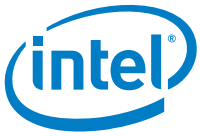


Table 10-23 Status Info (Continued)

Field	Bit(s)	Description
UDPCS	1	UDP Checksum Calculated on Packet.
Reserved	0	Reserved.

Table 10-24 MNG Status

Name	Bits	Description
Pass RMCP 0x026F	0	Set when the UDP/TCP port of the MNG packet is 0x26F.
Pass RMCP 0x0298	1	Set when the UDP/TCP port of the MNG packet is 0x298.
Pass MNG Broadcast	2	Set when the MNG packet is a broadcast packet.
Pass MNG Neighbor	3	Set when the MNG packet is a neighbor discovery packet.
Pass ARP req / ARP Response	4	Set when the MNG packet is an ARP response/request packet.
Reserved	7:5	Reserved.
Pass MNG VLAN Filter Index	10:8	
MNG VLAN Address Match	11	Set when the MNG packet matches one of the MNG VLAN filters.
Unicast Address Index	14:12	Indicates which of the 4 unicast Ethernet MAC addresses match the packet. Valid only if the unicast address match is set.
Unicast Address Match	15	Set when there is a match to any of the 4 unicast Ethernet MAC addresses.
L4 port Filter Index	22:16	Indicate the flex filter number.
L4 port Match	23	Set when there is a match to any of the UDP / TCP port filters.
Flex TCO Filter Index	26:24	
Flex TCO Filter Match	27	
IP address Index	29:28	Set when there is a match to the IP filter number. (IPv4 or IPv6).
IP address Match	30	Set when there is a match to any of the IP address filters.
IPv4 Packet	31	Set to 0b when packet is IPv4 (regardless of address match).
Decision Filter Match	39:32	Set when there is a match to one of the decision filters.

10.5.2.2.3 Read Status Command

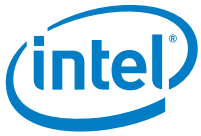
The BMC can read the 82599 status. The 82599 asserts an alert prior to the BMC reading the status bytes. There can be two reasons for the 82599 to send status to the BMC (described in Section 3.2.3):

1. The external BMC asserts a request for reading the 82599 status.
2. The 82599 detects a status change as described in Section 3.2.3.

Note that commands 0xC0/0xD0 are for backward compatibility. 0xD0/0xC0 can be used for other payloads the 82599 defines in the opcode, which payload this transaction is. When 0xDE command is set, the 82599 always returns opcode 0sDD with the 82599 status. The BMC reads the event causing the notification, using the Read Status command as follows:

Function	Command
Read Status	0xC0 or 0xD0 or 0xDE

Function	Byte Count	Data 1 (Op-Code)	Data 2 (Status data 1)	Data 3 (Status data 2)
Receive TCO Partial Status	3	0xDD	See the following table	



The following table lists the status data byte 1:

Bit	Name	Description
7	LAN Port	0b = Alert came from LAN port 0. 1b = Alert came from LAN port 1
6	TCO Command Aborted	0b = A TCO command abort event has not occurred since the last read status cycle. 1b = A TCO command abort event has occurred since the last read status cycle. See Section 3.2.5.2 for command abort flow.
5	Link Status Indication[1]	0b = LAN link down. 1b = LAN link is up.
4	PHY Link Forced Up	Contains the value of the MMNGC.MNG_VETO bit.
3	Initialization Indication[2]	0b = An EEPROM reload event has not occurred since the last read status cycle. 1b = An EEPROM reload event has occurred since the last read status cycle.
2	Reserved	Reserved as 0b.
1:0	Power State[3]	00b = Dr state. 01b = D0u state. 10b = D0 state. 11b = D3 state.

1. When the 82599 is working in teaming mode, and presented as one SMBus device, the link indication is 0b only when both links (on both ports) are down. If one of the LANs is disabled, its link is considered to be down.
2. This indication is asserted when the 82599 manageability block reloads the EEPROM and its internal database is updated to EEPROM default values. This is an indication that the external BMC should re-configure the 82599, if other values besides the EEPROM default should be configured.
3. In single address mode, the 82599 reports the highest power-state modes in both devices. The D state is marked in this order: D0, D0u, Dr, and D3.

Status data byte 2 is used for the BMC for an indication whether the LAN driver is alive and running.

The driver valid indication is a bit that is set by the driver when it is coming up, and cleared when it goes down to Dx state or cleared by the hardware on PCI reset.

Bits 2 and 1 indicate that the LAN driver is not stuck. Bit 2 indicates whether the interrupt line of the LAN function is asserted, and bit 1 indicates whether the driver between the last read status cycle dealt the interrupt line.

The following table lists status data byte 2:

Bit	Name	Description
7	Reserved	Reserved.
6	Reserved	Reserved.
5	Reserved	Reserved.
4	LinkSec Indication	If set, indicates that a LinkSec event has occurred. Use the read LinkSec parameters with the LinkSec interrupt cause parameter to read the interrupt cause

Bit	Name	Description
3	Driver Valid Indication	0b = LAN driver is not alive. 1b = LAN driver is alive.
2	Interrupt Pending Indication	0b = LAN interrupt is not asserted. 1b = LAN interrupt is asserted.
1	ICR Register Read/Write	0b = ICR register was not read since the last read status cycle. 1b = ICR register was read since the last read status cycle. Reading the ICR means that the driver has dealt with the interrupt that was asserted.
0	Reserved	Reserved.

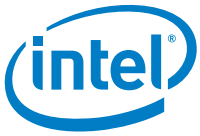
Note: When the 82599 is in teaming mode, these bits represent both cores:

- The driver alive indication is set if 1b of the driver is alive.
- The LAN interrupt is considered to be asserted if one of the interrupt lines is asserted.
- The ICR is considered to read if one of the ICRs was read (LAN0 or LAN1).

The following table lists the possible values of bits 2, 1 and what the BMC can assume according to that:

Previous	Current	
Don’t care	00b	Interrupt is not pending – OK.
00b	01b	New interrupt is asserted – OK.
10b	01b	New interrupt is asserted – OK.
11b	01b	Interrupt is waiting for reading – OK.
01b	01b	Interrupt is waiting for reading by the driver more than one read status cycle – Not OK (possible driver hang state).
Don’t Care	11b	Previous interrupt was read and current interrupt is pending – OK.
Don’t Care	10b	Interrupt is not pending – OK.

Note: The BMC reads should consider the time it takes for the driver to deal with the interrupt (a few microseconds), too frequent reads give false indications.



10.5.2.2.4 Get System Ethernet MAC Address

The Get System Ethernet MAC Address returns the system Ethernet MAC Address (RAL0, RAH0) over the SMBus. This command is a single fragment Read Block transaction, with the following format:

Function	Command
Get system Ethernet MAC Address	0xD4

Data returned from the 82599:

Function	Byte Count	Data 1 (Op-Code)	Data 2	...	Data 7
Get system Ethernet MAC Address	7	0xD4	Ethernet MAC Address MSB	...	Ethernet MAC Address LSB

10.5.2.2.5 Read Management Parameters

In order to read the management parameters the BMC should execute two SMBus transactions. The first transaction is a block write that sets the parameter that the BMC wants to read. The second transaction is block read that reads the parameter.

This is the block write transaction:

Function	Command	Byte Count	Data 1
Management Control Request	0xC1	1	Parameter Number

Following the block write the BMC should issue a block read that reads the parameter that was set in the Block Write command:

Function	Command
Read Management Parameter	0xD1

Data returned from the 82599:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data N
Read Management Parameter	N	0xD1	Parameter Number (PN#)	Parameter Dependent		

The returned data is as follows:

Parameter	PN#	Parameter Data
Keep PHY Link Up	0x00	A single byte parameter — Data 2: Bit 0 = Reflects the setting of the MMNGC.MNG_VETO bit. Bit [7:1] = Reserved.
Wrong Parameter Request	0xFE	the 82599 only: This parameter is returned on a read transaction, if in the previous Read command the BMC sets a parameter that is not supported by the 82599.
the 82599 Not Ready	0xFF	the 82599 only: Returned on Read Parameters command when the data that should have been read is not ready. The BMC should retry the read transaction.

Note: It might be that the parameter that is returned is not the parameter requested by the BMC. The BMC should verify the parameter number (default parameter to be returned is 0x1).

It is BMC's responsibility to follow the procedure Previously defined. If the BMC sends a Block Read command (as previously described) that is not preceded by a Block Write command with bytecount=1b, the 82599 sets the parameter number in the read block transaction to be 0xFE.

10.5.2.2.6 Read MNG RCV Filter Parameters

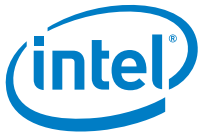
In order to read the MNG RCV filter parameters, the BMC should execute two SMBus transactions. The first transaction is a block write that sets the parameter that the BMC wants to read. The second transaction is block read that reads the parameter.

This is the block write transaction:

Function	Command	Byte Count	Data 1	Data 2
Update MNG RCV Filter Parameters	0xCC	1 or 2	Parameter Number (PN#)	Parameter Data

The following table lists the different parameters and their contents**:**

Parameter	PN#	Parameter Data
Filters Enable	0x1	None.
MNG2HOST Configuration	0xA	None.
Fail-Over Configuration	0xB	None.
Flex Filter 0 Enable Mask and Length	0x10	None.



Parameter	PN#	Parameter Data
Flex Filter 0 Data	0x11	Data 2 – Group of Flex filter's bytes: 0x0 = bytes 0-29. 0x1 = bytes 30-59. 0x2 = bytes 60-89. 0x3 = bytes 90-119. 0x4 = bytes 120-127.
Flex Filter 1 Enable Mask and Length	0x20	None.
Flex Filter 1 Data	0x21	Same as parameter 0x11 but for filter 1.
Flex Filter 2 Enable Mask and Length	0x30	None.
Flex Filter 2 Data	0x31	Same as parameter 0x11 but for filter 2.
Flex Filter 3 Enable Mask and Length	0x40	None.
Flex Filter 3 Data9	0x41	Same as parameter 0x11 but for filter 3.
Filters Valid	0x60	None.
Decision Filters	0x61	1 byte to define the accessed manageability decision filter (MDEF). Data 2 = Decision filter number.
VLAN Filters	0x62	1 byte to define the accessed VLAN tag filter (MAVTV). Data 2 = VLAN filter number.
Flex Ports Filters	0x63	1 byte to define the accessed manageability flex port filter (MFUTP). Data 2 = Flex port filter number.
IPv4 Filter	0x64	1 byte to define the accessed IPv4 address filter (MIPAF). Data 2 = IPv4 address filter number.
IPv6 Filters	0x65	1 byte to define the accessed IPv6 address filter (MIPAF). Data 2 = IPv6 address filter number.
MAC Filters	0x66	1 byte to define the accessed Ethernet MAC address filters pair (MMAL, MMAH). Data 2 = Ethernet MAC address filters pair number (0-3).
Wrong Parameter Request	0xFE	Returned by the 82599 only. This parameter is returned on read transaction, if in the previous Read command the BMC sets a parameter that is not supported by the 82599.
the 82599 Not Ready	0xFF	Returned by the 82599 only, on Read Parameters command when the data that should have been read is not ready. This parameter has no data.

Following the block write the BMC should issue a block read that readS the parameter that was set in the Block Write command:

Function	Command
Request MNG RCV Filter Parameters	0xCD

Data returned from the 82599:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data N
Read MNG RCV Filter Parameters	N	0xCD	Parameter Number (PN#)	Parameter Dependent		

The returned data is in the same format of the Update command.

Note: If the parameter that is returned is not the parameter requested by the BMC, the BMC should verify the parameter number (default parameter to be returned is 0x1).

If the parameter number is 0xFF, it means that the data that the 82599 should supply is not ready yet. The BMC should retry the read transaction.

It is BMC's responsibility to follow the procedure previously defined. If the BMC sends a Block Read command (as previously described) that is not preceded by a Block Write command with bytecount=1b, the 82599 sets the parameter number in the read block transaction to be 0xFE.

10.5.2.2.7 Read Receive Enable Configuration

The BMC uses this command to read the receive configuration data. This data can be configured in the Receive Enable command or through EEPROM loading at power up.

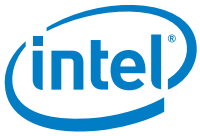
Read Receive Enable Configuration command format (SMBus Read Block Protocol):

Function	Command
Read Receive Enable	0xDA

Data returned from the 82599:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data 8	Data 9	...	Data 12	Data 13	Data 14	Data 15
Read Receive Enable	15 (0x0F)	0xDA	Receive Control Byte	Ethernet MAC Address MSB	...	Ethernet MAC Address LSB	IP Address MSB	...	IP Address LSB	BMC SMBus Address	Interface Data Byte	Alert Value Byte

The detailed description of each field is specified in the Receive Enable command description in Section 10.5.2.1.3.



10.5.2.2.8 Read LinkSec Parameters

In order to read the MNG LinkSec parameters, the BMC should execute two SMBUS transactions. The first transaction is a block write that sets the parameter that the BMC wants to read. The second transaction is block read that reads the parameter.

This is the block write transaction:

Function	Command	Byte Count	Data 1	Data 2
Update MNG RCV Filter Parameters	0xC9	1	Parameter Number (PN#)	Parameter Data

The following table lists the different parameters and their contents:

Parameter	PN#	Parameter Data
LinkSec Interrupt Cause	0x0	None.
LinkSec Rx Parameters	0x1	None.
LinkSec Tx Parameters	0x2	None.

Following the block write the BMC should issue a block read that reads the parameter that was set in the Block Write command:

Function	Command	Byte Count	Data 1	Data 2 — n
Read LinkSec Parameters	0xD9	2,18 or 22	Parameter Number (PN#)	Parameter Data

The following table lists the different parameters and their contents**:**

Parameter	PN#	Parameter Data
LinkSec Interrupt Cause	0x0	This command must return 1 byte (Data2). This byte contains the LinkSec interrupt cause, according to the following values: Data2: Bit 0 = Tx key packet number threshold met. Bit 1 = Host requested ownership. Bit 2 = Host released ownership. Bit 3...31 = Reserved.

Parameter	PN#	Parameter Data
LinkSec Rx Parameters	0x1	Data 2: Reserved. Data 3: LinkSec ownership status. See Table 10-25. Data 4: LinkSec host control status. See Table 10-26. Data 5: Rx host identifier (MSB). Data 6: Rx host identifier (LSB). Data 7: Rx SCI (MSB) … Data 12: (LSB). Data 13: Reserved. Data 14: Rx SA AN — The association number currently used for the active SA. Data 15: Rx SA packet number (MSB) … Data 18: (LSB). Rx SA packet number is the last packet number, as read from the last valid Rx LinkSec packet.
LinkSec Tx Parameters	0x2	Data 2: Reserved Data 3: LinkSec ownership status. See Table 10-25. Data 4: LinkSec host control status. See Table 10-26. Data 5: Tx port identifier (MSB). Data 6: Tx port identifier (LSB). *Note:* Tx port identifier is reserved to 0x0 for this implementation. Data 7: Tx SCI (MSB) … Data 12: (LSB). Data 13: Reserved. Data 14: Tx SA AN — The association number currently used for the active SA. Data 15: Tx SA packet number (MSB) … Data 18: (LSB). Data 19: packet number threshold (MSB) … Data 21: (LSB). Tx SA packet number is the last packet number, as read from the last valid Tx LinkSec packet. Data 22: Tx Control Status. See Table 10-27.

Table 10-25 LinkSec Owner Status

Value	Description
0x0	Host is LinkSec owner.
0x1	BMC is LinkSec owner.

Table 10-26 LinkSec Host Control Status

Bit	Description
0	Reserved.
1	Allow host traffic: 0b = Host traffic is blocked. 1b = Host traffic is allowed.
2..7	Reserved.

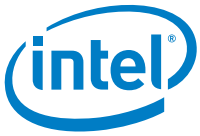


Table 10-27 Tx Control Status

Bit	Description
0..4	Reserved.
5	Include SCI: 0b = Do not include SCI in Tx packets. 1b = Include SCI in Tx packets.
6..7	Reserved

10.5.2.3 SMBus ARP Transactions

Note: All SMBus-ARP transactions include PEC byte.

10.5.2.3.1 Prepare to ARP

This command clears the Address Resolved flag (set to false). It does not affect the status or validity of the dynamic SMBus address. It is used to signal all devices that the ARP master is starting the ARP process:

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0001	0	[Data dependent value]	0	

10.5.2.3.2 Reset Device (General)

This command clears the Address Resolved flag (set to false). It does not affect the status or validity of the dynamic SMBus address.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0010	0	[Data dependent value]	0	

10.5.2.3.3 Reset Device (Directed)

The Command field is NACK-ed if the bits 7 through 1 do not match the current the 82599 SMBus address.

It clears the Address Resolved flag (set to false). It does not affect the status or validity of the dynamic SMBus address.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	Targeted slave address | 0	0	[Data dependent value]	0	

10.5.2.3.4 Assign Address

This command assigns the 82599's SMBus address. The address and command bytes are always acknowledged.

The transaction is aborted immediately (NACK-ed-) if any of the UDID bytes differ from the 82599 UDID bytes. If successful, the manageability interface updates the SMBus address internally. This command also sets the Address Resolved flag to true.

1	7	1	1	8	1	8	1	
	Slave Address	Wr	A	Command	A	Byte Count	A	•••
	1100 001	0	0	0000 0100	0	0001 0001	0	

8	1	8	1	8	1	8	1	
Data-1	A	Data-2	A	Data-3	A	Data-4	A	•••
UDID byte 15 (MSB)	0	UDID byte 14	0	UDID byte 13	0	UDID byte 12	0	

8	1	8	1	8	1	8	1	
Data-5	A	Data-6	A	Data-7	A	Data-8	A	•••
UDID byte 11	0	UDID byte 10	0	UDID byte 9	0	UDID byte 8	0	

8	1	8	1	8	1	
Data-9	A	Data-10	A	Data-11	A	•••
UDID byte 7	0	UDID byte 6	0	UDID byte 5	0	

8	1	8	1	8	1	8	1	
Data-12	A	Data-13	A	Data-14	A	Data-15	A	•••
UDID byte 4	0	UDID byte 3	0	UDID byte 2	0	UDID byte 1	0	

8	1	8	1	8	1	1
Data-16	A	Data-17	A	PEC	A	P
UDID byte 0 (LSB)	0	Assigned Address	0	[Data dependent value]	0	

10.5.2.3.5 Get UDID (General and Directed)

The Get UDID command depends on whether this is a directed or general command.

The General Get UDID SMBus transaction supports a constant command value of 0x03.

The Directed Get UDID SMBus transaction supports a dynamic command value equal to the dynamic SMBus address with the LSB bit set.

Note: Bit 0 (LSB) of Data byte 17 will always be 1b.

If the SMBus address has been resolved (Address Resolved flag is true); for a general command the manageability interface does not acknowledge (NACK) this transaction, for a directed command the manageability always acknowledges (ACK) this transaction.

This command does not affect the status or validity of the dynamic SMBus address nor of the Address Resolved flag.

The command returns the UDID bytes as defined in Section 3.2.7.

S	Slave Address	Wr	A	Command	A	S	•••
	1100 001	0	0	See below	0		

7	1	1	8	1	
Slave Address	Rd	A	Byte Count	A	•••
1100 001	1	0	0001 0001	0	

8	1	8	1	8	1	8	1	
Data-1	A	Data-2	A	Data-3	A	Data-4	A	•••
UDID byte 15 (MSB)	0	UDID byte 14	0	UDID byte 13	0	UDID byte 12	0	

8	1	8	1	8	1	8	1	
Data-5	A	Data-6	A	Data-7	A	Data-8	A	•••
UDID byte 11	0	UDID byte 10	0	UDID byte 9	0	UDID byte 8	0	

8	1	8	1	8	1	
Data-9	A	Data-10	A	Data-11	A	•••
UDID byte 7	0	UDID byte 6	0	UDID byte 5	0	

8	1	8	1	8	1	8	1	
Data-12	A	Data-13	A	Data-14	A	Data-15	A	•••
UDID byte 4	0	UDID byte 3	0	UDID byte 2	0	UDID byte 1	0	

8	1	8	1	8	1	1
Data-16	A	Data-17	A	PEC	~Ã	P
UDID byte 0 (LSB)	0	Device Slave Address	0	[Data dependent value]	1	

10.5.2.4 Example Configuration Steps

This section provides an overview and sample configuration settings for commonly used filtering configurations.Three examples are presented.

The examples are in pseudo code format, with the name of the SMBus command, followed by the parameters for that command and an explanation. Here is a sample:

```
Receive Enable[00]
```

Using the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering.

10.5.2.4.1 Example 1 - Shared MAC, RMCP Only Ports

This example is the most basic configuration. The MAC address filtering are shared with the host operating system and only traffic directed the RMCP ports (0x26F and 0x298) are filtered. For this simple example, the BMC must issue gratuitous ARPs because no filter is enabled to pass ARP requests to the BMC.

10.5.2.4.1.1 Example 1 Pseudo Code

Step 1: - Disable existing filtering:

```
Receive Enable[00]
```

Using the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering:

Receive Enable Control 0x00:

- Bit 0 [0] – Disable receiving of packets

Step 2: - Configure MDEF[0]:

```
Update Manageability Filter Parameters [61, 0, 00000C00]
```

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 0x61). This updates MDEF[0], as indicated by the 2nd parameter (0).

MDEF[0] value of 0x00000C00:

- Bit 10 [1] – port 0x298
- Bit 11 [1] – port 0x26F

Step 3: - Enable filtering:

```
Receive Enable [05]
```

Using the simple form of the Receive Enable command:

Receive Enable Control 0x05:

- Bit 0 [1] – Enable receiving of packets
- Bit 2 [1] – Enable status reporting (such as link lost)
- Bit 5:4 [00] – Notification method = SMBus Alert
- Bit 7 [0] – Use shared MAC

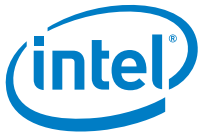


Table 10-28 Example 1 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
L2 Unicast Address	AND								
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR	x							
Port 0x26F	OR	x							
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

10.5.2.4.2 Example 2 - Dedicated MAC, Auto ARP Response and RMCP port filtering

This example shows a common configuration; the BMC has a dedicated MAC and IP address. Automatic ARP responses are enabled as well as RMCP port filtering. By enabling automatic ARP responses the BMC is not required to send the gratuitous ARPs as it did in Example 1. Since ARP requests are now filtered, in order for the host to receive the ARP requests, the manageability-to-host filter is configured to send the ARP requests to the host as well.

For demonstration purposes, the dedicated MAC address is calculated by reading the system MAC address and adding one do it, assume the system MAC is AABBCCDC. The IP address for this example is 1.2.3.4.

Additionally, the XSUM filtering is enabled.

Note that not all Intel Ethernet controllers support automatic ARP responses, please refer to product specific documentation.

10.5.2.4.2.1 Example 2 - Pseudo Code

Step 1: - Disable existing filtering:

```
Receive Enable[00]
```

Using the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering:

Receive Enable Control 0x00:

- Bit 0 [0] – Disable receiving of packets

Step 2: - Read System MAC Address

```
Get System MAC Address []
```

Reads the System MAC address. Assume returned AABBCCDC for this example.

Step 3: - Configure XSUM Filter

```
Update Manageability Filter Parameters [01, 00800000]
```

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) register.

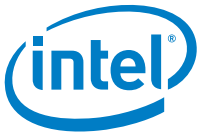
MANC Register 0x00800000:

- Bit 23 [1] – XSUM Filter enable

Note: Some of the following configuration steps manipulate the MANC register indirectly, this command sets all bits except XSUM to zero. It is important to either do this step before the others, or to read the value of the MANC and then write it back with only bit 32 changed. Also note that the XSUM enable bit might differ between Ethernet controllers, refer to product specific documentation.

Step 4: - Configure MDEF[0]

```
Update Manageability Filter Parameters [61, 0, 00000C00]
```



Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 0x61). This updates MDEF[0], as indicated by the 2nd parameter (0).

MDEF value of 0x00000C00:

- Bit 10 [1] – port 0x298
- Bit 11 [1] – port 0x26F

Step 5: - Configure MDEF[1]:

```
Update Manageability Filter Parameters [61, 1, 00000080]
```

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This updates MDEF[1], as indicated by the 2nd parameter (1).

MDEF value of 0x00000080:

- Bit 7 [7] – ARP requests

When enabling automatic ARP responses, the ARP requests still go into the manageability filtering system and as such need to be designated as also needing to be sent to the host. For this reason a separate MDEF is created with only ARP request filtering enabled.

Refer to the next step for more details.

Step 6: - Configure the Management to Host Filter

```
Update Manageability Filter Parameters [0A, 00000002]
```

Use the Update Manageability Filter Parameters command to update the Management Control-to-Host (MANC2H) register.

MANC2H Register 0x00000002:

- Bit 2 [1] – Enable MDEF[1] traffic to go to the host as well

This enables ARP requests to be passed to both manageability and to the host. Specified separate MDEF filter for ARP requests. If ARP requests had been added to MDEF[0] and then MDEF[0] specified in management-to-host configuration then not only would ARP requests be sent to the BMC and host, RMCP traffic (ports 0x26F and 0x298) would have also been sent to both places.

Step 7: - Enable filtering:

```
Receive Enable [8D, AABBCCDD, 01020304, 00, 00, 00]
```

Using the advanced version Receive Enable command, the first parameter:

Receive Enable Control 0x8D:

- Bit 0 [1] – Enable receiving of packets
- Bit 2 [1] – Enable status reporting (such as link lost)
- Bit 3 [1] – Enable automatic ARP responses
- Bit 5:4 [00] – Notification method = SMBus alert
- Bit 7 [1] – Use dedicated MAC

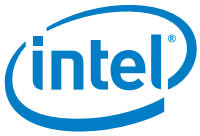
Second parameter is the MAC address (AABBCCDD).

Third parameter is the IP address (01020304).

The last three parameters are zero when the notification method is SMBus Alert.

Table 10-29 Example 2 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		0	1	2	3	4	5	6	7
L2 Unicast Address	AND								x
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR		x						
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR	x							
Port 0x26F	OR	x							
Flex Port 15:0	OR								
Flex TCO 3:0	OR								



10.5.2.4.3 Example 3 - Dedicated MAC and IP Address

This example provides the BMC with a dedicated MAC and IP address and allows it to receive ARP requests. The BMC is then responsible for responding to ARP requests.

For demonstration purposes, the dedicated MAC address is calculated by reading the system MAC address and adding one to it, assume the system MAC is AABBCCDC. The IP address for this example is1.2.3.4. For this example, the Receive Enable command is used to configure the MAC address filter.

In order for the BMC to be able to receive ARP requests, it needs to specify a filter for this, and that filter needs to be included in the manageability-to-host filtering so that the host operating system can also receive ARP requests.

10.5.2.4.3.1 Example 3 - Pseudo Code

Step 1: - Disable existing filtering:

```
Receive Enable[00]
```

Using the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering:

Receive Enable Control 0x00:

- Bit 0 [0] – Disable receiving of packets

Step 2: - Read system MAC address:

```
Get System MAC Address []
```

Reads the system MAC address. Assume returned AABBCCDC for this example.

Step 3: - Configure IP address filter:

```
Update Manageability Filter Parameters [64, 00, 01020304]
```

Use the Update Manageability Filter Parameters to configure an IPv4 filter.

The 1st parameter (0x64) specifies that we are configuring an IPv4 filter.

The 2nd parameter (0x00) indicates which IPv4 filter is being configured, in this case filter 0.

The 3rd parameter is the IP address – 1.2.3.4.

Step 4: - Configure MAC address filter:

```
Update Manageability Filter Parameters [66, 00, AABBCCDD]
```

Use the Update Manageability Filter Parameters to configure a MAC Address filter.

The 1st parameter (0x66) specifies that we are configuring a MAC Address filter.

The 2nd parameter (0x00) indicates which MAC address filter is being configured, in this case filter 0.

The 3rd parameter is the MAC address - AABBCCDD

Step 5: - Configure manageability filters valid to select the IPv4 [0] and MAC[0]filters:

Step 3 configured one of possibly many IP address filters, this step indicates which of those filters should be used when filtering incoming traffic.

```
Update Manageability Filter Parameters [60, 00010001]
```

Use the Update Manageability Filter Parameters to configure the MVFAL register.

The 1st parameter (0x60) specifies that we are configuring the MFVAL register.

The 2nd parameter (0x00010001) is the new value of the MFVAL register.

MFVAL value of 0x00010000:

- Bit 1 [1] – MAC Address Filter 0
- Bit 16 [1] – IPV4 Filter 0

For more information regarding Manageability Filters Valid, see section Section 8.2.3.25.5.

Step 6: - Configure MDEF[0] for IP and MAC filtering:

```
Update Manageability Filter Parameters [61, 0, 00000009]
```

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 0x61). This will update MDEF[0], as indicated by the 2nd parameter (0).

MDEF value of 0x00000040:

- Bit 1 [1] - MAC Address Filtering
- Bit 3 [1] – IP Address Filtering

Step 7: - Configure MDEF[1]:

```
Update Manageability Filter Parameters [61, 1, 00000080]
```

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 0x61). This will update MDEF[1], as indicated by the 2nd parameter (1).

MDEF value of 00000080:

- Bit 7 [7] – ARP Requests

When filtering ARP requests the requests go into the manageability filtering system and as such need to be designated as also needing to be sent to the host. For this reason a separate MDEF is created with only ARP request filtering enabled.

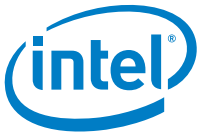
Step 8: - Configure the management to host filter:

```
Update Manageability Filter Parameters [0A, 00000002]
```

Use the Update Manageability Filter Parameters command to update the Management Control-to-Host (MANC2H) register.

MANC2H Register 00000002:

- Bit 2 [1] – Enable MDEF[1] traffic to go to the host as well



Step 9: - Enable filtering:

`Receive Enable [05]`

Using the simple form of the Receive Enable command,:

Receive Enable Control 0x05:

- Bit 0 [1] – Enable receiving of packets
- Bit 2 [1] – Enable status reporting (such as link lost)
- Bit 5:4 [00] – Notification method = SMBus Alert

The resulting MDEF filters are as follows:

Table 10-30 Example 3 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
L2 Unicast Address	AND	x							
Broadcast	AND								
Manageability VLAN	AND								
IP Address	AND	x							
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR		x						
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 15:0	OR								
Flex TCO 3:0	OR								

10.5.2.4.4 Example 4 - Dedicated MAC and VLAN Tag

This example shows an alternate configuration; the BMC has a dedicated MAC and IP address, along with a VLAN tag of 0x32 is required for traffic to be sent to the BMC. This means that all traffic with VLAN a matching tag is sent to the BMC.

For demonstration purposes, the dedicated MAC address is calculated by reading the system MAC address and adding one do it, assume the system MAC is AABBCCDC. The IP address for this example is 1.2.3.4 and the VLAN tag will be 0x0032.

It is assumed the host is not using the same VLAN tag as the BMC. If they were to share the same VLAN tag then additional filtering would need to be configured to allow VLAN tagged non-unicast (such as ARP requests) to be sent to the host as well as the BMC using the manageability-to -host filter capability.

Additionally, the XSUM filtering is enabled.

10.5.2.4.4.1 Example 4 - Pseudo Code

Step 1: - Disable existing filtering:

```
Receive Enable[00]
```

Using the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering:

Receive Enable Control 0x00:

- Bit 0 [0] – Disable receiving of packets

Step 2: - Read system MAC address:

```
Get System MAC Address []
```

Reads the system MAC address. Assume returned AABBCCDC for this example.

Step 3: - Configure XSUM filter:

```
Update Manageability Filter Parameters [01, 00800000]
```

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) register.

MANC Register 0x00800000:

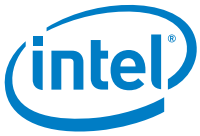
- Bit 23 [1] – XSUM Filter enable

Note: Some of the following configuration steps manipulate the MANC register indirectly, this command sets all bits except XSUM to zero. It is important to either do this step before the others, or to read the value of the MANC and then write it back with only bit 32 changed. Also note that the XSUM enable bit can differ between Ethernet controllers, refer to product specific documentation.

Step 4: - Configure VLAN 0 filter:

```
Update Manageability Filter Parameters [62, 0, 0032]
```

Use the Update Manageability Filter Parameters command to configure VLAN filters. Parameter 0x62 indicates update to VLAN Filter, the 2nd parameter indicates which VLAN filter (0 in this case), the last parameter is the VLAN ID (0x0032).



Step 5: - Enable VLAN 0 filter:

```
Update Manageability Filter Parameters [60, 00000100]
```

The previous step configured a VLAN filter, this step enables it.

Use the Update Manageability Filter Parameters command to enable the VLAN filter (VLAN filter 0) configured in the previous step, this information is written to the Manageability Filters Valid (MFVAL) register. See Section 8.2.3.25.5 for more details about MFVAL.

MFVAL value of 00000100:

- Bit 8 [1] – VLAN Filter 0

Step 6: - Configure MDEF[0]:

```
Update Manageability Filter Parameters [61, 0, 00000040]
```

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 0x61). This updates MDEF[0], as indicated by the 2nd parameter (0).

MDEF value of 00000040:

- Bit 2 [1] – VLAN AND

Step 7: - Enable filtering:

```
Receive Enable [85, AABBCCDD, 01020304, 00, 00, 00]
```

Using the advanced version Receive Enable command, the first parameter:

Receive Enable Control 0x85:

- Bit 0 [1] b – Enable receiving of packets
- Bit 2 [1] b – Enable status reporting (such as link lost)
- Bit 5:4 [00]b – Notification method = SMBus Alert
- Bit 7 [1]b – Use dedicated MAC

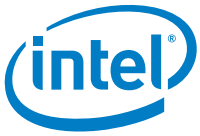
Second parameter is the MAC address: AABBCCDD.

Third parameter is the IP address: 01020304.

The last three parameters are zero when the notification method is SMBus Alert.

Table 10-31 Example 3 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		0	1	2	3	4	5	6	7
L2 Unicast Address	AND								x
Broadcast	AND								
Manageability VLAN	AND	x							
IP Address	AND								
L2 Unicast Address	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 15:0	OR								
Flex TCO 3:0	OR								



10.5.2.5 SMBus Troubleshooting and Recommendations

This section outlines the most common issues found while working with pass-through using the SMBus sideband interface.

10.5.2.5.1 SMBus Commands Are Always NACK'd

There are several reasons why all commands sent to the 82599 from a MC could be NACK'd. The following are most common:

- Invalid NVM Image — The image itself might be invalid or it could be a valid image and is not a pass-through image, as such SMBus connectivity is disabled.
- The MC is not using the correct SMBus address — Many MC vendors hard-code the SMBus address(es) into their firmware. If the incorrect values are hard-coded, the 82599 does not respond.
- The SMBus address(es) can be dynamically set using the SMBus ARP mechanism.
- Bus Interference — the bus connecting the MC and the 82599 might be unstable, consult the reference schematics for correct pull-up resisters.

10.5.2.5.2 SMBus Clock Speed Is 16.6666 KHz

This can happen when the SMBus connecting the MC and the 82599 is also tied into another device (such as an ICH) that has a maximum clock speed of 16.6666 KHz. The solution is to not connect the SMBus between the 82599 and the MC to this device.

10.5.2.5.3 A Network Based Host Application Is Not Receiving any Network Packets

Reports have been received about an application not receiving any network packets. The application in question was NFS under Linux. The problem was that the application was using the RMPC/RMCP+ IANA reserved port 0x26F (623) and the system was also configured for a shared MAC and IP address with the operating system and MC.

The management control to host configuration, in this situation, was setup not to send RMCP traffic to the operating system (this is typically the correct configuration). This means that no traffic sent to port 623 was being routed.

The solution in this case is to configure the problematic application NOT to use the reserved port 0x26F.

10.5.2.5.4 Unable to Transmit Packets from the MC

If the MC has been transmitting and receiving data without issue for a period of time and then begins to receive NACKs from the 82599 when it attempts to write a packet, the problem is most likely due to the fact that the buffers internal to the 82599 are full of data that has been received from the network but has yet to be read by the MC.

Being an embedded device, the 82599 has limited buffers that are shared for receiving and transmitting data. If a MC does not keep the incoming data read, the 82599 can be filled up This prevents the MC form transmitting more data, resulting in NACKs.

If this situation occurs, the recommended solution is to have the MC issue a Receive Enable command to disable more incoming data, read all the data from the 82599, and then use the Receive Enable command to enable incoming data.

10.5.2.5.5 SMBus Fragment Size

The SMBus specification indicates a maximum SMBus transaction size of 32 bytes. Most of the data passed between the 82599 and the MC over the SMBus is RMCP/RMCP+ traffic, which by its very nature (UDP traffic) is significantly larger than 32 bytes in length. Multiple SMBus transactions may therefore be required to move data from the 82599 to the MC or to send a data from the MC to the 82599.

Recognizing this bottleneck, the 82599 handles up to 240 bytes of data in a single transaction. This is a configurable setting in the NVM. The default value in the NVM images is 32, per the SMBus specification. If performance is an issue, increase this size.

10.5.2.5.6 Losing Link

Normal behavior for the Ethernet controller when the system powers down or performs a reset is for the link to temporarily go down and then back up again to re-negotiate the link speed. This behavior can have adverse affects on manageability.

For example if there is an active FTP or Serial Over LAN (SOL) session to the MC, this connection may be lost. In order to avoid this possible situation, the MC can use the Management Control command detailed in Section 10.5.2.1.5 to ensure the link stays active at all times.

This command is available when using the NC-SI sideband interface as well.

Care should be taken with this command, if the driver negotiates the maximum link speed, the link speed remains the same when the system powers down or resets. This may have undesirable power consumption consequences. Currently, when using NC-SI, the MC can re-negotiate the link speed. That functionality is not available when using the SMBus interface.

10.5.2.5.7 Enable XSum Filtering

If XSum filtering is enabled, the MC does not need to perform the task of checking this checksum for incoming packets. Only packets that have a valid XSum is passed to the MC. All others are silently discarded.

This is a way to offload some work from the MC.

10.5.2.5.8 Still Having Problems?

If problems still exist, contact your field representative. Be prepared to provide the following:

- A SMBus trace if possible
- A dump of the NVM image. This should be taken from the actual 82599, rather than the NVM image provided by Intel. Parts of the NVM image are changed after writing (such as the physical NVM size).