if page20 has the page control bit set to 1, the file name would be:

/virtualun/modepages/port2/alias5lun2.page20pc1

If the submode page example was using page control 2, the file name would be:

/virtualun/modepages/port2/alias5lun2.page20sp4pc2

If a page control of 0 or 3 is being used, nothing needs to be appended to the filename.

Adding Mode Page Data:

Once the filename has been properly formatted, the mode page text file can be created with the appropriate data.

Mode Pages and SubMode Pages must be created using the Page_0 format and sub_mode_page format as defined in the SCSI-3 spec. To create page 0x20 with the following contents in it: 0x1,0x2,0x3,0x4,0x5,06, the contents of the mode page text file would look like this:

20 06 01 02 03 04 05 06

The first byte correlates to the page number (20), the second byte is the length of the mode parameters (6 in our case), and the rest is the data. For more data, the mode parameter length byte must be adjusted accordingly.

To create page 0x20, submode page 0x4 with the same data as above, the page file would look like:

60 04 00 06 01 02 03 04 05 06

In this case, bit 6 in byte 0 is set to indicate the use of the sub_page format (60). The second byte indicates what subpage is defined (04), the next two bytes indicate the length of the data (0006), and the rest is the page data.

Once the pages have been created properly, the emulated lun will return or modify the user-defined page whenever it is requested via ModeSense(6) or ModeSelect(6).

Diagnostic Pages:

User defined diagnostic pages are stored/created in a similar manner to the mode pages. They are stored on the Virtualun system in the directory:

/virtualun/diagpages/portX/ (where X is the port being accessed)

To create page 0xC for alias5 lun2 on port 1, create a file with the following file name:

/virtualun/diagpages/port1/alias5lun2.pagec

Once the proper filename has been created, fill in the page with the desired contents. Pages must be created using the Diagnostic Page format as defined in the SCSI-3 spec.

For example, to create page 0xC with data 00,11,22,33,44,55,66,77,88,99, the page contents would look like:

```
0c 00 00 0a 00 11 22 33 44 55 66 77 88 99
```

In this case, byte 0 contains the page code (0xc), byte 3 contains the page size (10 bytes), and the rest is the page data.

After the page is saved, it will be returned by the lun whenever it is accessed using a Receive Diagnostic command.

The lun can also handle Send Diagnostic commands for a particular page. If it receives a Send Diagnostic command with control data, the lun will simply take the data and store it in a file on the system. This can be useful for scripting, etc.

For example, if alias5 lun2 on port 1 receives a Send Diagnostic command for page 0xc with control data (00,11,22,33,44) a file will be created on the Virtualun system in the diagpages directory with the following file name:

/virtualun/diagpages/port1/alias5lun2.pagec.ctrl

Viewing the contents of this file would result in the following data:

```
# cat /virtualun/diagpages/port1/alias5lun2.pagec.ctrl
```

```
0c 00 00 05 00 11 22 33 44
```

For this example, the data received in Send Diagnostic command was dumped into the appropriate file.

As an example, a script running on the Virtualun system could then potentially look for changes in the ctrl page file and take some action (e.g. Change contents of diagnostic page, etc).