Functional description of the TOPS-20 MSCP Server, PHYMVR

1-Sep-83 Revised: 24-NOV-84

Version 1, Revision 2

1.0 ARCHITECTURAL POSITION

PHYMVR is the TOPS-20 MSCP server module. It is responsible for granting access to massbus disks across the CI-20. The only supported user of the MSCP server is the TOPS-20 MSCP driver (PHYMSC). The MSCP server implements only the subset of MSCP functions required for the operation of PHYMSC.

The server appears to the MSCP driver to be an intelligent disk controller much like the HSC50. The MSCP server interfaces to TOPS-20 as a device independant disk driver a la the DSKOP Jsys. It functions at a higher architectural level than PHYSIO but at a lower level than PAGEM/PAGUTL. That is, the MSCP server has no knowledge of the file system format.

Because of the desirability of restricting the set of disks which can be accessed through the MSCP server, a small JSYS level interface has been provided as a means of limiting the set of disks available through the MSCP server. The SETSPD program has been expanded to include 2 new commands, ALLOW and RESTRICT, to provide the system administrator with a means to select the disk drives which are handled by the MSCP server.

The server is a SCA SYSAP. It communicates with the MSCP driver by sending messages and data across the CI using the facilities provided by the TOPS-20 SCA interface described in [3]. The server is a generally passive device, responding to requests issued by the MSCP driver.

The MSCP server is needed only when the TOPS-20 Common File System is in use. It is loaded under a conditional, CFSCOD, which also controls the use of CFS.

Figure 1 - Software Components

2.0 RELATED STANDARDS AND SPECIFICATIONS

This document describes, at a functional level, those aspects of the MSCP server not specified in other documents. In order for you to understand this functional specification, you must be familiar with at least the terminology, if not the details, of the following:

1. Mass Storage Control Protocol Version 1.2, 8-Apr-82 (Gardner)
2. TOPS-20 MSCP Driver Design Specification 20-Sep-83 (Mclean)
3. TOPS-20 SCA Functional Specification 21-Nov-83 (Dunn)
4. Systems Communications Architecture, 20-July-82 (Strecker)
5. TOPS-20 Coding Standard, 23-Mar-83 (Murphy)
6. PHYSIO.MAC TOPS-20 Physical IO module

3.0 GENERAL LIMITATIONS AND RESTRICTIONS

The MSCP server will only communicate with 4 MSCP drivers. This can easily be changed to allow any number of drivers.

Only one MSCP server can run at a time. The changes required to allow several MSCP servers are fairly minor. There is probably no advantage to running more than 1 MSCP server since the code is mainly interrupt driven and very compact.

4.0 SETSPD / SMON% / TMON% INTERFACE FOR SYSTEM ADMINISTRATORS

The SMON% Jsys provides a method for a given site to limit the set of disks available through the MSCP server. This interface is available to a site manager through the use of the RESTRICT and ALLOW commands of SETSPD.

When a system is booted all disks are treated as CI unavailable (RESTRICTed). To set them available (ALLOWed) the following must be done:

```
MOVEI AC1,.SFMSD
MOVEI AC2,<address of argument block>
SMON%
```

where <argument block> has the following format

OFFSET	VALUE	DESCRIPTION
.SVCNT	0	length of the block including this word
.SVTYP	1	flags and drive type
.SVDSH	2	High order serial number word
.SVDSN	3	Low order serial number word

NOTE: Currently the high order serial number word for massbus disks is computed by the monitor. The formula is:

```
        MOVEI AC1,<drive type>
        IORI AC1,400
        LSH AC1,20
```

Drive type symbol may be one of the following:

```
    .MSRP4==:1                  ;RP04
    .MSRP5==:5                  ;RP05
    .MSRP6==:6                  ;RP06
    .MSRP7==:7                  ;RP07
    .MSR20==:24                 ;RP20
```

or use the SETSPD command

ALLOW <drive type> <serial number (low)>

or the alternate form

ALLOW <drive type> <serial number (high)> <serial number (low)>

Drive type name may be one of the following:

```
    RP04
    RP05
    RP06
    RP07
    RP20
```

Examples:

```
          ALLOW RP06 1234
          ALLOW RP20 3327
```

RETURNS: +1 ALWAYS disk drive available

Illegal instruction trap on the following conditions

MSCPX1 - No MSCP server in current monitor
MSCPX2 - Drive type error
MSCPX3 - Requested drive not found
SMONX1 - WHEEL or OPERATOR capability required

To set a disk drive unavailable use the following instruction sequence:

```
        MOVEI AC1,.SFMSD
        MOVEI AC2,<address of argument block>
        SMON%
```

where <argument block> has the following format

OFFSET	VALUE	DESCRIPTION
.SVCNT	0	length of the block including this word
.SVTYP	1	flags and drive type. To restrict specify MS%DDU
.SVDSH	2	High order serial number word
.SVDSN	3	Low order serial number word

or use the SETSPD command

RESTRICT <drive type> <serial number (low)>

or the alternate form

RESTRICT <drive type> <serial number (high)> <serial number (low)>

Examples:

```
RESTRICT RP06 17170432 1234
RESTRICT RP20 3327
```

RETURNS: +1 ALWAYS disk drive restricted

Illegal instruction trap on the following conditions

MSCPX1 - No MSCP server in current monitor
MSCPX2 - Drive type error
MSCPX3 - Requested drive not found
SMONX1 - WHEEL or OPERATOR capability required

There is a corresponding TMON% interface for reading the status of disk drives it is:

```
MOVEI AC1,.SFMSD
MOVEI AC2,<address of argument block>
TMON%
```

where <argument block> has the following format

OFFSET	VALUE	DESCRIPTION
.SVCNT	0	length of the block including this word
.SVTYP	1	flags and drive type
.SVDSH	2	High order serial number word
.SVDSN	3	Low order serial number word

RETURNS: +1 ALWAYS T2/ 0 drive is available through the MSCP server
T2/ 1 drive is not available

Illegal instruction trap on the following conditions

MSCPX1 - No MSCP server in current monitor
MSCPX2 - Drive type error
MSCPX3 - Requested drive not found

There is no limitation on the number of disks which may be set "CI available" since the server is designed to handle an infinite number of disks. The MSCP driver may have a maximum number of disks that it can handle but the server cannot determine this limit. Since a site can potentially rebuild their monitor to handle more or less disks than the default, the server does not return an error nor make any checks regarding the number of disks serviced. It is up to the system administrator to insure that the number of disks serviced does not exceed the number of disks accessable by the MSCP drivers on the CI. At the present time the TOPS-20 MSCP driver will support up to 24 disks.

5.0 MSCP DRIVER INTERFACE

The complete interaction rules for an MSCP server and an MSCP driver are described in [1]. The TOPS-20 MSCP server provides access only to disk drives and does not provide any tape device service. The functions implemented by the TOPS-20 MSCP server are described in section 6 of [1] entitled "Minimal Disk MSCP subset". Therefore the following section describes only limitations, differences and restrictions of the TOPS-20 MSCP server implementation.

5.1 Unit Flags

The following restrictions apply to the unit flags described in section 6.1 of [1].

1. Compare reads and Compare writes - These flags are unsupported under TOPS-20. These flags are not used by the MSCP driver and would require space in the UDB to implement.

2. Controller initiated Bat Block replacement - This flag is always returned clear since the MSCP server does not do BAT block replacement. The server expects that the driver will do all management of BAT blocks.

3. Inactive shadow set unit - The MSCP server does not support shadowing and does not examine this flag. It is always returned clear.

4. Cache flags - The MSCP server does not support cacheing and does not examine these flags. They are always returned clear.

5. Write protect (software) - This flag is unsupported under TOPS-20 as it is unused by the MSCP driver, and would require space in the UDB to implement.

6. 576 byte sectors - This flag is always returned set, as all TOPS-20 disks have logical sectors of 576 (32 bit) bytes.

All other unit flags are fully supported as described.

5.2 Controller Flags

The following restrictions apply to the controller flags described in section 6.2 of [1].

1. Error log flags - These flags are fully supported in the sense that they are allowed to be set, but the MSCP server does not send error log datagrams.

2. Controller initiated bat block replacement - This flag is always returned clear since the MSCP server does not handle BAT blocks.

3. Shadowing - This flag is always returned clear since the MSCP server does not support shadow units.

4. 576 byte sectors - This flag is always returned set as all TOPS-20 formatted disks have logical sectors of 576 (32 bit) bytes.

All other controller flags are fully supported as described.

5.3 Command Modifiers

The following restrictions apply to the command modifiers described in section 5.4 of [1].

1. Clear Serious Exception - This modifier is ignored.

2. Compare - This modifier is partially supported. If this modifier is specified on a READ or WRITE command then the IORB function "read validity" or "write validity" is used instead of "read" or "write". However the data is not reobtained from the source and compared with the destination data.

3. Express Request - This modifier is ignored.

4. Force Error - This modifier is ignored.

5. Suppress Error Correction - This modifier is ignored.

6. Suppress Error Recovery - This modifier is ignored.

Modifiers which affect caching and shadowing are ignored, as the server does not support these features. All other command modifiers are fully supported as described.

5.4 Services Provided To The MSCP Driver

5.4.1 Supported Functions -

All supported functions contain a section labelled "Description" which is copied verbatim from [1]. This description is provided as an aid to the reader and is NOT intended to provide a full description of the functionality and interactions of the described command. A complete understanding of these functions and interactions may be found in [1]. Please remember that all references in the "Description" section of the supported commands are references to [1] and not to this document.

1. ABORT command

   Description:

   The ABORT command causes a specified command to be aborted a the earliest time convenient for the controller. Th specified command must, however, either be aborted or b completed within the controller timeout interval. Se Section 4.10, "Command Timeouts", for a discussion of th interaction between ABORT and command timeouts.

   The ABORT command always succeeds. If the command to b aborted is not known to the controller, this implies that i has already completed and the ABORT command will be ignored The controller always returns the "Normal" status code in th ABORT command's end message.

   The controller may ignore the ABORT command if the comman being aborted will always complete within the controlle timeout interval.

   The class driver must wait for the aborted command's en message, or else re-synchronize with the MSCP server, befor reusing the aborted command's command reference number o releasing the aborted command's context. If the command wa aborted, its end message will contain the "Command Aborted status code. Otherwise, the command was completed. Th class driver may ignore the ABORT command's end message Note that the class driver may receive the ABORT command' end message either before or after the aborted command's en message.

   The ABORT command functions exactly as described.

2. AVAILABLE attention message

   Description:

   An MSCP server sends an AVAILABLE attention message to "Controller-Online" class driver when a unit asynchronousl becomes "Unit-Available" to that class driver, unles AVAILABLE attention messages have been suppressed for tha unit by an AVAILABLE command with the "Spin-down" modifier o

by an error with similar side effects. Changes to the "Unit-Available" state due to the class driver itself issuing an AVAILABLE command are synchronous. All other changes to "Unit-Available" are asynchronous. See Section 4.3, "Unit States".

The actual sending of an AVAILABLE attention message may be delayed for an arbitrarily long time, due to communications mechanism flow control, from the time that the unit actually becomes "Unit-Available". The message must not be sent if the class driver ceases to be "Controller-Online" during this delay. The message must be sent anyway if the unit, or any unit with which it shares an access path, becomes "Unit-Online" via another controller during this delay. The message may or may not be sent, at the controller's option, if the unit ceases to be "Unit-Available" for any other reason during this delay.

Note that, due to these delays, it is possible for an AVAILABLE attention message to be received after the class driver has already brought the unit "Unit-Online". Therefore class drivers must not use AVAILABLE attention messages to flag "Unit-Online" units as having become "Unit-Available". The proper procedure is to issue a command, such as a GET UNIT STATUS, to a "Unit-Online" unit for which an AVAILABLE attention message has been received, and only flag the unit as having become "Unit-Available" if the command returns that status code.

AVAILABLE attention messages are not sent for units that are already "Unit-Available" when a class driver enables attention messages. Class drivers that need to be aware of all "Unit-Available" units must enable attention messages, then scan all units via the GET UNIT STATUS command with the "Next Unit" modifier set to locate all units that are already "Unit-Available". All units that subsequently become "Unit-Available" will be reported with an AVAILABLE attention message.

An MSCP server may send redundant or erroneous AVAILABLE attention messages at any time. The frequency of such messages must be low enough that they do not represent a significant overhead for either hosts or the communications mechanism. The information contained in such messages (unit number, unit identifier, media type identifier, etc.) must correspond to an actual, physical unit that is potentially accessible via that MSCP server (i.e., connected to the controller), although the unit need not be "Unit-Available". Note that hosts must be able to handle seemingly erroneous AVAILABLE attention messages in any case, since the unit's state may change before the host can act on an otherwise correct message.

The AVAILABLE attention message is sent when a disk drive is set "CI-available" through the SMON% monitor call if the disk is

locally online. Otherwise, it is sent when a disk which has been set "CI available" comes online locally. This message is not sent to a driver if it has not enabled attention messages or if the unit is already "Unit online" to a driver. It is possible for the driver to recieve multiple attention messages for single unit until that unit is ONLINEd by the driver.

3. AVAILABLE Command
Description:

All outstanding commands for the specified unit ar completed, then the unit becomes "Unit-Available". If th "Spin-down" modifier was not specified, the unit is no already "Unit-Available", and no other units that share thi unit's access path are "Unit-Online" (i.e., the "Stil Connected" status sub-code bit flag is clear), then a AVAILABLE attention message is sent by any other controlle to which the unit is connected. The controller to which thi command was sent need not itself send an AVAILABLE attentio message.

If the "Spin-down" modifier is specified, the disk spins dow and its heads are unloaded, unless some other unit with whic this unit shares a spindle is "Unit-Online". The disk may b spun up with an ONLINE command or by operator intervention The "Spin-down" modifier also suppresses AVAILABLE attentio messages for this unit, both for this controller and an other controllers to which the unit may be connected. Se Section 4.3, "Unit States", for a discussion of suppressin AVAILABLE attention messages.

This command will be accepted if the unit is "Unit-Online" o "Unit-Available". It is nugatory to issue this command to unit that is "Unit-Available" unless the "Spin-down" modifie is specified. Assuming no other errors occur, the "Success status code will be returned regardless of whether the uni was previously "Unit-Online" or "Unit-Available".

If the unit was "Unit-Online" but had a duplicate unit numbe prior to the AVAILABLE command being issued, the AVAILABL command may complete, at the controller's option, with eithe a "Success" or a "Unit-Offline" status code. Th "Unit-Offline" status code must have the "Duplicate Uni Number" sub-code flag set. The "Success" status code may o may not, at the controller's option, have the "Duplicate Uni Number" sub-code flag set. Subsequent attempts to access th unit will return "Unit-Offline" status with the "Duplicat Unit Number" sub-code flag set unless the duplicate uni number has been eliminated.

The AVAILABLE command makes the unit "unit available" to the driver which issued the command. If the "spin-down" modifier is specified the sub-code "Spin-down ignored" is returned. It is currently not possible to initiate an unload of an massbus disk at a software level. The sub-code "Still connected" is never

returned.

4. GET COMMAND STATUS Command

Description:

The GET COMMAND STATUS command is used to monitor the progress of a command towards completion. The command status measures the "doneness" of the command. The value returned in the command status field is guaranteed to not increase over time. Furthermore, the command status of an MSCP server's oldest outstanding command is guaranteed to decrease within the controller timeout interval. This last feature may be used by a host class driver to detect an insane or malfunctioning controller. See Section 4.10, "Command Timeouts", for more details.

The GET COMMAND STATUS command always succeeds. If the command referenced by the "outstanding reference number" is not known to the MSCP server or has been aborted, then the MSCP server should return zero for its command status. The MSCP server may also return zero as the command status of any command that will always complete within the controller timeout interval. The MSCP server always returns the "Normal" status code in the GET COMMAND STATUS command's end message.

The GET COMMAND STATUS command functions exactly as described. The command status returned is a function of the command's timeout value. It is guaranteed to decrease so long as the timeout counter is incremented at shorter intervals than the controller timeout period.

5. GET UNIT STATUS Command

Description:

The GET UNIT STATUS command returns the current state of a unit plus certain unit characteristics. In particular, the GET UNIT STATUS command is used to obtain host settable characteristics and those fixed unit characteristics that are not normally needed by the class driver.

Class drivers can determine which of the returned unit characteristics are valid by examining the returned "status" and "unit identifier" fields. The following cases exist:

1. "status" is "Success", implying that the unit is "Unit-Online". All characteristics are valid.

2. "status" is anything other than "Success", and "unit identifier" is not zero. All unit flags except for the "Removable media" flag are undefined. All other characteristics are valid.

3. "status" is anything other than "Success", and "uni identifier" is zero. Only the "shadow unit" and "shado status" characteristics are valid. All othe characteristics are undefined.

The three cases listed above are the only cases that ca occur. Note that if "status" is "Success" then "uni identifier" cannot be zero.

Rather than testing the entire quadword unit identifier, i is sufficient to merely test the high order word of the uni identifier, containing the class and model code bytes, to se if it is zero or not.

Controllers must supply valid values for all characteristic whenever the unit is "Unit-Online". Controllers must suppl a non-zero unit identifier and valid values for al characteristics except those noted above whenever the unit i "Unit-Available" or the unit is "Unit-Offline" solely due t being disabled or known. Controllers may or may not, at th controller's option, provide valid characteristics when th unit is "Unit-Offline" for any other reason or any othe status code is returned.

The rules in the above paragraphs can be restated as follows

1. If "status" is "Success", then "unit identifier" must b non-zero and all characteristics must be valid.

2. If "status" is "Unit-Available", then "unit identifier must be non-zero and almost all characteristics must b valid.

3. If "status" is "Unit-Offline" and the sole causes of th unit being offline are it being disabled or known, the "unit identifier" must be non-zero and almost al characteristics must be valid.

4. If "unit identifier" is zero, then "status" must eithe be "Unit-Offline" with some reason other than the th unit being disabled or known indicated, or "status" mus be "Controller Error" or "Drive Error". Virtually n characteristics need be valid.

The GET UNIT STATUS command functions exactly as described.

6. ONLINE Command

Description:

The ONLINE command is used to bring a unit "Unit-Online", se host settable unit characteristics, and obtain those uni characteristics that are essential for proper class drive operation. The unit is spun-up, if necessary, and its head are loaded prior to returning the ONLINE command's en

message. Host settable characteristics are set exactly as if a SET UNIT CHARACTERISTICS command were issued. See the description of that command. Host settable characteristics are set after the unit has been successfully spun-up and any other validity checks have succeeded. Note that the unit's host settable characteristics are NOT altered if the unit is already "Unit-Online".

The class driver must check if the "Controller Initiated Bad Block Replacement" unit flag is set or clear after bringing a unit "Unit-Online". If it is clear (implying that the host is performing bad block replacement), then the class driver must invoke a process that will access the unit's Replacement and Caching Table to determine if a bad block replacement operation has been partially performed or if the unit must be write protected. The details of this check and its consequences are described in Section 4.12, "Bad Block Replacement", and DEC Standard Disk Format. Controllers that support Controller Initiated Bad Block Replacement perform these checks themselves.

Note that the format of the ONLINE command's end message is identical to the SET UNIT CHARACTERISTICS command's end message.

The ONLINE command functions exactly as described. It is never necessary for a unit to be spun up. It is a requirement that the disk be online at the local system to become "unit available" to the MSCP driver. Massbus disks that are not spinning cannot be online at the local system. For a complete description of MSCP unit states see the section entitled "MSCP unit states" under "Algorithms".

7. READ Command

Description:

Data is read from the unit and transferred to the host buffer.

The read command functions as described. If the "compare" modifier is used then data is read using the IORB function "read validity".

8. SET CONTROLLER CHARICTERISTICS Command

Description:

The SET CONTROLLER CHARACTERISTICS command is used to set and obtain controller characteristics. The default value for "cntrlr. flags" is all flags clear (i.e., all messages disabled). The default value for "host timeout" is 60 seconds. These default values are used from the time that the controller becomes "Controller-Online" to a host until it stops being "Controller-Online" or until the host issues a

The SET CONTROLLER CHARACTERISTICS command functions exactly as described. The controller identifier is composed of the CPU serial number in the first word and a magic number in the second word. This magic number is required because there is no assigned identifier number for a TOPS-20 MSCP server.

9. WRITE Command

Description:

Data is fetched from the host data buffer and written to th unit.

The write command functions as described. If the "compare" modifier is used then data is written using the IORB function "write validity".

5.4.2 Unsupported And Illegal Functions -

The following functions are part of the Minimal disk MSCP subset described in [1]. They are not implemented by the MSCP server for the following reasons:

1. They are not required for the operation of the TOPS-20 MSCP driver and, in fact, are unused by the MSCP driver.

2. There is no existing method of testing these functions.

3. Implementation of these functions would require significant ammounts of code and data to be added to the TOPS-20 monitor.

4. These commands can be implemented should they be required in the future.

Commands specifying an unsupported function are treated like commands which specify an illegal function. An "Invalid command end message" is returned with sub code "Invalid opcode" specified. The unsupported functions are:

1. ACCESS Command

2. ACCESS PATH Attention Message

3. COMPARE CONTROLLER DATA Command

4. COMPARE HOST DATA Command

5. DETERMINE ACCESS PATHS Command

6. DUPLICATE UNIT NUMBER Attention Message

7. ERASE Command

8. FLUSH Command

9. SET UNIT CHARACTERISTICS Command

6.0 PHYSIO INTERFACE

The PHYSIO system of TOPS-20 provides the means by which the MSCP server accesses massbus disks. The MSCP server also stores disk unit specific data in the Unit Data Block (UDB) for each disk. This data is used only by the MSCP server.

6.1 Services Provided By PHYSIO

The following is a list of PHYSIO services used by the MSCP server. A one line description follows each subroutine name. For calling conventions see [6].

1. ADVCKS - Used to step to the next Channel, Kontroller, and Unit

2. CDSCCW(CDB) - Used to generate a CCW for a channel

3. CHKCKS - Used to check the validity of a Channel, Kontroller and Unit number

4. DGUMAP - Used to execute an instruction for each unit on a channel

5. FNDCKS - Converts from datablock addresses to C K U numbers

6. PHYSIO - Initiate a data operation (Read or Write)

6.2 PHYSIO Entries To The MSCP Server

1. When a disk goes offline PHYSIO calls the MSCP server. This insures that no driver sees that disk as "controller online" until another ONLINE command is issued by that driver. NOTE: This applies to the PHYOFL label in PHYSIO and does not indicate that the drive is offline because of the port being locked to the other side.

2. PHYSIO also calls the MSCP server when a disk comes online so that the MSCP server may send an AVAILABLE attention message to all drivers if the disk is marked as "CI Available". If the disk is not "CI available" when it comes online the MSCP server flags job 0 to run SETSPD at a special entry point to process ALLOW commands. This is the same entry point that processes tape drives coming online.

3. PHYSIO calls MSSIRD at IORB completion. The address of the routine to call (MSSIRD) is stored in the IORB at location IRBIVA.

7.0 SERVICES REQUIRED OF SCAMPI

A full description of the services provided by SCAMPI may be found in [3]. The MSCP server utilizes the following functions and callbacks.

7.1 SCAMPI Callbacks

1. Function 1 - .SSMGR - Message recieved
2. Function 2 - .SSPBC - Port broke connection
3. Function 3 - .SSCTL - Response to listen
4. Function 11 - .SSOSD - OK to send data
5. Function 12 - .SSRID - Request disconnect
6. Function 13 - .SSCIA - Credit available
7. Function 14 - .SSDMA - Named buffer transfer complete

Certain other SCA callbacks are ignored and do not generate errors. They simply return to SCA immediatly these are "don't care" conditions to the server and are expected to occur during normal operation. They are:

1. Function 5 - .SSMSC - Message or datagram send complete
2. Function 7 - .SSLCL - Little credit left

Finally there is a set of SCA callbacks which should never occur. These callbacks cause a BUGHLT, MSSSCA. They are:

1. Function 0 - .SSDGR - Datagram recieved
2. Function 4 - .SSCRA - Connection response available
3. Function 6 - .SSDDG - Datagram dropped
4. Function 10 - .SSNCO - Node coming online

7.2 Services Provided By SCAMPI

1. SC.ABF - Allocate a buffer
2. SC.ACC - Accept a connection
3. SC.DIS - Close a connection
4. SC.LIS - Listen for a connection
5. SC.NOD - Return node number given CID
6. SC.MAP - Map a named buffer
7. SC.RCD - Return configuration data for a node
8. SC.REJ - Reject a connection
9. SC.REQ - Request a named buffer
10. SC.SMG - Send a message
11. SC.SND - Send a named buffer
12. SC.UMP - Unmap a named buffer

8.0 PERIODIC CHECK

The MSCP server is called periodically by the scheduler as part of the normal CLK2 scheduler cycle and when requested during the short scheduler cycle. During this periodic check the MSCP server does the following:

1. Retries all queued commands and closes the connection if the command has timed out.
2. Checks that all drivers have communicated within the drivers timeout interval. The connection is closed if the driver has not communicated.
3. Broadcasts AVAILABLE attention messages to drivers which may not have recieved the messages.
4. Checks the state of all SCA connections.

9.0 ERROR HANDLING

9.1 Error Logging Datagrams

The MSCP server does not generate error logging datagrams, nor is it required to by [1]. All relevant error information is generated automatically by existing mechanisms in PHYSIO and by the TOPS-20 BUG. facility. All relevant device error information is generated as a part of the normal operation of PHYSIO.

9.2 IORB Error To MSCP Error Mapping

The following table shows the mapping of IORB errors to the MSCP status bits which appear in the commands end packet.

IORB ERROR BIT	MSCP STATUS BIT
IS.DVE	ST.DVE
IS.WGU	ST.DVE
IS.NRT	ST.DVE
IS.DTE	ST.DAT
IS.RTL	ST.DAT
IS.WLK	ST.WPR

10.0 ALGORITHIMS

10.1 Unit Number Calculation

The MSCP server uses the following formula to determine unit numbers.

```
<Unit number> = <channel number> * CHNCOD
              + <<kontroller number> + 1> * KONCOD
              + <local unit number> + 1
```

Where CHNCOD = 420 octal and KONCOD = 20 octal.

10.2 Unit States

The following table describes the MSCP unit states.

MASSBUS UNIT STATE	ONLINE COMMAND GIVEN? AND IN EFFECT	MSCP UNIT STATE
Nonexistant	N/A	Offline
Channel Offline	N/A	Offline
Offline	NO	Offline

Offline	!	YES	! Online **
Online	!	NO	! Available
Online	!	YES	! Online

** NOTE: When a unit is "massbus offline" it may be a transient condition due to the drive being dual ported. If "MSCP online" is still in effect then the unit is treated as "MSCP online" to save overhead. MSCP online is always cleared on an offline interrupt.

11.0 GLOBAL DATA STRUCTURES

11.1 UDB Changes

The MSCP server uses 2 bits in the UDBSTS word of the UDB. The first of these bits, US.BDK, indicates that a broadcast of an AVAILABLE attention message is needed. There is also a bit, US.CIA, which is set and cleared by the SMON Jsys. This bit, when set, indicates that the disk is CI available through the MSCP server.

The UDB characteristics word, UDBCHR, for disk devices contains a group of bits, UC.OLB. These bits are the rightmost bits in the word and indicate which drivers have issued online commands for the unit. There is one bit for each driver (currently 4).

11.2 IORB Changes

Word IRBLEN of the long form IORBS contains the address of the command packet and the connection id index. The CCW list starts at IRBLEN+1.

12.0 TESTING

The MSCP server will be tested by three major efforts: DVT (Design Verification Testing), by the PAGES and MULTIO programs and by software fault insertion and performance evaluation. Additional testing will occur by using the disks for general timesharing. DVT is conducted by hardware engineering and includes a comprehensive fault insertion effort aimed at validating the operation of the hardware, microcode and software in the face of failures. This should provide adequate assurance that the MSCP servers error recovery procedures are functioning correctly.

PAGES and MULTIO are programs traditionally used to verify PHYSIO interfaces to disks. They provide heavy load to the PHYSIO/DISK structures and when used on disks services by the MSCP server they will provide a considerable load for it also.

The software fault insertion testing is an informal process aimed at

testing infrequently used paths through the code. The performance evaluation will be aimed at determining the optimal parameters for MSCP server operation, and determining the typical access time for a page of data on a disk handled by the MSCP server. This program of testing and evaluation will include running the MSCP server while there is a heavy CI load and a heavy disk load.

13.0 DOCUMENTATION IMPACT

The MSCP server is largely transparent to the user and system administrator. Most existing documentation will be unnaffected except for the following:

1. Monitor Calls Manual - SMON% and TMON% Documentation should reflect the new function .SFMSD.

2. System Managers Guide - Should reflect the new SETSPD commands, ALLOW and RESTRICT, and explain their use.

3. BUGHLT document - Should reflect the new BUGxxx gentrated by the MSCP server.

4. Installation Guide - Should reflect the new SETSPD commands.